UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Faculty of Behavioural,
Management & Social Sciences

Detecting Combosquat Domains using Active DNS Measurements

Communication of Incident Severity between Customers and Analysts in a SOC

Joost Jansen

Combined MSc Thesis
August 2019

FOX IT
part of nccgroup

Supervisors:
prof. dr. M. Junger (chair) (IEBIS/BMS)
prof. dr. ir. A. Pras (DACS/EEMCS)
O. van der Toorn MSc (DACS/EEMCS)
S. Rog MSc (Fox-IT/MSS)
M. van Hensbergen MSc (Fox-IT/TI)

Preface

In front of you lies the combined master thesis that was written to finalize two master's programmes attended at the University of Twente; Computer Science (CS) with specialization 4TU Cyber Security and Business Information Technology (BIT) with specialization IT Management & Innovation. Although this combined thesis is written as a whole, **it contains two distinct parts that reflect the two distinct programmes**. For the CS part, it was investigated whether a detection model for combosquat domains was possible based on active DNS measurements. For the BIT part, the communication between analysts and customers in a Security Operations Center was analyzed, specifically focusing on possible differences in the perception of 'severeness' of incidents between these two groups. I have started working on the CS part of the thesis from September 2018, and from February 2019 on I have worked on the BIT part. When I first approached Fox-IT in the spring of 2018, and got in contact with Christian & Krijn, I could only hope that things would turn out the way they did. Together with the graduation committee, which consists of prof. dr. M. Junger, prof. dr. ir. A. Pras and O. van der Toorn MSc, multiple research topics were discussed that eventually resulted in the two subjects included in this thesis.

I would like to thank Christian & Krijn for helping me find a graduation assignment and their monthly feedback. Furthermore, I would like to thank Martin, Sanne & Ruud for their helpful feedback and guidance on a daily basis, countless colleagues at Fox-IT who made it fun to go to Fox-IT every day, and from whom I learned a lot over the past year. Special thanks for my graduation committee, who regularly provided really helpful feedback and provided me with new insights when I got stuck. Besides the feedback on this thesis, I really enjoyed the conversations we had about all kinds of topics related to cyber security. Lastly, I would like to thank my family, friends and my girlfriend Dorian who have always motivated me, especially at times when I needed it the most.

I hope you will enjoy reading this combined thesis and that you will gain new insights that can be used to make the (digital) world more secure!

Joost Jansen
Delft - August 12, 2019

Summary

Detecting Combosquat Domains using Active DNS Measurements

Domain squatting is a phenomena where attackers register domains that mimic popular domains and/or trademarks, in order to trick people into believing they are visiting a legitimate website. A distinct form of domain squatting is combosquatting; adding one or more words to an existing domain/trademark to craft a new domain. Think of *http://utwente-login.nl* as a combosquat domain for the original domain *utwente.nl*. A literature study revealed that a lot of research was performed in the field of malicious domain detection, however not specifically tackling the problem of combosquatting domains. Given this information, combined with the active DNS measurements available from the OpenINTEL project, a research was initiated that aimed at creating model to detect these combosquat domains.
At first, it was investigated whether a generic detection model for combosquat domains existed. After a validation, implementation and evaluation phase involving a ground truth dataset of 10.548 labeled domains, it became clear that no generic fingerprint of combosquat domains could be created given the data that was available. This led to the conclusion that it is extremely difficult to construct a generic model for detecting combosquat domains without a predefined list of trademarks.
The next part of the research focused on the lifecycle of combosquat domains, more specifically in which stages of the killchain they reside and which features could be used to determine when a combosquat domain turns into a malicious state.
Finally, a model that was trained on the information from the sub-questions was designed and validated in a real-world context. The results showed that the detection of combosquat domains turning malicious based on active DNS measurements is not sufficient. Future work includes the use of additional data sources and a bigger responsibility for registrars.

Communication of Incident Severity between Customers and Analysts in a Security Operations Center

The Security Operations Center (SOC) of Fox-IT monitors and analyzes computer networks of their customers in order to detect and respond to digital incidents. The initial reason for this research was the suspicion that the perception of 'incident severity' might be different for customers and analysts, while the two groups regularly have contact with each other about this concept. During the problem investigation, several other problems were identified around the escalation process in the SOC.

In order to standardize the escalation process, a new standard for recording types of incidents was chosen and a clear definition of the concept 'impact label' was created. Furthermore, a process model of the workflow in the SOC was constructed to gain a high-level overview of the interactions in the SOC.

To solve the initial problem (the possible differences in perception of the incident severity between customers and analysts), the problem was split into four sub-questions. A survey was constructed based on these sub-questions and was filled in by 53 customers and 22 SOC analysts.

Results showed that significant differences exist between the perception of severity of several types of incidents between customers and analysts. It also became clear that customers sometimes have different notification preferences, and would respond differently to changes in a situation to be assessed.

In the end, an advice for Fox-IT was produced that answers the research questions and provides useful insights into how they can continue to improve their communication to customers, as well as the (cost) efficiency of their internal processes.

Contents

Appendices

List of acronyms

2LD	Second-Level Domain
APT	Advanced Persistent Threat
CDM	Combosquatting Detection Model
CKC	Cyber Kill Chain
CTD	Cyber Threat Detection
CTMp	Cyber Threat Management platform
DNS	Domain Name System
ML	Machine Learning
MSS	Managed Security Services
OSINT	Open Source Intelligence
RR	Resource Record
TI	Threat Intelligence
TLD	Top-Level Domain

Chapter 1

General Introduction

In the current information age, being connected to the internet is a major part of life. Social media, online shopping, movie streaming; all examples of services that are frequently used on the modern-day internet. Despite providing a lot of convenience to people, these services also bring unwanted side-effects [1]. Privacy and identity are at stake when personal data gets compromised, stolen credit card information may result in illegitimate transactions and ransomware may infect one's device, encrypt the files and ask for ransom.

Not only individuals are targeted in this harsh world of cyber crime. Businesses and governments are attacked on a regular basis by a diverse set of attackers. These attackers can be individuals looking for personal gain, hacktivists attacking for propaganda purposes, organized crime groups looking for financial benefits or even military cyber units, disrupting and degrading an adversary's capabilities [2]. Although it is hard to accurately calculate the global cost of cyber security, several models estimate the costs into hundreds of billions of US dollars. In the future, the global costs of cybersecurity will grow even more. Therefore, the academic community as well as business around the world are providing knowledge, methods or services to minimize the impact of cybercrime.

On the one hand, businesses need to protect themselves in advance to minimize the risk of a being a victim of cybercrime. On the other hand, a business should also acknowledge that 100% security can never be achieved and thus, sooner or later the business might become victim.

Fox-IT [3] provides managed & professional services to protect businesses and governments against cyber attacks. The Threat Intelligence (TI) department is responsible for detecting and reporting threats on external networks, that is, out on the internet (e.g. DDoS attacks, phishing campaigns, Advanced Persistent Threat (APT)). More specifically, the Managed Intelligence Service of Fox-IT automatically collects Open Source Intelligence (OSINT) and performs the corresponding analysis/triage for customers. The Managed Security Services (MSS) department is

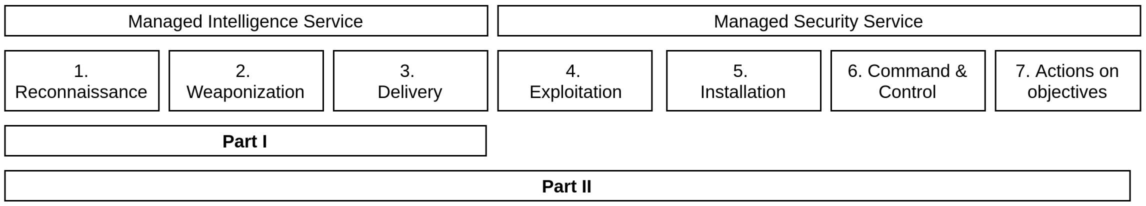


Figure 1.1: Relation of both parts of the thesis to the CKC and Fox-IT departments

responsible for detecting and reporting attacks and other anomalies on an internal network and individual endpoints (e.g. propagating virusses & trojans, insider crime). Part of this is the Security Operation Center, which actively monitors customers' networks and reports if anything suspicious is discovered. New technical developments and the emergence of new malicious actors in the cyber crime domain force Fox-IT to evaluate and improve their products on a continuous basis.

Since this thesis covers two separate but related parts, it's useful to plot the scope of both parts on some scale. A common way to do this in the academic & professional community is to use an attack modelling technique. An attack modelling technique is useful to understand the characteristics of an attack and the objectives of the attackers, in order to gain information about how and when the attack can be stopped. Over the years, several attack modelling techniques have been developed [1], [4], one of which is the Cyber Kill Chain (CKC). The CKC that has been introduced by Lockheed Martin [5] provides 7 common stages of an attack. This CKC will be a connecting thread throughout the thesis so that at any stage of detecting & reporting an incident, a reference to the CKC can be made. Within Fox-IT, the CKC is also widely adapted. In Figure 1.1, the 7 stages of the CKC are shown in relation to two parts of the thesis and the aforementioned Fox-IT departments in order to provide a high-level overview of the thesis.

The CKC consists of 7 stages, which are briefly described below:

1. *Reconnaissance*: An attacker searches for any publicly available information on the victim, in order to prepare the attack.

2. *Weaponization*: An attacker selects the malicious payload that will be sent to the victim. This payload usually contains code that is capable of performing some action on the victims machine, e.g an .exe file on Windows machines.

3. *Delivery*: An attacker delivers the malicious payload to the victim using any communication medium, e.g. providing a link to download the payload or attaching the payload to an email message.

4. *Exploitation*: The victim accidentally or deliberately stores the malicious payload on the victims machine.

5. *Installation*: The malicious payload on the victims machine gets executed, either automatically or by the victim performing some action.

6. *Command and control*: An attacker creates a communication channel to the victims machine to control the status and remotely execute commands. At this stage, the attacker is in control of the victims machine.

7. *Action on objectives*: An attacker performs the actions required to achieve his/her goals on the victims machine or the connected network. From the victims machine, the attacker can also launch a new attack to achieve a goal that requires more access.

At any stage in the killchain the attack can be stopped. In general, the earlier the attack is stopped in the killchain, the less damage is inflicted on the victim. The CKC will be addresses as the 'killchain' throughout the thesis for legibility.

1.1 Thesis structure

The thesis is split into two distinct parts: Part I covers the Computer Science-focused research on combosquat domains and subsequently, Part II covers the BIT-focused research on the communication of incident severity in a Security Operations Center. The reasoning behind this is the fact that besides the overlap of literature and killchain stages of the parts, the research objectives and methodology is unique for each part. Furthermore, each part consists of a distinct research question and corresponding subquestions. Still, both parts have a similar structure, as outlined in Table 1.1.

Part	I	II
Introduction	Chapter 2	Chapter 9
Background information	Chapter 3	Chapter 10
Methodology	Chapter 4	Chapter 11
Results	Chapter 5 Chapter 6 Chapter 7	Chapter 12 Chapter 13 Chapter 14
Conclusion	Chapter 8	Chapter 15

Table 1.1: Chapter structure of the thesis

Part I

Detecting Combosquat Domains using Active DNS Measurements

Chapter 2

Introduction

This introductory chapter will provide information about the current state of research into Cyber Threat Detection (CTD). Section 2.1 will provide a introduction in the research field of CTD and explain some of the basic (technical) concepts related to CTD. Section 2.2 describes the shortcomings in current literature and outlines the ideas that led to the motivation of this research. Next, in Section 2.4 the research questions that are used in this research are displayed. Finally, in Section 2.3 the requirements for the detection model are discussed.

2.1 Introduction into Cyber Threat Detection

The CTD research field covers a lot of subjects [6]. From a technical perspective, CTD looks at e.g. email indicators (email traffic, attachments, subject lines), host-based indicators (malware hashes, binaries, DLL's, registry keys) and network indicators (malicious URLs and domain names) to discover new or emerging threats. Furthermore, OSINT information can be used to identify potential threats in a stage long before it develops into an actual incident. CTD based on network indicators is a subject that has been researched for quite some time and is used to detect a wide range of threats in an early stage. The detection of malicious domain names implies the involvement of the Domain Name System (DNS), the backbone of the internet. DNS is a globally used protocol & system. In short, DNS translates human-readible domain names (e.g. *utwente.nl*) into IP-addresses used in the worldwide TCP/IP infrastructure (e.g. *130.89.3.249*). The aforementioned domain names are first registrered and configured through DNS, before attackers can make use of it. This provides DNS with the unique opportunity to detect malicious domains in the earliest stages of the killchain (*weaponization* and *delivery*) and prevent the attackers from exploiting their malicious domain name.

A subset of malicious domains are 'squatted' domains. These domains are designed in such a way that end users are tricked into believing they are connecting to a legit domain. Domain squatting exists in many forms (see Section 3.3), one of them being combosquatting. In short, combosquatting is the act of combining one or more arbitrary words with an existing trademark, to craft a seemingly legitimate domain. An example would be `utwente-login.nl`, which acts like a login-page for the University of Twente but in fact passes on these credentials to an attacker. Although the loss of university credentials might not be the end of the world, imagine losing credentials for a service authorized to perform financial transactions.

A study on combosquatting by Kintis et al. [7] suggests that combosquatting domains are currently observed 100 times more than typosquat domains. The lack of a generic model for combosquatting domains contributes to this problem, according to Kintis et al. In their large-scale empirical study, they furthermore found out that combosquat domains often remains undetected for a long period of time and that the abuse of combosquat domains is increasing by the year. An analysis of the attacker's usage of combosquat domains resulted in a list of many different forms of abuse, e.g. phishing, social engineering and affiliate abuse. Because of their findings, they called for more research into combosquatting domains & abuse.

It should be clear that malicious combosquat domains pose a serious threat to the mostly uninformed end users. Though some forms of domain squatting have been throughoughly studied, research into combosquatting is (besides the study by Kintis et a) still in its infancy. This leads us to the motivation of this research.

2.2 Motivation

The recent study by Kintis et al. called for the urge of further research after performing an empircal study and finding out that combosquatting domains are a growing threat. Before that, only one other empircal study was performed on combosquatting, which is an industry whitepaper published in 2008 [8]. Moreover, no actual detection models have been proposed in current literature. Kintis et al. state that in comparison to typo squatting, for which detection models have been created and validated, combosquatting lacks a generic model because of its nature; there are infinite amounts of possible combinations. While at the first glance this statement may look credible, no proof is provided for this claim. A frequently heard and simple solution to this would be a trademark search on newly registered domain names. This would however result in a lot of false positives, as Kintis et al. already stated. As an example, imagine the Dutch bank ING; if a substring search is performed on 'ing', the domains 'burgerking.com' and 'bing.com' would be flagged as malicious. Since investigating positives is a costly & time-consuming activity, a more refined model using multiple features is needed to minimize the amount of false positives.

Furthermore, this trademark search on new domain names limits the detection of combosquat domains to the trademarks included in a predefined list. This raises the desire for a generic model that is capable of detecting combosquatting domains regardless of the trademark involved.

For several types of domain abuse, not limited to domain squatting, DNS data can be used to create detection models. A recent study on the detection of snowshoe spam using active DNS data resulted in several characteristics that were useful to detect these domains [9]. Therefore, the question arose whether a model based on active DNS data could also be created for combosquatting domains.

Since the study by Kintis et al. (only) covers an empirical research, no real detection methods have been proposed for combosquat domains. Since attackers seem to be able to keep these domains off the blacklist for a long period of time, the early detection of combosquat domains is beneficial. While Kintis et al. present a temporal analysis of combosquat domains and their presence on blacklists, they do not specify which changes in DNS resource records correlate to the addition on a blacklist. The suspicion arises that a change of IP addresses related to the combosquat domain may be an indicator of a combosquat domain turning malicious. The majority of the detection models described in literature use *passive* DNS data, implying that the attackers have set-up and abused a malicious domain before it is detected. The latter also applies to the TI platform of Fox, which is fed by *passive* data on current attacks, e.g. malicious domains appearing in phishing campaigns. This means that businesses are warned only after the attack has been successfully set up. The desire is to actively detect combosquatting domains in order to warn the involved businesses in an earlier stage and thus, reduce the impact of attacks involving combosquatting domains.

Concluding, the motivation was to design a detection model that was better able to detect combosquatting domains using active DNS data, when compared to existing generic detection models.

2.3 Requirements

Section 3.2 gives an overview of detection methods that have been proposed to detect a wide array of domain abuse, using different data sources. These studies achieve False Positive rates of as low as 1%, and precision rates as high as 98%. However, no reference percentage is known for specifically detecting combosquat domains with active DNS measurements. Since this is an experimental study the lowest False Positive rate and highest precision rate as possible are desired, but as a bare minimum for the model to be of practical use, the following requirements are set:

REQ1: The model should have a False Positive rate of at most 5%.
REQ2: The model should have a precision rate of at least 90%.

2.4 Research Questions

Because there is a need for detecting combosquatting domains in an earlier stage, this will be the main focus of the thesis. An additional challenge is the fact that seemingly, it is difficult to design a generic model that can distinguish combosquat domains from legitimate domains. This leads us to the following technical research problem:

CTD: *How to develop an active combosquatting detection model that meets the requirements set in Section 2.3, so that businesses can be warned in an earlier stage within a threat intelligence platform?*

In order to provide an answer to this problem, first some sub questions have to be answered, which provide the basis for the Combosquatting Detection Model (CDM) design.

CTD1: *Is it possible to construct a generic model for detecting combosquat domains?*
CTD2: *In which stages of the killchain can a combosquat domain reside?*
CTD3: *Which features define the transitions between killchain stages?*

2.5 High level approach

This research approach is based on the engineering cycle, described in [10]. The main research question is a design problem; a treatment needs to be designed in order to a solve problem in a specific context. In this case, the treatment to be designed is a model that is able to detect combosquat domains as they turn malicious, and the context is a threat intelligence platorm.

Furthermore, the first sub-question about a generic model for combosquat domains is also a design problem, which has to be solved separately in the early phase of the research.

After all sub-questions have been answered, the approach consists of constructing a prototype, placing it in a model of the intended context and apply some scenarios to observe the responses, after which the model can be validated. These requirements are described in Section 2.3. It should also be noted that these requirements are not definitive; they may change based on the outcomes of the design cycles.

Chapter 3

Background Information

This chapter will provide background information on detecting squatted domains: domains that impersonate existing trademarks in order to let members of the public think the domain name legitimately belongs to the existing trademark. The DNS, discussed in Section 3.1, is an OSI Layer 7 level protocol & global system that seems inseparable with the detection of malicious domains. Afterwards, Section 3.2 will provide insight in the latest research into malicious domain detection in general. In Section 3.3 will this literature study will continue, but will be focused on squatted domain specifically.

3.1 Domain Name System

The Domain Name System DNS is a global protocol & system that is a major part of the internet [11]. Its most basic function is to translate human readable domain names (e.g. `people.utwente.nl`) into IP-addresses (e.g. `130.89.252.58`) used by the global TCP/IP network. DNS is set up as a distributed, hierarchical client-server system to ensure scalability and high availability. A client can perform a DNS query, which will result in a response from a resolver containing the requested information. Throughout this background information, the `people.utwente.nl` domain and corresponding IPv4-address `130.89.252.58` will be used as a reference.

A domain name consists of multiple levels, separated by a single dot.
The **Top-Level Domain** (**Top-Level Domain (TLD)**) is the rightmost level of the domain name; often called the domain suffix. TLD's can further be split up into country-code TLD's (ccTLD) and generic TLD's (gTLD). Where ccTLD's are allocated to specific countries (e.g. `nl` for The Netherlands), gTLD's are not bound to a country and are thus independent. In the example, `nl` is the TLD.
After the TLD, the domain is further identified by the **Second-Level Domain (2LD)**.

This level of the domain usually corresponds to the organization that has registered the domain name. Every individual person or business can register a 2LD under a TLD, only bound by regulation regarding trademark names. In the example, the 2LD is `utwente`.
After the 2LD, many more domains levels are possible. In the example, `people` refers to a 3LD. This makes it possible for organizations to have multiple 3LD or even 4LD domains for different services within the organization (e.g. `ftp.utwente.nl` for a FTP-server, `www.utwente.nl` for a webserver)

The functions and components of DNS will be described according to the DNS resolve sequence shown in Figure 3.1.

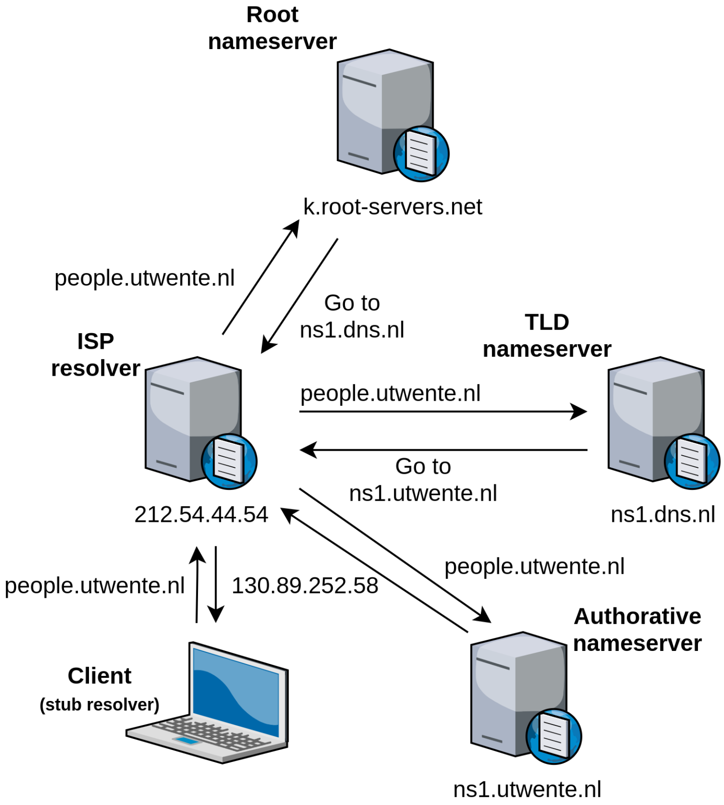


Figure 3.1: An example iterative DNS resolve for *people.utwente.nl*

Resolver

A resolver is the client-side application in the DNS system; it is responsible for initiating and finishing a full DNS query. Resolvers come in different forms; a simple stub resolver is a piece of software that can check whether the answer to a DNS query is available locally, or can pass the query onto another resolver. This second resolver can be hosted at the user's ISP (e.g. `212.54.44.54`); it may be a more complex system and can perform more difficult tasks. The resolver can make iterative or recursive requests to nameservers, which are explained next.

Since iterative requests are mostly used in resolvers, this type of request will be discussed. In Figure 3.1, the client's stub resolver cannot find the IP-address of `people.utwente.nl` locally, so it forwards the request to the recursive resolver. The ISP resolver is now responsible for returning the query to the client. The ISP resolver consecutively queries the root nameserver, TLD nameserver and authoritative nameserver.

Root nameserver
A root nameserver's function is to refer the resolver to the correct TLD nameserver. A total of 13 root nameservers exist geographically distributed around the globe, each one operated by a separate organization. In the example, the root server on `k.root-servers.net` is queried and responds with a TLD nameserver (`ns1.dns.nl`) for the `nl` TLD , to which the resolver heads next. One could say that the TLD part of the domain name has now been resolved.

TLD nameserver
The TLD nameserver holds references to all authoritative nameservers for a certain TLD. Usually TLD nameservers for ccTLD's are hosted by state-owned organizations, whereas gTLD nameservers can also be hosted by other organizations. A TLD nameserver responds to a query by providing the authoritative nameserver for a certain 2LD. In the example, the nameserver for the `nl` TLD responds to the resolver that one of the the authorative nameservers for `utwente` is located at `ns1.utwente.nl`.

Authorative nameserver
The authoritative nameserver holds all the information for certain 2LD, in the form of a Resource Record (RR). These records have a fixed layout, but the information in each of the RR's may be different. The most common RR's are:

Type	Description	Function
`A`	IPv4 address	Returns the IPv4 address for the domain
`AAAA`	IPv6 address	Returns the IPv6 address for the domain
`MX`	Mail exchange	Returns the location of the mail server for the domain
`NS`	Name server	Returns the authoritative name server for the domain
`CNAME`	Canonical name	Returns an alias to refer from one domain to another

The authoritative nameserver responds the requested RR's to the resolver, which in turn responds the completed DNS query to the client. In the example, the authoritative nameserver for `utwente`, which is `ns1.utwente.nl`, has a RR of type `A` for the `people` 3LD; `130.89.252.59`. Note that to the client, the process is recursive; the client only has to perform one action, after which the resolver iteratively queries the different nameservers on behalf of the client.

From an organizational perspective, there are three major roles when it comes to managing & registering new domain names. Figure 3.2 displays the sequence diagram provided in the paper by Kidmose et al. [12], along which the different roles will be explained.

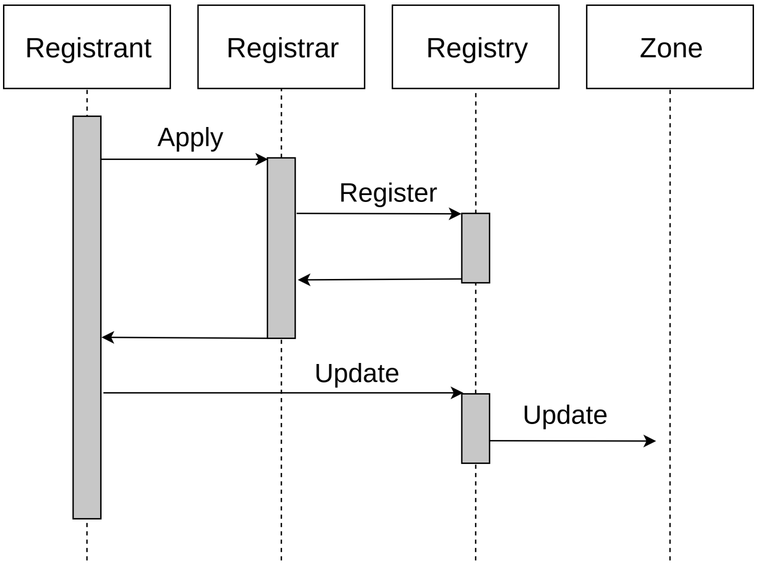


Figure 3.2: Sequence diagram for registering a new domain name

Registrant
The registrant is an individual person or organization that wants to register a 2LD. In the example, this is the University of Twente. Usually, a registrant registers a domain name at a registrar.

Registrar
A registrar is an organization that sells domain names on behalf of one or more registries. Registrars are typically webhosting providers or businesses providing internet services. Registrars have direct contact with the registries and function as an intermediary agency for the registrant.

Registry
A registry is the operator of a TLD and is responsible for taking care of the technical aspects of that operation. Furthermore, it takes care of meeting the requirements set by the ICANN and making the TLD available for commercial use. The latter is most often outsourced to the registrars. In the example, the `nl` TLD is operated by SIDN [13].

As can be seen in Figure 3.2, a registrant applies for a domain name at a registrar. The registrar passes the request on to the registry, which makes sure the domain name does not violate any of the abuse rules. The registry approves and charges a fee for the registration; the registrar on its turn charges a fee to the registrant. Now that the registration is complete, the new domain gets included in the *zone file* update of the registry. Kidmose et al. call this stage of the registration the *pre-registration* stage. After the update is published, the new domain name gets propagated over the other nameservers on the internet and can be resolved everywhere around the globe as displayed in Figure 3.1. This stage is called the *post-registration* phase. The *post-registration* can, in regard to abuse, be split into a *pre-abuse* and *post-abuse* stage. Combining these different stages in the domain name registration process with the CKC introduced in Chapter 1 results in Figure 3.3.

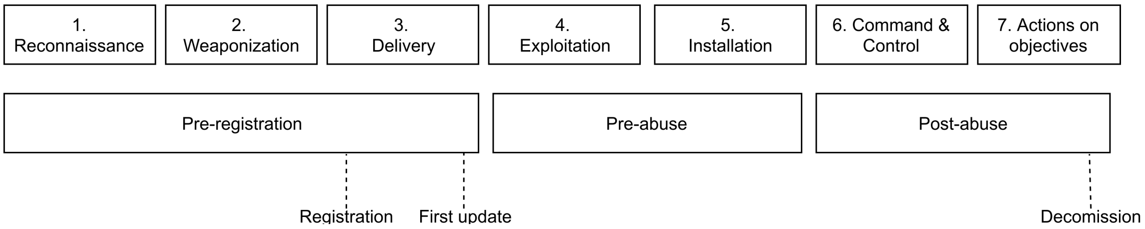


Figure 3.3: DNS registration process combined with CKC

3.2 Detecting Malicious Domains

A starting point for the literature review on the detection of malicious domains is the study performed by [12]. The authors perform an analysis on existing frameworks and theories. In addition, the study by Zhauniarovich et al. [14] provides a systematic review of malicious domain detection approaches based on this DNS data. This is a good starting point for enumerating state of the art research into malicious domain detection using DNS data. Several studies discussed in these overviews will now be discussed. EXPOSURE [15] is a passive DNS analysis service. It focuses on detecting DGA & C&C domains. The service was built en tested on billions of DNS requests, and during the 17 months of operation it detected over 100.000 malicious domains using a J48 decision tree algorithm. The classifier used multiple feature categories; time series based, DNS answer based, TTL value based and domainname based features were used. During the evaluation phase, the service managed to achieve a high detection rate on the training data (99.5%), as well as a low False Positive rate (0.3%).

Phoenix [16] is a system that focuses on detecting and distinguishing DGA and non-

DGA domains, and furthermore is able to find groups of DGA-generated domains that are used alongside in botnets. The system uses passive DNS data. It uses a combination of linguistic and IP-based features to do this ’fingerprinting’ of botnet DGA domain groups. During the evaluation phase, the system was able correctly distinguish DGA-generated domains from non-DGA-generated domains in 94.8% of the cases. Furthermore, the system was also able to detect these DGA domain groups in a real-world setting.

DFBotKiller [17] is another system that is able to detect botnet traffic to malicious C&C servers. This is done by analyzing passive DNS data within the network. Its main task is to assign a negative reputation score to a domain, that takes into account three suspicious measurements. These three metrics are then, along with a number of failed DNS queries, processed into a verdict. The system was evaluated in a test settings and resulted in interesting scores.

A system that acts in the domain pre-registration stage is the PREDATOR framework [18], which has a detection rate for new malicious DNS entries of 70%, in combination with a low false positive rate of 0.35%. This means that as early as in the pre-registration phase, a majority of the malicious domains can already be identified before they can be abused. The system uses features based on registrar data, characteristics of the domain name, previous registration history and correlation with registration bursts. This data is used, since no active nor passive DNS data is present in the pre-registration phase.

The study performed by Vissers et al. [19] is interesting since it provides an automated clustering process that analyzes the registration of malicious registrations in any TLD during the pre-registration stage. Using DNS registration data and publicly available blacklists, they found that at least 80.04% of the data corresponded to 20 malicious DNS registration campaigns. A remaining 19.30% of the traffic could also be related to these campaigns, after more rigorous inspection of the individual campaigns features, resulting in a false positive rate of only 0.92%.

Finally, another study outlines the malicious domain registration ecosystem and puts the different types of abuse in perspective regarding absolute numbers and total costs [20]. This study does not propose a concrete detection system, but gives insights into a malicious actor’s preferences and economic incentives.

Having covered most of the existing theory, Kidmose et al. state that future research should be into detecting malicious & abusive domains in the *pre-registration* stage and should not be limited to spam domains. They suggest to use new, currently unused, features in this stage of detection, namely a *lexicology analysis of a domain name*, the *registration history of domain name*, the *registrant information*, *contents of first zone update* and the *reputation of the registrar*.

3.3 Domain Squatting

A specific type of domain name abuse is called is domain squatting. In the case of domain squatting, an attacker registers a domain name that appears to be the legitimate domain name of a trademark, while in fact it hosts some malicious or abusive content. Domain squatting is particularly hard to detect since it involves no technical errors or flaws in the DNS protocol; in the end it is up to the user to spot a 'squatted' domain. Several types of domain squatting can be identified, each type with its own characteristics and features. To illustrate the different types of domain squatting, Figure 3.4 displays some examples. Additionally, the domain squatting types are briefly explained in Figure 3.4, with combosquatting outlined more in detail.

Type	Example	Literature
Typosquatting	utwent.nl	[21], [22]
Bitsquatting	utwenpe.nl	[23]
Homophone-Based squatting	youtwente.nl	[24]
Homograph-Based squatting	utvvente.nl	[25], [26]
Abbrevsquatting	ut.nl	[27]
Combosquatting	utwente-login.nl	[7], [8]

Figure 3.4: Different types of domain squatting targeting *utwente.nl*

Typosquatting

Typosquatting is a type of domain squatting where the domain names consist of typo variations of popular websites. This method of domain squatting requires an end user to make a mistake when entering a domain name in the browser. In the example, a user wants to visit `utwente.nl`, but accidentally forgets to type the `e` in the domain name. The user then ends up at a completely different website, which could be used for malicious purposes.

Typosquatting is phenomena that has been around for many years. The study by Wang et al. [21] from 2006 already presents a tool that is able to detect and monitor typosquat domains. In their study they list five different forms of typos, ranging from a missing dot typo (e.g. `wwwutwente.nl` to the character-omission typo mentioned in the example. With respect to combosquatting, it is worth mentioning that the amount of typosquat domains for a given popular domain is fixed; at a certain point, no further variations can be computed.

Bitsquatting

Bitsquatting is a type of domain squatting where the attacker anticipates on random bit-errors originating from the hardware in client devices (e.g. computers & smartphones). The study by Nikiforakis et al. [23] shows that these domains are actively being registered and used for abuse purposes. End users are often unaware of being redirected to a malicious website, since they have not performed a faulty action but rather are the victim of hardware errors and the attackers who anticipated for this. In the example, the users requests `utwente.nl`, but due to a random bit-error the client actually resolves `utwenpe.nl`.

Homophone-Based squatting

In another study by Nikiforakis et al. [24], homophone-based squatting or squatting based on words that sound exactly like the original domain, but are written in a distinctive matter. In the example, imagine someone telling the end user to visit the 'utwente' website, which then misinterprets the URL as `youtwente.nl`. Another example would be `weather.com` and `whether.com`; two URL's who sound exactly the same but would resolve to different hosts.

Homograph-Based squatting

Homograph-based squatting is a type of domain squatting where an attacker registers a domain that is visually (almost) indistinctable from the original popular domain. Research on this topic has been done by Holgers et al. [25]. An example attack (using this thesis' font) would be replacing a Latin lower case letter `l` with a number `1`; this would make `paypal.com` hard to distinguish from `paypa1.com`. A study that used homograph-based domain squatting was conducted at the University of Twente, where the original domain was replaced by `utvvente.nl` [26]; in this study the double `v`'s in the domain name were impersonating a `w`. Since the adoption of International Domain Names, which allow non-ASCII characters to be used, homograph-based squatting has become harder to detect since many Latin letters have similar looking characters in different alphabets.

Abbrevsquatting

Abbrevsquatting is a type of domain squatting where an attacker uses the abbreviation of popular domains to trick users into believing they are visiting the legitimate website. In the example, an attacker registers `ut.nl` because the University of Twente is often abbreviated as UT. The study by Lv. et al [27] shows that attackers are aware of the principles of abbrevsquatting and are already leveraging them in malicious ways.

Combosquatting

Combosquatting is the act of combining one or more arbitrary words with an existing trademark, to craft a seemingly legitimate domain. The first research into combosquatting dates back to 2008, when an industry whitepaper by Fair-Winds Partners, LLC was published [8]. An initial set of 30 trademarks was selected based on their ”strength” and number of search terms that were regularly associated with the trademarks. A keyword suggestion tool was used to generate the top 50 most popular keywords associated with the trademarks, and these were then combined into a total of 1500 domain names. Using the major search engines at that time (Google, Yahoo! and MSN), the daily searches for these domain names were analyzed per month. Furthermore the traffic for each domain name was registered. Afterwards, the domains were ordered by their traffic/search ratio and the top and bottom 500 were excluded, leaving 500 domains for manual testing purposes. Results showed that 50.6% domains contained Pay Per Click (PPC) advertisement, 22% of the domains were legitimately used for trademarking purposes and 75% of the trademark+keyword combinations that were not owned by the trademark contained PPC advertisements.

Nine years later, in 2017, the study by Kintis et al. [7] was published. This was and until now is the only academic research into combosquatting. For the first time, they introduce a definition of a combosquat domain; the domain contains a trademark and the domain cannot result by applying the five typosquatting models of Wang et al. [21]. Furthermore, they perform an empirical study on the presence of combosquat domains on the internet, using several large-scale datasets.

They conclude that current domain squatting detection techniques are not detecting combosquat domains properly due to the different threat models involved. They also state that no generative model can be constructed, since in theory an infinite amount of trademark+keyword combinations exist. This makes detection harder to perform than for example typosquatting, for which an exhaustive list of mutations regarding a trademark can be made. A temporal analysis of the detected combosquat domains shows that most domains were active for several months, before the domains were blacklisted. Combosquat domains are used in phishing campaigns, affiliate abuse and other types of abuse. All of these findings result in their call for future research regarding combosquat domains. Kintis et al. argue that not only registrants and registrars can help resolving this threat, but there is also a task for third parties to search for and monitor new combosquat domains.

Chapter 4

Approach

This chapter describes the approach that was used to provide the answer to the research questions. In Section 4.1 the approach that was used to answer the main research question is explained. In Section 4.2 the approach that was used to find out if a generic model could be designed is explained. Section 4.3 described how the different killchain phases are measured and assigned t the stages a combosquat domain can reside in. Finally, Section 4.4 describes the approach to find the most relevant features for the detection model.

