Master thesis
Computer Science

Radboud University

Defining Who Is Attacking by How They Are Hacking

A Classification of Current Remote Desktop Modus Operandi

Author:
Roland Middelweerd

First supervisor/assessor:
dr. Veelasha Moonsamy
email@veelasha.org

Daily supervisor:
Martijn Hoogesteger,
Teamlead CERT at Northwave
Martijn.Hoogesteger@northwave.nl

Second assessor:
prof. Frederik Zuiderveen Borgesius
frederikzb@cs.ru.nl

October 16, 2019

Abstract

Attacks where the Remote Desktop Protocol (RDP) is used to infiltrate systems and networks are becoming more and more prevalent. Selling RDP credentials and the usage of weak usernames and passwords contribute to the increase of these attacks. Computer Emergency Response Teams (CERTs) have a hard time finding out how an attacker acquired RDP credentials to access a system. This research will focus on whether it can be traced back how RDP credentials have been acquired, based on the most common cyber attacks. This will aid investigators in finding out how a system was compromised. To collect information about cyber attacks, a honeypot architecture is created which keeps track of RDP sessions and collects images of honeypot hard drives. RDP credentials are spread on various platforms and weak credentials are used to lure attackers. For each different way of acquiring credentials, a honeypot is deployed. The honeypots were deployed for five and a half weeks, resulting in the collection of 351 images. A classification is proposed to classify the gathered data. The classification also contributes to the improvement of the efficiency of forensic analysis on the Windows operating system, by prioritizing the analysis of digital artifacts. The proposed classification is a decision tree where the leaf nodes represent cyber attacks and the edges contain conditions based on the presence of digital artifacts. A subset of the collected images has been used to evaluate the proposed classification. The results of the evaluation show that PowerShell and the Command Prompt are often used during an attack, it is regularly checked if the browser automatically logs in at payment platform websites and port scanning is performed often. Moreover, there is a considerable difference between the amount of attention each way of acquiring credentials receives.

Acknowledgements

I would like to thank all my colleagues at Northwave who have helped me, with their knowledge and support, during my research. In particular I would like to thank:

- Tycho van Marle and John Fokker for providing me with information about the current state of affairs of the RDP credential market and their thoughts on the creation of honeypots.
- Luc van den Ackerveken, Tijme Gommers and Alex Rommelse for using their experience as hackers to improve the created honeypot setup.
- Marinus Boekelo and Peter Wagenaar for their expertise in the field of digital forensics, contributing to the creation of the proposed cyber attack classification.

I would also like to thank my supervisors, Martijn Hoogesteger and Veelasha Moonsamy, for their time, knowledge and mentoring to support me in successfully completing my master thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Attacks via the Remote Desktop Protocol (RDP) are rising since 2016, according to a public service announcement of the FBI [1]. The rise is associated with the growth of RDP markets on the dark web selling credentials. This trend was observed in the SANS Cloud Security Survey of 2019 [2]. While *Account or credential hijacking* was on the third place of factors involved in successful cloud attacks in 2017, it took the number one spot in the survey of 2019. The emergence of attacks performed via RDP was also experienced by the Northwave Computer Emergency Response Team (CERT). The Northwave CERT handles cyber security incidents, where their goal is to get a customer back to business as soon as possible. To be able to do this, it must be known how an attacker gained access and what has been damaged by the attack. Especially the point of entry and how the attacker gained access, are important pieces of information for preventing repetition of the attack. The problem with attackers gaining entry via RDP, is that it is difficult for a CERT to track down how the attacker acquired the credentials. Credentials can be acquired in different ways. For example, by performing a brute force attack. They are also often spread on platforms such as hacking forums and websites where text can be shared. Not knowing how the credentials have been acquired is problematic for preventing repetition of an attack. Knowing which systems have been affected by an attack requires reconstruction of the behavior of an attacker, also known as the *modus operandi*. The reconstruction is done by performing forensic analysis on collected digital evidence. The evidence can include any digital source which may contain information about the attack, ranging from log files to complete hard drive images.

While the modus operandi cannot directly provide information about where an attacker acquired RDP credentials from, it might be able to provide information indirectly. If the same modus operandi is used by attackers which acquire RDP credentials in the same way, the modus operandi could indirectly provide information about how the credentials have been acquired. This helps the investigation of the CERT, in preventing repetition of an attack. The following research question will be examined in this research:

Can a distinction be made between different ways through which attackers acquire RDP credentials, based on most common cyber attacks?

The following sub-questions have been formulated to guide answering the research questions:

1. What kind of data needs to be collected, to be able to recover the modus operandi of an attacker, and how can this data be gathered?
2. Where do attackers acquire RDP credentials from?
3. How can cyber attacks be classified?

Honeypots have been used to gather information about common cyber attacks, originating from attackers which acquired credentials. A custom honeypot has been created to gather information about the modus operandi of attackers. Multiple honeypots have been deployed where its RDP credentials were spread on a website, or a weak username and password combination was used. A separate honeypot was deployed for each credential acquisition scenario, to make a distinction between where the attackers acquired their credentials from. The created honeypot architecture collects images of the honeypot hard drives. Forensic analysis is performed on these hard drives to discover which attacks have been performed. A new classification is proposed which classifies cyber attacks based on the present digital artifacts. This is a novel approach for classifying cyber attacks. The classification is not only created to structure the forensic analysis, it is also created to improve the efficiency of forensic analysis by prioritizing the search for relevant digital artifacts. The method used for the creation of the classification can be used to create classifications for other domains and to expand the proposed one. The evaluation of the gathered data will test the performance of the proposed classification and will provide information about the performed attacks on the honeypots. The latter will be used to answer the research question.

Chapter 2 introduces background information on the different aspects of this research. Then, Chapter 3 will discuss related research. The research setup, including the creation of the honeypot and the creation of the classification, will be elaborated on in Section 4. Next, the evaluation and its results will be discussed in Chapter 5. Then, Chapter 6 will discuss the results of the research and Chapter 7 will outline possible future work. Lastly, Chapter 8 will conclude the research by answering the research question.

Chapter 2

Preliminaries

In the coming sections information will be provided on the different aspects of the research setup. Section 2.1 will discuss honeypots in general and the different types of honeypots are presented in Section 2.2. Then, in Section 2.3 the communication protocol used for the honeypot will be explained. Next, the method for attracting attackers is described in Section 2.4. After attackers logged into the system, their behavior or *modus operandi* is important to reconstruct. What modus operandi is will be elaborated on in Section 2.5 and how it can be reconstructed will be explained in 2.6. In Section 2.7 the process of performing a forensic investigation will discussed, including the challenges which are currently faced. Lastly, cyber attack classifications will be elaborated on in Section 2.8.

2.1 Honeypot

The first time the concept of a honeypot was described, was in a book titled *The Cuckoo's Egg* [3], published in 1989. The book describes the tracking of a hacker. At the time, the term **honeypot** did not exist yet. One of the first formal definitions of a honeypot was given by Lance Spitzner [4] in 2002:

> *"A honeypot is very different from most traditional security mechanisms. It's a security resource whose value lies in being probed, attacked, or compromised."*

Even though the definition is quite old, it is still applicable to modern-day honeypots. A honeypot is a system which can represent anything ranging from a Desktop PC to an Internet of Things (IoT) device. Honeypots can be used to gain information about existing attacks. One of the first honeypots was created by the SANS Institute to capture a malware sample [5]. Worms infected Windows and Unix systems in a large quantity at the time and they were difficult to analyze because they only existed in memory or they were hiding themselves. The SANS institute captured a sample of the sub7 malware by emulating an infected Windows system. The sample was captured within minutes.