4.1 Main research question

Following the problem statement & objectives in Chapter 1, this chapter will the describe the research approach that was used during the research. This provides structure in the research and gives an overview of the steps that were taken. In Figure 4.1 the research approach is displayed. This research approach is based on the engineering cycle, described in [10]. Since this approach focuses on answering knowledge questions and solving design problems, it fitted the needs of this research. The five individual phases of the engineering cycle, interpreted in the context of this research, are outlined below. This approach also keeps in mind the framework provided by [14], which outlines a general framework to design a detection model primarily based on DNS data. Steps included in the framework are data collection, data enrichment, algorithm design & evaluation; these steps will be identifiable in the engineering cycle as well. Before continuing, let us first define a combosquat domain. The definition of a combosquat domain is based on the definition provided by Kintis et al. as shown in Section 3.3, but is extended to fit the needs of this research. Below, the formal definition is outlined alongside a few examples to make the definition more tangible. This definition is expressed in Python code in Section B.2, which is used for validation purposes throughout the thesis.

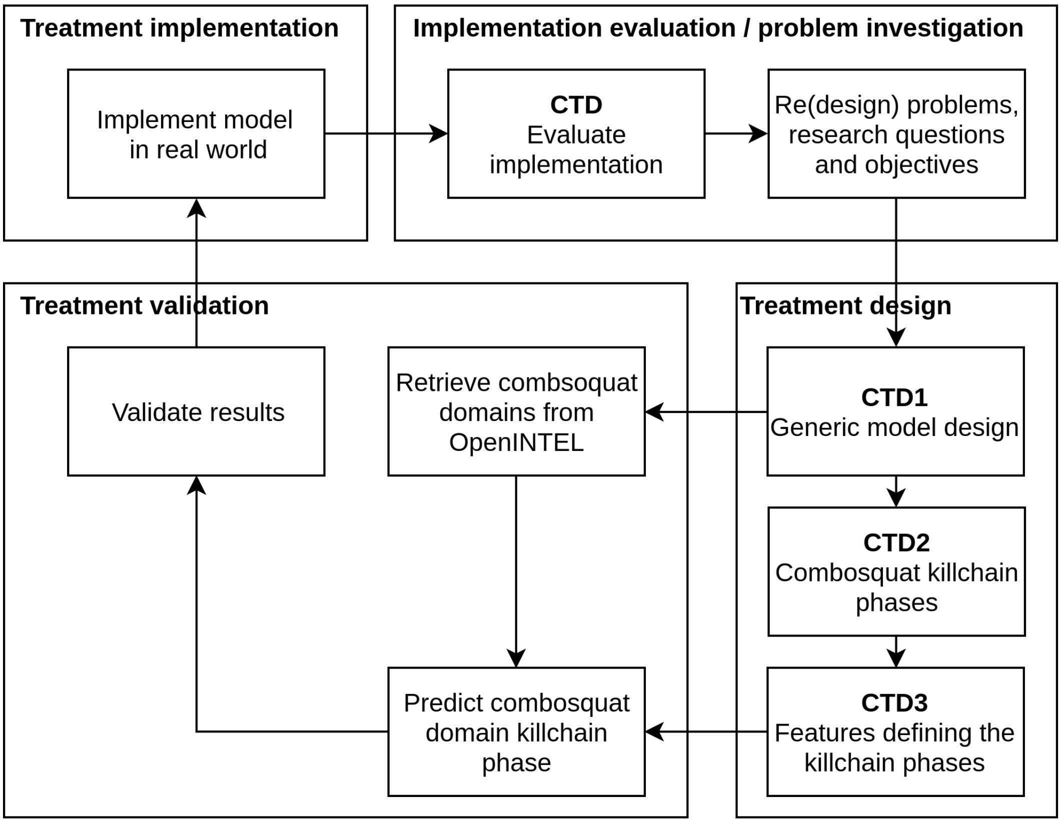

Treatment implementation
Implement model in real world
Implementation evaluation / problem investigation
CTD
Evaluate implementation
Re(design) problems, research questions and objectives
Treatment validation
Validate results
Retrieve combsoquat domains from OpenINTEL
Predict combosquat domain killchain phase
Treatment design
CTD1
Generic model design
CTD2
Combosquat killchain phases
CTD3
Features defining the killchain phases

Figure 4.1: Schematic view of the research approach

Domain name C is considered a combosquat domain of trademark T, if:

1) T is the original trademark name, without a spelling deviation

2) T is left intact within a set of other characters, in this case C

3) T is a standalone word in C

4) The owners of domain names C and trademark T are different

5) C can not be classified as any other form of domain squatting as listed in Section 3.3

Next, the four phases of the design cycle will be explained in the context of this research.

Problem investigation

During the first stage of the research, the initial problems are defined, research questions are created and objectives are set. This stage is described in Chapter 2.

Treatment design

In the treatment design phase, the sub questions will be answered. It should be noted that since CTD1 in itself is an extensive design problem it is only answered once when iterating over the design cycle multiple times. The different methodologies to answer the sub questions are described in the subsections of this chapter. The results from CTD2 (a list of killchain phases that a combosquat domain can reside in) will function as input for CTD3.

Treatment validation

After the sub questions have been answered, it is time to construct the prototype and validate the results. This is done by checking if the prototype matches the requirements. More specific, this means that it is validated that the classifier can distinguish between features that define a 'benign' combosquat and a `malicious` combosquat. The validation of the prototype is done by constructing a *confusion matrix*, as shown in Table 4.1. This is a common method to validate results regarding predicted and actual values.

		Prediction outcome		
		p	**n**	**Total**
Actual value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	Total	P	N	

Table 4.1: Example confusion matrix

From the confusion matrix, three important metrics can be calculated:

$$FP\,rate = \frac{FP}{FP + TN}$$

The *FP rate* represents the amount of wrongly predicted positives compared to the total amount of actual negatives. In this research, a low False Positive rate is desirable since every domain that is predicted as combosquat needs to be investigated; when the domain turns out to be a False Positive, the time spent on the analysis is 'wasted'.

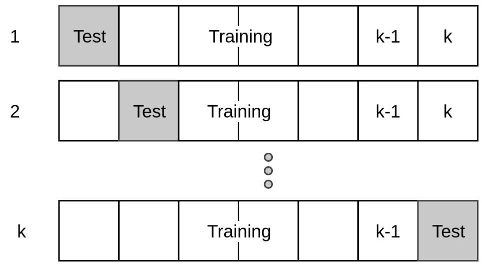


Figure 4.2: k-fold cross validation

$$Accuracy = \frac{TP + TN}{P + N}$$

The *accuracy* score represents the amount of correctly classified labels out of the total. A high accuracy score means that the classifier does not make many errors in relation to the total amount of predictions.

$$Precision = \frac{TP}{TP + FP}$$

The *precision* score represents the amount of labels that are correctly labeled positive relative to the wrongly labeled positives. A high precision score means that the classifier makes little mistakes when labeling positives. On the other hand, a low precision score implies that a lot of positives labels are predicted while in facts, these are negative. In this research a high precision score is one of the main requirements, in order to keep the amount of False Positives as low as possible.

A common problem with ML classifiers is overfitting: a classifier performs perfectly on the training data, but it does not perform well on newly, previously unseen data. To get a better picture of the performance of the classifier, a process called *k-fold cross validation* is performed on the training data. This process is shown in Figure 4.2.

The training data upon which the model is trained is split into k sections. In each one of the k iterations, a different section is used for training & testing purposes. This approach has another advantage, namely that the scores for the classifiers are calculated over the total training data and do not rely on a randomly chosen test sample. Each one of the k iterations produces a *False Positive*, *accuracy* and a *precision* score, which are in the end summed up and divided by k to get the average scores of the *False Positive rate*, *accuracy* and *precision*.

Treatment implementation & Implementation evaluation
In this phase, the validated prototype is placed in its intended context: a threat intelligence platform. An external Virtual Private Server, functioning as an abstraction of such a platform was chosen for this purpose.
Afterwards, the implementation is evaluated and will answer the main research question. A question to be asked is: Did the CDM setting function according to its requirements in the real world context? In most design researches, the design cycle is iterated over multiple times. When a new iteration is started, this phase redefines the problems, research questions and objectives in order to improve the quality of the CDM. The treatment implementation phase, along with the implementation evaluation is shown in Figure 4.3.

As can be seen, all combosquat domains from a specific day `X` are retrieved from OpenINTEL and fed to the classifier, which predicts the domain as either 'benign' or `malicious`. To evaluate this decision, it is checked whether the domain was actually on a blacklist on day `X` or not. Note that this validation completely relies on the presence of a domain on a blacklist; if the classifier manages to predict a malicious domain while it was undetected at that moment, it cannot be verified. Therefore, an extra check is performed; if the domain is not listed on a blacklist on day `X`, the features of the day it actually got detected are obtained and verified against the features of day `X`. If the features match the domain is labeled `malicious` and 'benign' if the features do not match, If no appearance on a blacklist can be observed after the prediction, the domain is also labeled 'benign'.

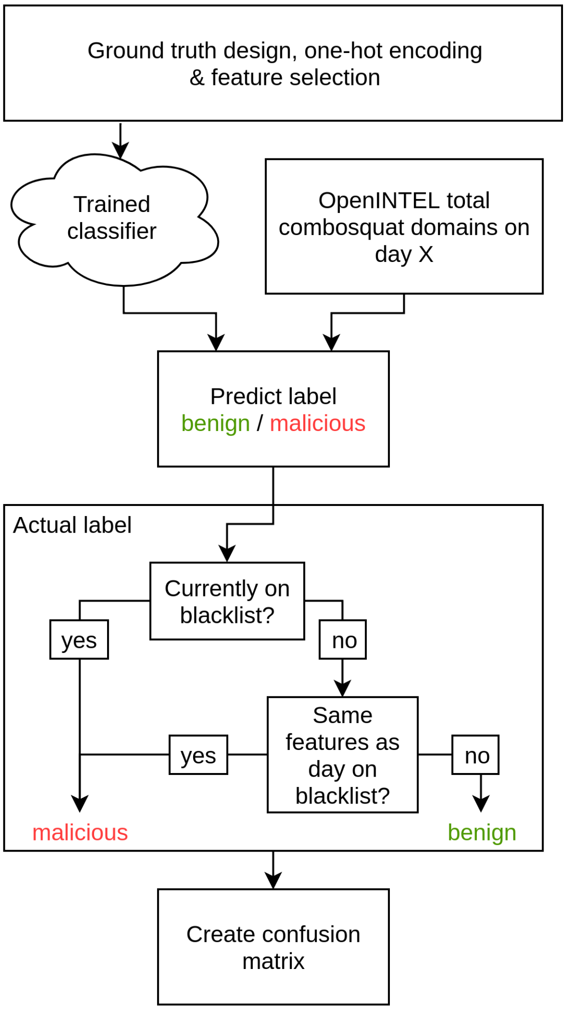

Ground truth design, one-hot encoding & feature selection
Trained classifier
OpenINTEL total combosquat domains on day X
Predict label
benign / malicious
Actual label
Currently on blacklist?
yes
no
Same features as day on blacklist?
yes
no
malicious
benign
Create confusion matrix

Figure 4.3: Treatment implementation & evaluation

4.2 Generic model design

This subsection describes the approach that was used to answer sub question CTD1: *Is it possible to construct a generic model for detecting combosquat domains?* Since this sub question in itself is a design problem, another design cycle has been constructed with the sole purpose of providing an answer to this sub question. This design cycle is shown in Figure 4.4.

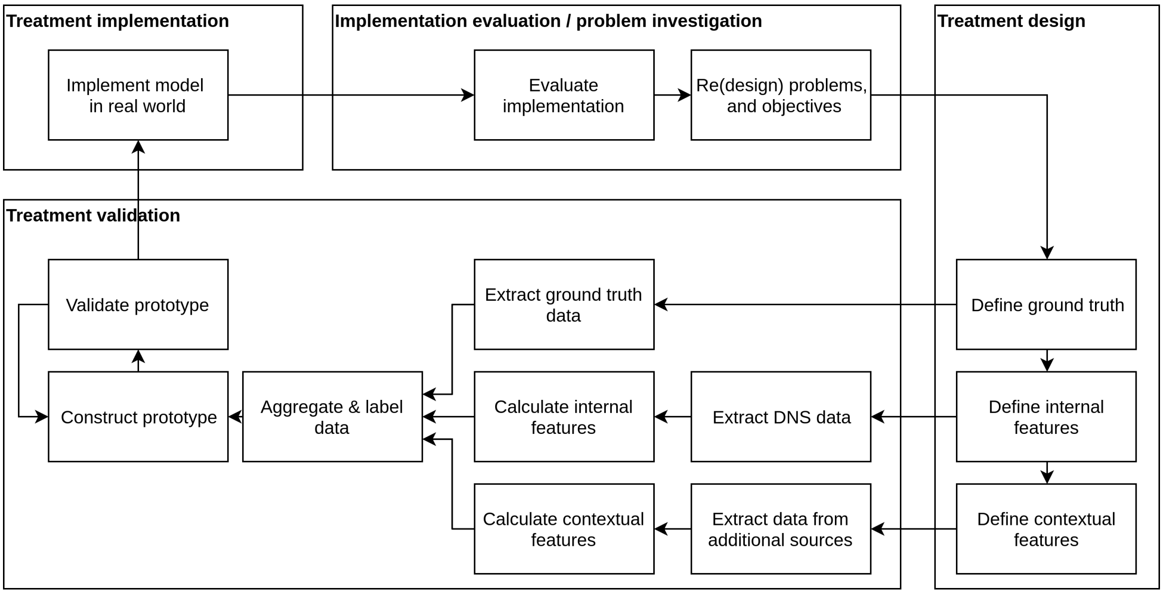

Treatment implementation
Implement model in real world
Implementation evaluation / problem investigation
Evaluate implementation
Re(design) problems, and objectives
Treatment design
Treatment validation
Validate prototype
Construct prototype
Aggregate & label data
Extract ground truth data
Calculate internal features
Calculate contextual features
Extract DNS data
Extract data from additional sources
Define ground truth
Define internal features
Define contextual features

Figure 4.4: Design cycle used to answer CTD1

4.2.1 Problem investigation

This problem is also described in Chapter 2; it is a direct result of the claim that there is no generic model possible for combosquat domains [7]. Since no proof was provided to support this claim, an attempt was made to design a generic model to check if it was indeed the case.

4.2.2 Treatment design

The design of the generic model is based on the approach used by van der Toorn et al. [9]; first a ground truth has to be composed of combosquat domains and non-combosquat domains because labeled data is needed for the model to be based on. Since no labeled dataset was available regarding combosquatting domains, this was created manually using the approach described below.

Before diving into detail regarding the ground truth creation, the different datasets need to be defined. **D** is the set of domain names in the `.com` TLD. **T** is the set

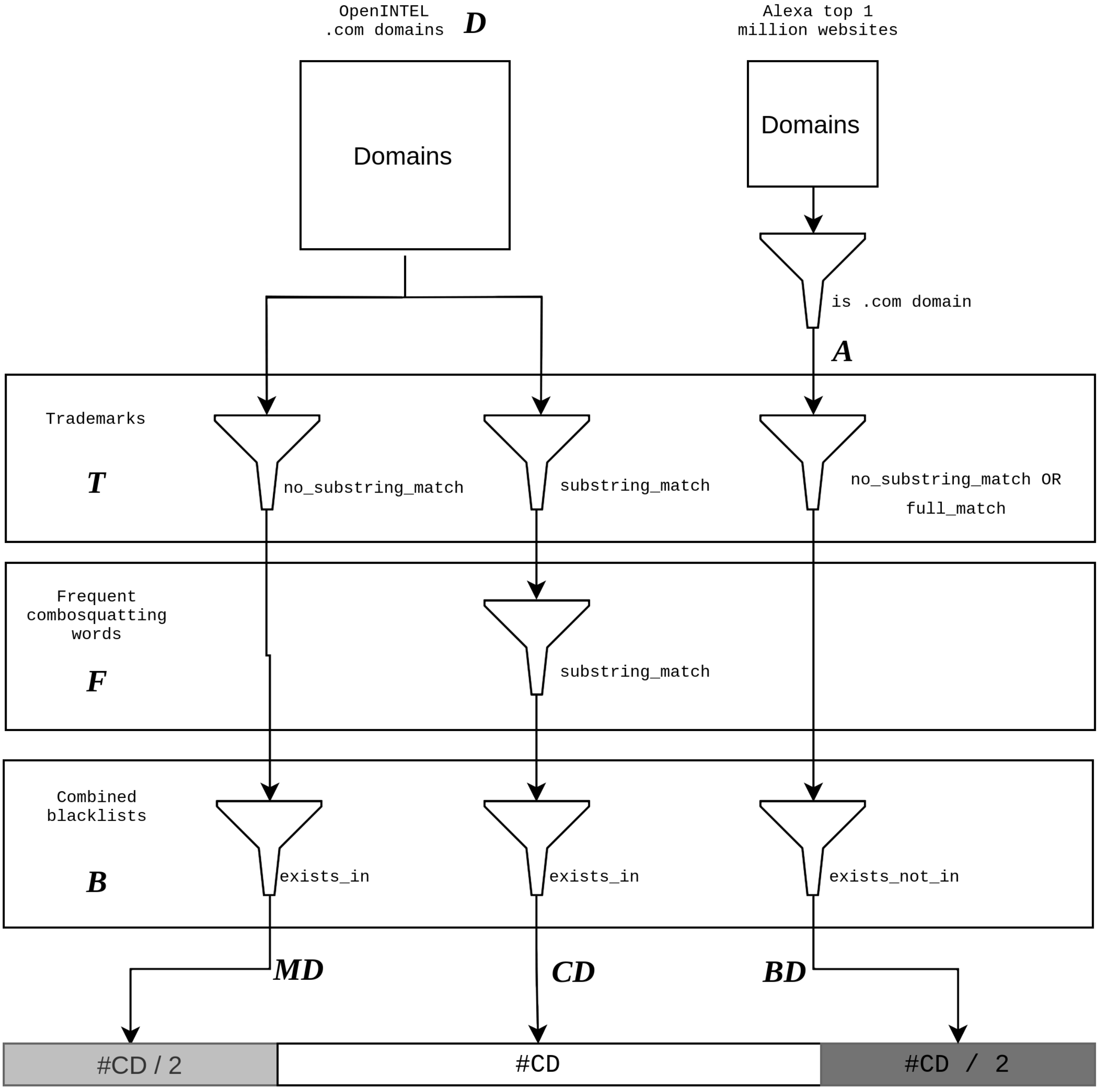


Figure 4.5: Approach to construct the ground truth

of trademarks. This set is based on the global Alexa top 500 domains, retrieved from [28]. Ambiguous domain names are excluded, as well as short names ($<$ 4 chars). This manual selection of domain names resulted in 106 unique domain names, displayed in Appendix A.1. **F** is the set of frequently used combosquatting words, as displayed in the paper by Kintis et al. [7]. **B** is the set of the blacklisted domains. This set is constructed out of the blacklists as shown in Appendix A.3. **A** is the set of `.com` domains appearing on the Alexa top 1M list [28].
Next, two string operations should be clarified. Consider a string as a sequence of characters, represented as: $S = c_1, c_2...c_n$. Then, a *substring* B is formally defined as $B = c_{1+i}...c_{m+i}$ where $0 \leq i$ and $m + i \leq n$. For example, `ent` is a valid substring of `utwente`, but `twete` is not. In the same way, a *full match* is when two strings are equal; `utwente` is a valid full match of `utwente`.

Now that the different datasets and functions are defined, the ground truth is created using a filtering process. The approach used in this filtering process is shown in Figure 4.5.
First, a list of combosquat domains (**CD**) is created; for each domain in **D**, it is checked if there exists at least one substring match with a trademark from **T** and a word from **F**. Afterwards, for each of these domains, a check is performed whether it is present on a blacklist. If this is the case, then it is added to **CD**. In order to create a proper ground truth that is usable for training & testing purposes, an equally long list of 'non-combosquatting' domains should be appended. This 'whitelist' is built out of two sources. 50% of the 'whitelist' consists of malicious domains which do not contain a trademark, called Malicious Domains (**MD**). The other 50% consists of Benign Domains (**BD**). These domains are first extracted from **A**. Then, the domains containing a trademark from **T** are excluded, except when there is a full match. The rationale behind this is that according to Kintis et al., popoular combosquat domains frequently make their way into the Alexa top list. Since the **BD** set should definitely not include any combosquat domains, these popular combosquatting domains are filtered out. For example, `youtube.com` must be included in the set of benign domains, but `youtubedownloader.com` should not. In the end, a ground truth consisting of **#CD** combosquat domains, **#CD/2** 'whitelisted' domains and **#CD/2** malicious but non-combosquat domains is present.

After the groundtruth is constructed, the ground truth needs to be enriched with features. Features are essentially datapoints used to train Machine Learning (ML) classifiers. ML is a technique that enables automated binary classification and prediction of entities and is commonly used to pick new features out of a large data pool. Following the terminology introduced by Zhauniarovich et al. [14], *internal features* are features based on DNS data, while *contextual features* are features based on additional, non-DNS data.

The main data source for the *internal* features will be historical measurements of domains extracted from the OpenINTEL project. The available variables (for example `domain_name`) must be transformed into features ready to be processed (for example `number_of_characters`, `number_of_digits`). Moreover, the data source may contain a large amount of features and since they are not equally interesting to the model, the most relevant features have to be selected. These features can be found in literature, but also arise as a result of statistical analysis on the ground truth data. A list of most valuable internal features will be the answer to this question. For the purpose of this research the internal features are further split up in *lexical features*, which are extracted from the domain name itself, and *DNS features*, all other features extracted from the OpenINTEL project. The study by Kintis et al. [7] already provides several lexical features that are commonly observed at combosquatting domains, which can be of use.

The *contextual* features are extracted from the Certificate Transparency Log, as well as the WHOIS service. The Certificate Transparency Log is an append-only Merkle-hash tree, which is used to verify the validity of a SSL/TLS certificate. Since new certificates are added on a continuous basis, Google stores information about these certificates and provides reports for all domains [29]. The WHOIS-servers provide information about a domain name through a special WHOIS query. Usually, these queries hold information about for example the registrant, registrar and nameservers. While there are standards in place for WHOIS queries & responses, it is up to the registries and registrars to determine what is inserted in the fields. While the WHOIS-information might not be fully reliable, it may still be a significant feature. Because WHOIS information is also used in relevant literature, it is initially included.

4.2.3 Treatment validation

In this phase, the a prototype of the CDM will be constructed and validated. All of this is performed on the ground truth sample dataset, functioning as a model of the intended context. The prototype is validated against the requirements that will be shown at the end of this section. The construction & validation of the prototype is itself an iterative process; by applying small changes to the prototype, the outcome of the validation model will slightly be changed.

The corresponding requirements which the generic model has to meet are specified as follows:

Functional requirements

FRQ1: The CDM should be able to classify a combosquat domain into the correct killchain phase.
FRQ2: The CDM should not include any other form of domain abuse other than combosquatting.
FRQ3: The CDM should be able to detect combosquat domains without the use of a predefined list of trademarks.

Non-functional requirements
NRQ1: The CDM should have a False Positive rate of at most 5%.
NRQ2: The CDM should have a precision rate of at least 90%.
NRQ3: If either NRQ1 and NRQ2 can be met, NRQ1 should be given a higher priority when selecting a single classifier.

The 5% and 90% percentages are set as a bare minimum. Since no reference percentages are known due to the lack of research into generic combosquatting detection models. The values will however be adjusted according to performance of the first prototype, since a generic model is new ground and realistic threshold values are not known up front.

4.2.4 Treatment implementation

In this phase, the validated prototype is placed in its intended context: a threat intelligence platform. The initial idea was to use Fox's threat intelligence platform for this purpose. In the end a simplified solution was chosen, in which the model was hosted on a Virtual Private Server, with a live connection to the OpenINTEL system. This VPS was used as an abstraction of a real threat intelligence platform.

4.2.5 Implementation evaluation

In this stage, a check was performed whether the model in context met the requirements set earlier in this section. For this research, it meant that the model should be able to distinguish newly added combosquats from newly added non-combosquats according to the requirements. This stage of the design cycle was used to answer directly sub question CTD1; whether it is possible to create a generic model to detect combosquat domains.

4.3 Domain lifecycle analysis

Since a generic model is not available for detecting combosquat domains, at this point the list of 106 trademarks as listed in Appendix Section A.1 is used; combosquat domains impersonating those trademarks are taken into account.
A starting point for answering this subquestion is a collection of blacklists, that have been scraped from 2016-07-08 until 2019-01-11. For the analysis, the blacklists up to 2018-12-31 were included to leave some data untouched for later validation usages. More information about the blacklists that have been scraped can be found in Appendix C.
The blacklisted domains functioned as the starting point for the analysis. First, out of the total domains on the blacklists, the combosquat domains were filtered according to the definition of a combosquat domain provided in Chapter 4. This means that each domain on the blacklist was checked against the five specified requirements, and those who did not match all of the requirements were dropped. The code that performed this combosquat filtering is shown in B.2.

Since in this analysis the complete combosquat domain is being researched, combosquat domains that were present on a blacklist on the first and last day of the selected blacklists were left out.

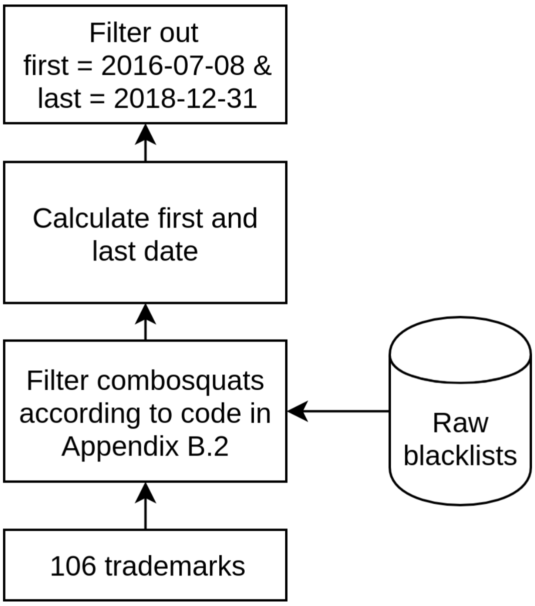

Filter out
first = 2016-07-08 &
last = 2018-12-31
Calculate first and last date
Filter combosquats according to code in Appendix B.2
Raw blacklists
106 trademarks

Figure 4.6: Blacklist filtering process

This resulted in a list of blacklisted combosquat domains, not present on the first and last day of the blacklist collection period. The domains on this list were then enriched with data from OpenINTEL. For every day, starting on 2015-02-20 (the starting day of OpenINTEL `.com` measurements) until 2018-12-31 (the fixed end date for

the analysis) the presence of the domain in OpenINTEL was checked. The check was performed by checking if any records for the domain were present in OpenINTEL. It should be noted that a domain can be registered without any record being related to it; if absolutely no records related to the domain are present in the zone file it is not measured by OpenINTEL. In this case, while the domain is registered, it is considered inactive and thus, not present in OpenINTEL. Afterwards, a list of domains and the corresponding first and last day in OpenINTEL is present.
Before continuing, only the domains of which the full lifecycle could be observed were taken into account; domains that were already active at the start of the OpenINTEL measurements, or still active at the last day of the analysis were filtered out. This means that the domains whose first date was 2015-02-20 and/or last date was 2018-12-31 were removed from the dataset.

For convenience, the OpenINTEL data combined with the blacklists is shown in Figure 4.7; the parts marked in grey are included in the dataset.

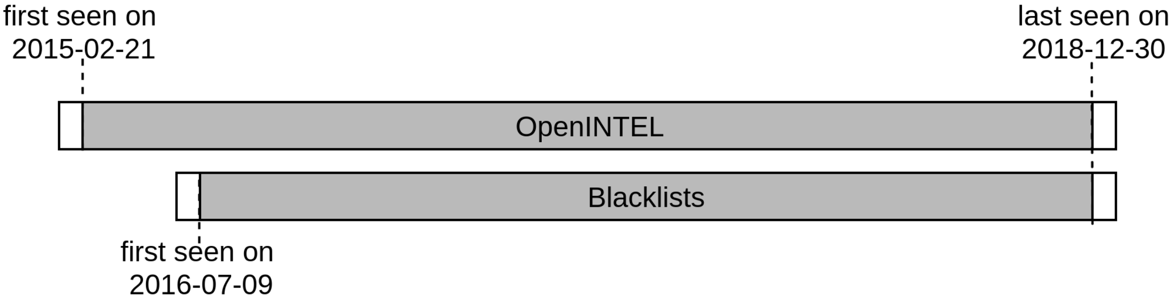


Figure 4.7: Overview of the selected dataset

At this point, a dataset containing only combosquat domains that could be observed throughout their full lifecycle are present. This dataset was used to calculate several graphs and metrics regarding the lifecycle of the domains. The results are described in Section 6.1.

4.4 Feature selection

Following the analysis in sub question CTD2, a distinction can be made between combosquat domains in the different killchain phases. More specifically, combosquat domains can be detected in two phases; the *Weaponization* and *Delivery* phase. It was also stated that domains in the *Weaponization* phase are considered benign and domains in the *Delivery* phase are considered malicious.

The dataset containing only combosquat domains that could be observed throughout their full lifecycle, produced by CTD2, is again used and enriched with the features described in Section B.1. This dataset consisted of 13693 domains. All these domains appeared in OpenINTEL, have gotten blacklisted and disappeared from either OpenINTEL or the blacklist during the measurement period.

Historical DNS & blacklist data was used in favor of creating a new dataset containing more data sources (for example HTTP), since the limited time available for this research would result in a dataset that only contains domains with a lifetime up to a few months, excluding the undetected long-living domains pointed out by Kintis et al. Note that this is only a preliminary study; it is only a starting point to see if additional measurements are feasible.

This means that the label `benign` or `malicious` has to be extracted from the blacklists alone. Since historical data is being processed, it cannot be verified whether the blacklisted domains were actually malicious. This also means that this detection can only be based on changes in the DNS records of that domain. Therefore, **a change in the DNS records of a domain is considered an indicator of an attacker's activity**. Keeping this in mind, a ground truth can be created as shown in Figure 4.8.

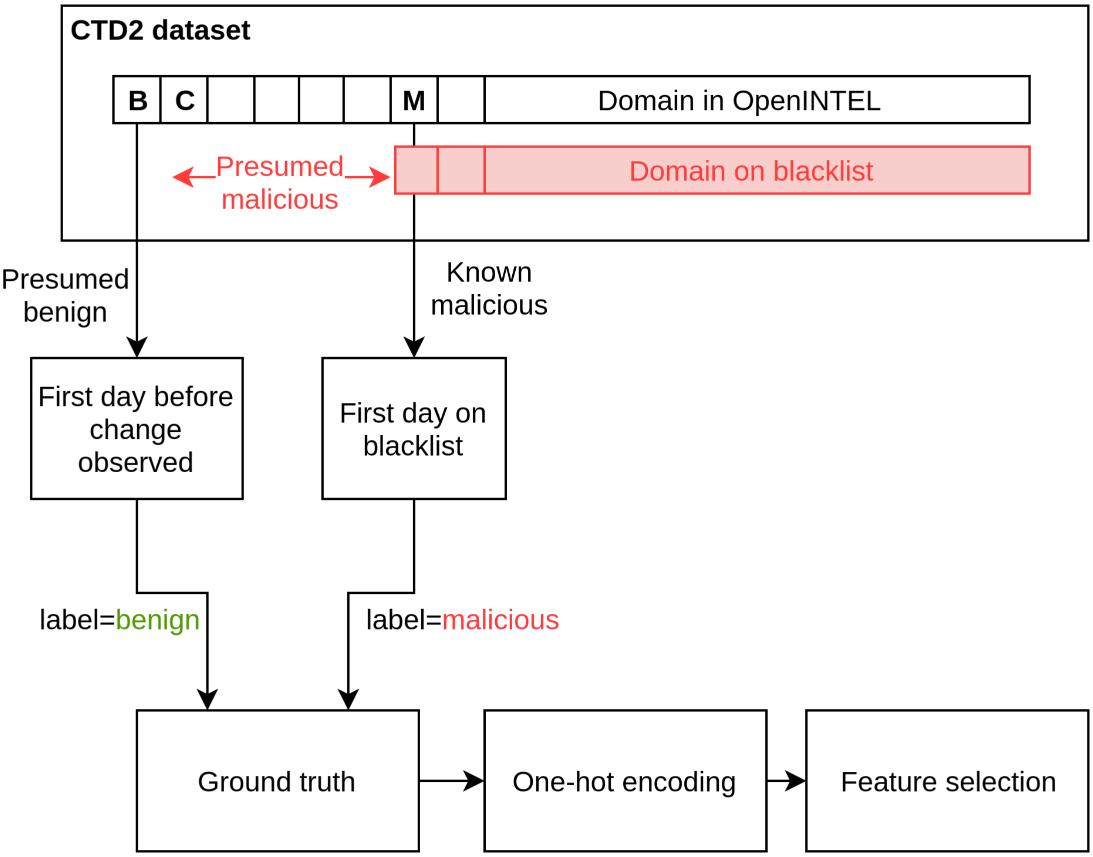


Figure 4.8: Constructing the ground truth and extracting features

Regarding the lifecycle of a combosquat domain, three important days can be marked. **M** is the day that a domain is known to be **M**alicious; this is the first day the domains appeared on a blacklist. **C** is the day when the **C**hange to the current DNS settings was observed. This means that the features on day M and C are equal. **B** is the day **B**efore the change was observed; the features of this day differ from M and C. The assumption that was previously made can not be narrowed down; **a change in the DNS records of a domain, with the new DNS records matching the DNS records at the time of blacklisting, is an indicator of a benign domain turning malicious**. The time between B and M is the time to be 'won' with earlier detection; in this period the domain is presumed to be malicious.

The features obtained from OpenINTEL on day B are then labeled as `benign`, while features obtained on day M are labeled as malicious. This was then aplied to all 13693 domains of whose B and M days, with a ground truth dataset as a result. One-hot encoding is performed to transform the data for Machine Learning, after which the most important features will be extracted by using a *DecisionTreeClassifier*.

Note that in this approach, combosquat domains that are being registered and do not change their DNS records before being blacklisted are excluded; since no change can be observed, there are no indicators based on DNS data that the domain is turning malicious. Since OpenINTEL has a resolution of one measurement per day, the DNS data that was analyzed is oblivious to quick changes in the DNS records. Therefore, if a domains has a malicious lifespan of less than one day (or even a few hours) and is pointed to a 'domain is blocked' page afterwards, the DNS records belonging to the 'domain is blocked' page are labeled `malicious` in the ground truth. This problem might be mitigated by investigating HTTP data, but for now this remains future work.

Chapter 5

Generic model design

In this chapter, the outcomes of the generic model design are displayed and discussed. First, in Section 5.1 the ground truth is shown. Afterwards, Section 5.2 provides an overview of the features that were selected and used. The generic model is then designed, validated and implemented in respectively Section 5.3, Section 5.4 and Section 5.5. Finally, the outcomes are discussed in Section 5.6.

5.1 Designing a ground truth

According to the approach described in Section 4.2, first all the `.com` domains having a substring match with both a trademark and a frequent combosquatting word were selected, which resulted in 285.327 domains. Next, the blacklists were retrieved and combined. After filtering only the `.com` domains out of the the blacklists, as displayed in Appendix Section A.3, 433.757 `.com` domains were present on the list. out of the 285.327 domains retrieved from OpenINTEL, 5274 were also present in the combined blacklist dataset. According to the approach another 2637 domains were added from the Top Alexa list, along with 2637 blacklisted domains not containing a trademark. In the end, this resulted in a labeled ground truth dataset of 10548, in which 5274 domains were labeled `combosquat` and 5274 domains were labeled `not-combosquat`.

5.2 Defining features

In this section, the features that were extracted are described. This is done according to the analogy by Zhauniarovich et al. [14]; *internal features* are features based on DNS data, while *contextual features* are features based on additional, non-DNS data.

Lexical feature	Type	Source
domainname_words	Ordinal	[7]
domainname_segments	Ordinal	[7]
domainname_popular_combosquatting_words	Ordinal	-
percentage_lms_of_total	Ordinal	[15], [18]
domainname_characters	Ordinal	[7], [18]
contains_minus_char	Boolean	[18]
contains_digit	Boolean	[18]

Table 5.1: Lexical features (total 7)

5.2.1 Defining internal features

For the purpose of this research the internal features are further split up in *lexical features*, which are extracted from the domain name itself, and *DNS features*, all other features extracted from the OpenINTEL project.

Lexical features

The lexical features that are used are displayed in Table 5.1. There are simple features (contains_digit, contains_minus_char, domainname_characters) and more complex features. The more complex features are explained below
domainname_words and domainname_segments are based on the *word segmentation algorithm*, originally proposed by [30] and also used in the paper by Kintis et al. [7]. Using this algorithm, the domain name is split into different sections based on their probability to be standalone sections. For example, 00fr-youtubevideos would result in the sections 00fr, youtube and videos. Following the classification by Kintis et al., if a section is present in one of multiple dictionaries [31]–[34] it is considered a *word*; otherwise it is considered a *segment*. In the example above, youtube and videos are considered words and 00fr is considered a segment. The two features count the number of these *words* and *segments* in a domain name.
The study of Kintis et al. furthermore provides a list of most frequent combosquatting words per category. All of these words are added to a dictionary set, and if a word matches one of these words the domainname_popular_combosquatting_words value increases by 1. This dictionary of words is displayed in Appendix Section A.2. Finally, the percentage_lms_of_total is calculated as the Longest Meaningful String of the total domain. The largest word in the domain name is selected and calculated as a percentage of the total domain name. In the example above, youtube is the longest meaningful string. Since it has length 7 and the total length of the domain name is 18, the value of percentage_lms_of_total will be 39%.