Besides gaining knowledge about existing attacks, honeypots can also be used to gain knowledge about unknown attacks. In 2002, the Common Desktop Environment (CDE) Subprocess Control Service buffer overflow [6] was discovered, because of network traces provided by the Honeynet Project [7]. The vulnerability allows an attacker to gain access to systems running the UNIX or Linux operating system. This was the first time in history in which a honeypot was used to detect an unknown threat [5]. Learning about unknown threats allows to prepare for the future by taking measures if necessary.

Different kinds of information can be collected by honeypots. It depends on the domain and also on the objective of the honeypot [8]. Generally, information is collected to be able to reconstruct the activities of an attacker. Organizations can use the collected information for attack vector customization. Pay extra attention to company assets which are more likely to be targeted by attackers.

To gather useful information, the honeypot needs to be attractive and realistic [9]. When the honeypot is not attractive enough, it will not lure attackers, thereby not gathering information. It also needs to be realistic to prevent discovery of being a fake system. Finding the right balance between these two traits can be difficult, this will be discussed in greater detail in Section 2.2.

All traffic directed to a honeypot can be regarded as anomalous [5]. A honeypot is not a real system so nobody should interact with it for legitimate use. This property of honeypots reduces the number of false positives and false negatives. Creating a cost-effective honeypot can be difficult. A honeypot with extensive features collects more relevant information than a honeypot with limited features. However, there is always a risk of attackers discovering the true nature of the honeypot and spreading that information [9]. When the honeypot is discovered, it is rendered useless. The honeypot needs to be changed before it can be used again. When the implemented features deviate too much from the standard implementation, they can be used for fingerprinting. Fingerprints of known honeypots can be used to check the legitimacy of a system [5].

A honeypot consisting of multiple systems, applications and services is called a **honeynet** [10]. By providing a high level of interaction, extensive information about threats can be captured. The disadvantage of honeynets is that they are complex, can be time consuming to maintain and require advanced configuration. To simplify the deployment of honeynets, Spitzner [11] proposed the concept of a **honeypot farm**. Instead of deploying honeypots in multiple networks, install multiple honeypots in a centralized location. Redirectors are used in the different networks to redirect the traffic of the attacker to a honeypot deployed in the central honeypot farm. By centralizing the honeypots, the maintenance and deployment become easier. Moreover, adding a honeypot to a network becomes easier because you only have to add a redirector to the network. Besides simplifying the deployment, it also mitigates the risk of a honeypot with a high interaction level. This kind of honeypot inherently has a risk of when the attacker takes over the honeypot, other real systems in the network can be attacked. In a honeypot farm, there are no real systems to attack.

2.2 Types of Honeypots

Honeypots can be classified based on different characteristics. For example, based on the objective of the honeypot, they can be divided into research and production honeypots [5]. Research honeypots are used to gather information, while a production honeypot is used to divert the attention of an attacker. These two objectives can also be combined into a single honeypot.

Honeypots can also be classified based on their adaptability [12]. Static honeypots have a static configuration, their appearance and responses are always the same. Dynamic honeypots have a dynamic configuration, that means that they can have a different appearance with different responses every time they are deployed.

Another characteristic of a honeypot is how the honeypot is implemented [8]. A dedicated system can be used as a honeypot. The maintenance and development of such honeypots can be expensive. To solve this problem, virtualization can be used. This decreases the maintenance and development costs. A hybrid of both software and hardware is also a possibility.

Based on the activity of a honeypot, they can be divided into a client or a server honeypot [13]. The difference between the two is whether the honeypot passively waits to be attacked or actively engages interaction to find client-side intrusions.

Lastly, the characteristic which is used most often to classify honeypots is their level of interaction. In the coming subsections the different levels of interaction will be explained.

2.2.1 Low-Interaction

A **low-interaction honeypot** simulates a system with one or more services. The services are not genuine, they only have predefined answers. Information can be gained about how attackers interact with the system, but the information is limited because the services are limited. Using this kind of honeypot decreases the risk of an attacker misusing the system when it is compromised [5]. There is a lot more control on the capabilities of the system compared to other honeypots. Even though the information gained about the behavior of the attacker is limited, other information about, for example demographics and the use of known vulnerabilities, can be gained through these systems.

Deploying and maintaining a low-interaction honeypot is easy because of their basic functionality [4]. They require a low amount of resources and their deployment is relatively easy compared to honeypots with a higher level of interaction. It is possible to deploy multiple low-interaction honeypots on a single host by using virtualization. Honeyd [14] uses virtualization to deploy multiple low-interaction honeypots where each honeypot has its own IP address. Popular services used for low-interaction honeypots are SSH, FTP, MySQL and Telnet. Examples of low-interaction honeypots are Honeyd [14] and Glastopf [15].

2.2.2 Medium-Interaction

A **medium-interaction honeypot** offers a higher level of interaction, compared to a low-interaction honeypot, by increasing the level of interaction of the services [5]. Because of this, more in depth information can be gathered. Information about the interaction between the attacker and the services can be gathered, but it also increases the risk of an attacker misusing the system [4]. The simulated services have more depth, thereby giving an attacker more possibilities to break the system. Medium-interaction honeypots are less likely to be discovered compared to low-interaction honeypots. Examples of medium-interaction honeypots are Dionaea [16], Kippo [17] and Cowrie [18].

2.2.3 High-Interaction

A **high-interaction honeypot** is a genuine system with legitimate services. Using a real system provides a high level of interaction. This allows to gather information about more sophisticated attacks, but this comes at a cost. The deployment and maintenance of high-interaction honeypots is more time consuming and thereby more expensive. Moreover, if the attacker breaks out of the honeypot, it can be used as a bridge to other systems on the local network [19]. This is an important risk to consider when using a high-interaction honeypot.

When the honeypot is used in a production environment, it has to be properly secluded from the rest of network to prevent lateral movement. Preventative measures should be taken to prevent the honeypot being used as a stepping stone to harm other systems. Not only inside the local network but also systems outside the local network. Ethical considerations like this should be taken into account when deploying a high-interaction honeypot. An example of a high-interaction honeypot is Argos [20].

2.2.4 Hybrid

A **hybrid honeypot** system consists of multiple honeypots with different interaction levels. Generally, a low- or medium-interaction honeypot is used in combination with a high-interaction honeypot [21]. By combining honeypots with different interaction levels, the advantages of the different honeypots can be combined. The low- or medium-interaction honeypot can be used as a front-end, they have the advantage that they can be deployed in large numbers [13]. The high-interaction honeypot can be used as a back-end, which is more time consuming to deploy, in small numbers. Multiple front-end honeypots are connected to the same back-end honeypot. Guarnizo et al. [22] used such a hybrid system on a larger scale. A large number of low-interaction honeypots were used to expose only a few IoT devices to the internet. Multiple low-interaction honeypots forward data to the same IoT device. This is an example of how a hybrid honeypot system can be used to increase coverage and how to improve information gathering with a limited number of physical devices.

2.3 Remote Desktop Protocol

The **Remote Desktop Protocol** is a proprietary protocol created by Microsoft. It allows remote control over a network connection [23]. An RDP client can connect to an RDP server, transferring the graphical user interface (GUI) from the server to the client and transferring mouse and keyboard input from client to server. Special drivers are used at the client and server to exchange the previously mentioned data. The RDP protocol uses the RC4 stream cipher to encrypt the network packets. It is a multiple-channel capable protocol which allows separate virtual channels to be used for different data sources. The default port used by the server in the RDP protocol is 3389. The protocol has support for features such as print redirection, clipboard mapping and bandwidth reduction. There are other protocols available which allow remote control over a network connection. The Virtual Network Computing (VNC) protocol, for example, also allows remote control with a GUI and keyboard and mouse input. There are also protocols which provides access via a command line interface (CLI) instead of access via a GUI. Popular protocols are Secure Shell (SSH) and Telnet.

2.4 Credentials

Credentials are used for authentication. Generally, you receive some form of documentation or secret knowledge to confirm your identity. From a digital point of view, credentials can be a certificate, a public/private key pair, a username and password combination, a fingerprint, a face, an iris, a smart card, a key fob or a voice. Location and time can be used in combination with the previously mentioned credentials to improve the authentication process.

The username and password combination is used often for authentication. According to a report from LastPass [24], a company which creates password management software, the average employee has to keep track of 191 passwords. The report is based on users of the LastPass password manager. It is difficult to remember such a great number of usernames and passwords. Stobert and Biddle [25] showed in a small experiment that people are prone to reuse passwords, 26 out of the 27 participants admitted that they reused passwords between different accounts. Moreover, as NIST describes in Special Publication 800-63B [26], *"Humans, however, have only a limited ability to memorize complex, arbitrary secrets, so they often choose passwords that can be easily guessed."*. Often reusing passwords or using password which are easily guessable, are dangerous for the level of security of passwords.

To improve the level of security of a username and password, two-factor authentication can be used. Two-factor authentication adds a second step to the authentication process. The user retrieves a second piece of secret knowledge from a device which needs to be in possession of the user. The secret knowledge can for example be a text which is sent to a phone, or a code which is displayed on a dedicated hardware device. By using two-factor authentication, not only secret knowledge is required but also a device needs to be in possession of the user. This prevents direct usage of stolen or leaked credentials.

Credentials can be stolen or leaked in numerous of ways. The Emergency Agency of Hawaii had an incident where a password leaked because in the background of a public photo, a post-it note containing password was published on the internet [27]. Data breaches are unfortunately common, even for major companies such as Facebook, Yahoo and LinkedIn [28]. When a data breach happens and plaintext credentials are stolen, not only the account at the concerning company is in danger, it might also endanger other accounts of users which reused their username and password.

Malicious tools such as keyloggers and credential stealing malware, are used to steal credentials. They record passwords which are filled in by a user and send them to the attacker. Another way of stealing credentials is by using Phishing. A user is led to a fake website and is encouraged to log in. The stolen credentials are used by the attacker to log in at the real website. Lastly, using a weak password makes it easy for an attacker to perform brute force attacks. There are word lists available containing the most used passwords.

2.5 Modus Operandi

According to the Encyclopedia Britannica the definition of **modus operandi** is [29]:

> *"Modus operandi, (Latin: "operating method",)abbreviation Mo, in criminology, distinct pattern or manner of working that comes to be associated with a particular criminal. Criminologists have observed that, whatever his specialty—burglary, auto theft, or embezzling—the professional criminal is very likely to adhere to his particular way of operating."*

The definition contains examples of offline criminal activities, but it is also applicable to online criminal activities. A criminal can have a particular way of operating for stealing a car, but the same is possible for a criminal breaking into a digital system. Learning about the behavior of criminals can be beneficial to people who keep the car or digital system safe. By knowing how criminals break in, you can take measures to keep them out.

In the context of digital forensics incident response, the goal of learning about modus operandi is exactly the same. Learning about the modus operandi, allows to estimate the scale of the attack and how the damage done can be mitigated. In NIST SP 800-61 [30] this is described as the *Containment* and *Eradication* phases in the incident response life-cycle. Without knowledge of the modus operandi it is difficult to complete these phases and without this, the life-cycle cannot proceed to the next phase which is the *Recovery* phase. All these phases need to be completed before everything can go back to business. How the modus operandi of an attacker can be reconstructed will be explained in the next section about digital artifacts.

2.6 Digital Artifacts

There is no official definition for a **digital artifact**, but Oliver [31] created the following definition:

> *"A digital artifact is a sequence of bits that has (or represents) meaning. The meaning is often (but not always) determined by context."*

The definition shows the broad spectrum of what can be seen as a digital artifact. It can be as small as a single integer to as large as an executable. A digital artifact is for example the creation date of a file. As Oliver [31] emphasizes, the meaning of a digital artifact often depends on the context. The change of a security sensitive system setting is a digital artifact depending on who changed the setting. By collecting digital artifacts and connecting them in the context of where and when they are found, can provide information about the modus operandi of an attacker.

2.7 Forensic Investigation

Performing a forensic investigation is part of the **Digital Forensics Incident Response** (DFIR) process. There are multiple forensic investigation models which consist of different phases. Yusoff et al. [32] have investigated common phases of these forensic investigation models, they concluded that there are five generic phases:

1. Pre-Process
2. Acquisition & Preservation
3. Analysis
4. Presentation
5. Post-Process

The *Pre-Process* phase includes everything that needs to be done before the acquisition of evidence begins, for example, requesting necessary approval or setting up required infrastructure. The *Acquisition & Preservation* phase collects and preserves all relevant data. In the *Analysis* phase, the acquired evidence is analyzed and the source of the crime is investigated. The results of the *Analysis* phase are documented and presented in the *Presentation* phase. The final phase, *Post-Process*, consists of closing the investigation. Evidence is properly archived or is returned to the rightful owner. The focus of this research will be on the *Analysis* phase.

Digital forensics is facing difficult challenges due to technological advancement [33]. The number of devices and their diversity has increased [33]. This has made the acquisition of evidence more difficult and the required level of specialized knowledge has increased. The shift to cloud computing and storage becoming cheaper, has increased the volume of data which has to be analyzed during a forensic investigation [33]. Forensic tools are not always built to efficiently handle a large volume of data, resulting in a long processing time. Privacy has also introduced new challenges [33]. There has to be a balance between collecting evidence and invading someone's privacy. This balance is different for each investigation.

2.8 Cyber Attack Classification

A wide variety of cyber attack classifications and taxonomies have been proposed over the past years. The difference between a classification and a taxonomy is that a classification groups things together while a taxonomy only gives a name to things. A taxonomy can name individual entities but also groups, which is why it goes hand in hand with classifications. Taxonomies also try to include all possible variations. A classification or taxonomy can be represented in different forms, for example a list of definitions, a hierarchical tree structure or a decision tree.

Classifications and taxonomies have been proposed to classify cyber attacks in domains such as smart grids [34], embedded systems [35], cyber-physical systems [36], SCADA [37], Advanced Persistent Threats (APTs) [38] and the internet [39]. These classifications and taxonomies differ in the number of dimensions used. While Ahmad et al. [39] used a single dimension, in other words, looked at the attacks from a single point of view, Yampolskiy et al. [36] used three dimensions. When multiple dimensions are included, different perspectives are used to approach the cyber attacks.

Different levels of abstraction can be used for a classification or taxonomy, it depends on its purpose. The previously mentioned research all proposed classifications or taxonomies for a specific domain, trying to improve the understanding of that specific domain. Harry and Gallagher [40] created a taxonomy which classifies cyber events based on the primary effect on the target of the cyber attack. This taxonomy is created for a higher level of abstraction, it does not focus on a specific domain.