DNS feature	Type	Source
number_of_A_records	Ordinal	-
number_of_AAAA_records	Ordinal	-
number_of_NS_records	Ordinal	-
number_of_MX_records	Ordinal	-
number_of_SOA_records	Ordinal	-
number_of_CNAME_records	Ordinal	-
number_of_DNSKEY_records	Ordinal	-
number_of_TXT_records	Ordinal	-
number_of_ipv4_addresses	Ordinal	[9], [15]
list_of_ipv4_addresses	Categorial	[15]
AS_number	Categorial	[7], [9], [14]
response_name_matches	Ordinal	[9]
country_code	Categorial	[9], [15]
soa_refresh	Ordinal	-
soa_retry	Ordinal	-
soa_minimum	Ordinal	[9]

Table 5.2: DNS features (total 16)

DNS features

The DNS features are all extracted from the OpenINTEL project. In Section 3.1 background information is provided on the DNS. OpenINTEL stores many DNS record types and DNS fields[1]. Initially, all fields were taken into account when defining the features. Based on relevant literature, a selection of initial fields was made. In addition to that some experimental features were also added, such as the number of certain types of records. The fields were then converted into features, as shown in Table 5.2. It should be noted that this selection is broad; feature selection should later on filter out the less significant features so that only truly distinctive features remain.

Most of these features count occurrences of different types of RR's, IP addresses, AS numbers etc. The `response_name_matches` field indicates whether the query name & response name of a DNS query match.

5.2.2 Defining contextual features

Given the defintion in [14], contextual features are obtained when DNS data is combined with external data sources. In this research, the external data sources are

[1] https://openintel.nl/background/dictionary/

WHOIS feature	Type	Source
whois_registrar	Categorial	[15], [18]
whois_number_of_nameservers	Ordinal	-
whois_registrant_name	Categorial	-
whois_registrant_organization	Categorial	-
whois_registrant_country	Categorial	-

Table 5.3: WHOIS features (total 5)

CTL feature	Type	Source
ctl_number_of_current_certs	Ordinal	-
ctl_list_of_providers	Categorial	-

Table 5.4: CTL features (total 2)

respectively the WHOIS-servers available on the internet and the Certificate Transparency Logs.

WHOIS information

10 features were chosen from the total list of WHOIS-information response fields. Features such as `whois_zipcode` were left out, since they provide too specific information. Similarly some features were left out because they were based on time. For example, the `whois_created_day_ago` held the number of days that had passed since the domain was registered; a feature that is not useful to detect combosquatting domains at any fixed point in time. In the end 5 features were left out, leaving a total of 5 features as shown in Table 5.3.

Certificate Transparency Log

The reports as described in the approach function as data input for the features. For a given domain, the CTL outputs a list of certificates that are currently issued to a domain. Here, the certificates issued to the 2LD and the `www` 3LD are taken into account. These certificates may be issued by multiple providers, for example `Let's Encrypt Authority X3` and `TERENA SSL High Assurance CA 3`. Both the list of certificate providers, as well as the number of current certificates associated with a domain are used as features. If a certificate for a `www` 3LD is issued, the domain is stripped down to its 2LD equivalent. The used features are shown in Table 5.4.

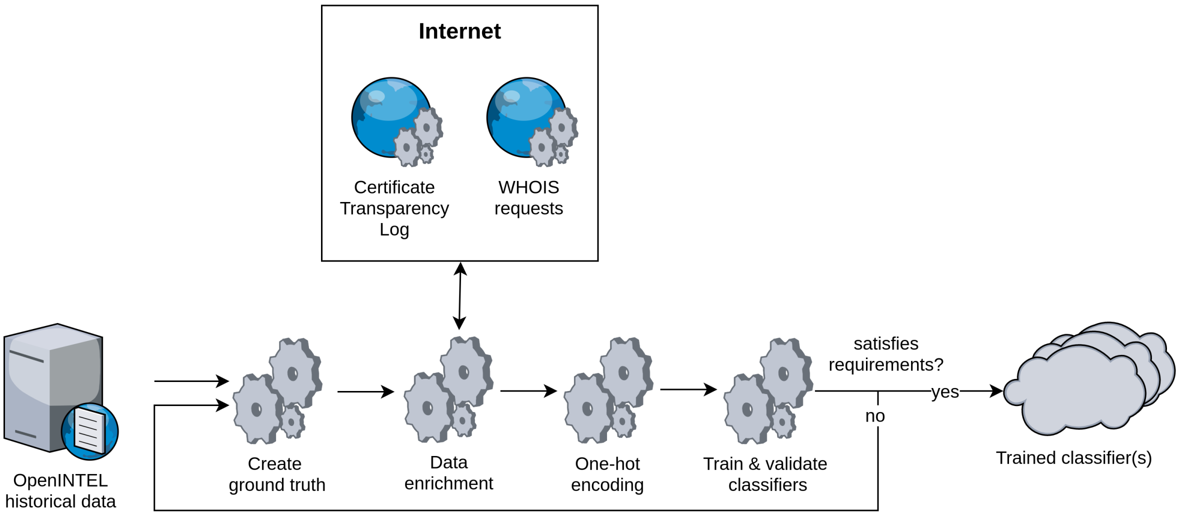


Figure 5.1: Schematical overview of the CDM training phase

5.3 Prototype construction

After constructing a ground truth and having it enriched with the **30** features described in Table 5.1, Table 5.2, Table 5.3 and Table 5.4, a prototype was constructed in order to be able to validate the performance. The Python[2] programming language was chosen as the main language because of the author's proficiency with the language and the availablity of data processing & machine learning libraries, such as `scikit-learn`[3], `pandas`[4], `numpy`[5] and `scipy`[6]. In Figure 5.1 and Figure 5.2 a schematic overview of both the training & test phase of the prototype is displayed.

As described earlier, the ground truth resulted in a list of 10548 domains Figure 4.5. Afterwards, the ground truth domains are enriched with the 32 selected features. This involves both invoking the OpenINTEL system to obtain the internal features, as well as querying two public systems to obtain the contextual features. After the data is enriched, a matrix of 10548 rows and 30 columns is present.
In the next step, several ML classifiers are trained on the enriched data. ML classifiers can only handle ordinal or binary values. Therefore, the categorial values in the ground truth need to be converted into ordinal or binary values. A technique known as *one-hot encoding* is applied on the data to fulfill this need. One-hot encoding transforms categorial data into binary data, by creating a new column for every categorial value and setting a 1 if the row contains this category, or a 0 otherwise. An

[2] https://www.python.org/
[3] http://scikit-learn.org/stable/
[4] https://pandas.pydata.org/
[5] http://www.numpy.org/
[6] https://www.scipy.org/

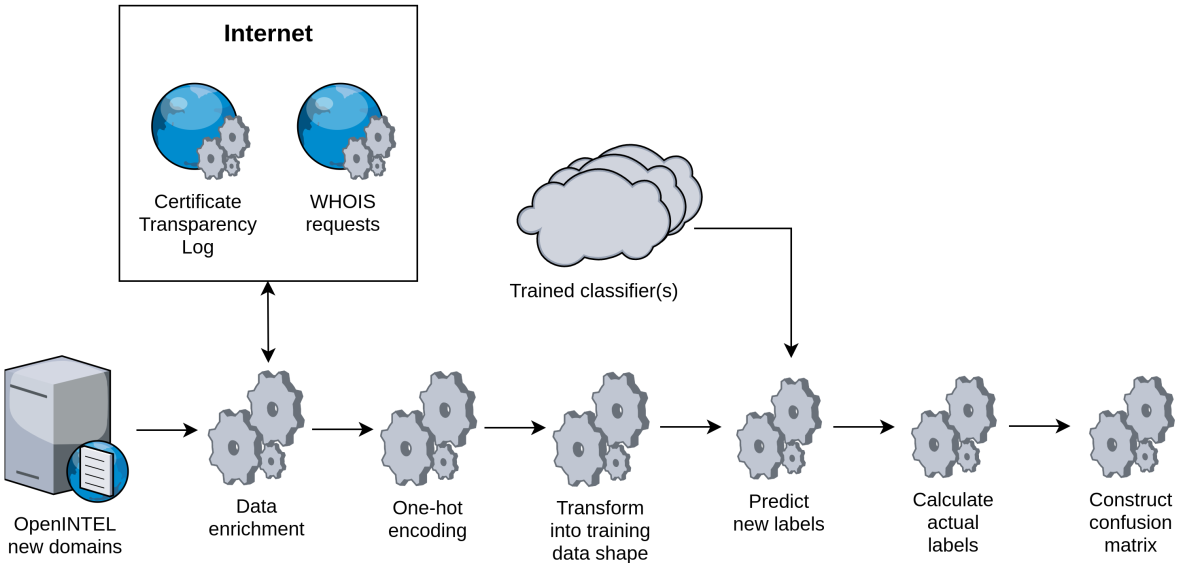


Figure 5.2: Schematical overview of the CDM test phase

domainname	list_of_AS_numbers
example1.com	[12282, 28892]
example2.com	[3853]

Table 5.5: Before one-hot encoding

example transformation is shown in Table 5.5 and Table 5.6

In this example, the list in the `list_of_AS_numbers` column is transformed into multiple columns containing only binary information. All categorial features are transformed in this way, except for the `list_of_ipv4_addresses` column. Since there are 2^{32} possible IPv4 addresses, performing one-hot encoding on this column would result in the same amount columns. This is undesirable, since each newly added column increases the memory usage during execution. Therefore, a more efficient way of using IPv4 addresses as features is proposed in the paper by Chiba et al. [35]. Following their transformation based on 'octets', this resulted in 1024 extra columns instead of the possible 2^{32} columns. After the transformations, the matrix consists of 10548 rows and 6106 columns.

For efficiency purposes, a technique called *feature selection* is often applied to a enriched dataset. During feature selection, features that are not used in the clas-

domainname	AS_number_3853	AS_number_12282	AS_number_28892
example1.com	0	1	1
example2.com	1	0	0

Table 5.6: After one-hot encoding

sification process are ommitted. In this research, feature selection is performed by training a *DecisionTreeClassifiers* with unlimited `max_depth` on the entire dataset and extracting the features that are used. Under the hood, the *DecisionTreeClassifier* tries to minimize the uncertainty based on the Gini impurity of the features. The Gini impurity is based on the probability that a domain is labeled incorrectly based on a certain feature. A high Gini value corresponds to a feature that is significant in the classification process; lower Gini values do not significantly contribute in the decision making process. By default, `sklearn`'s *DecisionTreeClassifier* constructs a tree and keeps adding new leafs to the tree until it reaches a point where it realizes that adding extra leafs is no longer beneficial to the classification process. By constructing such a tree and extracting the features that were used in the tree, not-significant features can be filtered out. Because a *DecisionTreeClassifier* is initialized with a random starting point, the selected features differ per execution round. On average, the total amount of features is reduced from 6106 to around 650, which greatly improves performance. This means that on average, the 650 selected features have the same significance in the classification process as the full set of 6106 features.

5.4 Prototype validation

According to the approach described in Section 4.2, the prototype is first validated in a model of its intended context by applying k-fold cross validation. The value of k can be arbitarily chosen, however several sources show that a value of $k = 10$ is often chosen as a default value[7,8].

An average of the scores form the 10 iterations is calculated and shown in Table 5.7. These were the first results and no classifier satisfies the requirements; a minimum precicion rate of 90% and a maximum *FP rate* of 5%. Our requirements state that if no classifier meets the requirements, the classifier with the lowest *FP rate* should be chosen. In this case, this was the *GaussianNB* classifier with a *FP rate* of 6%.

5.5 Real world validation

By retrieving all newly added domains of today, yesterday and the day before yesterday, a list of newly added domains can be obtained from OpenINTEL. The reason that the day before yesterday is included is to make sure that if a measurement error

[7] https://magoosh.com/data-science/k-fold-cross-validation/

[8] https://machinelearningmastery.com/k-fold-cross-validation/

Classifier	**FP rate** %	**Accuracy** %	**Precision** %
DecisionTreeClassifier	22	78	77
RandomForestClassifier	17	80	81
AdaBoostClassifier	18	80	80
KNeighborsClassifier	26	76	74
GaussianNB	6	51	54
BernoulliNB	26	73	73
MLPClassifier	31	67	69
SGDClassifier	76	48	48
GradientBoostingClassifier	23	81	77
ExtraTreesClassifier	20	81	79

Table 5.7: Classifiers and their scores

occured in OpenINTEL (and thus, one measurements for a certain domain is missing), it is not immediately considered a 'newly added domain'. The newly added domains were then enriched with features listed in Table 5.1, Table 5.3, Table 5.4 and Table 5.2. Afterwards, one-hot encoding was applied on the enriched data. However, the one-hot encoded newly added domains cannot be fed directly into the trained *GaussianNB* because of two reasons:
1) Categorial values present in the test data but not present in the training data are not known to the classifier.
2) Categorial values present in the training data but not present in the test data is missing.
The solution that has been chosen is to first iterate over the test data columns and remove the columns that are not present in the training data. Aftewards, when iterating over the training data columns, the columns that are not yet present in the test data are added and filled with zeroes. Finally, the prototype as described in Figure 5.2 was deployed in a real-world setting.

All newly added `.com` domains of 26-01-2019 were retrieved. This resulted in a list of 112.647 newly added domains. Subsequently, the trained *GaussianNB* classifier was used to predict the labels of each the new domains. Since the process of retrieving the WHOIS and CT log features is lengthy, the first 10.000 domains were used as a sample for the total set of 112.647 newly added domains.

Out of the total **10.000** domains, this resulted in **75** domains predicted as combosquat, and **9925** as not-combosquat. Now that the predicted labels are available, new 'actual' labels need to be calculated as well in order to calculate the confusion matrix. Since these domains were not part of the ground truth, the code that is shown in is shown in Appendix B.2 was used to calculate the actual labels. Afterwards, every predicted domain had a 'predicted' label as well as an 'actual' label assigned. These two values could then be used to create the confusion matrix, as shown in Table 5.8.

		Prediction outcome	
		cs	**b**
Actual value	**cs′**	TP 0	FN 15
	b′	FP 75	TN 9910
	Total	75	9925

Table 5.8: Confusion matrix for the real-world validation

Based on the confusion matrix, the *precision*, *FP rate* and the *accuracy* can be calculated as explained in Subsection 4.2.3; these scores indicate how well the prediction on the new unseen data performed:

$$\begin{aligned} Precision &= \frac{0}{0+75} \\ &= 0\% \end{aligned}$$

$$\begin{aligned} FP\,rate &= \frac{75}{75+9910} \\ &= 0.75\% \end{aligned}$$

$$\begin{aligned} Accuracy &= \frac{0+9910}{0+15+75+9910} \\ &= 99.1\% \end{aligned}$$

5.6 Discussion

Results indicated that the detection of new combosquat domains was not sufficient. As shown in the previous section, out of the 10.000 new `.com` domain names, 75 domain names were flagged as combosquatting domains. At first, the frequency of the detected domains is in line with distribution in the ground truth. 75 out of 10.000 means that 0.75% was flagged as combosquat. In the ground truth 285.327 domains

Domain name
sevenoakscommhomes.com
mybackyardrelaxation.com
smarthomegadgetguru.com
anthonyandmikayla.com
silvercloudinvestments.com
retroelectromotors.com
mizikmalemusic.com

Table 5.9: Selection of domain names that were falsely labeled as combosquat (False Positives)

were used as combosquat domains on a total dataset of 137M `.com` domains; here they make up $\frac{285327}{137000000} * 100 = 0.2\%$ of the total domains.
The *FP rate* and *accuracy* scores seem pretty satisfying at first. However, the *GaussianNB* classifier was unable to detect even one of the 75 actual combosquats in the 10.000 newly added domains. The *precision* score is therefore 0%, which is obviously not sufficient. When looking more in detail into the trained *GaussianNB* classifier, it becomes clear why this happened.

A selection of the domain names falsely flagged as combosquat is shown in Table 5.9. On the other hand, Table 5.10 shows the combosquat domains that were missed by the classifier. Since the GaussianNB classifier is trained based on the features selected by the *DecisionTreeClassifier* shown in Figure 5.3. the tree can be used to explain the results. For convenience, the `max_depth` has been set to 3, since this provides insight in the features with the highest Gini impurity value. The most significant feature is *domainname_characters*, which represents the total length of the domainname. This means that the prediction is greatly based on this feature. Furthermore, it can be observed that the DNS features are not as important as the lexical features; features like *percentage_lms_of_total* and *number_of_minus_chars* are present, while only one DNS feature is present (*response_name_matches*). This means that combosquat domains cannot easily be fingerprinted by distinct DNS entries. The last observation is that the classifier seems unable to distinguish trademarks from regular words. When looking at Table 5.9 and Table 5.10, the classifier did not learn that 'samsung' is a trademark and 'backyard' is not.

Domain name
samsungblockchaincore.com
facebookdekatsuyaku.com
linkedinclassroom.com
shopifywebsitedesignerbuilder.com
godaddyholdings.com
ihategodaddy.com
mywalmartcoupons.com

Table 5.10: Selection of domain names that were falsely labeled as not-combosquat (False Negatives)

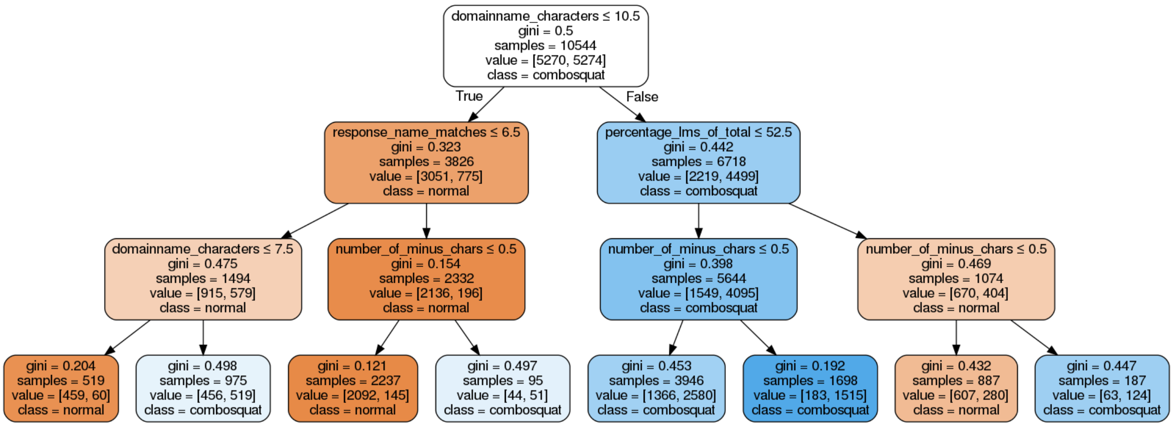


Figure 5.3: Decision tree of depth 3

Chapter 6

Domain lifecycle analysis & feature feature selection

This chapter will display and discuss the results of subquestions CTD2 and CTD3 in respectively Section 6.1 and Section 6.2.

6.1 Defining the killchain phases

According to the approach described in Section 4.3, the filtering was performed. 13693 blacklisted domains were available to be analyzed. In this stage a list of 106 trademarks was used and using a simple grouping function, the trademarks that were targeted most frequently could be analyzed. The top 10 of targeted trademarks is shown in Table 6.1.

Rank	Trademark	Number of domains
1	Apple	8751
2	Paypal	1241
3	Microsoft	711
4	Netflix	592
5	Facebook	372
6	Amazon	323
7	Instagram	265
8	Google	213
9	Whatsapp	166
10	Wellsfargo	115

Table 6.1: Top 10 most targeted trademarks

The full list of trademarks and the corresponding frequencies is displayed in Section D.2. It can be observed that 'Apple' takes up the majority of malicious combosquat domains; as much as 8751 out of the 13693 domains. Following was 'Paypal' with 1241 hits. The top 5 is completed with 'Microsoft', 'Netflix' and 'Facebook', respectively with 711, 592 and 372 combosquat domains. Next, OpenINTEL was used to add more context to the domains. The next filtering process was applied, leaving 12115 domains for the analysis. These domains were then used to calculate four metrics: the total days of a domain in OpenINTEL, the days of a domain in OpenINTEL before it got blacklisted, the amount of domains that were still present in OpenINTEL after being removed from a blacklist and finally, the amount of days the domains was present in OpenINTEL after being blacklisted.

The first graph that could be created was the total days of a domain in OpenINTEL, as seen in Figure 6.1.

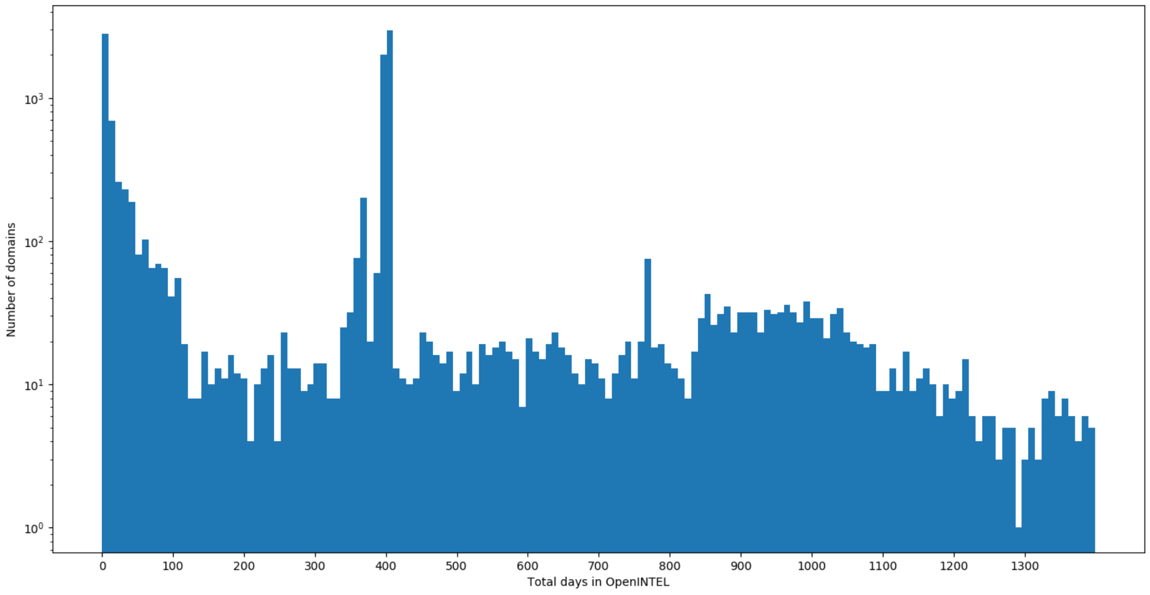
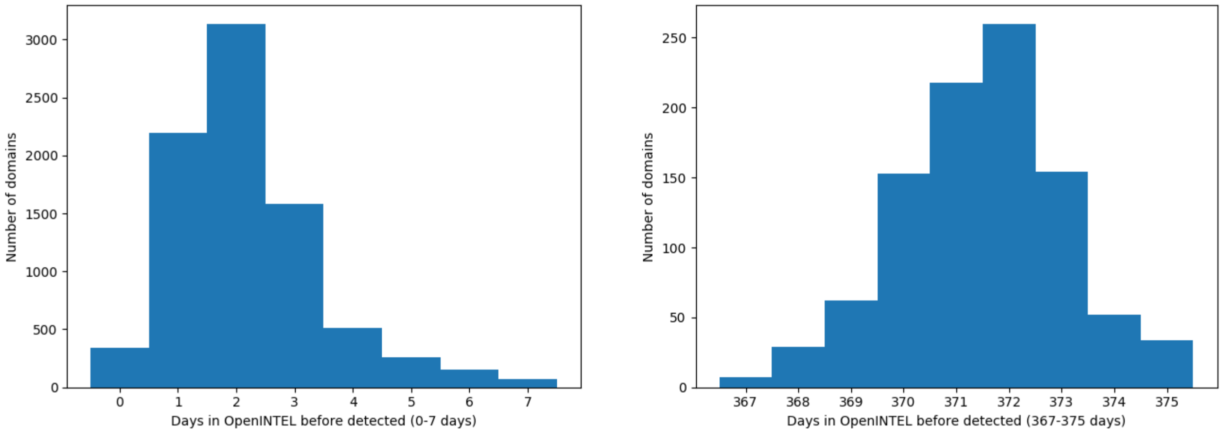


Figure 6.1: The total number of days a combosquat domain is present in OpenINTEL, with the y-axis in logarithmic scale.

When calculating and plotting the total days before a domain is detected two large peaks could be observed; the first peak lies at 2 days and the second peak lies at 372 days. These peaks are shown in Figure 6.2. The graph that shows all data can be found in Appendix D.1.



(a) First peak

(b) Second peak

Figure 6.2: A graph showing the first and second peak from the graph showed in Appendix D.1

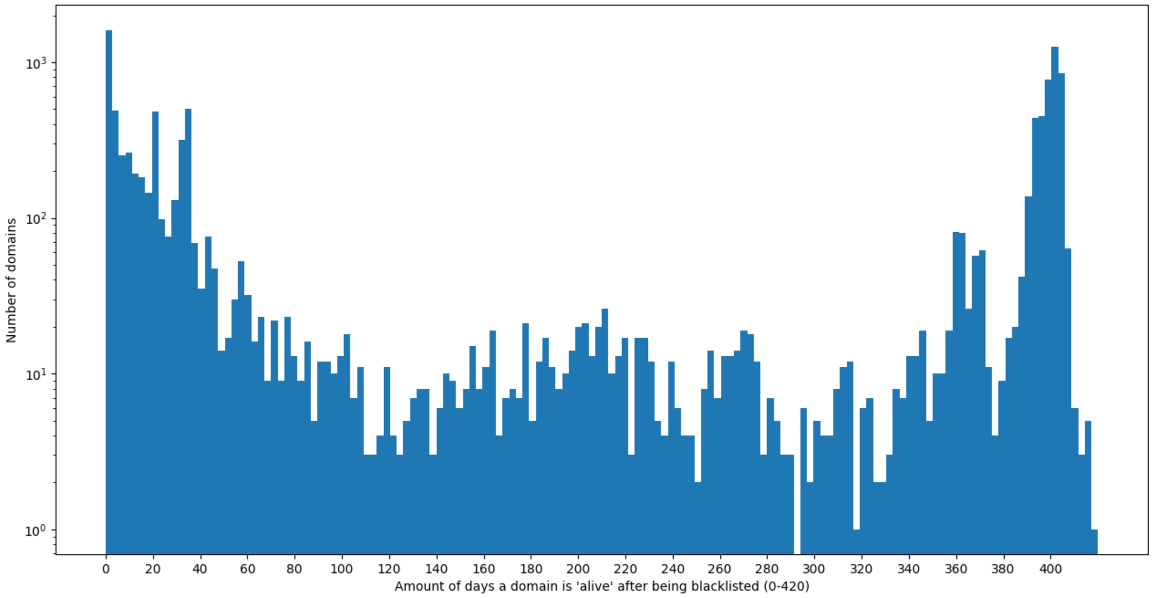


Figure 6.3: The number of days a domain is present in OpenINTEL after being detected, with the y-axis in logarithmic scale.

Next, it was calculated how many domains were still present in OpenINTEl after being removed from a blacklist. 86.7% of the domains were not present anymore, and 13.3% were still present in OpenINTEL after being removed from a blacklist. In this case, the reputation of the domain has improved in such a way that it got removed from the blacklist. Finally, Figure 6.3 shows the amount of days a domain is present in OpenINTEL after being detected.

Given the fact that most malicious combosquat domains are only active and detected after a few days, in order to define the killchain phases the domains needed to be measured more frequently. Since the historical DNS data was not suitable for this purpose, a additional small dataset consisting of active HTTP data was created using the Certstream[1] library which provide a real-time feed of newly signed SSL certificates.Since combosquat domains are used for phishing purposes, and phishing websites more frequently use SSL certificates to fake their legitimacy to users, it is expected that also SSL certificates for combosquat domains could be observed. A few combosquat domains were actively queried for their HTTP response during their lifecycle and the observations are summarized in Figure 6.4. This lifecycle sometimes only covered a few hours instead of a few days, which made it impossible for OpenINTEL to detect even the DNS changes. Note that this (extra) dataset was primarily used for exploring the possible stages the short-living domain could reside in. Although no conclusions can be based on this small dataset, future work could focus on the detection of malicious short-living combosquat domains primarily based on HTTP data.

6.1.1 Discussion

As a basis for the discussion, the domain lifecycle diagram provided by the ICANN[2] is provided in Figure 6.5.

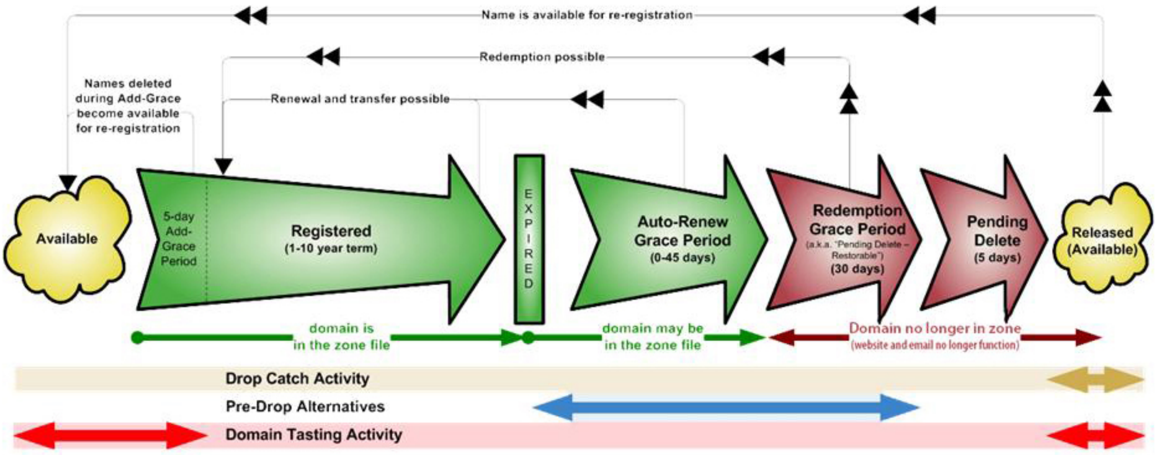

Name is available for re-registration
Redemption possible
Renewal and transfer possible
Names deleted during Add-Grace become available for re-registration
Available
5-day Add-Grace Period
Registered (1-10 year term)
EXPIRED
Auto-Renew Grace Period (0-45 days)
Redemption Grace Period (a.k.a. "Pending Delete – Restorable") (30 days)
Pending Delete (5 days)
Released (Available)
domain is in the zone file
domain may be in the zone file
Domain no longer in zone (website and email no longer function)
Drop Catch Activity
Pre-Drop Alternatives
Domain Tasting Activity

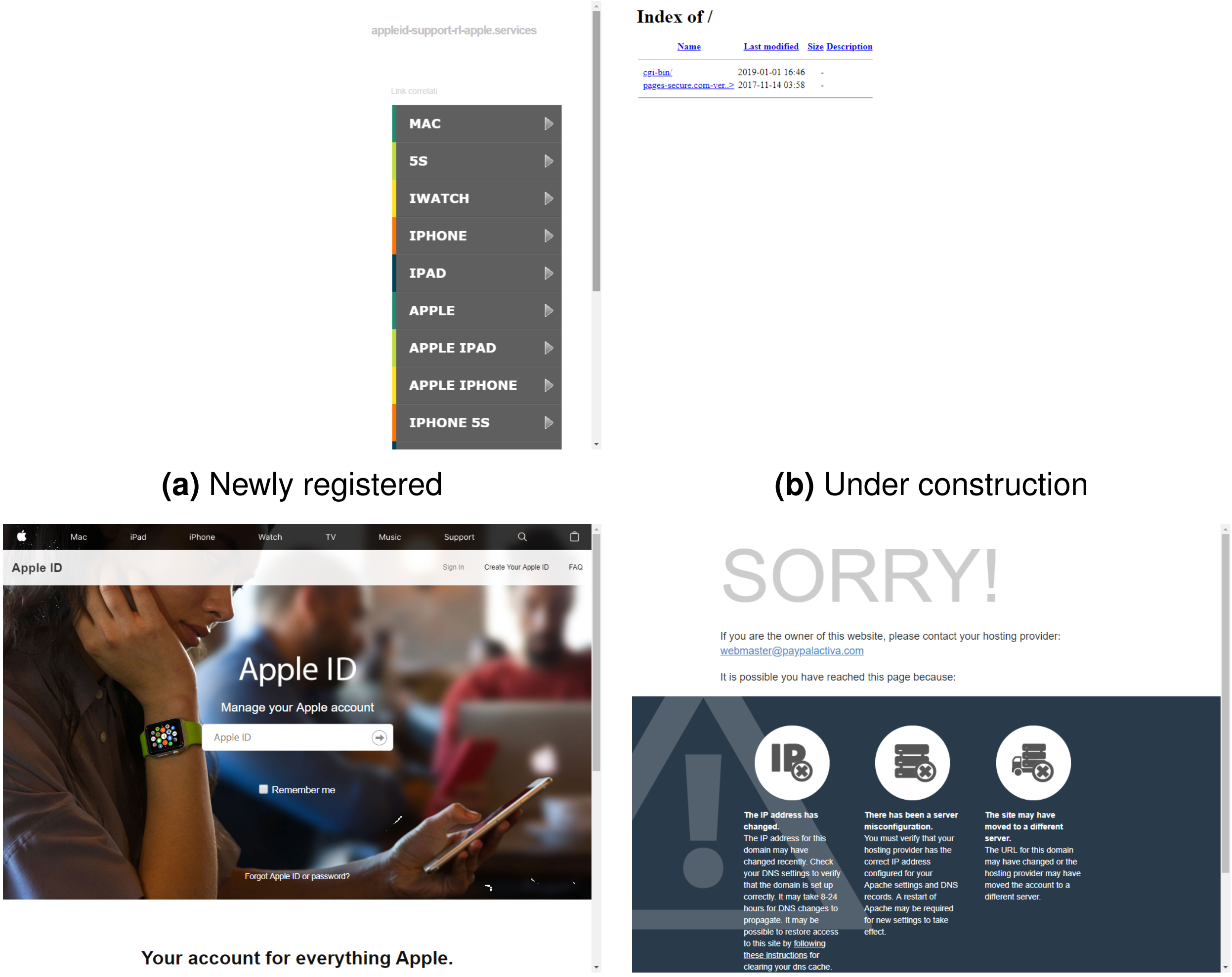
Figure 6.5: ICANN Domain lifecycle diagram

First, all the graphs that were created during the analysis are discussed.

Figure 6.1 shows three interesting peaks. The first peak (around 0-7 days) are short-living domains; domains that are being registered and disappear within a few days.

[1] https://certstream.calidog.io

[2] Internet Corporation for Assigned Names and Numbers



Index of /

Name	Last modified	Size	Description
cgi-bin/	2019-01-01 16:46	-	
pages-secure.com-ver..>	2017-11-14 03:58	-	

(a) Newly registered **(b)** Under construction

Your account for everything Apple.

SORRY!

If you are the owner of this website, please contact your hosting provider: webmaster@paypalactiva.com

It is possible you have reached this page because:

The IP address has changed. The IP address for this domain may have changed recently. Check your DNS settings to verify that the domain is set up correctly. It may take 8-24 hours for DNS changes to propagate. It may be possible to restore access to this site by following these instructions for clearing your dns cache.

There has been a server misconfiguration. You must verify that your hosting provider has the correct IP address configured for your Apache settings and DNS records. A restart of Apache may be required for new settings to take effect.

The site may have moved to a different server. The URL for this domain may have changed or the hosting provider may have moved the account to a different server.

(c) Up & running **(d)** Inactive

Figure 6.4: Screenshots of a short-living malicious combosquat domain

The second (small) peak is around 365, which correlated to a registration period of 1 year. However, the third peak lies around 405 days. This could be explained by summing the registration period (365 days) and the Auto-Renew Grace Period offered by a lot of registrars (40 days). In this Auto-Renew Grace Period, the domains is put 'on hold' to give the registrant some extra time to decide on continuing the registration or not, while the domain remains present in the zone file (see Figure 6.5. This means that no indication of an attackers' incentive to reuse or 'clean' a malicious domain can be seen; it seems that they just register combosquat domains in bulk and let them expire.

Figure 6.2 shows the two peaks in the total amount of days before domains are detected. It is interesting to see that the majority of detected domains are detected within ~ten days. Contrary to what Kintis et al. reported, the findings suggest that these types of domains are quickly blacklisted, often within necten days instead of the long-living domains reported by Kintis et. al. Figure 6.2b shows that the peak lies at 2 days. The second peak lies around 372. This peak has a similar shape as the peak around 2, however it has moved 370 days up front. This could be explained as a one-year registration period of 365 days + five additional days. It is assumed that these five days correspond to the Add-Grace Period (see Figure 6.5) of the registrar; within five days after registering a domain, the registration may be reversed by the registrar which in turn receives a full refund of the registry. After the registration is reversed, the domain becomes immediately available for re-registration. An explanation could be that the registration is reversed after five days, someone else registers that particular domain and it lives for another 365 days. Then, after 370 days, the domain expires and is drop-catched by a malicious registrant. This malicious registrant then uses the domain for malicious purposes, after which it gets blacklisted after an average 2 days.

Next are the amount of domains that are still present in OpenINTEL after being removed from a blacklist. A majority of 86.7% of the domains is not present in OpenINTEL after it has been removed from a blacklist, again indicating that there is no (economic) incentive for attackers to reuse the abused domains.

Lastly, Figure 6.3 shows the amount of days domains are alive after being blacklisted. This graph can be explained by taking Figure 6.1 and extracting the 2 days taken from Figure 6.2b in Figure 6.2. Therefore, this graph confirms the correlation between the different graphs.

Figure 6.4 shows the four phases that were frequently observed when actively obtaining HTTP information about short-living combosquat domains. Figure Figure 6.4a shows the domain being registered and parked; domains are not actively misused in this stage. Figure 6.4b shows the first signs of activity; a webserver is set-up and a folder containing malicious content is uploaded. Usually after this stage, it does

not take long before Figure 6.4c can be observed; the combosquat domains is up & running and its intentions are malicious. Finally, after some time the domain gets blacklisted / blocked / banned and the domain resolves to an error page, as shown in Figure 6.4d. Note that in this example, the domains is used for a phishing page, while several other types of abuse can also be present. Since users need to be directed to the phishing page, it is publicly promoted via email and/or the World Wide Web which possibly results in the fast detection.
Considering all of the above, OpenINTEL is not able to detect the really short-living combosquat domains (with a lifecycle of a few hours). Active HTTP would be needed for this purpose, which is unfortunately unavailable in this research. However, when examining Figure 6.2, some domains manage to remain undetected from a few days to a few hundred days. OpenINTEL is suitable to detect changes in DNS records related to a domain turning malicious (for example a change in `A`, `AAAA` or `MX` records).