Chapter 3

Related Work

In the coming sections related research will be elaborated on. First, proposed honeypot architectures will be discussed in Section 3.1. Then, in Section 3.2 research using credential leaks is examined. Next, previous attempts to improve the efficiency of forensic analysis are explained in Section 3.3. Lastly, cyber attack classifications are discussed in Section 3.4.

3.1 Honeypots

Honeypots have been used frequently to gather information about cyber attacks. Architectures have been proposed for a broad range of domains including the Internet of Things [22] [41] [42] [43], Industrial Control Systems [44], Supervisory Control and Data Acquisition (SCADA) [45], the World Wide Web [46] [47], mobile devices [48] and even robot systems [49]. To gather as much information as possible, it is important to keep the attacker engaged. Barron and Nikiforakis [50] concluded that by making the honeypot as real as possible, the amount of interaction between the attacker and the honeypot increases. Pohl et al. [51] proposed a honeypot architecture where extra honeypot fields are added to a web form to detect web related attacks. They prevented detection of the extra fields by taking the content of the web form into account. Their research emphasizes the importance of using relevant content, to prevent detection of the honeypot and to keep an attacker engaged.

Preventing honeypot detection can be difficult because deploying a honeypot has ethical considerations. Moreover, the people deploying the honeypot have to abide by the law as opposed to the attacker [8]. Tsikerdekis et al. [8] give an example where a honeypot should not be allowed to participate in a denial of service attack. The prevention of misuse can be used by attackers to identify honeypots. Tsikerdekis2019 et al. [8] also propose approaches to prevent honeypot detection, such as automatic redeployment, using dedicated hardware and using honeypots with dynamic behavior. Multiple dynamic honeypot architectures have been proposed [52] [53] [54].

The geographical location of where a honeypot is deployed, can influence the amount of attention it receives, according to the research of Guarnizo et al. [22]. Barron and Nikiforakis [50] also deployed honeypots in different locations. Their results showed that the company where the honeypot was hosted had a much larger impact on the amount of attention compared to the geographic location. They deployed their honeypots on Linode and Amazon Web Services on the same locations with the same number of honeypots. The honeypots hosted on Amazon Web Services received more attention than the honeypots hosted on Linode.

3.2 Leaking Credentials

To collect information about the behavior of attackers, they need to be lured to the honeypot. A method for luring attackers is by spreading RDP credentials. Different kinds of credentials have been spread by previous research, for example email credentials [55] [56], cloud environment credentials [57], bank details [58], SSH credentials [50] and FTP credentials [59].

Different strategies can be used to generate fake credentials. Barron and Nikiforakis [50] performed an experiment which consisted of three rounds, where each round used a different generation method for generating fake credentials. The different generation methods used, provided a different level of difficulty for breaking into the system. The first round accepted any username and password combination after trying between one and three combinations. This allowed any attacker to log in easily. For the second round, the username *root* and popular passwords from a data breach were used, which increased the difficulty of breaking in. By using popular passwords from the data breach, a brute-force attack was still possible. Lastly, in the third round random usernames from a list of celebrities were used, in combination with randomly selected passwords from the same data breach as in the previous round. The usernames and passwords were spread on various platforms. Onaolapo et al. [55] used a different strategy, random combinations of popular first and last names were used for their email addresses. Bermudez Villalva et al. [56] used a database of random names for their email addresses.

Credentials can be spread in many different ways. By far the most popular platform for spreading credentials, are paste websites such as `pastebin.com` [57] [50] [55] [56] [58]. These kinds of websites can be used to share text with other people. Barron and Nikiforakis [50] posted credentials two times per day on Pastebin to increase the chance of crawlers accessing the leaked credentials. Besides posting on normal paste websites, Bermudez Villalva et al. [56] also leaked credentials on dark web paste websites.

Besides paste websites, hacking forums are frequently used to spread credentials [57] [50] [55] [56]. To increase the credibility of the messages posted on the hacking forums, Onaolapo et al. [55] and Barron and Nikiforakis [50] claimed that they had thousands of credentials for sale and provided a small sample of credentials as proof. The sample contained the honeypot credentials. Just as with the paste websites, there are also hacking forums on the dark web. Bermudez Villalva et al. [56] leaked credentials on dark web hacking forums. Along with the previously mentioned platforms, black markets [56], social media [57], dark web image boards [57] and credential stealing malware [55] [59] are also used as platforms to leak credentials.

To be able to distinctly identify where attackers acquired their credentials from, Fraunholz et al. [57] used a different URL for each platform where credentials were leaked. This solution obviously depends on the context of the honeypot. Akiyama et al. [59] used a unique combination of account name, password, IP address and the fully qualified domain name to be able to make the distinction where the attackers acquired credentials from. Lastly, Barron and Nikiforakis [50] and Onaolapo et al. [55] used different credentials for each platform where credentials were leaked.

3.3 Improvement of Forensic Analysis

Multiple frameworks have been proposed to improve performing a digital forensic investigation [60] [61] [62] [63] [64]. Du et al. [65] recently evaluated existing digital forensic investigation frameworks, showing the evolution and specific characteristics of the frameworks. A strategy which is applied often for the creation of a framework, is to focus on a specific domain, for example, Industrial Control Systems [66]. The previously mentioned frameworks all use a high level of abstraction, focusing on the complete process of performing a digital forensic investigation from the initial triage to the visualization of results.

As mentioned in Section 2.7, forensic analysis is challenged by the increasing volume of data which has to be analyzed during an investigation. To tackle this problem of having too much digital evidence, Brady et al. [67] proposed to use an ontology. The ontology allows an examiner to discover what kind of artifacts may be available on a device and to classify extracted data. The classification of the data allows equivalent artifacts across devices to be compared and thereby create connections. Lim et al. [68] use a strategy where only data is collected which enables the examiner to gain information about the usage history of the system. This reduces the amount of data which has to be examined. String searches can be used in a forensic investigation to help answer investigative questions. Beebe et al. [69] propose a method to improve information retrieval effectiveness of string searches, thereby decreasing the time spent on analyzing string search results.

Creating a timeline of gathered evidence is often used in a forensic analysis. The problem with using this method, is that it becomes more difficult to analyze the extracted events when the amount of gathered evidence increases. An approach is proposed by Hargreaves and Patterson [70] to combine multiple low-level events into a single high-level event. Using this approach produces a summary of the gathered data, allowing to prioritize the investigation by focusing on areas of interest. This reduces the time spent on low-level events which are not relevant for the forensic analysis.

3.4 Cyber Attack Classification

Cyber attack classifications have been used before to classify data gathered with a honeypot. Onaolapo et al. [55] created a taxonomy based on the behavior of the attacker. A distinction was made between *Curious Attackers*, *Gold Diggers*, *Spammers* and *Hijackers*. Bermudez Villalva et al. [56] extended the work of Onaolapo et al. [55] and used the same taxonomy.

A multi-dimensional taxonomy was proposed by Kjaerland [71] to classify cyber incidents. The dimensions of the taxonomy are based on cyber incident aspects categorized by CERT Coordination Center employees. The aspects used are *Source Sectors*, *Method of Operation*, *Impact* and *Target Sectors*. This research shows a different approach for classifying cyber incidents.

Gadelrab et al. [72], use a method to extract a sequence of actions that represent the execution of a malware sample. The actions used for the sequence are general attack steps, for example: *Reconnaissance*, *Execute Program* and *Gain Access*. Even though the method which is used to derive the unique patterns has been created to analyze malware samples, it can also be used in other domains.

The taxonomy proposed by Simmons et al. [73] uses five dimensions to classify an attack. The five dimensions used are *Attack Vector*, *Operational Impact*, *Defense*, *Informational Impact* and *Attack Target*. The *Operational Impact* and *Informational Impact* dimensions provide an overview of existing cyber attacks. That is also the case for the *Method* dimension of the taxonomy proposed by Joshi and Singh [74]. The focus of their proposed taxonomy is on network and computer attacks.

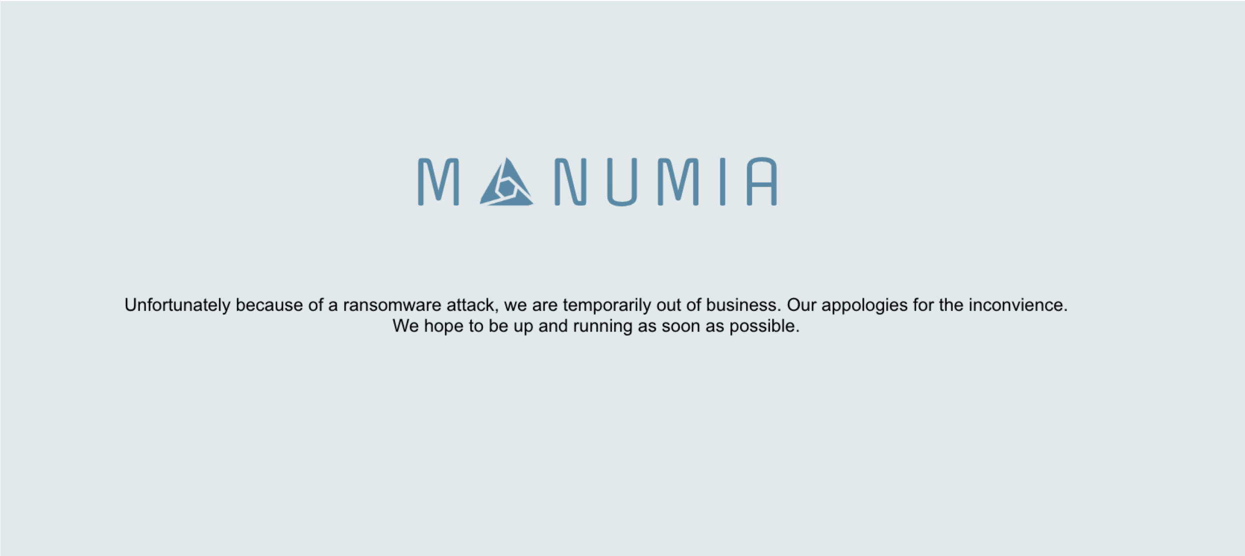
Chapter 4

Research

To gather information about cyber attacks, a honeypot architecture has been created. This architecture, including all the different design choices, will be explained in the coming sections. Moreover, how attackers are lured to the honeypots will be elaborated on. During the creation of a honeypot, the law and ethical considerations have to be taken into account. These two aspects will be discussed. While the honeypots were deployed, problems and set up flaws were encountered, these will be elaborated on. The creation of the classification and how the evaluation will be performed will also be discussed.

4.1 Honeypot Architecture

The goal of the honeypot is to collect information about attacks which are performed via RDP. To be able to do this, it needs to be attractive enough to attack and realistic enough to not raise any suspicion of being a honeypot [9]. A high-interaction honeypot was created to collect information about cyber attackers. The high level of interaction was chosen because detailed information needed to be collected to be able to reconstruct the modus operandi. A fake company called **Manumia** was created which supposedly was located in Belgium. The fake company was created to increase the credibility of the honeypot. A logo was created, the domain `www.manumia.com` was registered, prepaid sim cards were bought for the imaginary CTO of Manumia and the honeypots were hosted using the fake information of the imaginary CTO. A static HTML page was shown on the registered domain, stating that the company was temporarily out of business because of a ransomware attack, see Figure 4.1. This story was also included in the leaked credential messages. The story increased the credibility of spreading the temporary credentials. Furthermore, it explained why the data on the honeypot was not very old and why the hardware of the server was limited: the server was only temporary deployed because of the ransomware attack. The created honeypot architecture will be discussed in the coming subsections. Lessons learned from the development of the honeypot architecture can be found in Appendix A.



Unfortunately because of a ransomware attack, we are temporarily out of business. Our appologies for the inconvience.
We hope to be up and running as soon as possible.

Figure 4.1: Static HTML page of the Manumia website.

4.1.1 Point of Entry

Because of the emergence of attacks performed via RDP [1], the Remote Desktop Protocol has been chosen as point of entry. The Northwave CERT have also experienced the increase of incidents where RDP was used to gain access to a system. Using RDP requires a low level of technical skills, compared to other remote control protocols such as SSH. The danger of the low level of technical skills required, is that RDP can be insecurely configured unintentionally. For example, allowing access to RDP directly from the internet. The level of security is in this case dependent on the complexity of the username and password. For the proposed honeypot architecture, RDP accesses is allowed directly from the internet. This design choice has been made to also be able to gather information about attacks where credentials are acquired via a brute force attack. This insecure configuration is seen frequently in incidents handled by the Northwave CERT.

4.1.2 Content

The content of the honeypot is an important factor for the credibility of the system. As already mentioned in Section 3.1, the content of the honeypot can influence the amount of interaction between a system and an attacker. The content should fit to the created context. For a honeypot to be interesting for an attacker, it needs to have assets which are valuable to the attacker. The content of a honeypot can influence the kind of attacks which are performed on the system. Different kinds of assets can be used, depending on what kind of information should be collected with the honeypot. If information needs to be collected about lateral movement, the honeypot should be connected to multiple other systems. If information needs to be collected about data breaches, valuable information needs to be present on the honeypot. Which kind of assets are used and how the rest of the content of the honeypot has been chosen, will be explained in the coming paragraphs. Feedback on the appearance of the honeypot has been given by employees of the Northwave Red Team. The feedback can be found in Appendix B.

The RDP protocol is a proprietary protocol created by Microsoft. So, using a Microsoft operating system is a logical choice. RDP clients and servers are available for other operating systems, but they are not as widely used as on the Windows operating system. No recent statistics about the distribution of Windows operating system versions have been found, which were not older than a year. Because of this, the choice of operating system was based on which was seen most often in incidents handled by the Northwave CERT. The operating system which was seen most often is Windows Server 2012.

As previously mentioned, a honeypot needs interesting assets. The asset used for this honeypot is the installation of knowledge base software called *Confluence* [75]. Confluence allows to share knowledge with people inside and outside a company. The Confluence installation contains information about the internal infrastructure of the company, including fake regulations for employees, a fake organizational chart of the company and the internal network infrastructure. This is security sensitive information for the fake company. Moreover, all the names and email addresses of the users of Confluence can be accessed. A total of 131 randomly generated names and email addresses have been included. All the users are employees of Manumia. The randomly generated names come from French and Flemish name databases, to support that the company is located in Belgium.

The Confluence installation is customized with the colors of the Manumia logo. Moreover, the logo itself is used in multiple places in the layout of Confluence. The default page of the Confluence installation contains a message stating that the server is a temporary solution for the ransomware attack on the actual server containing Confluence. This is to increase the credibility of the ransomware attack story. It is also mentioned that not all Confluence spaces have been recovered from the backups yet, so not all pages are online. The information stored in Confluence and the names and email addresses of employees are the valuable assets of the honeypot.