The killchain is used to define the different stages a combosquat domain can reside in. Based on the discussed figures and lifecycle phases, Table 6.2 was created. This table shows the first killchain stages in relation to combosquat domains, and whether it is possible to detect the transitions between killchain phases using either OpenINTEL or active HTTP measurements.

		Detection with	
Killchain phase	**Combosquat appliance**	**OpenINTEL**	**HTTP**
Reconnaissance	Attacker checks free domains	False	False
Weaponization	Attacker registers a combosquat domain	True	False
Delivery	Attakcker configures malicious webserver	True*	True
Exploitation	Attacker directs users to domain	False	False
Installation & later	User interacting with malicious page	False	True

Table 6.2: Killchain phases in combosquat perspective

The asterisk means that this is True when a change in DNS records is present. Usually, when registering a domain the registrar sets the DNS records to the registrar's defaults. Therefore, when an attacker for example changes the `A` or `AAAA` records to point to the malicious webserver, this can be observed in OpenINTEL. However, if the DNS records point to the malicious webserver from the start, OpenINTEL is not able to detect this change. Furthermore, if this change is made within one day after registration, OpenINTEL is not able to detect this because it only measures once per day.

6.2 Feature selection

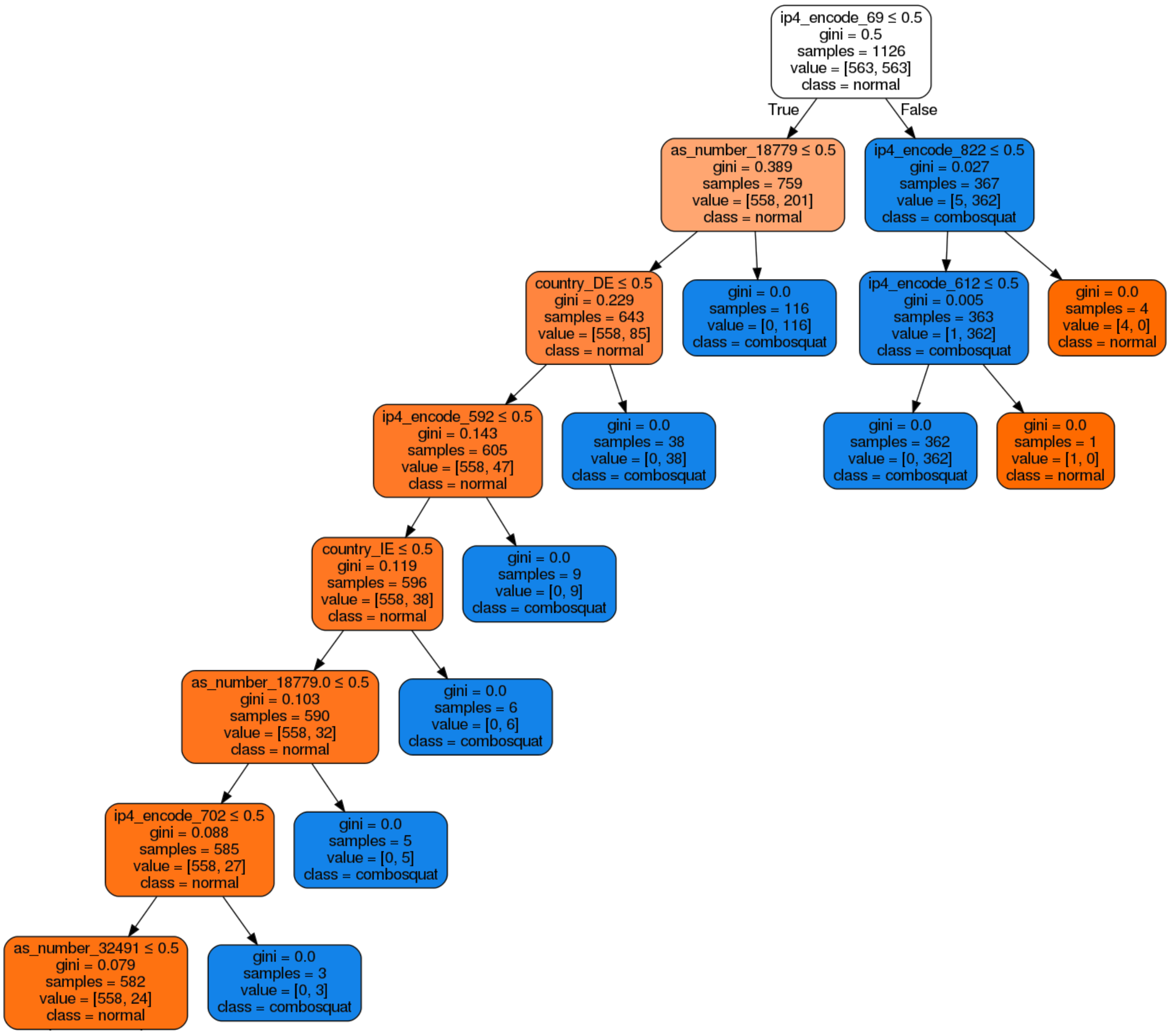


Figure 6.6: Decision tree with the selected features

The ground truth has been created in the manner described in Section 4.4.M was defined as the day the domain first appeared on a blacklist, C as the day the change to the malicious DNS records was observed and B was the day before that change. The domains and the corresponding features from OpenINTEL were stored. For every domain, all dataframes were concatenated in chronological order. A hash was calculated over all dataframes and thus, over all features. First, the hash of day $M-1$ checked against the hash of M. If the hashed would differ, $M-1$ would be labeled as B. If the hashed matched, the iteration would continue until a day $M-n$ would arise where the hash was not maching. The maximum value of n was set to 100 to increase the performance of the process; this means that a change in the DNS records had to be present in the 100 days before domain got blacklisted, which covers the first peak displayed in Figure 6.2 (a).

This resulted in a total of 5282 rows containing 'benign' and 'malicious' rows. This meant that 2641 domains were present that had the same features on C and M, and where the features of both B and M were available. This means that out of the 12115 domains, in 9474 cases no change was observed in the last 100 days, the features of C and M did not match, or the features of day B were not available.
The 5282 rows were then one-hot encoded, resulting in a ground truth dataframe with 5282 rows and 1533 columns. A *DecisionTreeClassifier* was then trained on the ground truth in the same way as described in Section 5.3. On average, this resulted in a decrease of columns from 1533 to 356. Finally, in Figure 6.6 the actual *DecisionTreeClassifier* that was used for the feature selection is shown.

6.2.1 Discussion

The top features outlined in Figure 6.6 are almost all based on the IP-addresses and AS numbers that correspond to the domains. The most significant features is *ipv_encode_69*, and the second-most significant features are *as_number_18779* and *ipv4_encode_822*. This means that no unique signature is observed for combosquat domains turning malicious except for a change in the IP address or AS range. This means that attackers change their DNS entries in order to point the domains to certain malicious IP addresses and AS ranges.

Chapter 7

Detection model design & validation

In this chapter, the results regarding the main research question are described. In Section 7.1 the model is designed and validated. Section 7.2 describes how the model was placed in the intended context and how the implementation was evaluated. Section 7.2 discusses the final outcomes of the combosquat detection model.

7.1 Treatment design and validation

The ground truth that was designed while answering CTD3 can again be used. This ground truth was used to extract the features as described in Figure 4.8.
A total of 10 classifiers were selected based on their usage in related work. For each of the 10 classifier, 10-fold cross validation was performed on the training i.e ground truth data. After training and validating the classifiers, the average scores of the 10 classifiers can be found in Table 7.1

Since the *RandomForestClassifier* has the highest *precision* and the lowest *FP rate*, this classifier is picked to be used in the real-world implementation.

7.2 Treatment implementation and evaluation

The trained *RandomForestClassifier* was used to detect malicious combosquat domains. The approach as described in Figure 4.3 was applied on the total set of combosquat domains on 30-07-2017. This date was chosen because it is in the middle of the total timespan of the dataset.
All combosquat domains on the date were extracted from OpenINTEL, resulting in a list of `173189` combosquat domains. For every domain in this list, the features were retrieved, one-hot encoding was applied and the columns of this test set were shaped to the training columns that were used by the trained *RandomForestClassifier*.

Classifier	**FP rate** %	**Accuracy** %	**Precision** %
DecisionTreeClassifier	24	70	72
RandomForestClassifier	22	70	73
AdaBoostClassifier	25	70	71
KNeighborsClassifier	25	68	70
GaussianNB	83	52	51
BernoulliNB	37	65	64
MLPClassifier	47	61	62
SGDClassifier	60	46	46
GradientBoostingClassifier	23	67	70
ExtraTreesClassifier	24	70	73

Table 7.1: Classifiers and their scores

After the predictions were made, the actual labels were also calculated for every domain and consecutively a confusion matrix was constructed.

		Prediction outcome	
		m	**b**
Actual value	**m'**	TP 2227	FN 5719
	b'	FP 30422	TN 134810
	Total	32649	140529

Table 7.2: Confusion matrix for the real-world validation

Based on the confusion matrix, the *precision*, *FP rate* and the *accuracy* can be calculated as explained in Subsection 4.2.3:

$$\begin{aligned} Precision &= \frac{2227}{2227 + 30422} \\ &= 6.8\% \end{aligned}$$

$$\begin{aligned} FP\,rate &= \frac{30422}{30422 + 134810} \\ &= 18.41\% \end{aligned}$$

$$\begin{aligned} Accuracy &= \frac{2227 + 134810}{2227 + 5719 + 30422 + 134810} \\ &= 79.13\% \end{aligned}$$

The scores are insufficient to be used for efficient detection of combosquat domains turning malicious. The *Precision* score is too low to be of practical use, as well as the high *FP rate*.

Chapter 8

Conclusion

The chapter provides answers for the research questions that were formulated in Section 2.4. In Section 8.1 the conclusions from the subquestion are listed, and the main research question is anwered. Finally, in Section 8.2 the recommendations for future work are discussed.

8.1 Conclusion

Based on the literature study there was a hypothesis that, with enough data in the form of active DNS measurements, a generic combosquat detection model could be designed that was able to warn customers in an early stage. To test the hypothesis, a research question was defined, that was further divided into multiple subquestions: *How to develop an active combosquatting detection model that meets the requirements set in Section 2.3, so that businesses can be warned in an earlier stage within a threat intelligence platform?*

The first sub question was about investigating whether a generic model was possible for the detection of combosquat domains using active DNS measurements. The first observation was that the key feature is `domainname_characters`, thus the total length of the domainname. Since the addition of words to a trademarks often results in a lengthy domain name, the classifier was mainly trained on this one feature. While this holds for combosquat domains, this also holds for other general purpose domain names, such as *smarthomegadgetguru.com*, not an unique domain name in itself. The second observation was that lexical features (based on the domain name itself) were more important than features selected from DNS data. Combosquatting domains do not significantly differ from benign domains and/or other malicous domains in a distinctive manner regarding the selected DNS features. The third observation was that based on the data & selected features, no clear distinction could

be made between a regular word and a trademark. Apple illustrates this problem clearly; it is both a well-known trademark, but also a regular word frequently used in domain names. These trademarks make it difficult for an automated approach to distinguish domains where Apple is used as a trademark and where not.
Through the observations it can be concluded that it is extremely difficult to construct a generic model for detecting combosquat domains without a predefined list of trademarks.

To answer the second subquestion, a temporal analysis was performed on the combined OpenINTEL & blacklist data. This resulted into new insights regarding combosquat domain usage by malicious users and the (economic) incentives behind it. These findings were then translated into actions that malicious users could perform in the separate killchain stages.
A combosquat domain can reside in all of the killchain phases, however the first five phases (Reconnaissance, Weaponization, Delivery, Exploitation and Installation) are most useful for detection purposes. Detection based on DNS data is under certain circumstances possible between the Weaponization & Delivery phase. Short-living domains are more difficult to detect with OpenINTEL because of the measurement frequency of one time per day; domains that are registered, turn malicious and are abandoned within a day remain undetected in this way. Detection of combosquat domains turning malicious based on OpenINTEL data should therefore be focused on domains living longer than 1 day. If one would want to detect also the short-living domains, other data sources such as active HTTP measurements should be added. Combosquat domains in the Weaponization phase are considered benign, while domains in the Delivery phase are considered malicious.

The last subquestion is answered by looking at the features selected in Section 6.2. These features were related to the IP-addresses and AS numbers of the domain. So based on changes in the IP-addresses and AS numbers of a domain, a benign combosquat domain turning malicious can be identified.

The results show that the detection of combosquat domains turning malicious based on active DNS measurements is not sufficient, when considering acceptable scores for the *False Positive rate*(5%) and the *precision*(90%). The *False Positive rate* of 18.41% is too high to be of practical use, similar to the low *precision* rate of 6.8%. Therefore, the first observation is that based on only active DNS measurements and given the features that were used, it is not possible to detect when a combosquat domain transforms from an 'inactive' state to an 'active' state. The second observation is that when looking at the features that define a legitimate domain turning malicious, is that no unique indicator of change can be observed. As can be seen in the decision tree in Figure 6.6, the most relevant features consist of IP-addresses and AS numbers. This method of detection has been widely researched and is also actively being used in practice; IP-addresses and AS numbers are getting rated and blacklisted regularly. Thus, the classifier that uses IP-addresses and AS numbers is not considered 'new' and it matches the current detection methods.

8.2 Future work & recommendations

One of the main conclusions is that the detection of combosquat domains turning malicious is not sufficient when it is only based on active DNS measurements. This obviously does not mean that this is not possible at all. By using & combining other data sources, for example HTTP data, detection may be possible. Although a lot of research is also done in this field, it has not been applied to combosquat domains specifically.

It should also be noted that registrars at this point are not succeeding in declining combosquat registrations. One the one hand, it is because this is very hard to do, as this research shows. Another aspect is that it is against the nature of a registrant selling domain names is their core business and generates revenue. Limiting the numer of registrations implies less revenue. Although currently there is legislation in place to prevent combosquatting abuse, it is obviously not enforced properly. An option might be to make the domain registration procedure more restricted by law; registrars may ask for more information about the registrants in order to verify the actual identity, such as passport numbers, legal entity numbers and more. Since every registrar is only responsible for a selection of TLD's, this may even be different per country.

To conclude, the detection results from this thesis can be used for future work, as an effective way of detecting combosquat domains is strongly desired.

Part II

Communication of Incident Severity between Customers and Analysts in a SOC

Chapter 9

Introduction

This chapter provides an introduction into the subject & the research. In Section 9.1 the motivation for this research is outlined, and in Section 9.2 the research questions are listened.

9.1 Motivation

Fox-IT offers many products and services to their customers, two of them being a Managed Security Service (MSS) and a Managed Intelligence Service (MIS). The MSS consists of network and endpoint monitoring within a company's internal network, while MIS consists of gathering open-source intelligence that is out on the internet. Originally, the Security Operations Center (SOC) only handled the incidents from the MSS. For this purpose, a platform called the Cyber Threat Management platform (CTMp) was designed and implemented. Originally, the MIS used its own interface, but since January 2019 the MIS incidents are also processed by the SOC, thus handled through the CTMp system. On a 24/7 basis, SOC-analysts process incidents that pop up and take several steps to classify and possibly report these incidents to the customer.

The crucial step in this process is when the triage for the incidents is being done; the incident is manually classified as either a False Positive (the incident is not actually an incident) or a True Positive (the incident is actually malicious). When it is a True Positive, an impact label has to be assigned to the incident. Currently this assignment of an impact label is mainly based on the SOC-analysts' knowledge & experience, with no fixed decision making process. This implies that different analysts escalate cases in a different way, which in turn leads to inconsistent reporting towards the customer. An example would be a coinminer[1]; some analysts would assign a `High risk` label to the potential impact because a trojan horse is actually

[1] A coinminer is a trojan horse that uses the compromised computer for cryptocurrency mining.

installed on a system, while other analysts would assign a `Low risk` label since the trojan is not directly harming the user.

After this impact classification, the threat category to which the incident belongs is also set. Currently, the choice of labels for this category is limited; it consists of generic labels such as *Compromised Asset* or *Misuse of systems*. Again, there is no formal consensus on labeling the data which leads to inconsistent reporting to the customer; the coinminer discussed earlier could be placed in either category. This results in a need to evaluate the potential impact labels & threat categories, as well as a standardized decision making process to be able to escalate cases more consistent.

Another major problem is that the labels are currently not validated with the customer; Fox-IT determines whether an incident is classified as `High risk` or `Low risk`. Of course, customers can request exceptions to the default policy, but by default this policy is provided by Fox-IT. Therefore Fox-IT wants to know whether the assigned impact labels differ from the impact labels the customer would have assigned to it. In other words; would the customer classify incidents with the same potential impact as Fox-IT does? Or would customers classify some incidents more severe than Fox-IT? These preferences are discussed with the customer during Quarterly Meetings, however this is on an individual basis and does not influence the overall default model.

The aforementioned reasons increase the urge to create a fixed decision making process validated from a customer perspective, that can be used in practice by the SOC analysts to improved the consistency of the decision making process. Creating such a decision making process starts by analyzing the current situation and observing the differences that exist between customers and analysts, as well as taking into account the cost of incidents.

To conclude, the scope of this research is not to design a complex new risk framework that covers all potential threats for all possible customers based on a lot of assumptions. The objective also also not to design a practical tool that needs to be validated. The main objective is to create an advice for Fox-IT by answering several research questions. Since this is an academic study the results will not only be beneficial to Fox-IT's SOC, but could be of interest to any other SOC. Ultimately this advice can be transformed into a decision making tool, but as stated this is not within the scope of this research.

9.2 Research Questions

RQ: Which actions should Fox-IT take in order to align their incident reporting decision making with the analysts & customers preferences?

SRQ1: How do the current processes & preferences regarding incident reporting look like?

SRQ1.1: Which dimensions are relevant during the escalation analysis?
SRQ1.2: Which impact labels can be used to cover all levels of severity?
SRQ1.3: Which threat categories can be used to cover all incident types?

SRQ2: Is there a significant difference in the mindset of customers and analysts regarding incident reporting?

SRQ2.1: Would customers and analysts assign significantly different impact labels to similar situations?
SRQ2.2: Do customers and analysts have significantly different notification preferences?
SRQ2.3: Do customers and analysts assign significantly different weights to the dimensions that influence the assessment of an incident?
SRQ2.4: Could any significantly different results regarding the assessment of incidents be observed when looking at the business category, business size or job title?

Subquestion 1 is used for getting a clear overview of the current processes in the SOC, while subquestion 2 is used to check if these processes differ from what the customer would expect. Both subquestions in the end contribute to the answer to the main research question.

Chapter 10

Background information

This chapters covers the background information & literature for the research. Section 10.1 gives a brief overview of the activities that are being performed in the SOC and lists the different stages of the analysis. Afterwards, Section 10.2 and Section 10.3 discuss two of these stages more in detail.

10.1 Security Operation Center

A Security Operations Center is a central facility that houses a team of analysts that monitor and analyze security alerts for a given organization. The goals of a SOC include the detection, analysis and response to cybersecurity incidents. Fox-IT's Security Operations Center has the task of processing network & endpoints alerts, as well as alerts coming from the Managed Intelligence Service. Figure 10.1 illustrates this, and provides a high-level overview of the three types of incidents that are handled by the SOC via the Cyber Threat Management platform CTMp.

Scientific research is also performed in the area of Security Operation Centers. Already in 2005, Pirolli et al. [36] studied the analysis of intelligence sources from a cognitive point of view, basically the task performed in a SOC. They already foresaw the emerge of huge data streams and the potential usage of technology to help converting the raw data into actionable data. In their opinion, the analysis of intelligence can be seen as 'a form of sensemaking and expert skill'. Some time later, in 2008, D'Amico and Whitley [37] performed research in the actual work that is performed by Computer Network Defense Analysts. They point out that the analysts' work can be defined in six tasks: triage analysis, escalation analysis, correlation analysis, threat analysis, incident response and forensic analysis [37].

Out of these six tasks, the analysts at the Fox-IT SOC are responsible for performing the *triage analysis* and the *escalation analysis*. The correlation analysis and threat analysis are automatically performed by the CTMp itself, while incident

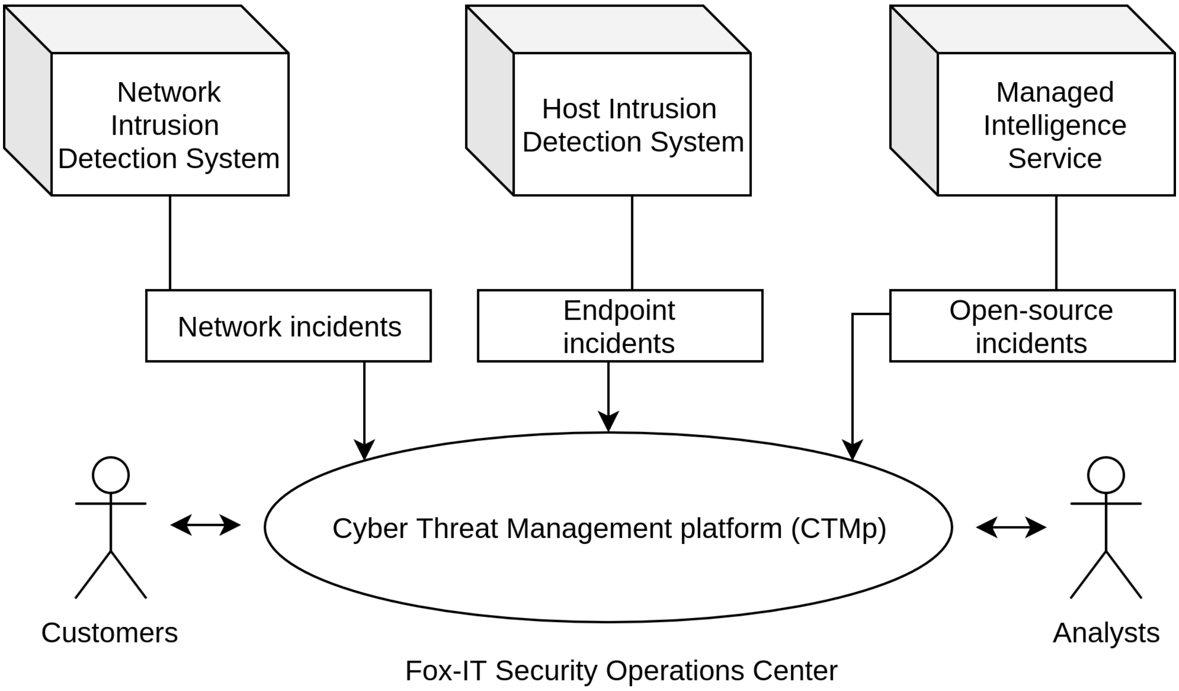

Network Intrusion Detection System
Host Intrusion Detection System
Managed Intelligence Service
Network incidents
Endpoint incidents
Open-source incidents
Cyber Threat Management platform (CTMp)
Customers
Analysts
Fox-IT Security Operations Center

Figure 10.1: Figure showing an abstract overview of the three incident types handled in CTMp

response and forensic analysis are part of the bigger Managed Detection & Response proposition. The triage analysis generally results in a conclusion in the form of a report, which is then used as input for the escalation analysis. The following subsections will elaborate more on the two distinct tasks performed by the analysts at the Fox SOC.

10.2 Triage analysis

As stated in by Zhong et al. [38], data triage is a routine task performed by analysts. During the data triage, an analysts decides what the actual threat is. Originally, the data triage was a highly specialized job manually performed by analysts. More recently, new data analysis techniques in the field of machine learning have transformed this into a more automated process, performed by computers using a variety of algorithms. Bierma et al. [39] noticed that because of the large number of alerts, analysts sometimes are overloaded with alerts and come up with a machine-learning backed prioritization of alerts in order to allocate the analysts time and energy to the most important alerts. More recently, Zhong et al. also came up with methods to reduce the analysts workload by automating the triage. In their work, a graph-based trace mining approach is used in order to model the triage process as a finite-state machine. When comparing the performance of the trained finite-state machines with the ground truth created by real human analysts, a satisfactory false positive rate could be achieved.

10.3 Escalation analysis

Communicating the severity of incidents to users is a field of study that originates from the medical world. A 2013 paper by Sendelbach & Funk [40] stated that 72% to 99% of clinical alarms are false. This leads to a sensory overload for the persons who have to anticipate to these alarms. The term 'alarm fatigue' is used to describe a situation where an analyst receives too much alarms and is no longer able to distinguish between alarms that should receive attention, and alarms that are a false positive. This was also researched by Bonafide et al. [41], who performed a study on the effects of alarm fatigue on nurses. They found that the response time of the nurses increased when the amount of non actionable alarms in the 120 minutes before the incident increased. In cyber security terms, this means that if a SOC is overloaded with false positive alerts from e.g. network sensors, the response time of the analysts may increase, which is obviously not a desired effect.

Studies are therefore focusing on the human factors in Cyber Network Defence and more specifically, how these human factors can have positive effects on the analysis & communication tasks. The paper by Gutzwiller et al. [42] highlight research areas that would be beneficial to the operations in a SOC. They include; training & feedback, cognitive biases, situation awareness and interface design, multi-tasking, vigilance and automation interaction. Although they mainly focus at the analysts' job, this information may also be of interest to customers of Managed Security Services; they too can be subject to alarm fatigue when alarms are just forwarded to them. The handbook by Wogalter et al. [43] provides insight into understanding the human errors that occur and how these human errors make their way to actual accidents & incidents.

The design principles suggested by Norman [44], [45] could also be useful in designing or adpting the existing communication between systems and SOC analysts, and between SOC analysts and customers. The principle developed by Norman is called Activity Centered Design (ACD). The ACD paradigm focuses on the activities that a user would actually need to perform in given a situation.

Chapter 11

Methodology

This chapter describes the methodology that was followed in order to answer the research questions. In Section 11.1 a high-level overview is provided, and the correlation between the sub questions and research question is explained.

11.1 Methodology

The main research question is being answered by first answering the three subquestions. A high-level overview of the methodology is shown in Figure 11.1.

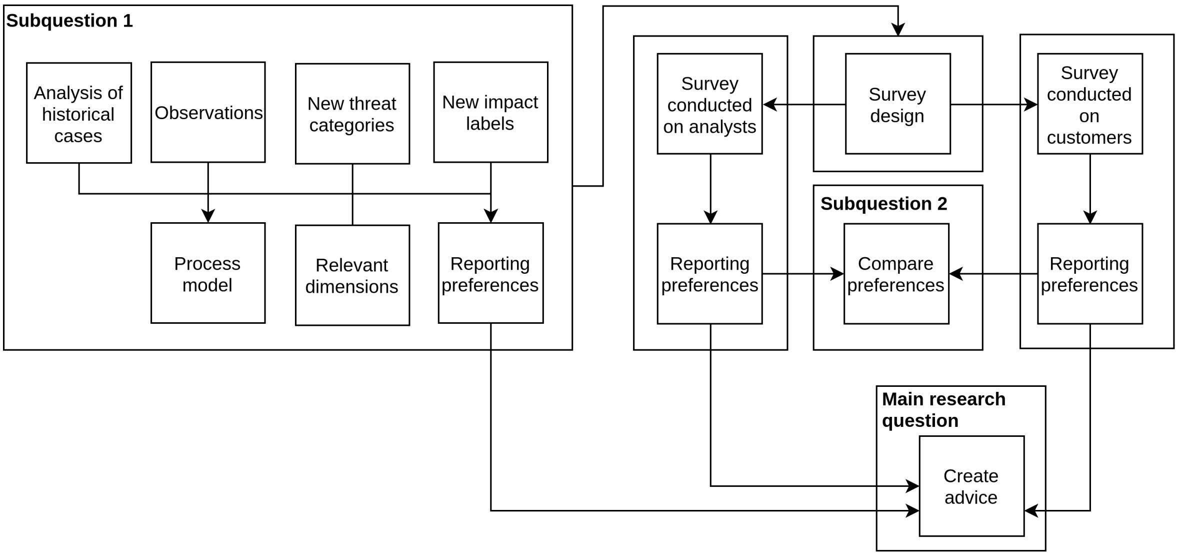


Figure 11.1: A high level overview of the methodology

During the problem investigation described in Section 9.1, the lack of more detailed impact labels was identified based on initial interviews and observations. Since currently escalated incidents are addressed as either `Low risk`, `High risk` or `Successful hack attack`, new impact labels needed to be designed to tackle this problem. The design choices for these new impact labels followed the guidelines provided by the Activity Centered Design (ACD) [44] paradigm; the naming of the labels should be in line with the response that is required from the customer, i.e. the activity that the customer has to perform.

Furthermore, during the initial problem investigation it turned out that the current threat categories that are assigned to the incidents in the SOC are too generic. Think of a label *Compromised asset*, that is assigned to both malware and adware incidents; in the current process of handling and documenting an incident it is not possible to assign a different threat category to these two incidents. At this point, subquestion 1 could be answered. The answers to this subquestion were then used to construct a survey, which was used to check for differences in the perception of incident severity between customers and analysts.

After the new impact labels & threat categories were defined, observations were made in the SOC; how did analysts perform their tasks, which dimensions did they take into account and how did they report different threat categories? The observations were gathered by looking at the routine actions performed by the analyst, and by performing some of the routine actions myself, to get an impression of the typical workflow. From these observations, the relevant dimensions were also extracted. These observations were further supported by data derived from historical, already handled incidents. This analysis of historical incidents was done manually by labeling historical incidents using the new threat categories and the old impact labels, such as `High risk` and `Low risk`. A mapping was created to convert the old impact labels to the newly defined ones. In the end, everything combined resulted in a BMPN process model of the SOC operations and the escalation procedure. The current reporting preferences could also be defined, which were used later on to compare the current decision making with the customers' survey outcomes. At this point, the first subquestion could be answered.

The results from subquestion 1 were used during the construction of the survey. This survey was used to check whether significant differences existed between the customers and analysts. Multiple statistical tests were formulated & performed, which were used to answer subquestion 2. The transformation of the findings from answering subquestion 1, as well as operationalizing subquestion 2 can be found in Chapter 13. The final conclusions that answer the main research question can be found in Chapter 15.

Chapter 12

Current processes & preferences

This chapter describes the results of the first subquestion: *How do the current processes & preferences regarding incident reporting look like?*. Section 12.1 describes the initial observations that were made and is used as a starting point for answering the subquestion. Next, Section 12.2 identifies the dimensions that are relevant during the escalation analysis. Afterwards in Section 12.3 a new framework to be used for the threat categories was selected. In Section 12.4 an analysis based on historical cases is made and finally in Section 12.5 a new design for the impact labels is made.

12.1 Observations

The work by D'Amico et al. [37] made a distinction between the triage analysis and the escalation analysis. Because of that, the current processes in place to perform these analyses are now discussed separately in the context of Fox-IT. The descriptions of both the triage & escalation analysis is based on long-term observations in the Security Operations Center, starting from 01-01-2019.

12.1.1 Observing the triage analysis

At Fox-IT, the triage analysis is partially automated by the Cyber Threat Management platform, which already greatly reduces the amount of incidents that need to be manually analyzed. In the case of network & endpoint monitoring, a sensor containing a daily-updated rule-set raises alerts whenever the network traffic matches one of the specific rules. These rules are partially obtained through an external party, but a lot of rules are written by Fox's own Security Research Team (SRT) from the Threat Intelligence department. This team of security researchers actively tracks down APTs, criminal organizations and other actors and writes rules to detect these threats. Whenever the network traffic or endpoint module matches a certain rule, an alert is raised. Alerts are assigned with different priority levels, where PRIO-1 has the highest priority and PRIO-5 the lowest. These priority levels are defined by the security researchers, and are outside the scope of the SOC. These raw alerts are picked up by the Cyber Threat Management platform, which uses an algorithm to cluster different alerts into distinct incidents. This clustering task is again outside of the SOCs scope. A incident is therefore only created under certain circumstances, e.g. a high amount of low-priority alerts or just a single high-priority alert. This relates to the correlation analysis and threat analysis as defined by D'Amico [37]. In the end, the Cyber Threat Management platform provides the analyst with a incident, that holds one or more alerts.

In the case of open-source incidents, data is delivered through the Managed Intelligence Service. This system actively looks for credentials, company code and other sensitive information that is out on the internet, only to collect it when it matches a certain customer profile (e.g. a company domain name, keywords). Though some filtering is in place to filter out duplicate data, most of this collected data is presented in raw form to the analyst, meaning that it has no priority label assigned to it. It therefore is a distinct type of incident.

At this point, the analyst receives the incident and the manual triage has to be performed. In the case of a network or endpoint incident, it is checked whether the alerts in the incident are True Positives or False Positives. Because the alerts are raised based on certain rules, an analyst can easily categorize the threat. For example, if an incident is present that was triggered on the network rule ”ET MALWARE Adware.InstallCore.B Checkin”, the analyst has to verify whether actual adware is sending a check-in signal to its Command and Control server, or that the rule triggered on some random stream of data. If it turns out to be a false positive, the analyst writes down the findings and labels the incident as `False Positive`. The incident gets checked a second time by a fellow analyst, and if the findings match the incident is closed. In the incident of a True Positive, the threat category is determined. The threat category is again almost always extracted from the alert(s) that are linked to the incident. In the example regarding the adware, it is clear to the analyst that this is an adware check-in. The analyst then assigns one of the labels listed in Table 12.1 to the incident.

Compromised asset	*Live monitoring unavailable*
Compromised information	*Undeterminable*
Misuse of systems	*Business Interruption*
Data loss	

Table 12.1: Current threat category labels

When a incident from the MIS system is presented (which is currently being done in an external application, not the Cyber Threat Management platform), the analyst first determines whether the intelligence is interesting or not. This triage is mostly manual and based on the analysts experience and research. For example, if publicly available information about a customer is found (e.g. email addresses), this might be considered not interesting. Furthermore, false positives might arise when data is found from a similar-named company. If the analyst is not sure whether the incident is interesting or not, or when the incident is clearly interesting the analyst enters the incident in the Cyber Threat Management platform, assigns one of the impact labels as shown in Table 12.1 to the incident and it proceeds to the escalation analysis phase.

12.1.2 Observing the escalation analysis

Before proceeding to the description of the escalation analysis process, it should be noted that 'escalating' an incident to a customer means in the case of Fox-IT that:

1. A detailed digital report including (technical) evidence, a justification of the impact label and an advice is sent to the customer.
2. Email and SMS messages are sent to all analysts on the customers' side, so that they can start resolving the incident.
3. (Optional) The customer's emergency contact person receives a call from the SOC, in which the customer is notified about the incident.

At this stage it is decided which impact label is assigned to the incident. Currently, the analysts assign one of the impact labels described in Table 12.2. For every impact label, a very broad description of the label is provided, along with the procedures that need to be followed for that specific impact label (writing a report and/or phoning a customer). These escalation procedures are standard, however individual customers can set individual preferences for the impact labels (e.g. "do not phone for `Low risk` incidents"). Note that the `False positive` label is considered the least severe and the `Successful hack attack` is the most severe. If a

Impact label	Description	Write report	Phone customer	Email/ SMS
Successful hack attack	The damage has already been done	✓	✓	✓
High risk	High potential damage / fast response required	✓	✓	✓
Low risk	Medium potential damage / needs to be checked soon	✓	✓	✓
Malicious	Medium potential damage / needs to be checked soon	✗	✗	✗
Not malicious	Not related to malicious behavior	✗	✗	✗
False positive	Presumed threat was a false positive	✗	✗	✗

Table 12.2: Current impact & corresponding escalation procedure

High risk	Low risk	Malicious
Internal scans	Adware	Already reported incidents
Ransomware	Hacktool updates	Adware without download
Spyware	Suspicious User-Agent	External scans
Malware	Telnet internal	
Telnet to external	Bittorrent traffic	
Successful phishing		
Remote Access Tool		
Coinminer		
Hacktool usage		

Table 12.3: A guideline for assigning impact labels to different types of incidents

(part-time) analyst is not sure about the impact label, he or she consults the senior analyst who is available at that moment.

During the problem investigation it became clear that there is no fixed decision making process for choosing the appropriate impact label for an individual incident. A senior SOC-analyst provided a list of different threat categories and the frequently preferred impact label, which is listed in Table 12.3. Note that this is only a guideline; besides the threat category there are other dimensions that play a role in making this decision. For example, the Coinminer that is listed in the `High risk` column can also be reported as a `Low risk` under some circumstances. These dimensions will further be discussed in Section 12.2.

At this stage, an analyst has assigned a threat category, as well as an impact label to the incident. The analyst then looks up the default escalation procedures

(also displayed in Table 12.2). Note that only incidents classified as `Low risk`, `High risk` and `Successful hack attack` need to be ’escalated’ to the customer; incidents classified as either `False positive`, `Not malicious` or `Malicious` are archived and closed. These incidents will still be visible to the customer (they can view all handled incidents on the Cyber Threat Management platform), but a detailed report will not be written and/or the customer will not be phoned about the incident.
The last step the analyst performs is checking the customers’ Customer Security Policy (CSP). The CSP holds all customer-specific preferences mentioned earlier. If no CSP is present, the analysts will use the default escalation procedure. The analyst then escalates the incident to the customer.

The combined triage & escalation analysis was converted into a BPMN[1] model, as shown in Appendix E. This model provides a clear overview of the incident handling process, starting as a raw alert end possibly ending escalated at a customer.

12.2 Analysis of relevant dimensions

Recall that the process of assigning an impact label to a incident is not standardized. Though there are some guidelines that map several threat categories to an impact label (as shown in Table 12.3), the decision making process is often influenced by multiple dimensions. Using the literature from the background information and three initial interviews conducted with senior analysts, nine dimensions were discussed. They were asked about the relevance of the dimension in their current decision making process, and how the the dimensions is defined concretely.

- **Threat category**: The actual threat that needs to be handled. Ranges from leaked credentials to adware, and even ransomware. Was identified as the primary source for determining the impact label by all three analysts (as shown in Table 12.3).
- **Similar incidents**: Indicates whether a incident with similar characteristics (e.g. same source IP, same alerts, similar timestamp) was escalated in the last period; the period is not strictly defined. If an incident is considered a duplicate of another and the original incident has already been escalated, it is usually not escalated again. This really depends on the length of the period; if it is one day, it is considered a duplicate. A similar case one month ago is usually not considered a duplicate.
- **Raised awareness**: Knowing that e.g. a company is targeted or threatened in a recent campaign influences the decision. If for example, a malware cam-

[1] Business Process Modeling Notation

paign is targeting a customer and an incident that is normally assessed as `Low risk` occurs, it might be classified as `High risk` instead because of the increased potential threat. Only one out of three senior analysts acknowledged this influenced the decision making.

- **Timing**: Indicates whether the incident is created during office hours (when the customer is likely present and ready to act) or outside office hours (meaning that the customer has to be disturbed / woken up in their free time). Three out of three analysts acknowledged the importance of this dimension.
- **Involved asset**: In practice, no difference is made between devices on the internal network. However sometimes a separate guest network is present, which is given a lower priority. Three out of three analysts acknowledged that this is taken into account.
- **Type of attacker**: Indicates who is behind this attack; this might for example be a script kiddie, or a state actor. In practice however, it is hard to determine who is responsible for the attack. Customers themselves can sometimes be the 'attackers'; penetration testing & auditing is happening on a continuous basis and is not always announced to the SOC beforehand. Although this might look like an interesting dimension, it is hard to use in practice. All analysts acknowledged that this was not a significant dimension.
- **Potential costs**: This was not taken into account at all by all three analysts, since this data has not yet been processed and made available to the analysts.
- **Triage confidence**: The confidence of the triage analysis. Sometimes, an analyst is unable to fetch all the necessary data needed to conduct the triage (e.g. due to encryption). If there is any doubt during the triage process, all three analysts stated that they better wanted to be safe than sorry.
- **Age of source data**: The age of the underlying incident data. Sometimes, due to various reasons, incidents are based on outdated network data. In this case, it may not be useful to escalate the case since logging and other crucial data may be gone. Regarding open source data, sometimes really old password dumps may be presented as a credential leak, which is in turn not very interesting anymore (assuming the customer is already aware of the password dump). This dimension was considered relevant by the three analysts.

The process of prioritizing one dimension over another, and assigning values to every single dimension all happens inside the mind of a single analyst. If a single analyst does not have all the experience required to make the decision, a second opinion by a senior analyst can be requested or some fellow analysts can be asked for advice. The senior analyst or fellow analysts then perform the same process inside their own minds. Eventually, this process results into one of the six labels shown in Table 12.2.

12.3 Defining new threat categories

One of the problems identified in the problem investigation was that the threat categories do not cover all types of incidents. The threat categories that are currently being used were listed in Table 12.1. Looking at the guidelines for mapping threat categories to impact labels in Table 12.3, it can already be observed that there is not a single match between threat categories listed in both tables. Because of that, there is a need to define new threat categories that cover all of the incidents handled in the SOC. Note that this includes incidents based on network, endpoint and open-source alerts.
Multiple threat category taxonomies have been proposed over time. The Malware Information Sharing Platform (MISP) project[2] provides a good starting basis. It provides open-source JSON files of existing taxonomies and mapping between them on[3]. After inspecting the different frameworks, five taxonomies remain interesting for further investigation.

1. **VERIS**: The Vocabulary for Event Recording and Incident Sharing (VERIS) framework [4], originally initiated by Verizon, is an effort to create create a standard for classifying incidents . It can be used to e.g. track incidents, provide demographics about victims and actors and perform an impact assessment. Anther purpose of the VERIS framework is *Incident description*, which can be of use when defining new threat categories. The VERIS framework is very detailed and differentiates between very specific threats.

2. **eCSIRT**: eCSIRT. The ECSIRT incident taxonomy was first introduced in 2003 and updated in 2012[5]. The taxonomy is a collaboration between a number of established European CSIRTs, and focused on improving the collaboration between these CSIRTs by making incident sharing easier. The objective was to make this process of incident sharing more efficient and enable the collection of shared data for research purposes.

[2] http://misp-project.org/

[3] https://github.com/MISP/misp-taxonomies

[4] http://veriscommunity.net/

[5] https://www.trusted-introducer.org/Incident-Classification-Taxonomy.pdf

3. **Europol**: The Europol Common Taxonomy for Law Enforcement and The Nation Network of CSIRT's[6], which was last updated in 2017, is focused on offering a common language between CSIRT's and Law Enforcement Agencies. They aim at 'filling the gap between CSIRT's and LEA's' by providing this international legislative framework.

4. **CIRCL**: The Computer Incident Response Center Luxembourg, or CIRCL[7] taxonomy for incidents is another framework that aims at providing one framework to be used in collaboration with other CSIRT's. The incident categories are rather technical and very specific, but there is no option to further extend the framework with custom, CSIRT specific incidents.

5. **ENISA**: The ENISA Reference Incident Classification Taxonomy[8] is again a combined effort by European Union countries to design an incident taxonomy that can be used to share data between CSIRT's. This latest ENISA taxonomy, which was updated in 2018, is built upon multiple other initiatives such as the eCSIRT.net classification listed above. The objective of the taskforce responsible for the development of the framework is "to centralize all current relevant taxonomies and to agree upon a small common set of taxonomies for specific use cases."

Because only one framework needs to be chosen, a comparison was made in Figure 12.1. The framework needed to be easily expandable to include all incidents and updated after 2017 to include the latest incident types. Furthermore, frameworks with a low depth (number of subcategories) are preferred, as well as categories that have a large number of top categories; this implies that a one-dimensional extensive framework is preferred over a deeply nested framework containing many layers and subcategories. In the end, the ENISA framework was chosen because it was the best fit to the needs.

12.4 Analysis of historical incidents

Now that a framework for the threat categories was chosen and the processes within the SOC were observed, it was time to do an analysis on historical data to retrieve statistics about the SOC.

[6] https://www.europol.europa.eu/publications-documents/common-taxonomy-for-law-enforcement-and-csirts

[7] https://www.circl.lu/pub/taxonomy/

[8] https://www.enisa.europa.eu/publications/reference-incident-classification-taxonomy

Framework	Expandable	Updated since 2017	# of category layers	# of top categories
VERIS	✓	✓	3	7
eCSIRT	✓	✗	2	11
Europol	✓	✓	2	9
CIRCL	✗	✓	1	17
ENISA	✓	✓	2	11

Figure 12.1: Comparison of taxonomy frameworks

First, a dump of all handled incidents on the Cyber Threat Management platform was made from from 22-02-2018 until 22-02-2019. This resulted in a total of 24667 incidents. Out of these 24667 incidents, 2470 were reported to the customer. This means around 10% of the created incidents were reported to the customer. Every incident also has the initial alert embedded that triggered the incident. Although multiple alerts can be assigned to a incident, the initial alert will be used to classify the incident since the alert is specified in the incident title and speeds up the manual process. For every incident in the total set of 2470, the title was observed and the ENISA threat category that came closest was assigned. Since the ENISA taxonomy allows non-standard categories to be inserted in the 'Other' category, these were added during the investigation. The results can be observed in Table 12.4. The data is also graphically displayed in Figure 12.2.

Category	Subcategory	Amount	Percentage	Low risk	High risk	Suc. h. attack
Availability	DoS	3	0.12%	3	0	0
Fraud	Masquerade	7	0.28%	6	1	0
	Phishing	69	2.79%	28	41	0
Information Content Security	Unauthorized access to information	9	0.36%	4	4	1
Information gathering	Scanning	253	10.23%	27	226	0
Intrusion attempts	Exploit known vulnerabilities	264	10.68%	92	170	2
	Login attempts	21	0.84%	13	8	0
Intrusions	Account compromise	103	4.16%	13	88	2
Malicious code	Adware	608	24.58%	3	605	0
	Coinminer	46	1.86%	11	35	0
	Ransomware	11	0.44%	2	9	0
	Spyware	27	1.09%	26	1	0
	Trojan	177	7.16%	61	108	7
Other	Domain abuse	6	0.24%	6	0	0
	Executable suspicious behavior	33	1.33%	27	6	0
	Leaked credentials	18	0.72%	12	6	0
	Malicious file download	36	1.46%	31	5	0
	Remote access policy violation	52	2.09%	34	18	0
	Sensitive data leaked	26	1.05%	16	10	0
	Suspicious traffic	238	9.54%	149	89	0
	Hacktool update	103	4.16%	99	4	0
	TOR usage	257	10.40%	26	231	0
	Torrent usage	14	0.57%	8	6	0
	USB executable started	39	1.57%	37	2	0
Vulnerable	Open for abuse	52	2.14%	46	7	0

Table 12.4: Threat categories identified at Fox-IT, based on the ENISA standard

It should be noted that the Figure 12.2 is capped at 250. The results show that

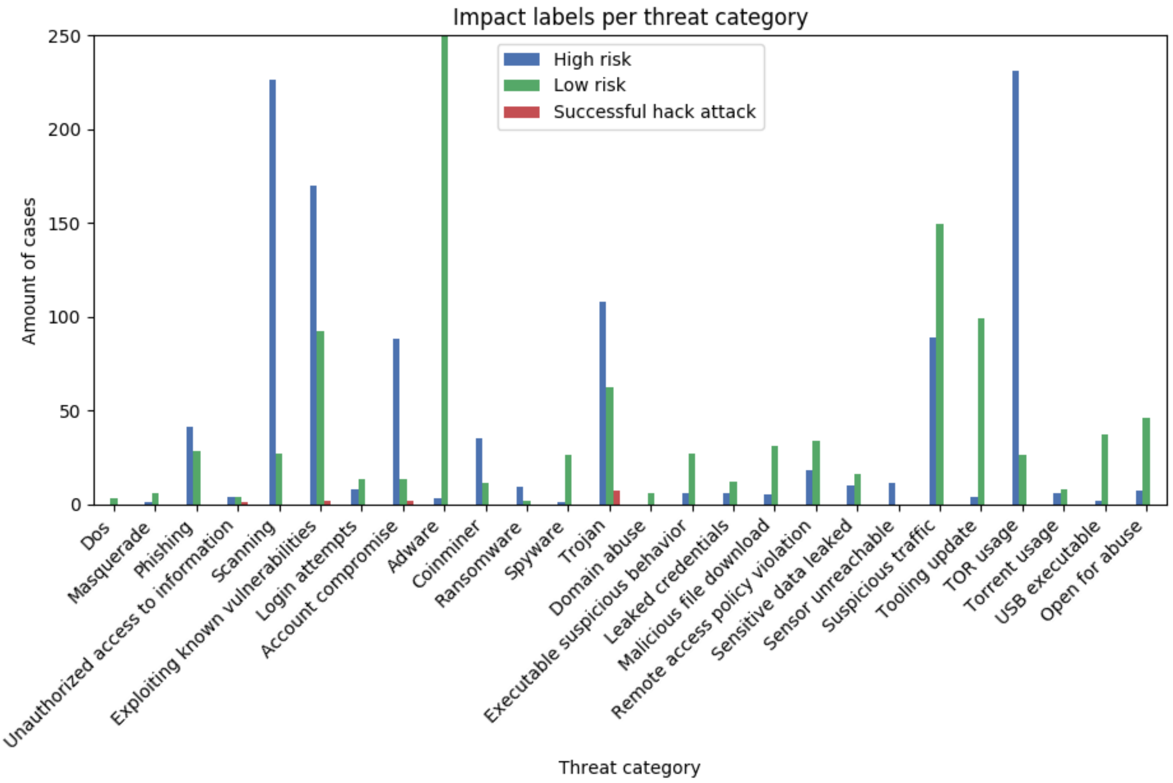


Figure 12.2: Graph showing the amount of incidents and corresponding impact labels. The graph is capped at 250.

a handful of threat categories is accountable for the majority of reported incidents. For example, the *Adware* threat category accounts for almost 25% of all reported cases. The *Scanning* (representing port scans), *Exploit known vulnerabilities* (representing exploit attempts / scans), *TOR usage* and *Suspicious traffic* each account for approximately 10% of the reported cases. In total, the top five incidents account for approximately 65%.

It also becomes clear that despite the guidelines provided in Table 12.3, similar threat categories are assigned a different impact label. This problem was already raised during the problem investigation, but can now be observed in the data. The dimensions discussed in Section 12.2 are probably responsible for this phenomena.

12.5 Defining new impact labels

Based on the initial interviews, there were some indications that he current labels were not self-explanatory and clear. For example, the label `Malicious` would have less priority than `Low risk` and they both do not explain on their own what the user should do. Possibly this is the case because the impact labels aim to describe the severity of the incident, and severity is a rather vague variable. To be able to investigate this variable in the remainder of the thesis, it needed to be specified more clearly.

This is visualized in Figure 12.3. To start off, risk calculations have been made in the past, which led to the default guidelines stated in Table 12.3. These are just guidelines which can be influenced by the dimensions mentioned in Section 12.2. In the end, the impact label thus represents the severity. When asked to further divide the variable severity, analysts came up with two indicators for severity:

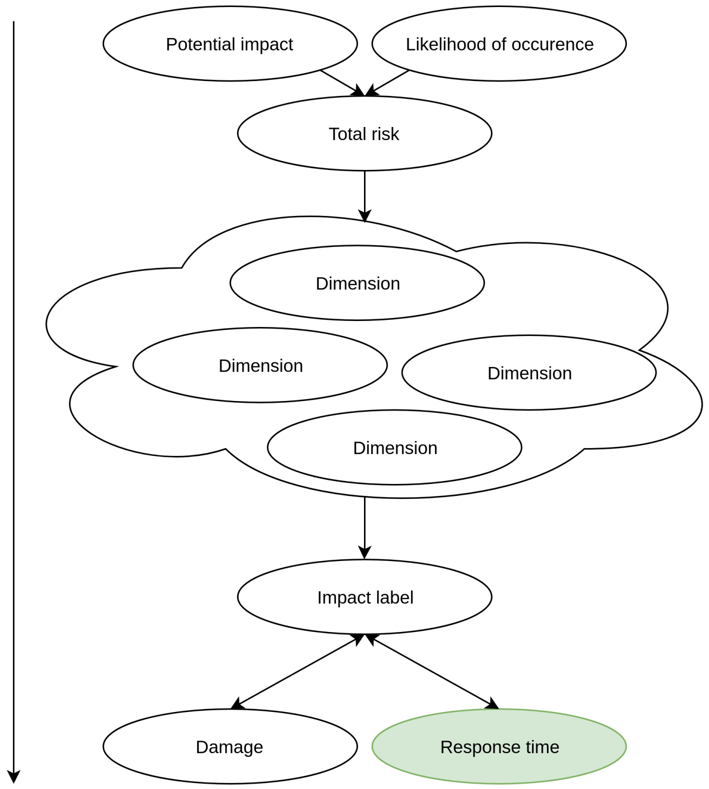


Figure 12.3: Defining the concept of incident severity for customers, as assessed by analysts

- **Damage**: much like the potential impact used in the first risk calculation, which represents the possible impact on the organization. An incident labeled `High risk` in general can cause more damage than a `Low risk` incident.
- **Response time**: indicates how quick the customer should respond to the incident. In general, a `High risk` incident needs to be resolved quicker than a `Low risk`.

This raises a problem, since the current impact labels can have a duplicate meaning; either damage or response time, or a combination of both. For the purpose of this research, the **response time** indicator is chosen to represent the severity variable. On the one hand, because this is easier to quantify (on a timescale) than damage (using monetary units). On the other hand because from several short interviews it became clear that this indicator was most frequently meant by analysts.

In a different scientific field, namely medicine, The National Confidential Enquiry into Patient Outcome and Death (NCEPOD) classification of intervention was used to standardize the impact labels that represented the response time. This classification provides labels for different severity levels of medical emergencies [9]. Although it was designed for the medical world, it provides clear statements for different response times; e.g. ”immediate” response for life threatening incidents that need to be resolved within minutes. The NCEPOD classification lists four different response times: minutes, hours, days and planned ahead.

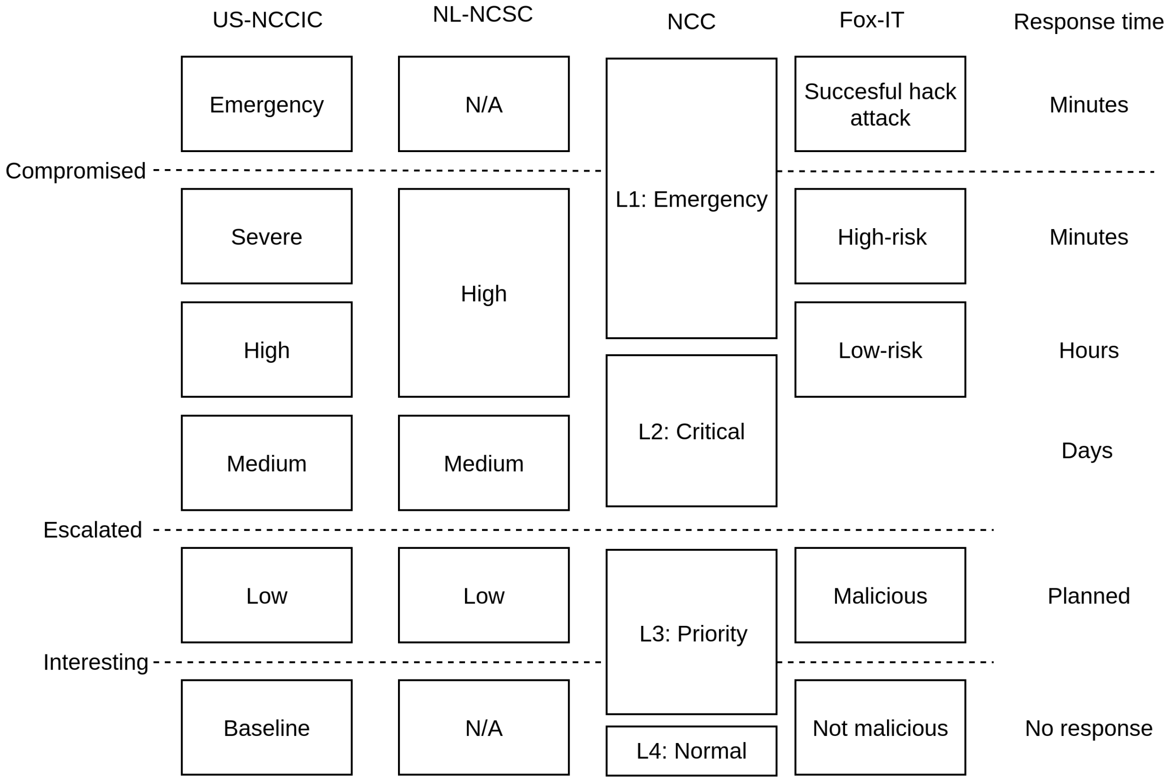


Figure 12.4: Comparison of impact labels from several frameworks

More concrete, an attempt has been made to plot four different impact label frameworks in one figure based on the response time, as shown in Figure 12.4. This includes the United States NCCIC Cyber Incident Scoring System[10], the Dutch National Cyber Security Center ematrix[11], the impact labels used by NCC Group

[9] https://www.ncepod.org.uk/classification.html

[10] https://www.us-cert.gov/NCCIC-Cyber-Incident-Scoring-System

[11] https://www.ncsc.nl/binaries/content/documents/ncsc-nl/dienstverlening/response-op-dreigingen-en-incidenten/beveiligingsadviezen/_data%5B2%5D/_data/ncsc%3Aresources%5B1%5D

(the parent company of Fox-IT) and lastly, the impact labels used by Fox-IT itself. This is nowhere near a baselined representation, but is displayed just to show the variation in impact labels and their different meanings.

However, it does reveal one interesting thing. During the initial interviews, an observation was that the SOC-analysts felt like they sometimes could not find an appropriate impact label with the correct escalation procedures for an incident. Furthermore, when asked for the response time for `High risk` and `Low risk` incidents, the response was often respectively minutes and hours. Since the `Malicious` label is not escalated to the customer, resolving these incidents can be planned and/or performed in spare time by the customer. This means that **the impact labels defined by Fox-IT lack an option to classify an incident that needs to be resolved in a few days.**

For the purpose of this research new impact labels were designed, as shown in Figure 12.5. These labels cover all of the six levels shown in Figure 12.4. Four of the labels (`Immediate response`, `Urgent response`, `Expedited response`, `Elective response`) correlate with the NCEPOD classification. On top, a label named `Start incident response` was added to indicate a successful compromise (previously called `Successful hack attack`) and at the bottom of the table, the `No response required` replaces the `Not malicious` label.

Label	Response time	Description	Old label
`Start incident response`	Minutes	Successful compromise	`Successful hack attack`
`Immediate response`	Minutes	Last stage before getting compromised	`High risk`
`Urgent response`	Hours	Resolve to reduce possibility of compromise	`Low risk`
`Expedited response`	Days	Situation requires early intervention	-
`Elective response`	Spare time	Resolving can be planned ahead	`Malicious`
`No response required`	-	No need to resolve this	`Not malicious`

Figure 12.5: Newly designed impact labels, based on the response time

As shown, the newly designed labels can still be mapped to the old impact labels; this is because they are baselined on the response time. Interviews with analysts revealed that `High risk` incidents were correlated to a response time of ’minutes’, while `Low risk` incidents needed to be resolved within hours. This mapping will be used later on the research.

In relation with the literature, the new labels are designed while keeping in mind the Activity Centered Design proposed by Norman [44], [45]. Grammatically the labels are written in the imperative mood; ”immediately respond” or ”start incident response” tells the customers exactly what to do in the situation. In the old situation if a customer received a `Low risk` incident notification, it was still unclear what action should be performed at that point; the new labels were designed to solve this problem.

Chapter 13

Designing, distributing & analyzing the survey

This chapter describes the results from subquestion 2. In Section 13.1 the subquestion is operationalized into variables and sub-subquestions. Afterwards, for each of the sub-subquestions a subsection is dedicated, specifying the objectives and hypotheses. In Section 13.2 the construction & distribution of the survey is described.

13.1 Operationalization of subquestion

The second subquestion was defined as: *Is there a significant difference in the mindset of customers and analysts regarding incident reporting?*

In this section, this subquestion is operationalized; it will be divided into even more questions and hypotheses that can be tested. The variables that are being investigated are listed, as well as the indicators that are used to measure the variables and the instruments that are used to conduct the tests in practice.

13.1.1 Defining sub-subquestions

In the escalation analysis (discussed in Subsection 12.1.2), Fox-IT has set several standards and/or guidelines that need to be validated. The guidelines for mapping threat categories into impact labels, provided in Table 12.3, being the major point of interest (**SRQ2.1**). Fox-IT does determine these guidelines, but has not validated these guidelines with the customers. Furthermore, the escalation procedures belonging to the impact labels are also static; they are listed in Table 12.2. Therefore, it is useful to check whether the current preferences regarding notifying the customer align with the actual customers' preferences (**SRQ2.2**). The dimensions that were identified influencing the escalation process in Section 12.2 are only enumerated, but have no weights assigned to them. For example, to which extent does a change in the dimension ”timing” influence the impact label assigned to the incident? (**SRQ2.3**). Lastly, it might be interesting to look for differences in business size, business category and job title of respondents (**SRQ2.4**). In the perspective of the second subquestion, all of the above can be transformed into the following four sub-subquestions:

> **SRQ2.1**: Would customers and analysts assign significantly different **impact labels** to similar situations?
> **SRQ2.2**: Do customers and analysts have significantly different **notification preferences**?
> **SRQ2.3**: Do customers and analysts assign significantly different **weights to the dimensions** that influence the assessment of an incident?
> **SRQ2.4**: Could any significantly different results regarding the assessment of incidents be observed when looking at the business category, business size or job title (**demographics**)?

From now on the questions will be referred to by means of the terms marked in bold.

13.1.2 Impact label questions

This is considered the most important sub-subquestion; it aims to reveal any significant differences between the perception of incident severity when looking at customers & analysts. This means that two sample groups need to be used to answer this question; customers and analysts.
It is practically impossible to test all possible incidents; this would make the survey too long to complete, possibly lowering the response rate. As a trade off, a total of ten incidents were chosen based on the historical case analysis in Section 12.4. Since Fox-IT’s SOC covers network incidents, endpoint incidents and open-source incidents, a mix of these incidents was picked. The resulting list of ten incidents with a distinct threat category is shown in Figure 13.1.

The mix includes several threat categories that are always classified as either `High risk` or `Low risk` using the old impact labels, which makes it interesting to see if these impact labels are still suitable. On the other hand, also threat categories that are classified `High risk` just as often as `Low risk` are chosen, since probably for these categories no clear guideline is available. Lastly, the top three most frequently occurring incidents are added. The *Exploit known vulnerabilities* threat category was excluded from this top three, since it includes both vulnerability

Threat category	Type of incident	Reason
Adware	Network	Most frequently occurring incident
Port scanning	Network	Third most frequently occurring incident
Trojan	Network	Mix of `High risk` and `Low risk` classifications
TOR usage	Network	Second most frequently occurring incident
Malicious tooling update	Network	Almost always `Low risk`
Open for abuse	Network	`Almost always Low risk`
Suspicious application behavior	Endpoint	Mix of `High risk` and `Low risk` classifications
USB executable started	Endpoint	Almost always `Low risk`
Leaked credentials	Open source	Mix of `High risk` and `Low risk` classifications
Sensitive data leaked	Open source	Mix of `High risk` and `Low risk` classifications

Figure 13.1: The ten selected incidents used in the survey

scanning behavior as well as individual exploit attempts. Therefore, *Port scanning* was chosen as the single *Scanning* threat category. The *Suspicious traffic* threat category was also dismissed in the survey, since this category is too broad and may be influenced by other variables that are not included in the survey (i.e. suspicious HTTP traffic is different from suspicious SMB traffic).

For each of the ten incidents, the participants will be asked to assign an impact label. In Section 12.2 however, it was discovered that the threat category of an incidents was only one of many dimensions that is taken into account when assigning an impact label. Therefore, each of the incidents needs extra contextual information of the dimensions. In Section 12.2 nine different dimensions were identified. Again, using all nine dimensions, the result would be a long survey potentially lowering the response rate. Therefore, six out of nine dimensions were included in the survey. Figure 13.2 shows which dimensions were included in the survey, and how the dimensions was actually represented in the survey.

It can be observed that every dimension has two entries for the three types of incidents. Every first (**#1**) entry of a dimensions is considered the 'default' value, and is used for the contextual information in this sub-subquestion. For example, the description of the incident along with contextual information for *Adware*, which is a network incident, is:

> Adware (**Threat category**) has been observed being installed on a client in the internal network (**Involved asset**) during office hours (**Timing**). Also, there have not been similar incidents involving the same client (**Similar incident**). Lastly, the network data is just a few minutes old (**Age of source data**) and you are 100% sure that it is adware (**Triage confidence**).

The dimensions "Raised awareness", "Type of attacker" and "Potential costs"

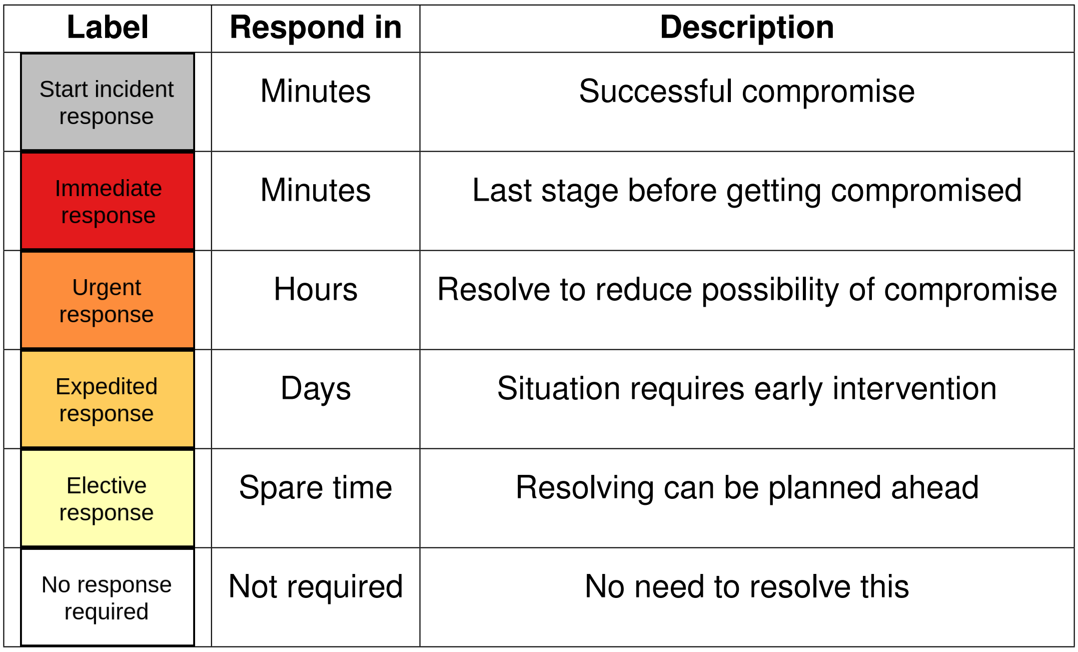
Dimension	#	Network incident	Endpoint incident	Open source incident
Timing	1	**During office hours**	**During office hours**	**During office hours**
	2	Outside office hours	Outside office hours	Outside office hours
Involved asset	1	**Internal client**	**Privileged account**	**Privileged account**
	2	Guest client	Unprivileged account	Unprivileged account
Similar incident	1	**No**	**No**	**No**
	2	Yes, last week	Yes, last week	Yes, last week
Age of source data	1	**Today**	**Today**	**Today**
	2	One month	One month	One month
Triage confidence	1	**Indisputable**	**Indisputable**	**Indisputable**
	2	Disputable	Disputable	Disputable
Raised awareness	Dismissed			
Type of attacker	Dismissed			
Potential costs	Dismissed			

Figure 13.2: Operationalization of dimensions, default values in bold

were all dismissed because they were hard to be used in practice and/or were not considered relevant by senior analysts. The ”Threat category” dimensions were already shown in Figure 13.1.

Recall that the participants (both customers and analysts) need to assign an impact label to the incident, e.g. the adware example above. For this purpose, the newly designed impact labels that represent the response time will be used (see Figure 13.3)

These new impact labels were given a color to indicate and emphasize the sever-



Label	Respond in	Description
Start incident response	Minutes	Successful compromise
Immediate response	Minutes	Last stage before getting compromised
Urgent response	Hours	Resolve to reduce possibility of compromise
Expedited response	Days	Situation requires early intervention
Elective response	Spare time	Resolving can be planned ahead
No response required	Not required	No need to resolve this

Figure 13.3: Newly designed impact labels, as shown to the participants

No response	Elective response	Expedited response	Urgent response	Immediate response	Start incident response
1	2	3	4	5	6

Table 13.1: Impact labels encoded to numerical values

ity levels. The color levels for `No response required` up to `Immediate response` would gradually turn from white into red[1]. The grey `Start incident response` color indicates that the damage has already been done and that the incident requires incident recovery and forensic investigation.

This will result in ten data points per participant; exactly one assigned impact label per incident. Since the incidents are the equal for all participants (both customers and analysts), these impact labels can be used to check whether the distribution of the impact label is the same in both sample groups. Since statistical tests require numerical values instead of impact labels, the impact labels are converted to numerical values as displayed in Table 13.1.

The test specifications for all ten incidents is shown below. The tests are based on the Mann-Whitney U test. This test is preferred, since it can be used to test whether a significant difference exists between two independent different distributions[2].

[1] Consecutive colors were picked from http://colorbrewer2.org/

[2] The webpage on https://stats.idre.ucla.edu/other/mult-pkg/whatstat/ provides a clear overview of the different statistical tests and when to use them

Test specifications for impact label questions

1. The samples are drawn randomly from the population
2. The collected data is considered ordinal
3. The data is that is collected from both groups is independent; a participant is either a customer or an analyst
4. The null hypothesis H_0 assumes that the impact labels of both samples are equally distributed.
5. The alternative hypothesis H_1 assumes that the impact labels of both samples are not equally distributed.
6. The significance level to test for is $\alpha = 0.05$
7. The Mann-Whitney U value is calculated using:

$$U = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1$$

   where n_1 is the sample size of the customers, n_2 is the sample size of the analysts and R_1 is the sum of ranks from the customer samples.

8. H_0 will be rejected when a two-sided Mann-Whitney U test results in a corresponding probability value $p < 0.05$. If $p > 0.05$ the null hypothesis will not be rejected.

Note that this data is considered ordinal instead of categorical; when assigning a label `Immediate response`, the incident is considered more severe than when an Urgent response label is assigned. The distances between the impact labels is not known; the distance between `Immediate response` and `Urgent response` might be smaller than the distance between `No response` and `Elective response`. A Mann-Whitney U test is capable of handling ordinal data.

13.1.3 Notification preference questions

To answer this sub-subquestion, data needs to be collected about the notification preferences from both the customers and analysts. In Subsection 13.1.2 the participants are already asked to assign an impact label to an incident. This means that they have assessed the situation and probably have an idea on how severe the incident is. The question can be extended with two questions: *Would you write a detailed report about this incident?* and *Would you notify the customer by phone* *about this incident?* This data can be used to calculate the amount of customers and analysts that want to receive a detailed report / phone call when a specific impact label is assigned. The data that is collected is nominal data; a 'Yes' is not better nor worse than a 'No', and there is also no fixed distance between these two categories. A test that is well suited for handling nominal data is the Chi-Square test. The twelve tests can be defined as follows:

Test specifications for notification preference questions

1. The samples are drawn randomly from the population
2. The collected data is considered nominal
3. The data is that is collected from both groups is independent; a participant is either a customer or an analyst
4. The null hypothesis H_0 assumes that the notification preferences of both samples are equally distributed.
5. The alternative hypothesis H_1 assumes that the notification preferences of both samples are not equally distributed.
6. The significance level to test for is $\alpha = 0.05$
7. The Chi-Square value is calculated using:

   $$\chi^2 = \sum_{k=1}^{n} \frac{(O_k - E_k)^2}{E_k}$$

   where O_k is the observed frequency, E_k the expected frequency, n the number of input array cells. The degrees of freedom is $df = 2 - 1 = 1$
8. H_0 will be rejected when a Chi-Square test results in a corresponding probability value $p < 0.05$. If $p > 0.05$ the null hypothesis will not be rejected.

13.1.4 Weights of dimensions

The main objective of this sub-subquestion is to determine to which extent the dimensions influence the escalation analysis. In Subsection 13.1.2 the original amount of dimensions was already reduced to five (excluding the threat category) to be used in the survey. This meant that when assigning the impact labels, the participants were taking into account the dimensions displayed in bold in Figure 13.2.

To answer this sub-subquestion, at the end of the survey three incidents will be repeated and the impact label that the participant had assigned to the incident will also be shown. This reminds the participant of the involved dimensions and possibly the reasoning why he/she had assigned that specific impact label.

All dimensions listed in Figure 13.2 have two possible values; the default values in bold, and an alternative option. The participants will be asked five times to reassign an impact label the incident, while in each of the questions exactly one of the dimensions will be changed from the default value to the alternative value. The alternative values are considered less severe than the default value, so a decline in severity (or 'lower' impact label) is expected. Afterwards, for every dimension the difference between the original impact label and the newly assigned impact label can be calculated, which is essentially the data that is collected in this question.

For example, consider that a participant has previously assigned a `Urgent response` impact label (numeric value of 4) to the *Adware* incident. The original situation stated that "there have not been similar incidents involving the same client". After repeating the original situation the participant is asked to reassign an impact to the incident, with the only change being that "a similar case involving the same client was already reported last week", which is considered 'less severe' by analysts. Assuming that the participant assigns a `Elective response` impact label (numeric value 2), the change resulted in a -2 difference in impact label. In the end, the differences for all questions for all participants is calculated. The three different types of incidents will be questioned to the participants, so that potential differences can be checked. A network incident is represented by the *Trojan* threat category, an endpoint incident by the *Suspicious application behavior* threat category and an open-source incident is covered by the *Leaked credentials* threat category. Since the data that is collected again consists of impact labels, which is ordinal data, the Mann-Whitney U test can be used to test for differences in the distributions:

Test specifications for dimension weight questions

1. The samples are drawn randomly from the population
2. The collected data is considered ordinal
3. The data is that is collected from both groups is independent; a participant is either a customer or an analyst
4. The null hypothesis H_0 assumes that the decline in severity is equally distributed.
5. The alternative hypothesis H_1 assumes that the decline in severity is not equally distributed.
6. The significance level to test for is $\alpha = 0.05$
7. The Mann-Whitney U value is calculated using:

   $$U = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1$$

   where n_1 is the sample size of the customers, n_2 is the sample size of the analysts and R_1 is the sum of ranks from the customer samples.
8. H_0 will be rejected when a two-sided Mann-Whitney U test results in a corresponding probability value $p < 0.05$. If $p > 0.05$ the null hypothesis will not be rejected.

13.1.5 Demographic questions

The objective of this question is to investigate whether customers of different size, operating in a different business category and/or employees with different job titles would assess incidents differently. Note that in order to answer this sub-subquestion, no analyst responses are required. Three multiple choice questions will be added to the start of the survey to collect this data. Obviously, these three questions are only included in the customers' instance of the survey. The multiple choice options that customers were allowed to pick from are displayed in Figure 13.5, Figure 13.6 and Figure 13.4.