Besides Confluence, the following software was installed: 7-Zip, Foxit Reader, Microsoft Silverlight, Mozilla Firefox, Notepad++, PDF Creator, Putty, VLC and WinDirStat. The software installed are all basic tools used on a Windows Server. The installed software allows to open PDFs, browse the internet, open video files and gain information about the storage of the server. Which software to install and what kind of files to store, was derived from previous incidents handled by the Northwave CERT. Images of previous incidents have been inspected to see what kind of software was installed and what kind of files were stored in the user folders. The inspection of images showed that servers generally have a dedicated purpose and only contain software and files to serve that purpose. The number of personal files of users on the servers were limited. This information has been used for the creation of the honeypot.

A variety of files have been stored on the Desktop and in the user folders. Not a lot of files were stored because according to the story of the server, it has been recently deployed. What is stored on the Desktop is of great importance because it is the first thing an attacker will see when logging into the honeypot. The attacker should have the feeling that the system is genuine. Moreover, the Desktop should show what the server contains in terms of interesting assets.

The files which are stored should portray a server which recently has been deployed. That is why a guide for Confluence is stored on the Desktop, just as some configuration files belonging to the Confluence installation. Moreover, a guide for improving the performance of Windows Server 2012 was stored on the Desktop, because the temporary server has limited resources. Furthermore, a PDF file containing descriptions of critical alerts for Windows Server monitoring was stored on the desktop, to give the impression that the server might not be secure yet. Besides the previously mentioned files, a folder containing a few maintenance scripts, shortcuts to installed software and shortcuts to Windows components are placed on the Desktop. The Documents folder contains a folder with Confluence Guides, again providing hints on that knowledge base software is installed on the system. The same performance guide as on the Desktop is also placed in the Documents folder. Moreover, a file called *TODO.txt* is stored in the Documents folder with the following content:

- *Add new user accounts*
- *Install virus scanner*
- *Retrieve backups of confluence*

The file contains hints that the system is not properly secured yet and that knowledge base software is installed. The Downloads folder contains setup files for 7-Zip, PDF Creator and VLC. A PDF file containing information about the basics of Office 365 is stored in the Downloads folder. The Music and Videos folders are empty. The Pictures folder contains the Manumia logo in an icon format. As already mentioned, servers generally do not contain lots of personal files. Besides that, the server is not very old. The content stored on the Desktop and in the Documents, Downloads and Pictures folders can be found in Figures 4.2 till 4.5.

Region settings are important to change because they should correspond with the settings of the country where Manumia is located. The locale settings of Windows and Firefox were changed to *Dutch - Belgium* to represent the right country.

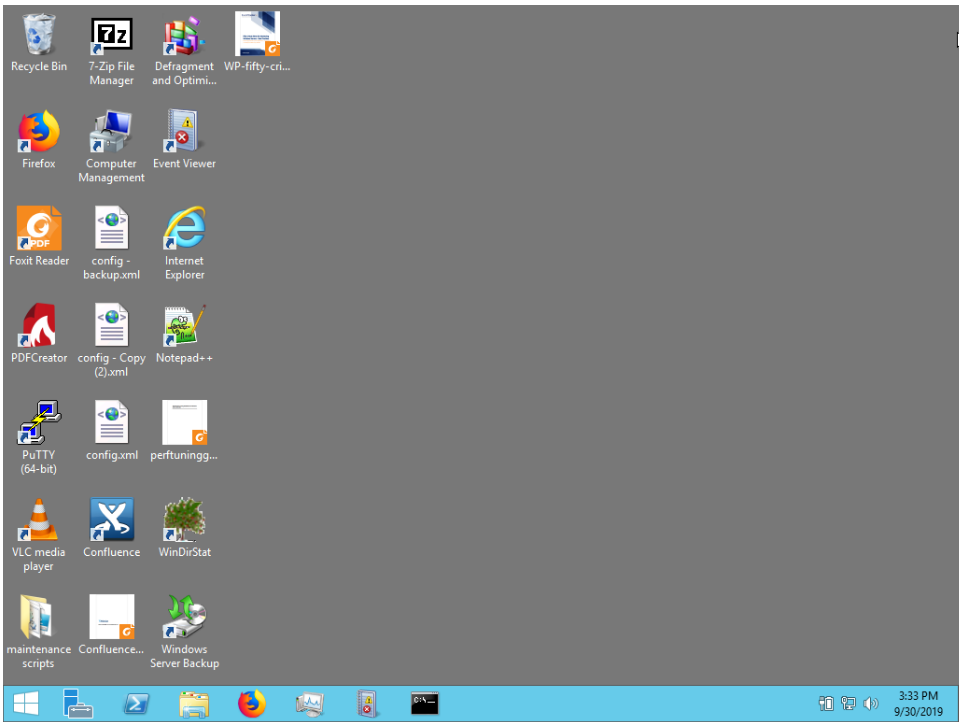
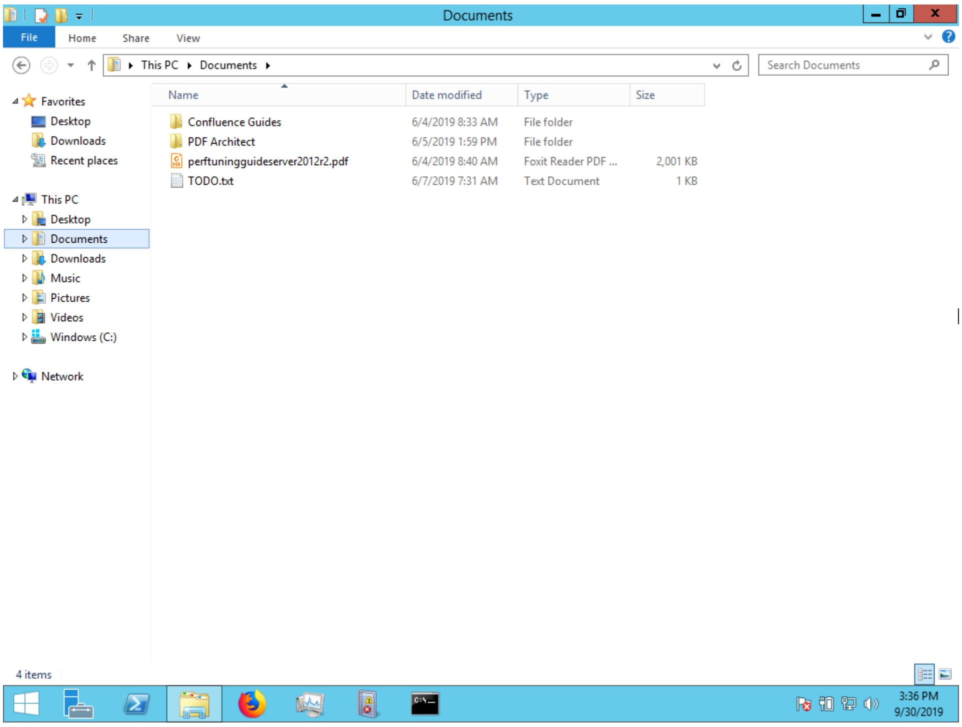
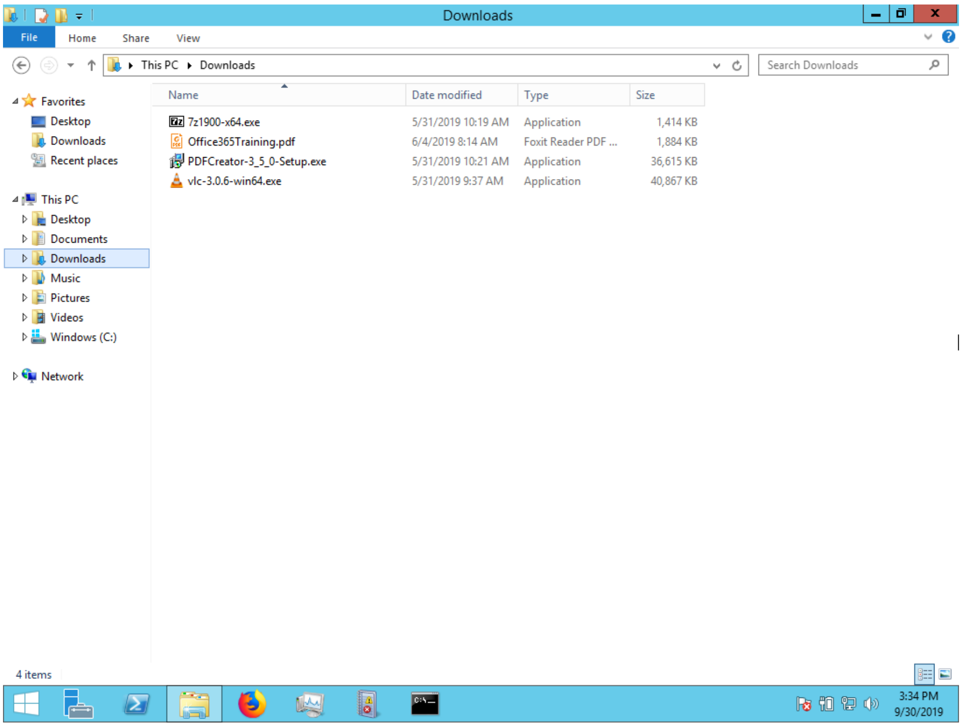