The data is again ordinal. However, the Mann-Whitney U test can not be used since it can only handle two sample groups. The Kruskal-Wallis test can be used to test whether the medians of two or more groups are equal; therefore it is used to find out whether any differences exist.

Job title	
Board member	CISO
(IT) Manager / Team Lead	Product Owner
Information Security Officer / Coordinator	Security Consultant
System Administrator	Network administrator
Developer / Operations Engineer (DevOps)	

Figure 13.4: Job title options

Business category			
Agriculture	Mining	Manufacturing	Energy
Water & waste	Construction	Wholesale & Retail	Transport
Hotels & bars	ICT	Finance	Real estate
Public sector	Education	Health & social work	Entertainment

Figure 13.5: Business category options

Number of employees
1 - 9
10 - 249
250 - 999
1000 +

Figure 13.6: Business size options

13.2 Constructing & distributing the survey

Construction procedure

Because Fox-IT does not share the names of their customers and wanted to avoid any possibility that these names would leak, the survey was designed to be conducted anonymously. This means that, except for three demographic questions and a timestamp of the recorded response, all (meta)data about the individual customers was not recorded. The platform that was used to construct the survey is Qualtrics[3], an online system suitable for the function of conducting surveys and made available by the University of Twente. A visual representation of the survey can be found in Appendix F. After clicking an anonymous link in the invitation, a small description of the study was again provided and the participants were introduced to the new impact labels as shown in Figure 13.3. On the next page, the demographic questions were displayed.

The demographic questions were followed by ten pages containing the incident de-

[3] https://utwentebs.eu.qualtrics.com

scriptions and questions to answer the impact label questions and the notification preference questions. For each of the ten incidents, first a small optional description of the threat category was displayed to make sure that the definition of the threat category is known to the participant. For example, in the case of adware, the description was:

> Adware belongs to a class of software that shows advertisements at varying times during general computer usage. The advertisements are shown to generate revenue for the creator of the adware program. Clicking on these advertisements would generate more revenue than just showing the advertisement in general. In general adware is bundled with legitimate software and will slow down the PC it is installed on because it uses the PC's resources to capture data and show advertisements.

Below the description of the threat category, Figure 13.3 was again displayed to ’refresh’ the participants’ knowledge of the new impact labels. Below the impact labels, the actual situation describing the incident was shown (see Subsection 13.1.2 for the description of the *Adware* incident) followed by a multiple choice question listing the impact labels. Below the impact label question, the two questions asking for the participants’ preferences regarding writing a detailed report & notifying the customer by phone, were listed. In the end, the participants had to select exactly one impact label and had to fill in the reporting and phone preferences in order to continue to the next question.
After filling in the question for the ten incidents, the data for the dimension weight questions was collected. This was done by again displaying the incident situation and showing the impact label that the participants had previously assigned to that situation. The five alternative dimension values, as discussed in Subsection 13.1.4 were listed and for every different dimension, a new impact label had to be assigned. After performing this task for a network incident (*Trojan*), endpoint incident (*Suspicious application behavior*) as well as an open-source incident (*Leaked credentials*, the survey was completed.

Distribution & responses

An anonymous link link, meaning that the link was the same for every participant, was sent to customers via the Cyber Threat Management platform’s question functionality. An initial list of customers was was discussed with Customer Success Managers and Service Delivery Managers of Fox-IT, and after dismissing customers related to the Dutch government & a small selection of other special customers, a list of 101 customers was left.

The invitation (which can be found in Appendix F) was first sent to a pilot group of 5 customers on 01-05-2019, followed by another pilot group of 20 customers on 06-05-2019. Also at 06-05-2019, the surveys were sent to a pilot group of 5 analysts. The analyst were asked if all the questions were clear, if they had fully understood the purpose of the survey and whether there were any practical problems. After receiving the feedback that the analysts had no problem filling in the surveys, combined with the fact that not a single customer had raised a question on the Cyber Threat Management platform regarding the survey, it was decided to send out the remaining 76 surveys to the customers as well as the 27 remaining surveys to the analysts.

In the analysts' version of the survey, the three demographic questions (business size, business category & job title) were dismissed, but the remainder of the survey was identical to the customers' version. This ensured that both sample groups was presented with the exact same information. The surveys for both the customers as well as the analysts were finally closed in Qualtrics on 10-06-2019.

	Customers	Analysts
Survey opened	69	23
Survey 100% completed	53	22

Figure 13.7: Response statistics

As can be seen in Figure 13.7, a total of **53** employees at customers participated in the study, as well as **22** out of the total 28 SOC analysts.Note that when a question was handed over to a customer, all of the analysts on the customers' side received the message. This means that multiple employees at the customer can fill in the survey; the 53 full responses are therefore not likely coming from 53 unique businesses. Some business have only granted one analyst access to the Cyber Threat Management platform, while other business may have granted access to up to 20 analysts.

13.2.1 Ethical considerations

The survey was constructed with the privacy and security of the participants as the number one priority [Endnote 1]. It should be noted that:

- Participation in the survey was completely voluntarily.
- Participants were able to stop their participation at any given point during the survey.

- The recorded data was completely anonymous and except for the business size, business category and job title of the participant no other demographic data was stored. The timestamp of the response was the only metadata that was recorded, so the individual responses could not be traced back to individual participants nor customers (e.g. correlating response IP-addresses to customers was not possible).
- No analysis and conclusions were based on individual responses; the results were aggregated and analyzed as groups (e.g. customers versus analysts); data analysis was therefore anonymous
- The invite link to Qualtrics was sent via Fox-IT's own Cyber Threat Management platform, which can only be reached via two-factor authentication. This is considered the most secure option of distribution, since (unsigned and unencrypted) email cannot be trusted. Qualtrics is a supplied and approved tool by the University of Twente and all guidelines for secure data collection have been taken into account[4].
- The password for the Qualtrics environment was a randomly generated string of 40 characters and stored on an encryped drive at Fox-IT. Data retention and management was therefore considered safe.
- Fox-IT attaches great importance to the anonymity & privacy of the participants, and the confidentiality of the data, as it is their core business. For example, the list of 101 customers to which the survey was sent is also not known to the graduation committee. The author was also not able to map individual responses to individual persons. Furthermore, it was also in the interest of the company to handle the participants with care, since it directly involves their customer base and they obviously want to keep in a good relationship with them.
- Participants were informed about the facts listed above

Therefore, the research is in line with the guidelines provided by the BMS ethics committee[5].

[4] https://www.utwente.nl/en/bms/datalab/datacollection/surveysoftware/qualtrics

[5] https://www.utwente.nl/en/bms/research/ethics/

Chapter 14

Analyzing & discussing the survey

This chapter describes and discusses the results of the survey analysis for each of the four sub-subquestions. First, the process of the data (pre)processing is described in Section 14.1, after which the four sub-subquestions are covered in respectively Section 14.2, Section 14.3, Section 14.4 and Section 14.5.

14.1 Data (pre)processing

As a result of the survey design, the data could be extracted in CSV format from Qualtrics. Note that there are two values for N; one for the amount of customer responses, and an N for the amount of analyst responses. Since not all respondents had completed the survey for 100%, the responses containing any empty values were dropped and not used in the analysis. The impact labels used in the survey were then converted to corresponding numerical values, as displayed in Table 13.1. This resulted in a total of 530 data points submitted by customers, as well as 220 data point submitted by analysts. Per question, two arrays were constructed consisting of respectively the customer and analyst data. Afterwards, the mean and standard error values were calculated. Furthermore, two arrays held the data points covering all questions, which were used to calculate the statistical tests over the aggregated data points.

14.2 Impact label questions

14.2.1 Results

Per question, the two arrays were used to perform the test specified in Subsection 13.1.2. In Table 14.1 the results of all these individual tests are displayed. Furthermore, Figure 14.1 visualizes the data point of both the customers and analysts in a bar chart.

Incidents	Customers			Analysts			Mann-Whitney test	
	N	Mean	Std. err	N	Mean	Std. err	U	p
Adware	53	3.15	0.12	22	2.63	0.12	390	0.016
Port scanning	53	4.03	0.12	22	4.31	0.13	677	0.232
Trojan	53	4.77	0.10	22	4.81	0.12	585	0.984
TOR usage	53	4.01	0.15	22	3.59	0.12	430	0.059
Hacktool update	53	3.73	0.16	22	3.95	0.15	628	0.587
Open for abuse	53	4.20	0.12	22	3.68	0.17	414	0.037
Suspicious application	53	4.52	0.12	22	4.86	0.16	679	0.232
USB executable	53	3.69	0.15	22	3.72	0.20	574	0.917
Leaked credentials	53	4.58	0.13	22	4.86	0.15	657	0.366
Sensitive data leaked	53	4.90	0.15	22	4.50	0.19	432	0.067
Aggregated	530	4.16	0.05	220	4.09	0.06	55450	0.272

Table 14.1: The calculated values for the mean, standard error and Mann-Whitney tests for the impact label questions

14.2.2 Discussion

As can be derived from Table 14.1, in the case of the *Adware* and *Open for abuse* incidents, the p-value for the Mann-Whitney U test is smaller than 0.05, which means that H_0 can be rejected and the alternative hypothesis can be accepted. It should be noted that the values for *TOR usage* and *Sensitive data leaked* approach the value of α, and could also

In the case of **Adware** and **Open for abuse** incidents, the distribution of impact labels is not equal in both samples. The distribution of impact labels is equal in the other eight incidents. The aggregated distributions of impact labels of both samples did not significantly differ ($p = 0.272 > 0.05$).

This means that customers and analysts have a different perception of the severity of *Adware* and *Open for abuse*. The distributions of two other incidents were not significantly different (*TOR usage* and *Sensitive data leaked*), but might still be worth looking into.

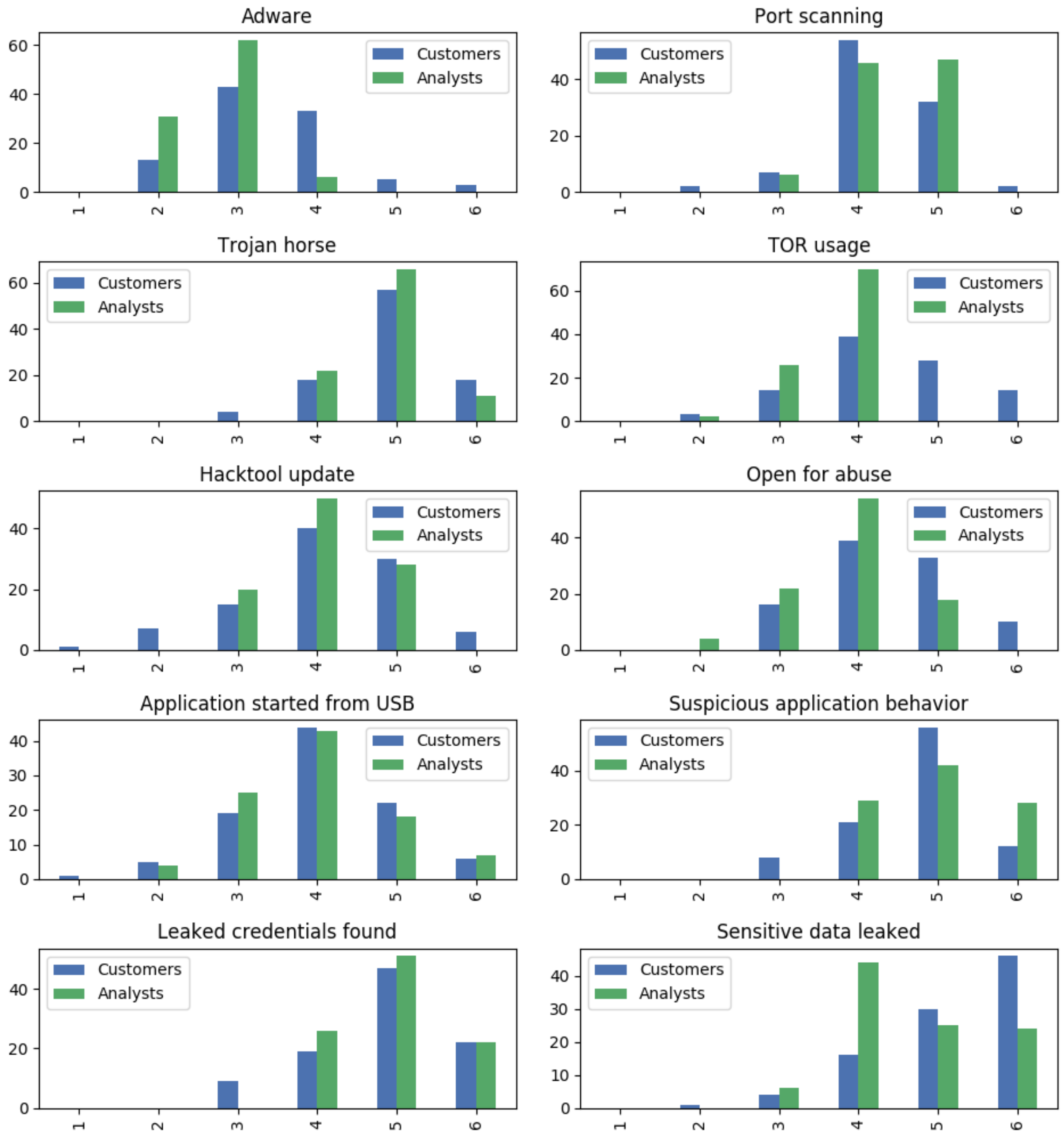


Figure 14.1: Bar chart showing the distribution of impact labels assigned by customers and analysts

14.3 Notification preference questions

14.3.1 Results

The statistical tests were performed and the results are shown in Table 14.2. Graphs representing the results for both the detailed report & phone notification can be found in Figure 14.2 and Figure 14.3.

Response	Type	Customers			Analysts			Chi-Square test	
		Yes	No	Total	Yes	No	Total	χ^2	p
No response	Report	1	6	7	0	0	0	–	–
	Phone	0	7	7	0	0	0	–	–
Elective	Report	7	27	34	6	8	14	1.49	0.222
	Phone	8	26	34	2	12	14	0.10	0.745
Expedited	Report	50	47	97	32	10	42	6.37	0.012
	Phone	25	72	97	19	23	42	4.27	0.039
Urgent	Report	135	41	176	76	13	89	2.24	0.134
	Phone	127	49	176	70	19	89	0.98	0.320
Immediate	Report	144	15	159	56	3	59	0.57	0.448
	Phone	147	12	159	59	0	59	3.37	0.066
Start IR	Report	53	4	57	16	0	16	0.21	0.640
	Phone	51	6	57	16	0	16	0.70	0.401

Table 14.2: The calculated values for the mean, standard error and Chi-Square tests for the impact label questions

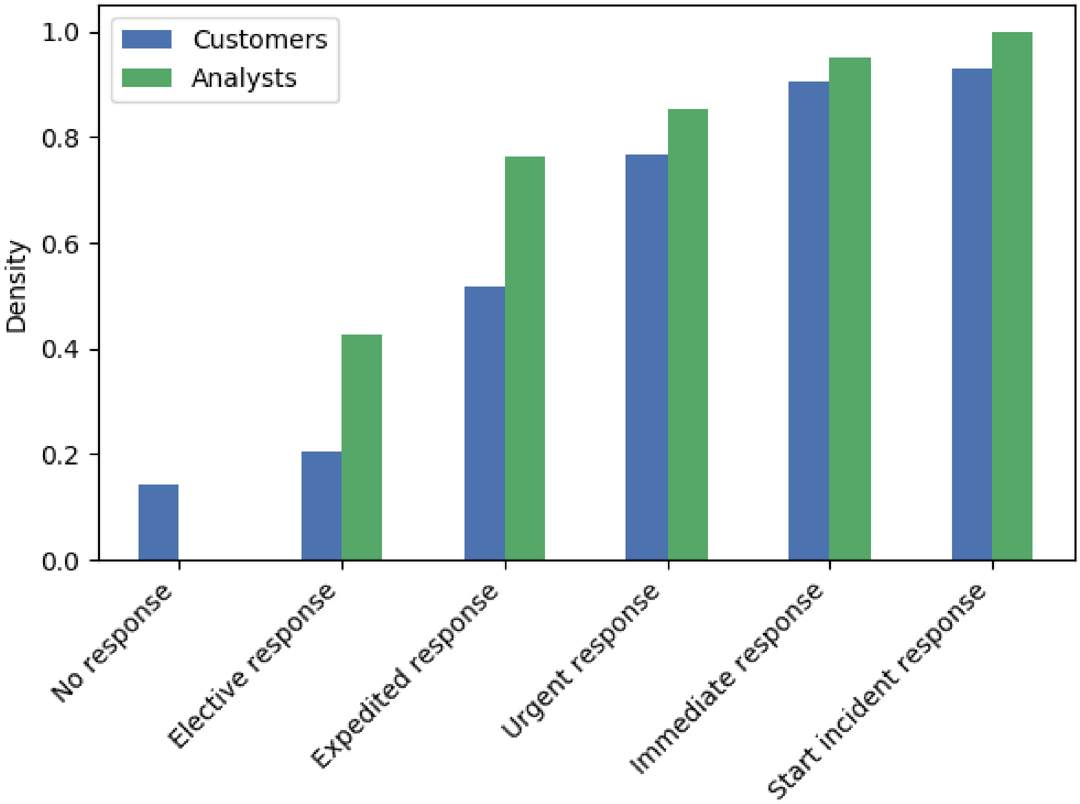


Figure 14.2: Bar chart displaying the answer to the question: *Would you write a detailed report about this incident*, with on the y-axis the density of the answers and on the x-axis the impact labels that were selectable.

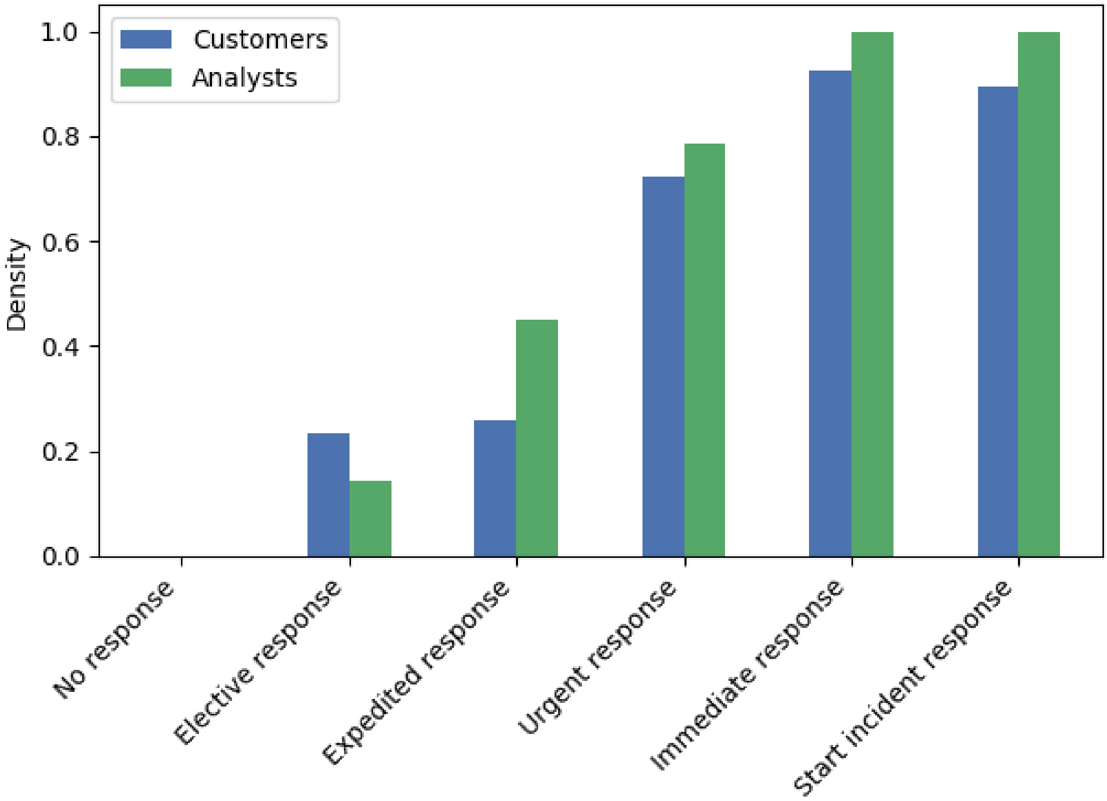


Figure 14.3: Bar chart displaying the answer to the question: *Would you notify the customer by phone about this incident*, with on the y-axis the density of the answers and on the x-axis the impact labels that were selectable.

14.3.2 Discussion

As can be seen in the Figure 14.2 and Figure 14.3, there are several differences in preferences between what analysts want to report, and customers want to receive. From the graphs it becomes clear that analysts more often want to write a detailed report, when compared to customers. This also holds for most of the phone notification preferences. Since the graphs give a good visual representation but are not useful for making statistical conclusions, Table 14.2 shows the outcomes for the statistical tests that were defined in Subsection 13.1.3.

When looking at the p-values for the ten performed tests, it becomes clear that in two cases the null hypothesis H_0 can be rejected:

In the case of an **Expedited response**, the distribution of notification preferences is not equal in both samples. This holds for both the detailed report ($p = 0.012 < 0.05$) and the phone notification ($p = 0.039 < 0.05$). The distribution of notification preferences is equal in the other eight incidents.

Note that there are not tests were performed for the `No response` impact label since not a single analyst has chosen this label at least once, thus making it impossible to perform a Chi-Square test on.

It is interesting to see that exactly the impact label `Expedited response`, which does not have a direct mapping to the old impact labels, holds different notification preferences.

14.4 Weights of dimensions

14.4.1 Results

As described in Subsection 13.1.4 the differences between the original impact labels and the reassigned impact labels was calculated according to the simple formula $\Delta = L_n - L_o$, where L_n is the numerically coded reassigned impact label, and L_o the previously 'old' assigned impact label. Next, the results were split into the three incident types (Network, Endpoint & Open-source) and for each of the five dimensions a Mann-Whitney U test was performed. This resulted in 15 different tests. In the end, all three incident types were combined for five additional tests to conclude the overall difference. In the end, this resulted in 20 statistical tests, which are displayed in Table 14.3. The table is also graphically represented in Figure 14.4.

14.4.2 Discussion

The results show that multiple significant differences exist. Several conclusions can be made:

For the network incidents, the null hypotheses related to the **Timing** and **Age of source data** dimensions can be rejected. For the endpoint & open-source incidents, the null hypothesis can be rejected for the **Similar case** dimension. With all three incident types combined, the null hypothesis H_0 is rejected for the **Timing** and **Similar case** dimensions.

Type	Dimension	Customers			Analysts			Mann-Whitney test	
		N	Mean	Std. err	N	Mean	Std. err	U	p
Network	Timing	52	−0.46	0.14	21	0.09	0.06	726	0.009
	Asset	52	−1.67	0.16	21	−2.04	0.23	441	0.185
	Similar case	52	−0.65	0.12	21	−1.09	0.21	414	0.087
	Old data	52	−1.44	0.14	21	−0.76	0.19	735	0.017
	Triage confidence	52	−0.88	0.10	21	−0.76	0.21	630	0.268
Endpoint	Timing	50	−0.16	0.12	21	0.04	0.08	568	0.496
	Asset	50	−0.92	0.18	21	−0.95	0.24	553	0.713
	Similar case	50	−0.22	0.12	21	−1.00	0.20	295	0.002
	Old data	50	−1.00	0.12	21	−0.95	0.20	559	0.661
	Triage confidence	50	−0.50	0.16	21	−0.71	0.14	476	0.515
Open-source	Timing	50	−0.18	0.11	21	−0.14	0.07	537	0.855
	Asset	50	−0.78	0.17	21	−1.09	0.18	435	0.237
	Similar case	50	−0.44	0.16	21	−1.57	0.27	268	0.001
	Old data	50	−0.46	0.15	21	−0.66	0.26	472	0.488
	Triage confidence	50	−0.64	0.14	21	−1.00	0.20	432	0.213
All	Timing	152	−0.26	0.07	63	0.00	0.04	5491	0.035
	Asset	152	−1.13	0.10	63	−1.36	0.14	4338	0.260
	Similar case	152	−0.44	0.08	63	−1.22	0.13	2936	0.0001
	Old data	152	−0.97	0.08	63	−0.79	0.12	5262	0.236
	Triage confidence	152	−0.67	0.08	63	−0.82	0.10	4593	0.615

Table 14.3: The analyzed data from the dimension weight questions. For both the customer & analyst data, the means and standard errors are calculated, after which the differences between the two groups are calculated using the Mann-Whitney test.

In the ”Timing” dimensions, the analysts sometimes **increased** the impact label. In other words; they sometimes assessed the incident more severe instead of less. In Section 12.2 the analysts indicated that they sometimes hesitate to phone a customer outside office hours because of a higher threshold, the opposite can however be concluded from the data. An explanation might be the fact that for the network incident, a *Trojan* was used as an example. If trojan activity is observed outside office hours, this might indicate that the trojan is already successfully installed. Maybe analysts would assess a `Start incident response` impact label here instead of a `Immediate response` because of this. This explanation also holds for the endpoint incident, but not for the open-source incident.

The difference in the ”Similar case” dimension could possibly be explained by the routines performed by the analysts on a daily basis; they are used to assess similar cases as `Elective response` where according to the original guidelines, they do not need to contact the customer. This partially has to do with the alarm fatigue discussed earlier; ’flooding’ customers with similar cases might cause inconvenience

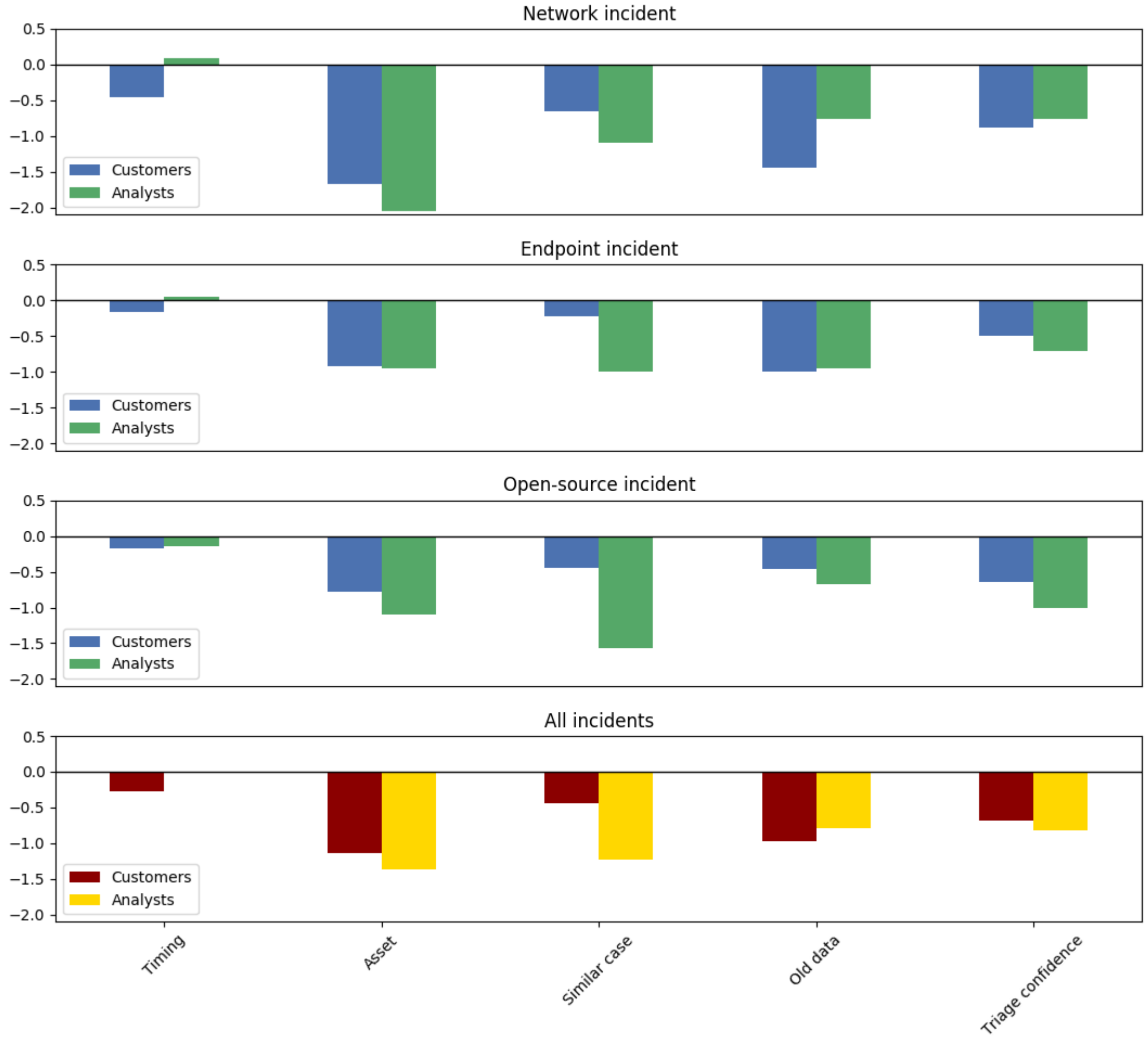


Figure 14.4: Graphs displaying the decrease in assigned impact albels for both customers and analysts, with on the x-axis the impact labels that were selectable and on the y-axis the average decrease in impact label

to the customer. An explanation might be that analysts are trained to prevent this alarm fatigue, while customers might not take the effect into account. Note that a ”Similar case” here means that it involves the same client and incident type, which from a analysts’ perspective looks like a duplicate and unnecessary case that has not been resolved yet, and therefore is not valueable to report again.

14.5 Demographic questions

14.5.1 Results

According to the methodology described in Subsection 13.1.5 the Kruskal-Wallis test was applied in this case. The data was processed in the same was as in SRQ2.1. The only difference was that no analyst responses were recorded and instead, the customer responses were split up in either the job titles, business sizes and business categories. Since there are ten incidents and three demographic questions, this resulted in 30 unique Kruskal-Wallis tests. The results of all the tests is displayed in Table 14.4.

Incidents	Job title	Business size	Business category
	p	p	p
Adware	0.39	0.25	0.23
Port scanning	0.01	0.26	0.19
Trojan	0.15	0.03	0.39
TOR usage	0.16	0.99	0.08
Hacktool update	0.27	0.77	0.17
Open for abuse	0.32	0.09	0.13
Suspicious application	0.37	0.67	0.37
USB executable	0.56	0.17	0.13
Leaked credentials	0.10	0.97	0.40
Sensitive data leaked	0.29	0.40	0.11

Table 14.4: The probability values (p) for the Kruskall-Wallis tests

14.5.2 Discussion

The results showed that only two out of 30 tests showed a significant difference (with $\alpha = 0.05$).

The test for differences in job titles regarding the *Port scanning* incident resulted in a very low p-value. However, after further examination, it turned out that there was one job titles that was only present once. This led to the conclusion that no actual difference was present, but rather a misleading calculation was made.

The other significant difference was observed at the test for different business sizes regarding the *Trojan* incident, and after double-checking this calculation turned out to be correct. This indicated a difference, but since it tests three samples it is unknown where the difference lies. Therefore, three additional Mann-Whitney U tests were performed among the three samples (10-249, 250-999, 1000+), in order to find the actual difference. The results of these tests are shown in Table 14.5.

#1	#2	Stats #1			Stats #2			Mann-Whitney test	
		N	Mean	Std. err	N	Mean	Std. err	U	p
10-249	1000+	12	5.25	0.13	22	4.50	0.18	66	0.009
250-999	1000+	18	4.77	0.19	22	4.50	0.18	164	0.327
250-999	10-249	18	4.77	0.19	12	5.25	0.13	144	0.087

Table 14.5: Three extra Mann-Whitney U tests

The results showed that the difference lies between the 10-249 the 1000+ size in employees, since $p = 0.009 < 0.05$. This leads to the following conclusion:

The distribution of impact labels for the **Trojan** incident is not equal for different business sizes. After further inspection using a Mann-Whitney U test, a significant difference between the sizes **10-249** and **1000+** was discovered.

Chapter 15

Conclusion

This chapter shows the conclusions that can be formulated from the results. Starting in Section 15.1, the answers to the initial problems are provided. Afterwards in Section 15.2 the conclusions regarding subquestion 2 are shown. Finally, in Section 15.3 the practical differences and advices for Fox-IT are displayed and suggestions for future work are given in Section 15.4.

15.1 Problems identified in current situation

Different impact labels were assigned to similar incidents, making the communication towards customers inconsistent
An analysis of historical incidents showed that this was indeed the case. It turned out that multiple dimensions influenced the decision making process for assigning an impact label to an incident, which caused similar incidents to be labeled differently.

The limited and too generic options to assign a threat category to an incident made comparison impossible
Multiple incident taxonomy frameworks were compared and in the end it was decided that the ENISA framework fulfilled all the requirements. For the remainder of the research, incidents were classified according to the ENISA framework.

The use of ambiguous and generic impact labels made it unclear to the customer what was actually meant with the impact label
To tackle this problem, new impact labels were proposed and put to the test in the survey. By choosing a clear definition of the incident severity (in this was, this was the ’time to respond’) maybe in the future incidents can be labeled more consistent.

Finally subquestion 1, defined as ”How do the current processes & preferences regarding incident reporting look like?” can be answered using the conclusions above. A combination of the current processes & preferences also resulted in the BPMN model shown in Appendix E, which summarized the conclusion to this subquestion.

15.2 Difference of preferences between analysts and customers

SRQ2.1: Would customers and analysts assign significantly different impact labels to similar situations?

In the case of **Adware** and **Open for abuse** incidents, the distribution of impact labels is not equal in both samples. The distribution of impact labels is equal in the other eight incidents. The aggregated distributions of impact labels of both samples did not significantly differ ($p = 0.272 > 0.05$).

SRQ2.2: Do customers and analysts have significantly different notification preferences?

In the case of an **Expedited response**, the distribution of notification preferences is not equal in both samples. This holds for both the detailed report ($p = 0.012 < 0.05$) and the phone notification ($p = 0.039 < 0.05$). The distribution of notification preferences is equal in the other eight incidents.

SRQ2.3: Do customers and analysts assign significantly different weights to the dimensions that influence the assessment of an incident?

For the network incidents, the null hypotheses related to the **Timing** and **Age of source data** dimensions can be rejected. For the endpoint & open-source incidents, this only holds for the **Similar case** dimension. With all three incident types combined, the null hypothesis H_0 is rejected for the **Timing** and **Similar case** dimensions. This means that in these cases, customers and analysts would assign significantly different weights to the dimensions.

SRQ2.4: Could any significantly different results regarding the assessment of incidents be observed when looking at the business category, business size or job title?

The distribution of impact labels for the **Trojan** incident is not equal for different business sizes. After further inspection using a Mann-Whitney U test, a significant difference between the sizes **10-249** and **1000+** was discovered. Regarding the business categories and job titles, no significant differences were found.

These sub-subquestion in turn provide an answer to subquestion 2 in general.

15.3 Creating the advice for Fox-IT

Now that data about the customer preferences is available, the current preferences and guidelines that are present at Fox can be compared to the customers' preferences.

First, the means of the impact labels assigned by the customers to the ten different incidents (discussed in Section 14.2) are shown in Figure 15.1. The mean values are accompanied by the values that are currently being used at the SOC. These values are constructed from the data about the historical cases, as listed in Table 12.4. This data is obviously built on the guidelines provided in Table 12.3. For example, *Adware* incidents were reported 605 times as the old label `Low risk`, three times as `High risk` and zero times as `Successful hack attack`. Using the converting scheme listed in Table 15.1, the old impact labels can be 'converted' to the new ones.

Old label	New label	Numeric value
Successful hack attack	Start incident response	6
High risk	Immediate response	5
Low risk	Urgent response	4

Table 15.1: Conversion scheme to compare old labels to new labels

Such a conversion is possible, since the new labels were designed in order to be compared to the old impact labels, as described in Section 12.5. Thus, the mean value for adware is calculated as:

$$\frac{(605 \cdot 4) + (3 \cdot 5) + (0 \cdot 6)}{608} = 4.01$$

Furthermore, the historically assigned labels can also be seen as a sample group which means that the distribution of the historically assigned impact labels can be compared to the distribution of the impact labels assigned by the customers. Again, the Mann-Whitney U test can be used to test the differences in the mean of both the customers and the historical data. The results of these tests is shown in Figure 15.1.

Threat category	**Mean of customers**	**Mean historical data**	U	p
Adware	3.16	4.01	6056	0.001
Port scanning	4.04	4.89	2605	0.001
Trojan	4.77	4.69	5081	0.128
TOR usage	4.02	4.89	3256	0.001
Hacktool update	3.73	4.03	2408	0.053
Open for abuse	4.21	4.13	1461	0.340
Suspicious application behavior	4.52	4.18	1151	0.004
USB executable started	3.69	4.05	8270	0.029
Leaked credentials	4.58	4.33	579	0.078
Sensitive data leaked	4.90	4.38	944	0.003

Figure 15.1: The ten selected incidents used in the survey

This leads to the following conclusion, which can be used to answer the main research question:

RQ: Which actions should Fox-IT take in order to align their incident reporting decision making with the analysts & customers preferences?

With a significance level of $\alpha = 0.05$ the distributions of impact labels is not equal between the customers and the historical data in the case of **Adware**, **Port scanning**, **TOR usage**, **Suspicious application behavior**, **USB executable started** and **Sensitive data leaked**.

In other words; the impact labels that were assigned to the five mentioned threat categories in the past do not align with the impact labels that customers currently would assign. The advice for Fox-IT is therefore to reconsider the guidelines for assigning impact labels to threat categories on a regular basis, since the preferences of the customers may change over time. Note that only ten out of the total 25 threat categories were tested in this research; there might be more threat categories that are currently being reporting using a wrong impact label.

This could be very beneficial to Fox-IT in terms of time & money, since it turns out that the top 5 of most frequent accounts for approximately 65% of the total incidents that are being reported! As an example, consider the *Adware* threat category that is currently being reported as `Low risk`, equivalent to `Urgent response` in terms of the impact labels used in this study. Adware incidents make up almost **25%(!)** of the total amount of reported incidents. Currently, for each adware incident, customers need to be notified by phone and a detailed report has to be written by one of the SOC analysts, costing time & money. The survey results showed that customers would classify adware as `Expedited response`, 52% of the customers would want to receive a report for `Expedited response` incidents and only 26% would want to be notified by phone. If Fox-IT decides to lower the severity of adware cases to a level where no report nor call have to be made, this would instantly save 25% of the reported incidents, leaving more time for SOC analysts to perform other tasks.

15.4 Future work

Further research into the communication between analysts and customers could include other threat categories, impact label design and other demographic variables. A lot of scientific research has been done in the field of alert processing by SOC analysts, but this does not focus on the communication between analysts and customers. Future research could also adapt other social science constructs into the cyber domain.

The recommendations for Fox-IT are to regularly repeat this research, since customers' and analysts' perceptions of incident severity may change over time. It is a good practice to continuously validate the guidelines for mapping the threat categories to impact labels. Since these guidelines are still rather generic and may not apply to every individual customer, adapting individual Customer Security Policies to the needs of specific customers will remain required.

Bibliography

[1] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen, and J. Disso, "Cyber-attack modeling analysis techniques: An overview," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, Aug 2016, pp. 69–76.

[2] P. Dreyer, T. Jones, K. Klima, J. Oberholtzer, A. Strong, J. W. Welburn, and Z. Winkelman, "Estimating the global cost of cyber risk," 2018.

[3] "Fox-it | for a more secure society," https://www.fox-it.com, accessed: 2018-09-07.

[4] Y. Ayrour, A. Raji, and M. Nassar, "Modelling cyber-attacks: a survey study," *Network Security*, vol. 2018, no. 3, pp. 13 – 19, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1353485818300254

[5] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, no. 1, p. 80, 2011.

[6] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Computers & Security*, vol. 72, pp. 212 – 233, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404817301839

[7] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis, "Hiding in plain sight: A longitudinal study of combosquatting abuse," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: ACM, 2017, pp. 569–586. [Online]. Available: http://doi.acm.org/10.1145/3133956.3134002

[8] "Combosquatting: The business of cybersquatting," Fairwind Partners LLC, Tech. Rep., 2008.

[9] O. van der Toorn, R. van Rijswijk-Deij, B. Geesink, and A. Sperotto, “Melting the snow: Using active dns measurements to detect snowshoe spam domains,” in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, pp. 1–9.

[10] R. Wieringa, *Design science methodology for information systems and software engineering*. Springer, 2014, 10.1007/978-3-662-43839-8.

[11] P. V. Mockapetris, “Domain names - concepts and facilities.” *RFC*, vol. 1034, pp. 1–55, November 1987. [Online]. Available: http://dblp.uni-trier.de/db/journals/rfc/rfc1000-1099.html

[12] E. Kidmose, E. Lansing, S. Brandbyge, and J. M. Pedersen, “Detection of malicious and abusive domain names,” in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, April 2018, pp. 49–56.

[13] “Sidn : Jouw wereld. ons domein.” https://www.sidn.nl/, accessed: 2018-09-21.

[14] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, “A survey on malicious domains detection through DNS data analysis,” *CoRR*, vol. abs/1805.08426, 2018. [Online]. Available: http://arxiv.org/abs/1805.08426

[15] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, “Exposure: A passive dns analysis service to detect and report malicious domains,” *ACM Trans. Inf. Syst. Secur.*, vol. 16, no. 4, pp. 14:1–14:28, Apr. 2014. [Online]. Available: http://doi.acm.org/10.1145/2584679

[16] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, “Phoenix: Dga-based botnet tracking and intelligence,” in *Detection of Intrusions and Malware, and Vulnerability Assessment*, S. Dietrich, Ed. Cham: Springer International Publishing, 2014, pp. 192–211.

[17] R. Sharifnya and M. Abadi, “Dfbotkiller: Domain-flux botnet detection based on the history of group activities and failures in dns traffic,” *Digital Investigation*, vol. 12, pp. 15 – 26, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287614001182

[18] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster, “Predator: Proactive recognition and elimination of domain abuse at time-of-registration,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’16. New York, NY, USA: ACM, 2016, pp. 1568–1579. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978317

[19] T. Vissers, J. Spooren, P. Agten, D. Jumpertz, P. Janssen, M. Van Wesemael, F. Piessens, W. Joosen, and L. Desmet, “Exploring the ecosystem of malicious domain registrations in the .eu tld,” in *Research in Attacks, Intrusions, and Defenses*, M. Dacier, M. Bailey, M. Polychronakis, and M. Antonakakis, Eds. Cham: Springer International Publishing, 2017, pp. 472–493.

[20] J. Szurdi and N. Christin, “Domain registration policy strategies and the fight against online crime,” *WEIS*, June 2018.

[21] Y.-M. Wang, D. Beck, J. Wang, C. Verbowski, and B. Daniels, “Strider typo-patrol: Discovery and analysis of systematic typo-squatting,” in *Proceedings of the 2Nd Conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*, ser. SRUTI’06. Berkeley, CA, USA: USENIX Association, 2006, pp. 5–5. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251296.1251301

[22] P. Piredda, D. Ariu, B. Biggio, I. Corona, L. Piras, G. Giacinto, and F. Roli, “Deepsquatting: Learning-based typosquatting detection at deeper domain levels,” in *AI*IA*, 2017.

[23] N. Nikiforakis, S. Van Acker, W. Meert, L. Desmet, F. Piessens, and W. Joosen, “Bitsquatting: Exploiting bit-flips for fun, or profit?” in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW ’13. New York, NY, USA: ACM, 2013, pp. 989–998. [Online]. Available: http://doi.acm.org/10.1145/2488388.2488474

[24] N. Nikiforakis, M. Balduzzi, L. Desmet, F. Piessens, and W. Joosen, “Soundsquatting: Uncovering the use of homophones in domain squatting,” in *Information Security*, S. S. M. Chow, J. Camenisch, L. C. K. Hui, and S. M. Yiu, Eds. Cham: Springer International Publishing, 2014, pp. 291–308.

[25] T. Holgers, D. E. Watson, and S. D. Gribble, “Cutting through the confusion: A measurement study of homograph attacks,” in *Proceedings of the Annual Conference on USENIX ’06 Annual Technical Conference*, ser. ATEC ’06. Berkeley, CA, USA: USENIX Association, 2006, pp. 24–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267359.1267383

[26] J.-W. Bullee, L. Montoya, M. Junger, and P. Hartel, “Spear phishing in organisations explained,” *Information and Computer Security*, vol. 25, no. 5, pp. 593–613, 7 2017.

[27] P. Lv, J. Ya, T. Liu, J. Shi, B. Fang, and Z. Gu, “You have more abbreviations than you know: A study of abbrevsquatting abuse,” in *Computational Science – ICCS 2018*, Y. Shi, H. Fu, Y. Tian, V. V. Krzhizhanovskaya, M. H. Lees, J. Dongarra,

and P. M. A. Sloot, Eds. Cham: Springer International Publishing, 2018, pp. 221–233.

[28] "Alexa top 1m," http://s3.amazonaws.com/alexa-static/top-1m.csv.zip, accessed: 2018-09-17.

[29] "Certificate transparency," https://transparencyreport.google.com/https/certificates, accessed: 2018-09-11.

[30] T. Segaran and J. Hammerbacher, *Beautiful data: the stories behind elegant data solutions*. " O'Reilly Media, Inc.", 2009.

[31] "Pyenchant tutorial," https://faculty.math.illinois.edu/~gfrancis/illimath/windows/aszgard_mini/movpy-2.0.0-py2.4.4/manuals/PyEnchant/PyEnchant%20Tutorial.htm, accessed: 01-11-2018.

[32] "Slang dictionary - internet & text slang," http://www.noslang.com/dictionary/, accessed: 01-11-2018.

[33] "List of swear words, bad words, & curse words - starting with a," http://www.noswearing.com/dictionary, accessed: 01-11-2018.

[34] "Sowpods scrabble word list," https://www.wordgamedictionary.com/sowpods/, accessed: 01-11-2018.

[35] D. Chiba, K. Tobe, T. Mori, and S. Goto, "Detecting malicious websites by learning ip address features," in *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet*, July 2012, pp. 29–39.

[36] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," in *Proceedings of international conference on intelligence analysis*, vol. 5. McLean, VA, USA, 2005, pp. 2–4.

[37] A. DAmico and K. Whitley, "The real work of computer network defense analysts," in *VizSEC 2007*. Springer, 2008, pp. 19–37.

[38] C. Zhong, J. Yen, P. Liu, R. F. Erbacher, C. Garneau, and B. Chen, *Studying Analysts' Data Triage Operations in Cyber Defense Situational Analysis*. Cham: Springer International Publishing, 2017, pp. 128–169. [Online]. Available: https://doi.org/10.1007/978-3-319-61152-5_6

[39] M. Bierma, J. D. J. E. Doak, and C. Hudson, "Learning to rank for alert triage," in *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, May 2016, pp. 1–5.

[40] S. Sendelbach and M. Funk, “Alarm fatigue: a patient safety concern,” *AACN advanced critical care*, vol. 24, no. 4, pp. 378–386, 2013.

[41] C. P. Bonafide, R. Lin, M. Zander, C. S. Graham, C. W. Paine, W. Rock, A. Rich, K. E. Roberts, M. Fortino, V. M. Nadkarni *et al.*, “Association between exposure to nonactionable physiologic monitor alarms and response time in a children’s hospital,” *Journal of hospital medicine*, vol. 10, no. 6, pp. 345–351, 2015.

[42] R. S. Gutzwiller, S. Fugate, B. D. Sawyer, and P. A. Hancock, “The human factors of cyber network defense,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 59, no. 1, pp. 322–326, 2015. [Online]. Available: https://doi.org/10.1177/1541931215591067

[43] M. S. Wogalter, K. R. Laughery, and C. B. Mayhorn, “Warnings and hazard communications,” *Handbook of human factors and ergonomics*, vol. 4, pp. 868–894, 2012.

[44] D. Norman, *The design of everyday things: Revised and expanded edition*. Constellation, 2013.

[45] ——, “Design, business models, and human-technology teamwork: As automation and artificial intelligence technologies develop, we need to think less about human-machine interfaces and more about human-machine teamwork,” *Research-Technology Management*, vol. 60, pp. 26–30, 01 2017.

Notes

1. This combined thesis first started with a (technical) design research (*Detection of Combosquat Domains using Active DNS Measurements*, see Chapter 2). At the end of the first assignment, there was a transition to this social science research. As a result of this transition from one assignment into another, it has accidentally been overlooked by both the author as well as the supervisors that this survey involved human participants and therefore had to be formally approved by the ethics committee.

Appendix A

Generic model data

A.1 Manual trademark selection

n=107
soundcloud dropbox nbcsports nordstrom premierleague twitch alipay mailchimp skype aliexpress paypal siemens dailymail googleplus indiatimes tomshardware norton theguardian netflix quora playstation yandex ieee pinterest airbnb huffingtonpost flickr salesforce bankofamerica pastebin instagram tripadvisor foxnews wikipedia github nvidia apple americanexpress youtube fox-it twitter stackoverflow dailymotion office365 facebook spotify usatoday snapchat microsoftonline hewlett steampowered duckduckgo homedepot hdfcbank thestartmagazine slideshare walmart bloomberg epicgames samsung amazonaws autodesk packard nytimes godaddy alibaba mediafire expedia wordpress linkedin breitbart amazon tumblr marktplaats fujitsu elsevier filehippo ladbible google wellsfargo reddit nokia mozilla symantec mcafee microsoft shopify whatsapp avast utwente gamepedia verizon cloudfront 4shared adobe stackexchange gitlab leagueoflegends lenovo wetransfer wiktionary

A.2 Frequent combosquatting words

Taken from the paper by Kintis et al. [7]
car universal square villa cheap marketing search porno account print office content vacation official listen wire hot shipping worldwide county services pilgrim net free videos san shop plus sex business fuck health group maps online delivery apps phone channel play princess watch kindle support post home com wireless mobile island theme freight inn news foundation posting president vote yeah archive service photography gift glass store photos club south sale express trump tube life jobs energy mortgage mike file sucks world elect center ground galaxy views live update user xxx campaign garden stores time cards deals page media zine university blog

hotel trust login family movies movie buy just card head best line video stop lay gay land love google real music chill themes beach cars estate followers porn school city paris download games prices credit hotels mail bank price property tex truth new phones canada chemical movie photo converter investment

A.3 Public blacklists

This table shows the blacklists that were used when answering CTD1.

Source
http://www.abuse.ch/
http://www.malwaredomains.com/
http://www.malwaredomains.com/
http://dns-bh.sagadc.org/
https://isc.sans.edu/suspicious_domains
http://www.urlvir.com/exporthosts/
http://www.nothink.org/blacklist/blacklist malware dns.txt.
http://www.joewein.net/dl/bl/dom-bl.txt.

Appendix B

Python code snippets

This appendix holds several code snippets, used in the thesis. They can be used to clarify the methodology and to see what was actually calculated. Note that the imports are not shown, as well as code that is not useful to show, e.g. code that loads and saves DataFrames, trademarks, words and more.

B.1 OpenINTEL queries

```
SELECT
lower(query_name) AS domainname,
count(case response_type when 'A' then 1 else null end) AS number_of_A_records,
count(case response_type when 'AAAA' then 1 else null end) AS number_of_AAAA_records
count(case response_type when 'NS' then 1 else null end) AS number_of_NS_records,
count(case response_type when 'MX' then 1 else null end) AS number_of_MX_records,
count(case response_type when 'SOA' then 1 else null end) AS number_of_SOA_records,
count(case response_type when 'CNAME' then 1 else null end) AS number_of_CNAME_recor
count(case response_type when 'DNSKEY' then 1 else null end) AS number_of_DNSKEY_rec
count(case response_type when 'TXT' then 1 else null end) AS number_of_TXT_records,
count(ip4_address) AS number_of_ipv4_addresses,
concat("[", group_concat(ip4_address), "]") AS ipv4_addresses,
count(case query_name when response_name then 1 else null end) AS response_name_matc
max(country) as country,
max(`as`) as as_number,
avg(soa_refresh) AS soa_refresh,
avg(soa_retry) AS soa_retry,
avg(soa_minimum) AS soa_minimum
FROM openintel.com_warehouse_parquet
WHERE year = [current_year] AND month = [current_month] AND day = [current_day]
AND lower(query_name) NOT LIKE '123-nonexistant-dnsjedi-456.%'
AND lower(query_name) NOT LIKE 'www.%'
AND regexp_like(lower(query_name), [manual_trademarks_regex])
AND regexp_like(lower(query_name), [frequent_combosquat_words_regex])
```

```
GROUP BY lower(query_name)
```

B.2 Validation of real-world domains

Every domain has a predicted value True or False, provided by the trained classifier. The code below calculates the `actual` value for the domain. In the end, every domain has a `predicted` and `actual` boolean, which are then used to construct the confusion matrix.

```
'''
Function for validating whether the predicted domains are actual combosquat domains or no
Consists of a few simple checks.
'''

def is_combosquat(domainname):
    contains_trademark = alexaregex.search(domainname)

    # Check for constraint 1) and 2)
    if contains_trademark:
        trademark = contains_trademark.group(0)
        # Rule out typosquatting, so a levenshtein distance of 1. This (
        # partially) checks constraint 5)
        if distance.levenshtein(trademark, domainname) == 1:
            return False

        segmented_domainname = wordsegment.segment(domainname)

        # Segmented_domainname contains a list of segments here. Now,
        # check for constraint 3)
        is_standalone_word = False

        for segment in segmented_domainname:
            if alexaregex.fullmatch(segment):
                is_standalone_word = True
                break

        if is_standalone_word:
            # Now we only have to check whether the two IP's are not
            # in the same range and the AS numbers do not match.
            # This checks constraint 4)
            return not ip_and_as_match(trademark, domainname):

        # If it's not a standalone word, dismiss it. E.g.
        # 'applejuice.com' should not be included.
        else:
            return False

    # If no trademark is present in the domainname, dismiss it already
    else:
        return False

'''
For a given trademark_domain (e.g. amazon) and new_domainname (amazon-secure-login),
this function checks whether the AS numbers match and the IP of the new_domainname
is in the /16 range of the IPv4 of the trademark_domain
'''
def ip_and_as_match(trademark_domain, new_domainname):
    originaldomain_as = domains_with_features.loc[domains_with_features['domainname'
        == trademark_domain + '.com.']['as_number'].values[0]
    originaldomain_ips = domains_with_features.loc[domains_with_features['domainname
        == trademark_domain + '.com.']['ipv4_addresses'].values
    domainname_as = new_domains.loc[new_domains['domainname']
        == new_domainname + '.com.']['as_number'].values[0]
    domainname_ips = new_domains.loc[new_domains['domainname']
        == new_domainname + '.com.']['ipv4_addresses'].values

    ip_in_original_range = False
    as_numbers_match = originaldomain_as == domainname_as

    for originaldomain_ip in originaldomain_ips:
            network = ip_network(originaldomain_ip + "/16", strict=False)
            for domainname_ip in domainname_ips:
                if ip_address(domainname_ip) in network:
                    # Here, the IP of the domain is in the original range
                    ip_in_original_range = True

    return ip_in_original_range and as_numbers_match
```

B.3 Running the prototype, training & test phase

```
No backup for today available, creating new one..
Today is: 2019-01-27 12:44:15.613630:
So, the ground truth is based on 2019-01-25 12:44:15.613631
Performing Kerberos authentication for OpenINTEL ...
found keytab: /home/jjansen/oi_jjansen.keytab
OK

Querying OpenINTEL for combosquatting domains
```

```
Got query response, now writing to csv...
Written new info to combosquatdomains_2512019.csv
Data transferred from voordeur!

Importing abuse.ch [Elapsed Time: 0:00:01] |################| (Time:  0:00:01)
Importing hosts-file.net [Elapsed Time: 0:00:17] |##########| (Time:  0:00:17)
Importing nothink.org [Elapsed Time: 0:00:00] |#############| (Time:  0:00:00)
Importing malwaredomainlist delisted [Elapsed Time: 0:00:00] || (Time:  0:00:00)
Importing malwaredomainlist blacklist [Elapsed Time: 0:00:00] |#| (Time:  0:00:00)
Importing malwaredomains immortal [Elapsed Time: 0:00:00] |#| (Time:  0:00:00)
Importing malwaredomains default [Elapsed Time: 0:00:00] |##| (Time:  0:00:00)
Importing joewein [Elapsed Time: 0:00:00] |#################| (Time:  0:00:00)
Importing isc_sans_edu 1/3 [Elapsed Time: 0:00:00] |########| (Time:  0:00:00)
Importing isc_sans_edu 2/3 [Elapsed Time: 0:00:00] |########| (Time:  0:00:00)
Importing isc_sans_edu 3/3 [Elapsed Time: 0:00:00] |########| (Time:  0:00:00)
Importing urlvir.com [Elapsed Time: 0:00:00] |##############| (Time:  0:00:00)

Out of the 285327 domains, a total of 5274 is present on passive blacklists!
Adding 2637 Alexa & Random malicious domains
Done
Starting adding lexical features..
Done
Start adding contextual features..
Done
Applying one hot encoding..
Done

Datagram shape: (10548, 6106)
is_combosquatting
False    5274
True     5274
Name: is_combosquatting, dtype: int64

Shape before features selection:
(10548, 6106)
Shape after feature selection:
(10548, 676)
Selected features:
Index(['number_of_a_records', 'number_of_aaaa_records', 'number_of_ns_records',
       ’number_of_mx_records’, ’number_of_txt_records’,
       ’number_of_ipv4_addresses’, ’response_name_matches’, ’soa_refresh’,
       ’soa_retry’, ’soa_minimum’,
       ...
       ’ip4_encode_995’, ’ip4_encode_998’, ’ip4_encode_999’, ’ip4_encode_1005’,
       ’ip4_encode_1006’, ’ip4_encode_1008’, ’ip4_encode_1009’,
       ’ip4_encode_1011’, ’ip4_encode_1012’, ’ip4_encode_1017’],
      dtype=’object’, length=676)

###################################################
DecisionTreeClassifier
###################################################

Average FP rate: 22 %
Average precision: 77 %
Average accuracy: 78 %
Raw FP: 118
Raw TP: 421
Raw TN: 408
Raw FN: 106


###################################################
RandomForestClassifier
###################################################

Average FP rate: 17 %
Average precision: 81 %
Average accuracy: 80 %
Raw FP: 92
Raw TP: 424
Raw TN: 434
Raw FN: 102


###################################################
AdaBoostClassifier
###################################################
```

```
Average FP rate: 18 %
Average precision: 80 %
Average accuracy: 80 %
Raw FP: 101
Raw TP: 434
Raw TN: 425
Raw FN: 92


###################################################
KNeighborsClassifier
###################################################

Average FP rate: 26 %
Average precision: 74 %
Average accuracy: 76 %
Raw FP: 144
Raw TP: 432
Raw TN: 382
Raw FN: 95


###################################################
GaussianNB
###################################################

Average FP rate: 6 %
Average precision: 54 %
Average accuracy: 51 %
Raw FP: 34
Raw TP: 54
Raw TN: 492
Raw FN: 472


###################################################
BernoulliNB
###################################################
```

```
Average FP rate: 26 %
Average precision: 73 %
Average accuracy: 73 %
Raw FP: 141
Raw TP: 392
Raw TN: 385
Raw FN: 135


###################################################
MLPClassifier
###################################################

Average FP rate: 31 %
Average precision: 69 %
Average accuracy: 67 %
Raw FP: 166
Raw TP: 357
Raw TN: 360
Raw FN: 170


###################################################
SGDClassifier
###################################################

Average FP rate: 76 %
Average precision: 48 %
Average accuracy: 48 %
Raw FP: 406
Raw TP: 393
Raw TN: 120
Raw FN: 133


###################################################
GradientBoostingClassifier
###################################################
```

```
Average FP rate: 23 %
Average precision: 77 %
Average accuracy: 81 %
Raw FP: 126
Raw TP: 459
Raw TN: 400
Raw FN: 68


###################################################
ExtraTreesClassifier
###################################################

Average FP rate: 20 %
Average precision: 79 %
Average accuracy: 81 %
Raw FP: 109
Raw TP: 446
Raw TN: 417
Raw FN: 81

Added all classifiers with a FP rate lower than 10 percent and a precision higher than

Starting the Combosquat Detection Model testing phase!
Loading new domains backup..
Done
Only processing first 1000 entries
Enriching new domains with lexical features
Start Alexa filtering
Done
Starting adding lexical features..
Done
Start adding contextual features
Done with the WHOIS requests, now heading to the CT Log features
Done with the contextual features
Applying one hot encoding..
Shape before one-hot encoding: (10000, 31)
Done
Removing columns not present in training columns..
```

```
Adding dummy test columns for missing training columns..
Done
Total new domains: 10000
75 combosquat domains found using GaussianNB
Done

Starting the Combosquat Detection Model valdation phase!
Validation complete!
TP: 0
FP: 75
TN: 9910
FN: 15
```

Appendix C

Scraped blacklists

This table shows the blacklists that were included in the scraped blacklists, starting from **2016-07-08** and ending at **2019-01-11**.

Source
http://www.malwaredomainlist.com/hostslist/hosts.txt
http://www.malwaredomainlist.com/hostslist/delisted.txt
http://mirror1.malwaredomains.com/files/justdomains
http://www.joewein.net/dl/bl/dom-bl.txt
http://malc0de.com/bl/ZONES
https://zeustracker.abuse.ch/blocklist.php?download=domainblocklist
https://ransomwaretracker.abuse.ch/downloads/RW_DOMBL.txt
https://hosts-file.net/hphosts-partial.txt
https://palevotracker.abuse.ch/blocklists.php?download=domainblocklist (until 06-12-2016)
https://feodotracker.abuse.ch/blocklist/?download=domainblocklist
http://www.networksec.org/grabbho/block.txt
https://openphish.com/feed.txt
https://www.threatcrowd.org/feeds/domains.txt
https://urlhaus.abuse.ch/downloads/text/
http://osint.bambenekconsulting.com/feeds/c2-dommasterlist.txt
http://vxvault.net/URL_List.php

On a daily basis, the blacklists were fetched and stored in the following format:

domainname	source	date

This resulted in a total of 870 files (8.3GB). Since the total days between the first and last date is 907 days, 37 days were missing due to switching to an other machine and/or temporary measurement failures. Out of the 870 files, 3 files turned out corrupt, leaving a total of **867** files to work with.

Appendix D

Trademark distribution

D.1 Total days before blacklisted

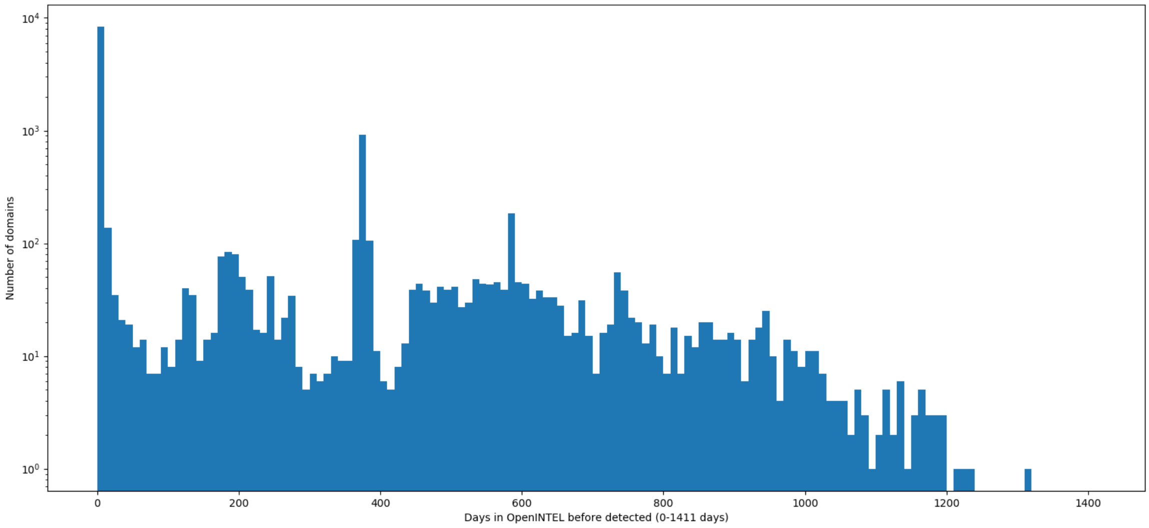
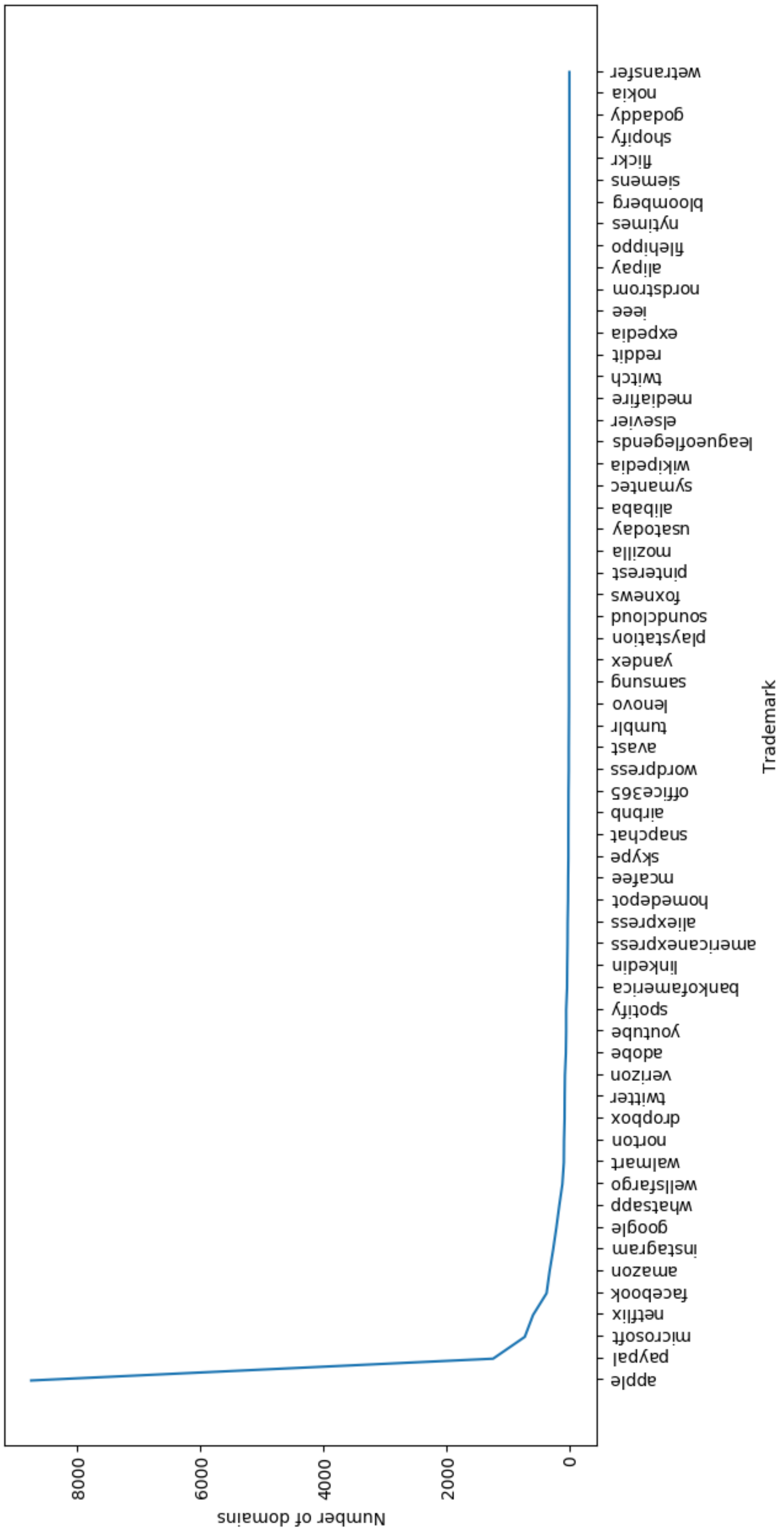


Figure D.1: The total number of days a combosquat domain is is present in OpenINTEL before being listed on a blacklist, with the y-axis in logarithmic scale.

D.2 Trademark frequency


Number of domains
0
2000
4000
6000
8000
Trademark
apple
paypal
microsoft
netflix
facebook
amazon
instagram
google
whatsapp
wellsfargo
walmart
norton
dropbox
twitter
verizon
adobe
youtube
spotify
bankofamerica
linkedin
americanexpress
aliexpress
homedepot
mcafee
skype
snapchat
airbnb
office365
wordpress
avast
tumblr
lenovo
samsung
yandex
playstation
soundcloud
foxnews
pinterest
mozilla
usatoday
alibaba
symantec
wikipedia
leagueoflegends
elsevier
mediafire
twitch
reddit
expedia
ieee
nordstrom
alipay
filehippo
nytimes
bloomberg
siemens
flickr
shopify
godaddy
nokia
wetransfer

Appendix E

Incident handling BPMN model

A BMPN process model of an analysts' incident handling task in the SOC is displayed on the next page.

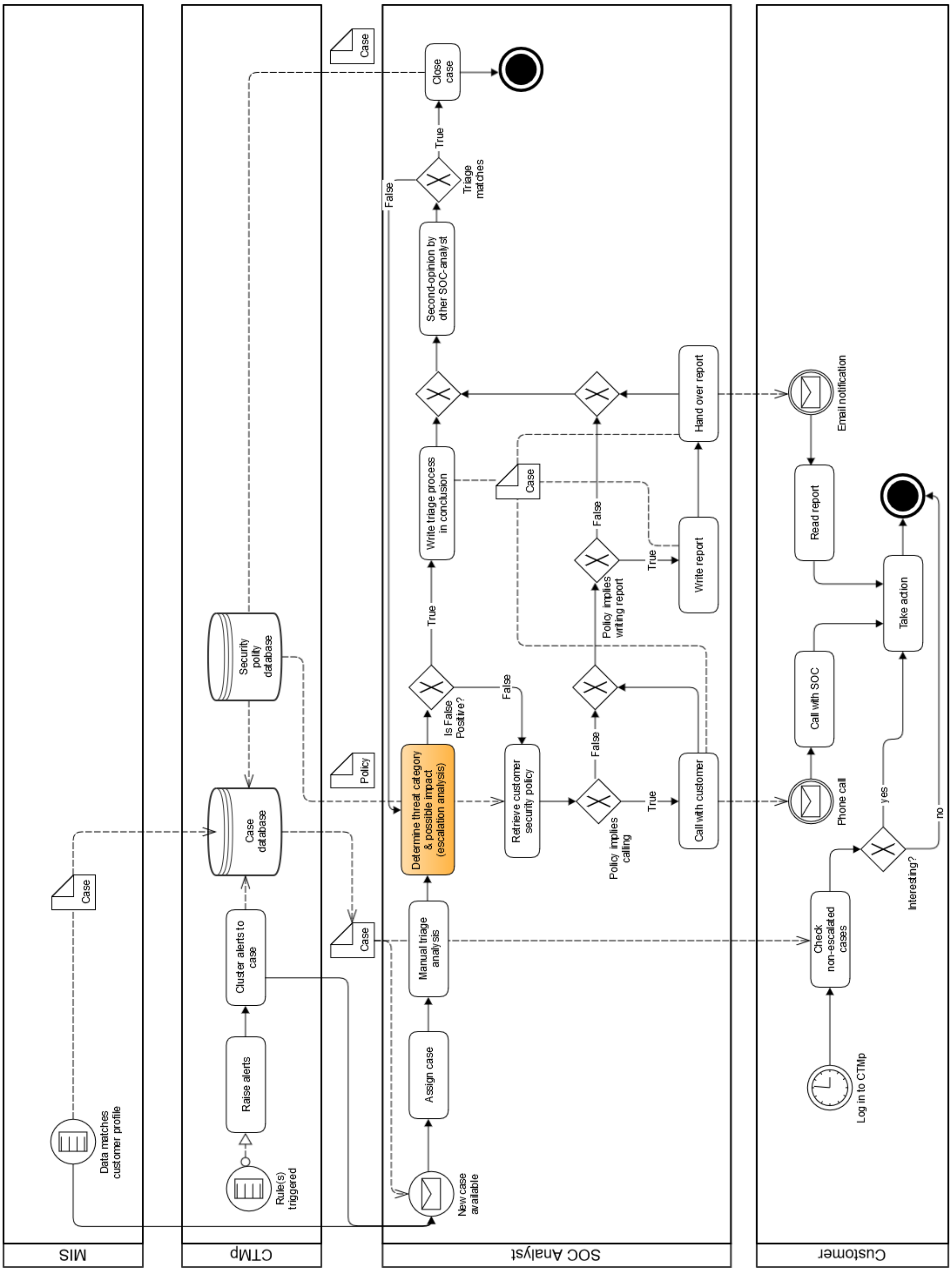
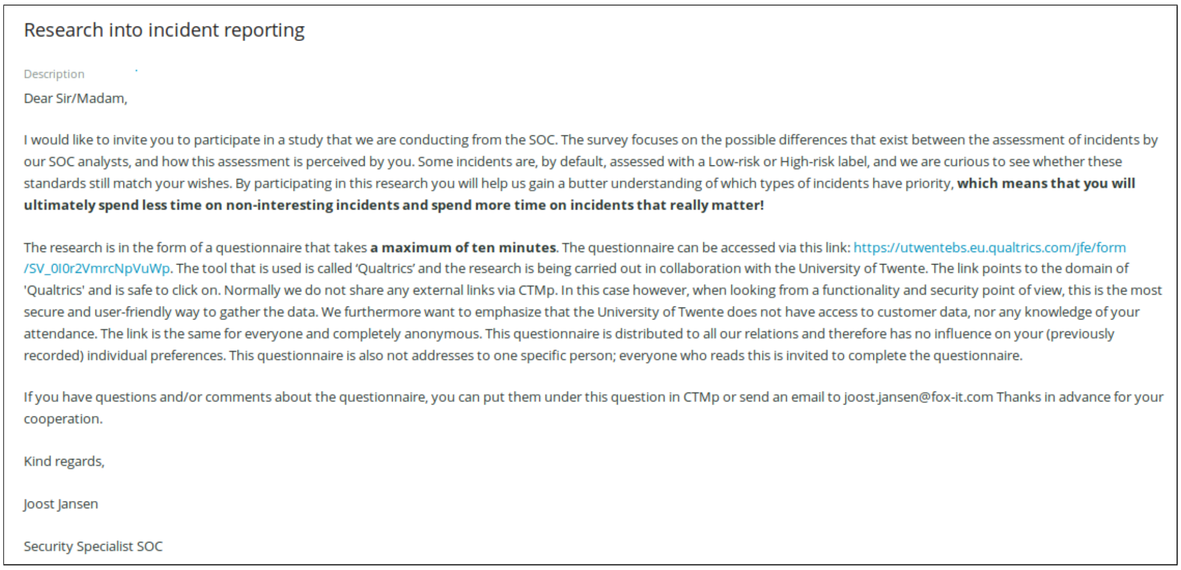


Figure E.1: BPMN model

Appendix F

Screenshots of the survey

This appendix provides some screenshots of the Qualtric survey, in order to visualize the experience. Figure F.1 shows the invitation that was sent via the CTMp. Next, Figure F.2 shows the introduction text, in which the new labels are presented to the participants. Following, Figure F.3 shows one of the ten pages on which a situation regarding a certain threat category was sketched, after which the participants were asked to assign an impact label. Furthermore, they were asked to indicate whether they would write a detailed report and/or notify a customer by phone. Lastly, Figure F.4 shows the page that listed the questions regarding the different dimensions.