Figure 4.2: The Desktop of the honeypot.



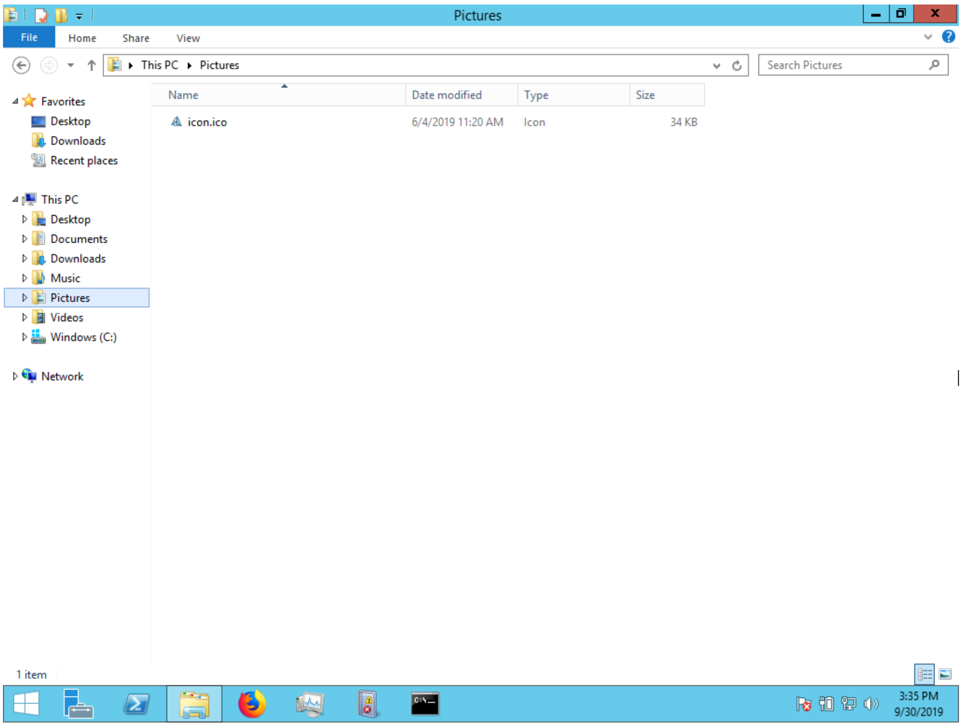
Name	Date modified	Type	Size
Confluence Guides	6/4/2019 8:33 AM	File folder	
PDF Architect	6/5/2019 1:59 PM	File folder	
perftuningguideserver2012r2.pdf	6/4/2019 8:40 AM	Foxit Reader PDF ...	2,001 KB
TODO.txt	6/7/2019 7:31 AM	Text Document	1 KB

Figure 4.3: The Documents folder of the honeypot.



Name	Date modified	Type	Size
7z1900-x64.exe	5/31/2019 10:19 AM	Application	1,414 KB
Office365Training.pdf	6/4/2019 8:14 AM	Foxit Reader PDF ...	1,884 KB
PDFCreator-3_5_0-Setup.exe	5/31/2019 10:21 AM	Application	36,615 KB
vlc-3.0.6-win64.exe	5/31/2019 9:37 AM	Application	40,867 KB

Figure 4.4: The Downloads of the honeypot.



Name	Date modified	Type	Size
icon.ico	6/4/2019 11:20 AM	Icon	34 KB

Figure 4.5: The Pictures of the honeypot.

4.1.3 Hosting & Location

The honeypots have been hosted in Microsoft Azure [76]. This platform has been chosen because it had to be possible to backup and restore the honeypots in a short period of time, to collect as much information as possible. Hosting virtual machines in the cloud provides more flexibility compared to dedicated servers. It also improves the scalability of the honeypot architecture. Moreover, Azure has a command line interface which allows to backup and restore virtual machines remotely with a script. Using Azure allowed to manage the firewall of the virtual machines individually. Even when the Windows firewall of the honeypot was changed by an attacker, these changes do not have an effect because of the firewall managed in Azure. This improves the control over the honeypot and limits the misuse.

As mentioned in Section 3.1, the geographical location can have an effect on the amount of attention a honeypot receives, but the company where the honeypot is hosted has a greater influence. In Azure it is possible to select the region where a virtual machine is deployed. The location *West Europe* has been chosen for the honeypots. That region corresponds to where the fake company is supposedly located. The IP address ranges of Azure are publicly available, so anyone can check that the IP addresses of the honeypots are part of the Azure IP range.

4.1.4 Data Collection

Changes have been made to Windows to improve data collection, PowerShell and TaskScheduler logging have for example been enabled. *System Monitor* (Sysmon) [77] has also been installed. It is a system service which logs detailed information to the Windows event logs. A configuration file has to be provided to Sysmon to declare what needs to be monitored. The configuration file of SwiftOnSecurity [78] has been used. The log sizes of the *Security*, *PowerShell*, *System* and *Sysmon* logs have been increased. Moreover, the Windows Audit Policy has been enabled for the following categories:

- Account logon events
- Account management
- Logon events
- Object access
- Policy change
- Process tracking
- System events

The Windows Audit Policy allows to enable the auditing of specific kinds of events, such as the ones previously mentioned. By enabling auditing, information will be logged about the events concerning the enabled category. The following object access auditing settings have been applied to the Desktop, Documents, Downloads, Music, Picture and Videos folder:

- List folder / read data
- Read attributes
- Read extended attributes
- Create files / write data
- Create folders / append data
- Write attributes
- Write extended attributes
- Delete
- Change permissions

These object access auditing settings have been applied to improve the visibility on activities performed in these folders.