Research into incident reporting

Description

Dear Sir/Madam,

I would like to invite you to participate in a study that we are conducting from the SOC. The survey focuses on the possible differences that exist between the assessment of incidents by our SOC analysts, and how this assessment is perceived by you. Some incidents are, by default, assessed with a Low-risk or High-risk label, and we are curious to see whether these standards still match your wishes. By participating in this research you will help us gain a butter understanding of which types of incidents have priority, **which means that you will ultimately spend less time on non-interesting incidents and spend more time on incidents that really matter!**

The research is in the form of a questionnaire that takes **a maximum of ten minutes**. The questionnaire can be accessed via this link: https://utwentebs.eu.qualtrics.com/jfe/form/SV_0I0r2VmrcNpVuWp. The tool that is used is called 'Qualtrics' and the research is being carried out in collaboration with the University of Twente. The link points to the domain of 'Qualtrics' and is safe to click on. Normally we do not share any external links via CTMp. In this case however, when looking from a functionality and security point of view, this is the most secure and user-friendly way to gather the data. We furthermore want to emphasize that the University of Twente does not have access to customer data, nor any knowledge of your attendance. The link is the same for everyone and completely anonymous. This questionnaire is distributed to all our relations and therefore has no influence on your (previously recorded) individual preferences. This questionnaire is also not addresses to one specific person; everyone who reads this is invited to complete the questionnaire.

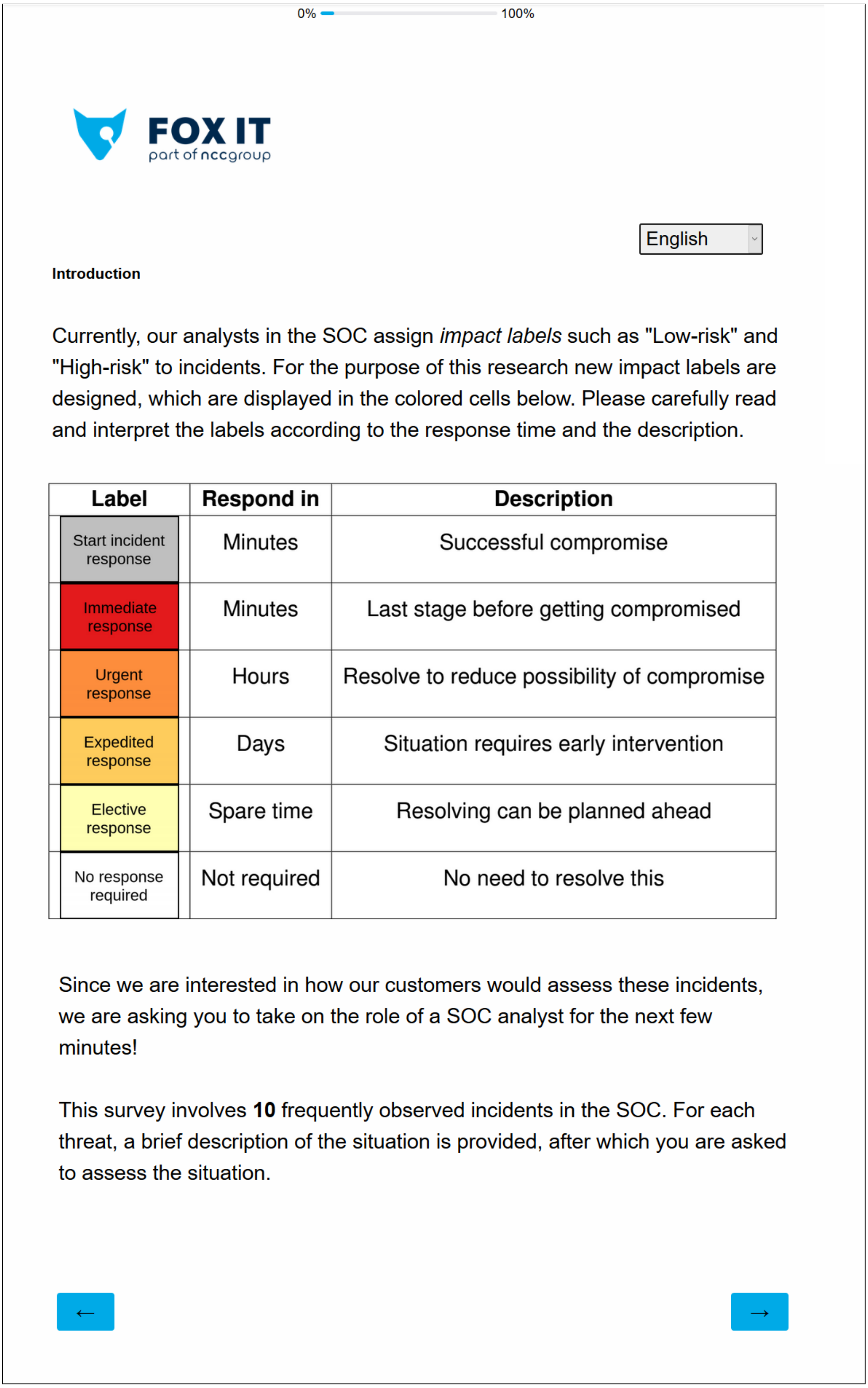
If you have questions and/or comments about the questionnaire, you can put them under this question in CTMp or send an email to joost.jansen@fox-it.com Thanks in advance for your cooperation.

Kind regards,

Joost Jansen

Security Specialist SOC

Figure F.1: The invitation in CTMp



Introduction

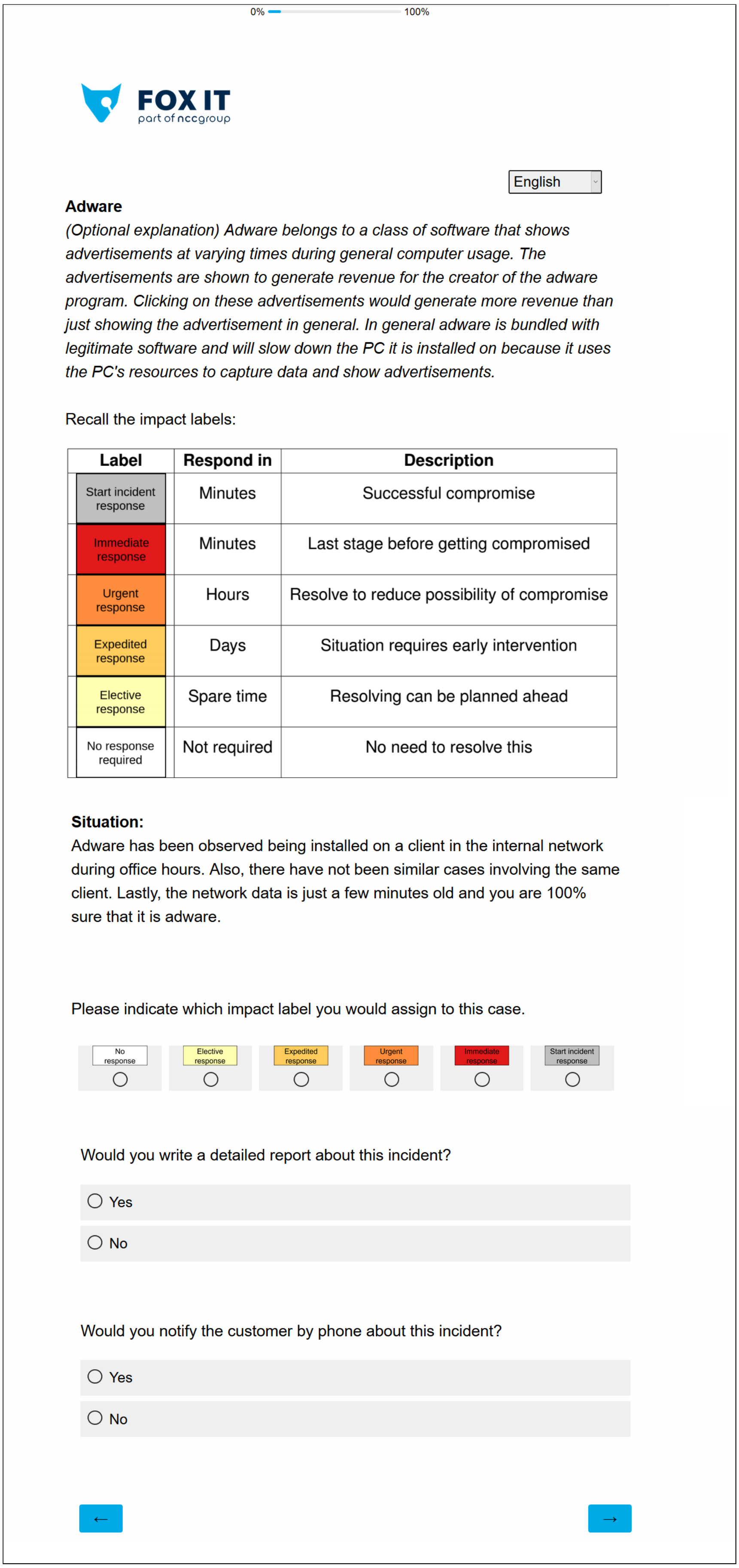
Currently, our analysts in the SOC assign *impact labels* such as "Low-risk" and "High-risk" to incidents. For the purpose of this research new impact labels are designed, which are displayed in the colored cells below. Please carefully read and interpret the labels according to the response time and the description.

Label	Respond in	Description
Start incident response	Minutes	Successful compromise
Immediate response	Minutes	Last stage before getting compromised
Urgent response	Hours	Resolve to reduce possibility of compromise
Expedited response	Days	Situation requires early intervention
Elective response	Spare time	Resolving can be planned ahead
No response required	Not required	No need to resolve this

Since we are interested in how our customers would assess these incidents, we are asking you to take on the role of a SOC analyst for the next few minutes!

This survey involves **10** frequently observed incidents in the SOC. For each threat, a brief description of the situation is provided, after which you are asked to assess the situation.

Figure F.2: Intro text displaying the new impact labels



FOX IT
part of nccgroup

English

Adware

(Optional explanation) Adware belongs to a class of software that shows advertisements at varying times during general computer usage. The advertisements are shown to generate revenue for the creator of the adware program. Clicking on these advertisements would generate more revenue than just showing the advertisement in general. In general adware is bundled with legitimate software and will slow down the PC it is installed on because it uses the PC's resources to capture data and show advertisements.

Recall the impact labels:

Label	Respond in	Description
Start incident response	Minutes	Successful compromise
Immediate response	Minutes	Last stage before getting compromised
Urgent response	Hours	Resolve to reduce possibility of compromise
Expedited response	Days	Situation requires early intervention
Elective response	Spare time	Resolving can be planned ahead
No response required	Not required	No need to resolve this

Situation:
Adware has been observed being installed on a client in the internal network during office hours. Also, there have not been similar cases involving the same client. Lastly, the network data is just a few minutes old and you are 100% sure that it is adware.

Please indicate which impact label you would assign to this case.

No response	Elective response	Expedited response	Urgent response	Immediate response	Start incident response
○	○	○	○	○	○

Would you write a detailed report about this incident?

○ Yes

○ No

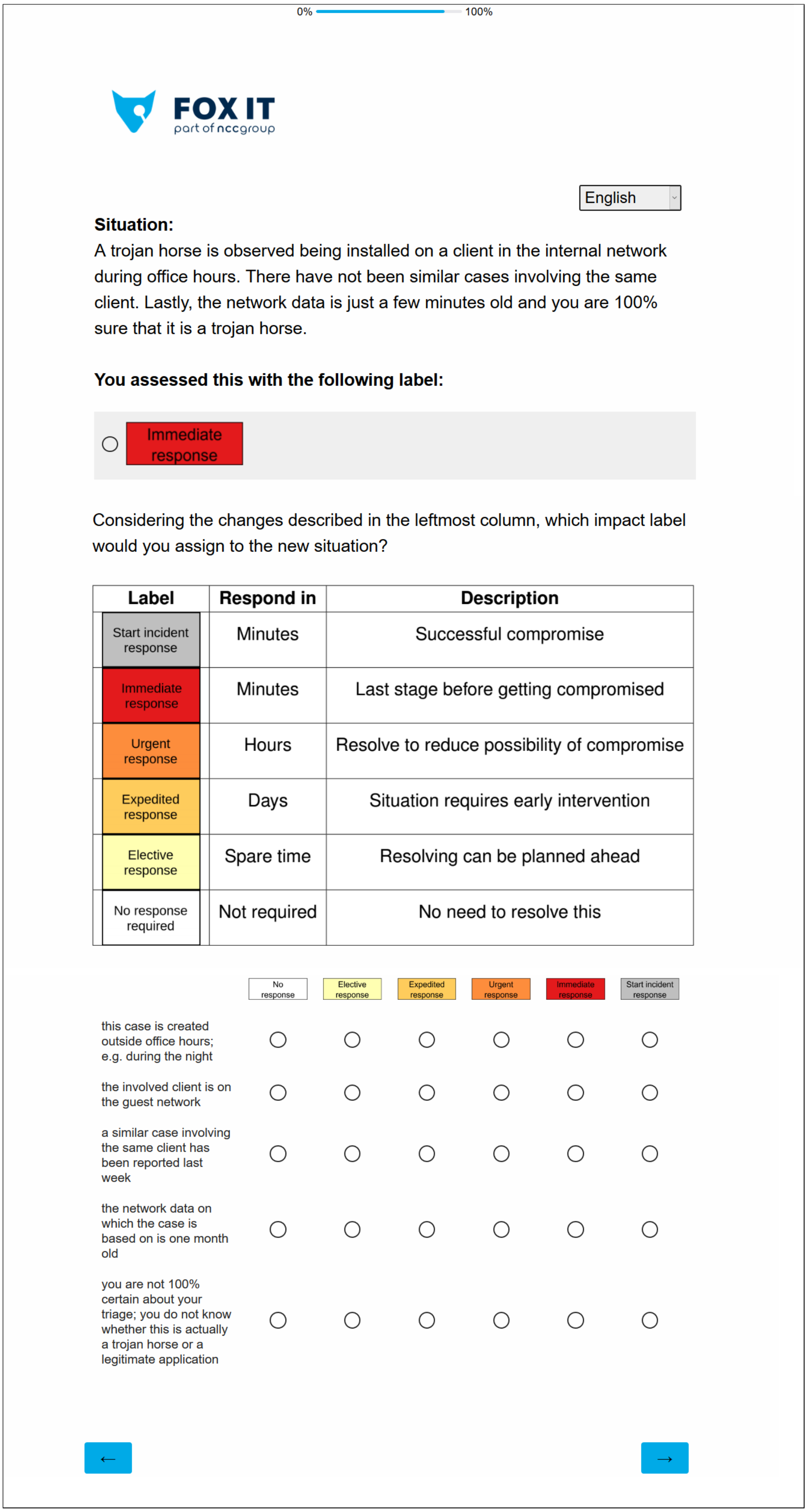
Would you notify the customer by phone about this incident?

○ Yes

○ No

← →

Figure F.3: Adware situation & questions



0% 100%

FOX IT
part of nccgroup

English

Situation:
A trojan horse is observed being installed on a client in the internal network during office hours. There have not been similar cases involving the same client. Lastly, the network data is just a few minutes old and you are 100% sure that it is a trojan horse.

You assessed this with the following label:

○ Immediate response

Considering the changes described in the leftmost column, which impact label would you assign to the new situation?

Label	Respond in	Description
Start incident response	Minutes	Successful compromise
Immediate response	Minutes	Last stage before getting compromised
Urgent response	Hours	Resolve to reduce possibility of compromise
Expedited response	Days	Situation requires early intervention
Elective response	Spare time	Resolving can be planned ahead
No response required	Not required	No need to resolve this

	No response	Elective response	Expedited response	Urgent response	Immediate response	Start incident response
this case is created outside office hours; e.g. during the night	○	○	○	○	○	○
the involved client is on the guest network	○	○	○	○	○	○
a similar case involving the same client has been reported last week	○	○	○	○	○	○
the network data on which the case is based on is one month old	○	○	○	○	○	○
you are not 100% certain about your triage; you do not know whether this is actually a trojan horse or a legitimate application	○	○	○	○	○	○

← →

Figure F.4: Situation describing a trojan infection and testing the different dimensions

Appendix G

Raw customer survey data

The raw data from the customers is displayed here. In the table to the right, the column headers are displayed, which correspond with the large table starting on the next page. So, in total there are 49 data columns and 53 full responses, ranging from index 2 to 68.

0	job_title
1	business_category
2	business_size
3	adware_label
4	adware_report
5	adware_phone
6	portscan_label
7	portscan_report
8	portscan_phone
9	trojan_label
10	trojan_report
11	trojan_phone
12	tor_label
13	tor_report
14	tor_phone
15	update_label
16	update_report
17	update_phone
18	open_label
19	open_report
20	open_phone
21	usb_label
22	usb_report
23	usb_phone
24	application_label
25	application_report
26	application_phone
27	creds_label
28	creds_report
29	creds_phone
30	dataleak_label
31	dataleak_report
32	dataleak_phone
33	trojan_var_timing
34	trojan_var_asset
35	trojan_var_similar
36	trojan_var_olddata
37	trojan_var_triageconf
38	application_var_timing
39	application_var_similar
40	application_var_olddata
41	application_var_triageconf
42	application_var_asset
43	creds_var_timing
44	creds_var_asset
45	creds_var_similar
46	creds_var_olddata
47	creds_var_triageconf

2	(IT) Manager / Team lead	Wholesale & retail	250 - 999	3	0	0	4	0	1	5
3	Network administrator	Public sector	1000+	5	1	1	4	1	0	5
4	(IT) Manager / Team lead	ICT	250 - 999	4	0	0	4	1	1	5
5	System administrator	Manufacturing	1000+	3	0	0	5	1	1	5
6	(IT) Manager / Team lead	Energy	250 - 999	3	1	0	4	1	1	4
7	Network administrator	Energy	250 - 999	2	0	1	4	1	1	5
8	Other (please specify)	Wholesale & retail	1000+	2	0	0	3	0	0	3
9	(IT) Manager / Team lead	Other services	10 - 249	3	0	0	5	0	1	5
10	Information security officer / coordinator	Finance	10 - 249	4	1	1	4	1	1	5
11	Network administrator	Public sector	250 - 999	6	0	0	2	0	1	4
12	System administrator	Energy	250 - 999	3	0	1	4	1	1	4
13	(IT) Manager / Team lead	ICT	10 - 249	3	0	0	4	0	1	5
14	CISO	ICT	10 - 249	3	1	0	4	1	0	5
15	Developer / Operations engineer (DevOps)	Finance	250 - 999	2	0	0	3	1	0	4
16	Information security officer / coordinator	ICT	250 - 999	4	1	0	4	1	1	5
17	Information security officer / coordinator	Transport	1000+	2	0	0	2	0	0	4
18	Information security officer / coordinator	Manufacturing	1000+	2	0	0	4	1	1	3
19	Information security officer / coordinator	ICT	1000+	3	0	0	5	1	1	5
20	Network administrator	Transport	250 - 999	3	0	1	5	1	1	6
21	Other (please specify)	ICT	250 - 999	4	1	0	2	0	0	4
22	CISO	Finance	10 - 249	4	1	1	4	1	0	5
23	(IT) Manager / Team lead	Entertainment	250 - 999	3	1	0	4	1	1	5
24	(IT) Manager / Team lead	Manufacturing	250 - 999	4	0	1	5	1	1	5
25	Information security officer / coordinator	Finance	250 - 999	2	0	0	4	1	1	3
26	(IT) Manager / Team lead	Wholesale & retail	1000+	2	0	1	4	1	1	5
27	Information security officer / coordinator	Finance	1000+	3	0	0	5	1	1	5
28	System administrator	ICT	1000+	5	1	0	5	1	0	5
29	Network administrator	ICT	10 - 249	4	1	0	4	1	0	5
30	Other (please specify)	Finance	1000+	1	1	0	3	1	1	4
31	Network administrator	Finance	1000+	3	0	0	4	0	0	4
32	Information security officer / coordinator	Other services	250 - 999	2	0	1	5	1	1	5
33	Network administrator	Health & social work	1000+	2	0	0	3	1	0	3
34	Information security officer / coordinator	Finance	1000+	3	0	0	4	1	1	5
35	NaN	NaN	NaN	3	0	0	1	0	0	5
36	System administrator	Finance	1000+	3	1	1	4	1	1	4
38	Information security officer / coordinator	Finance	1000+	4	0	0	5	1	1	5
39	Network administrator	Other services	1000+	3	0	1	4	0	1	4
41	(IT) Manager / Team lead	ICT	10 - 249	3	1	1	4	1	1	5
42	Other (please specify)	Finance	1000+	3	0	0	4	1	0	4
43	Information security officer / coordinator	Health & social work	1000+	3	0	0	4	1	0	5
45	CISO	Other services	1000+	4	1	0	4	1	1	6
46	(IT) Manager / Team lead	Finance	1000+	4	1	1	5	1	1	5
47	Other (please specify)	ICT	1000+	3	1	1	4	1	0	6
48	System administrator	Agriculture	1000+	2	0	0	4	1	1	4
49	(IT) Manager / Team lead	Other services	250 - 999	3	0	0	6	1	1	6
50	System administrator	Other services	250 - 999	4	0	1	4	1	1	5
58	Security consultant	Finance	10 - 249	3	1	0	5	1	1	5
60	CISO	Finance	10 - 249	3	0	0	4	1	1	6
62	(IT) Manager / Team lead	ICT	10 - 249	3	0	0	5	1	1	6
63	(IT) Manager / Team lead	Other services	10 - 249	4	0	1	5	1	1	6
64	Other (please specify)	ICT	250 - 999	2	1	1	4	1	1	6
67	System administrator	Agriculture	10 - 249	4	1	1	5	1	1	5
68	Network administrator	Health & social work	250 - 999	4	0	1	3	0	0	5

2	1	1	4	0	1	4	0	1	5	1	1	3	0	0	5	1	1	5	0	1
3	1	1	5	1	0	4	0	0	6	1	1	3	1	0	5	1	1	5	1	1
4	1	1	4	0	1	4	0	1	4	0	1	4	1	1	5	1	1	5	0	1
5	1	1	4	1	0	5	1	1	4	1	0	4	0	0	5	1	1	4	1	1
6	1	1	3	1	0	5	1	1	5	1	1	4	1	1	5	1	1	5	1	1
7	1	1	5	1	1	3	1	1	3	0	0	3	0	0	3	0	0	5	1	1
8	1	0	2	0	0	3	0	0	3	1	0	4	1	0	3	0	0	4	1	0
9	0	1	4	0	1	4	0	1	3	0	0	5	0	1	5	0	1	5	1	1
10	1	1	3	1	0	2	0	0	3	1	0	3	1	0	4	1	0	4	1	1
11	1	1	6	0	0	6	0	0	6	0	0	2	0	1	2	0	0	3	0	1
12	1	1	4	1	1	4	1	1	5	1	1	4	1	1	4	1	1	5	1	1
13	0	1	6	1	1	4	0	1	3	0	1	4	0	1	4	0	1	5	0	1
14	1	1	5	1	1	3	1	0	4	1	0	2	0	0	4	1	1	3	1	0
15	1	1	2	0	0	3	1	0	4	1	1	3	1	0	4	1	1	4	1	1
16	1	1	4	0	0	4	0	0	5	1	1	4	1	0	5	1	1	5	1	1
17	1	1	4	1	1	2	0	0	4	1	1	2	0	0	3	1	0	3	1	0
18	1	1	2	0	0	4	1	1	3	1	0	1	0	0	4	1	1	3	1	1
19	1	1	2	0	0	4	0	1	4	0	1	3	0	0	5	1	1	4	0	1
20	1	1	4	1	1	1	0	0	3	1	1	1	0	0	6	1	1	6	1	1
21	1	0	3	1	0	2	1	0	4	1	1	4	1	1	5	1	1	5	1	1
22	1	1	4	1	1	4	1	1	4	0	1	3	1	0	5	1	1	3	0	1
23	1	1	4	1	1	4	1	1	4	1	1	2	1	0	4	1	1	4	1	1
24	1	1	4	1	1	4	0	1	4	1	0	4	0	1	5	1	1	3	1	0
25	1	1	4	1	1	3	1	0	4	1	1	3	0	0	5	1	1	2	1	0
26	1	1	4	1	1	5	1	1	3	1	1	3	0	1	3	0	1	5	1	1
27	1	1	5	1	1	4	1	0	5	1	1	4	1	0	5	1	1	6	1	1
28	1	0	5	1	0	5	1	0	5	1	0	5	1	0	5	1	0	5	1	0
29	1	1	4	1	0	5	1	1	4	1	0	5	1	1	6	1	1	6	1	1
30	1	1	3	1	0	3	1	0	5	1	1	2	1	0	4	1	1	5	1	1
31	1	1	5	1	1	3	0	0	5	0	1	5	0	1	4	1	0	4	1	0
32	1	1	4	0	1	5	1	1	4	0	1	4	0	0	5	1	1	3	0	0
33	1	0	3	1	0	2	0	0	4	1	1	3	1	0	4	1	1	5	1	1
34	1	1	4	1	1	5	1	1	4	1	1	4	1	1	5	1	1	3	1	0
35	1	0	1	0	0	3	0	0	4	1	1	4	1	0	5	1	1	5	0	1
36	1	0	5	1	1	5	1	1	6	1	1	4	0	1	5	1	1	6	1	1
38	1	1	4	0	1	4	1	0	4	0	0	5	1	1	5	1	1	4	1	0
39	0	1	5	1	1	2	0	0	5	1	1	3	0	1	3	0	1	6	1	1
41	1	1	4	1	1	4	1	1	3	1	0	4	1	1	6	1	1	5	1	1
42	1	0	4	1	0	4	1	0	5	1	1	5	1	1	5	1	1	4	1	1
43	1	1	3	1	0	3	1	0	3	1	0	4	1	0	4	1	0	5	1	1
45	1	1	6	1	1	5	1	1	4	1	1	4	1	1	5	1	1	5	1	1
46	1	1	5	1	1	4	1	1	5	1	1	6	1	1	6	1	1	6	1	1
47	1	1	5	1	1	6	1	0	6	1	0	4	1	1	4	1	0	5	1	1
48	1	1	3	0	0	3	0	1	5	1	1	3	0	1	5	1	1	5	1	1
49	1	1	5	0	1	1	0	0	5	1	1	3	0	0	5	1	1	6	1	1
50	0	1	5	1	1	5	1	1	4	1	1	5	1	1	5	1	1	6	1	1
58	1	1	3	0	0	4	1	0	3	1	0	4	1	0	3	1	0	4	0	0
60	1	1	4	0	1	2	0	0	4	1	1	5	1	1	5	1	1	5	1	1
62	1	1	3	0	0	4	1	1	5	1	1	6	1	1	6	1	1	6	1	1
63	1	1	6	1	1	5	1	1	4	1	1	5	1	1	5	1	1	4	1	1
64	1	1	6	1	1	4	1	1	4	1	1	4	1	1	5	1	1	5	1	1
67	1	1	4	1	1	5	1	1	5	1	1	4	0	1	3	1	1	5	1	1
68	1	1	3	0	0	2	0	1	3	0	1	4	0	1	4	1	1	4	1	1

2	6	1	1	5	4	5	4	4	5	5	4	4	4	5	4	4	4	4
3	5	1	1	5	3	4	3	4	5	5	3	4	4	5	5	5	5	5
4	5	0	1	5	1	3	3	5	5	3	5	5	1	5	4	4	4	5
5	3	1	0	5	1	3	2	3	5	5	3	3	5	4	3	3	3	3
6	6	1	1	4	3	3	3	2	5	4	3	3	2	5	3	3	3	2
7	4	1	1	4	2	4	2	4	4	4	2	3	4	5	3	4	4	4
8	4	1	1	4	3	3	2	2	3	3	2	2	2	3	3	4	2	2
9	5	1	1	4	4	3	3	4	5	4	3	4	4	5	4	3	3	3
10	5	1	1	5	4	4	5	4	4	4	4	3	3	4	4	4	4	3
11	4	1	1	6	3	3	2	3	3	2	2	6	6	3	6	3	5	3
12	5	1	1	4	3	4	3	4	4	4	4	4	4	5	4	4	5	5
13	4	1	1	5	4	5	2	4	5	5	2	4	4	5	5	5	5	5
14	6	1	1	5	2	5	2	5	4	4	2	4	4	2	3	3	3	3
15	3	1	0	3	4	4	3	3	4	4	3	4	3	3	3	3	4	3
16	6	1	1	5	4	5	4	4	5	6	4	4	4	6	5	6	6	4
17	5	1	1	2	2	3	2	2	2	3	2	2	2	2	2	2	2	2
18	4	1	1	3	2	3	3	3	4	4	3	4	2	3	3	3	3	3
19	3	0	1	5	3	5	2	3	5	5	3	3	3	3	2	4	2	2
20	6	1	1	4	2	4	3	4	4	4	3	4	4	5	4	5	5	5
21	6	1	1	4	2	4	3	3	5	6	4	4	3	4	3	5	4	4
22	5	1	1	5	5	5	5	5	5	5	4	4	4	3	3	3	3	3
23	5	1	1	4	3	4	3	3	3	2	2	2	2	4	3	3	3	3
24	4	1	1	5	4	4	4	4	5	5	4	4	4	3	3	3	2	2
25	2	1	0	2	3	3	3	2	5	5	4	4	5	4	4	4	4	4
26	6	1	1	5	3	4	3	4	3	3	3	4	2	5	2	5	5	5
27	6	1	1	6	3	5	5	4	6	5	4	5	3	6	6	6	6	6
28	5	1	0	1	5	5	3	5	1	5	3	5	5	5	5	3	5	5
29	6	1	1	4	3	3	2	4	4	4	4	3	3	4	3	3	3	3
30	4	1	1	3	2	4	2	3	4	4	3	4	2	5	2	5	5	4
31	4	1	1	3	1	2	2	3	3	2	2	3	2	3	2	1	4	2
32	6	1	1	5	5	4	4	4	5	4	3	4	5	3	3	4	3	3
33	5	1	1	3	2	4	3	3	4	4	4	4	3	5	4	5	3	4
34	4	1	1	6	5	5	5	5	6	6	6	6	5	5	4	6	5	5
35	5	0	1	5	4	5	3	4	5	4	3	3	2	5	4	3	3	3
36	6	1	1	4	3	4	4	4	5	5	5	5	4	6	5	6	6	6
38	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	6	1	1	2	1	2	2	2	3	3	3	3	1	4	2	3	6	5
41	6	1	1	4	2	4	4	5	6	5	5	6	6	6	4	6	6	6
42	6	1	1	3	2	4	3	4	5	5	4	5	4	4	4	4	4	4
43	3	1	0	5	1	5	4	5	4	5	4	5	4	5	5	6	6	5
45	5	1	1	6	4	6	4	5	5	5	4	5	4	5	5	5	4	5
46	6	1	1	5	4	5	5	4	5	5	5	4	4	6	5	6	5	4
47	5	1	1	2	2	5	5	5	2	3	3	4	4	3	4	4	4	5
48	6	1	1	4	2	5	4	4	5	5	4	4	3	6	3	6	6	5
49	6	1	1	4	3	3	3	4	4	3	3	4	4	4	4	4	4	4
50	6	1	1	5	4	4	4	4	5	4	4	4	4	6	6	5	5	5
58	5	1	1	5	3	3	3	5	0	0	0	0	0	0	0	0	0	0
60	5	1	1	6	6	5	5	5	5	5	4	4	4	5	6	4	4	4
62	6	1	1	6	5	6	6	5	6	6	6	5	6	6	6	6	6	6
63	6	1	1	6	3	6	6	6	6	6	6	6	5	4	4	4	5	4
64	4	1	1	6	6	6	4	6	5	5	4	5	5	5	4	4	3	4
67	5	1	1	3	3	4	2	4	4	5	4	5	5	5	4	5	4	5
68	2	1	1	4	3	3	2	2	0	0	0	0	0	0	0	0	0	0

Appendix H

Raw analyst survey data

The raw data from the analysts is displayed here. In the table to the right, the column headers are displayed, which correspond with the large table starting on the next page. So, in total there are 45 data columns (the three demographic customer questions are missing) and 22 full responses, ranging from index 2 to 24.

0	adware_label
1	adware_report
2	adware_phone
3	portscan_label
4	portscan_report
5	portscan_phone
6	trojan_label
7	trojan_report
8	trojan_phone
9	tor_label
10	tor_report
11	tor_phone
12	update_label
13	update_report
14	update_phone
15	open_label
16	open_report
17	open_phone
18	usb_label
19	usb_report
20	usb_phone
21	application_label
22	application_report
23	application_phone
24	creds_label
25	creds_report
26	creds_phone
27	dataleak_label
28	dataleak_report
29	dataleak_phone
30	trojan_var_timing
31	trojan_var_asset
32	trojan_var_similar
33	trojan_var_olddata
34	trojan_var_triageconf
35	application_var_timing
36	application_var_similar
37	application_var_olddata
38	application_var_triageconf
39	application_var_asset
40	creds_var_timing
41	creds_var_asset
42	creds_var_similar
43	creds_var_olddata
44	creds_var_triageconf

2	2	0	0	5	1	1	5	1	1	4	1	0	5	1	1	4	0	1	4	0	1	5	1	1	6
3	3	0	0	4	1	1	5	1	1	4	1	0	3	1	1	4	1	1	4	1	1	4	1	1	6
4	2	1	0	4	1	1	5	1	1	3	1	0	4	1	1	5	1	1	4	1	0	5	1	1	5
5	2	0	0	3	1	0	5	1	1	3	1	0	4	1	1	4	1	1	2	0	0	5	1	1	5
6	2	1	1	4	1	1	4	1	1	2	1	1	3	1	1	3	1	1	3	1	1	4	1	1	4
7	3	1	1	5	1	1	5	1	1	3	1	1	3	1	1	4	1	1	3	1	0	5	1	1	5
8	2	0	0	5	1	1	4	1	1	4	0	1	4	1	1	4	1	1	4	1	1	6	1	1	5
9	2	1	0	5	1	1	5	1	1	4	1	1	5	1	1	4	1	1	4	1	1	5	1	1	5
10	3	1	0	4	1	1	5	1	1	4	1	1	5	1	1	4	1	0	3	1	0	4	0	0	4
11	3	0	0	4	1	1	5	1	1	3	0	0	4	1	0	4	1	1	5	1	1	5	1	1	5
12	4	0	0	5	0	1	5	1	1	4	0	0	4	0	0	2	1	0	5	0	1	5	1	1	4
13	3	0	0	5	1	1	4	1	1	3	1	1	4	1	1	4	1	1	3	1	0	4	1	1	5
14	3	1	0	4	1	1	5	1	1	4	1	1	3	1	1	3	1	0	4	0	0	4	0	1	6
15	2	1	0	4	1	1	4	1	1	3	1	0	4	1	1	3	1	1	3	1	1	4	1	1	4
17	3	1	0	4	1	1	5	1	1	4	0	0	5	1	1	4	1	0	4	1	0	6	1	1	5
18	3	1	0	5	1	1	5	1	1	4	1	1	4	1	0	5	1	1	3	1	1	5	1	1	5
19	3	0	0	3	0	1	5	1	1	3	0	0	5	0	1	4	1	1	3	1	1	4	1	1	5
20	3	1	0	4	1	1	6	1	1	4	1	0	4	1	1	3	1	1	4	1	0	6	1	1	4
21	2	0	0	5	1	1	4	1	1	4	1	1	4	1	1	3	1	1	5	1	1	6	1	1	5
22	2	0	0	4	0	1	5	1	1	4	1	1	3	1	1	3	1	1	4	1	1	5	1	1	4
23	3	0	0	5	1	1	6	1	1	4	1	1	4	1	0	5	1	1	6	1	1	6	1	1	6
24	3	0	0	4	1	1	4	1	1	4	0	1	3	0	0	2	0	0	2	0	0	4	1	1	4

2	1	1	3	1	0	5	2	4	3	5	5	5	5	5	6	6	4	5	4	5
3	1	1	6	1	1	5	4	4	6	4	3	3	3	3	3	6	4	4	3	4
4	1	1	5	1	1	5	4	4	5	3	5	4	5	3	4	5	4	4	4	3
5	1	1	4	1	1	5	3	3	5	4	5	3	2	4	5	5	5	2	5	4
6	1	1	4	1	1	4	2	4	3	4	4	4	2	4	3	4	4	4	5	4
7	1	1	4	1	1	5	2	2	5	5	5	2	5	5	4	5	5	2	4	5
8	1	1	4	1	1	5	1	3	2	3	6	4	6	5	5	5	3	2	5	4
9	1	1	4	1	1	6	4	5	5	4	6	5	5	4	4	5	4	3	5	5
10	1	0	5	1	1	5	3	5	3	4	4	3	3	3	3	4	3	4	5	4
11	1	1	6	1	1	5	4	4	5	5	5	4	4	5	4	5	4	5	5	5
12	1	1	4	1	1	5	3	4	3	4	5	4	3	4	3	3	4	2	4	3
13	1	1	3	1	1	4	3	4	3	4	4	4	3	4	4	5	4	3	4	4
14	1	1	5	1	1	5	3	3	4	4	4	2	2	3	4	5	4	2	4	3
15	1	1	4	1	1	4	2	4	3	4	5	4	3	4	3	4	3	2	3	3
17	1	1	6	1	1	5	3	4	4	5	6	5	4	5	5	5	5	4	3	5
18	1	1	4	1	1	5	2	2	4	3	5	2	3	3	4	5	2	2	4	3
19	1	1	5	1	1	5	5	3	5	5	4	3	3	4	3	5	3	4	5	5
20	1	1	4	1	1	6	1	4	4	2	6	6	6	5	1	4	3	5	6	1
21	1	1	4	1	1	4	1	4	4	4	6	5	6	5	5	4	4	4	3	4
22	1	1	6	1	1	5	3	3	4	5	5	4	4	4	5	4	3	3	3	3
23	1	1	5	1	1	6	4	6	6	5	6	6	6	6	5	6	5	4	5	5
24	0	1	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0