To have real-time visibility on the honeypots, without disturbing the attackers, Splunk [79] is used to gather and visualize real-time information. Splunk is a software solution which allows to collect, forward and index information. The indexed information can be used in many different ways, for example, scheduled searches can be performed, real-time data can be visualized, historic searches can be performed and reports can be made. To gather the information of all the different honeypots, Splunk Forwarders are installed which forward the Windows event logs partially to a Splunk Enterprise instance. An overview of the setup can be found in Figure 4.6. The Splunk Enterprise instance indexes all the log events which it receives. An encrypted connection is set up between the honeypot and the Splunk server to send the log events. The ports of the Splunk Forwarder and Splunk server have been changed from the default port to a random port number. This is to prevent easy recognition of the usage of Splunk. Besides the specific ports, no other communication is possible between the two virtual machines. This has been enforced in the firewall managed in Azure. Splunk has been chosen because it allows to perform actions, based on real-time information. This functionality has been used for the backup and restore process and to prevent misuse of the honeypots. Both will be elaborated on in the next two subsections. In Splunk you have to configure which Windows event logs are forwarded. Moreover, it is possible to create blacklists based on the event ID or the content of the log event. Both functionalities have been used to only send relevant events to the Splunk server.

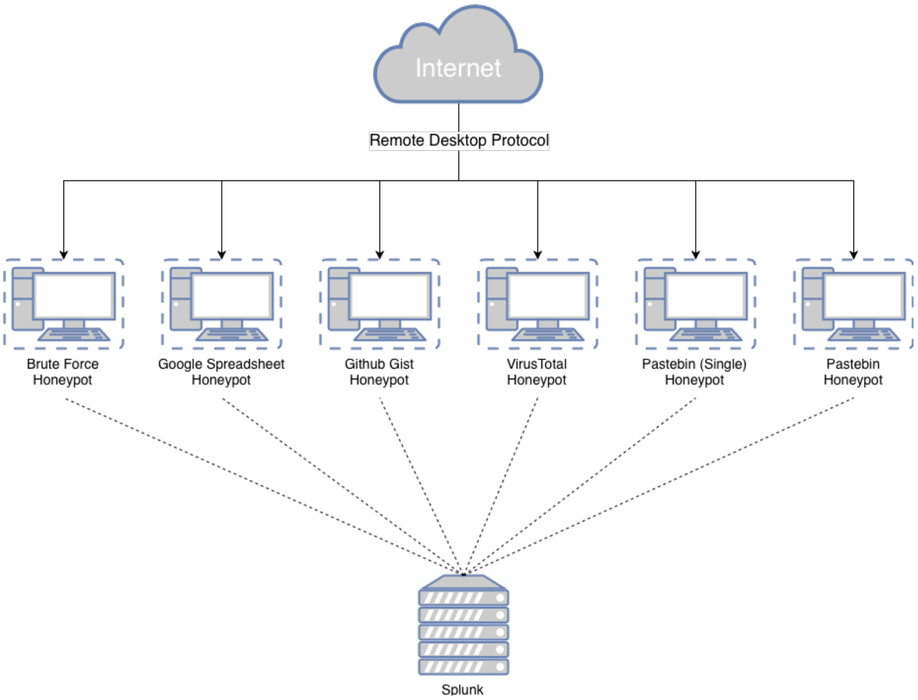


Figure 4.6: The honeypot architecture.

4.1.5 Automated Backup and Restore

An automated backup and restore process was necessary to be able to capture separate RDP sessions as virtual disk images. A real-time Splunk alert was created which monitored finished RDP sessions. A *transaction* was made between the Windows event of logging in and logging out of an account. When the session was at least 90 seconds long, a list of actions was started. The threshold of 90 seconds has been used to prevent triggering the backup when somebody only checks if the found credentials are correct. The first action was registering of the alert by the Splunk server. The second action was to send an email to an email address. This was done to keep track of how often alerts triggered. The third action was the execution of a Command Prompt script. This script started a PowerShell script, which used the Azure Command Line Interface (CLI) to backup and restore the honeypot.

To be able to restore the honeypots to a clean state, a snapshot was taken of each honeypot. From a snapshot, a new disk can be created. Azure allows to change the disk which is attached to a virtual machine. The combination of these functionalities were the base for the backup and restore process. The Azure CLI script performed the following actions:

1. Log in to Azure.
2. Stop and deallocate the virtual machine.
3. Create a new disk, based on the snapshot which belongs to the honeypot.
4. Change the disk used by the virtual machine to the newly created disk.
5. Start the virtual machine.
6. Log out of Azure.

Besides the previously mentioned alert, two other alerts were used to prevent misuse of the honeypot. These two alerts will be discussed in the next subsection. The use of the Azure CLI is not thread-safe. This is a problem because the setup uses three concurrent alerts which can trigger a backup and restore process. A Windows environment variable is used to prevent multiple scripts starting for the same honeypot. When the Azure CLI script starts, it first checks if any other scripts on the same honeypot are running. This prevents multiple scripts starting for the same honeypot. Another problem with the Azure CLI is that it uses a default folder to store received tokens from the login process. These tokens are used for the communication with Azure. When multiple scripts are started for different honeypots, information will be overwritten each time a new script starts. This will result in the previously active script failing to execute because the tokens are invalid. To solve this, a Windows environment variable is used to change the folder of the stored information to a unique folder for each honeypot. A combination of the Azure CLI and Splunk alerts allowed to create a fully automatic backup and restore process to automatically redeploy honeypots after being attacked.

4.1.6 Preventing Misuse

The honeypot architecture has to balance gathering information and preventing misuse. It is not a problem when the honeypot itself is attacked and destroyed but it becomes a problem when the honeypot is used to attack other systems. Not only from an ethical point of view, but also from a law abiding perspective. Splunk has been used to take preventative measures against misuse. The received Windows events and performance data of the honeypot were used to create two Splunk alerts:

- **Bandwidth Threshold:** The Splunk Forwarder is able to send performance data of the system to the Splunk server. Every ten seconds information about for example CPU, memory, storage and network usage is sent to the indexer. To prevent the honeypot being used for attacking other systems on the internet, a Splunk alert was used to monitor network usage. If the honeypot exceeded a threshold of two Mbit for five times in the past four minutes, where every ten seconds network usage is received, then the alert is triggered. The same backup and restore process is started as with normal RDP sessions, described in Section 4.1.5. The honeypot is taken offline for an extra 15 minutes to discourage the attacker of returning and starting the outgoing malicious activities again.

- **Session Length Threshold:** RDP session are monitored by the Splunk Enterprise instance by linking login and logout events in Windows. A Splunk transaction is used to link these two events together. The transaction allows to measure the time of an RDP session. It also allows to track incomplete transactions. This is used to check when RDP sessions are longer than 45 minutes. After the 45 minutes, the backup and restore process is started, the same process as described in Section 4.1.5 is applied. The threshold has been chosen because limiting the network usage alone does not prevent all different kinds of outgoing malicious behavior. To limit the malicious behavior, a maximum session length of 45 minutes has been chosen.

Restricting the honeypot too much can harm the credibility of the honeypot. As described in Section 3.1, the restrictions of a honeypot can be used as a detection mechanism. A honeypot which is easily detectable will not gather a lot of useful information. That is why the preventative measures have been limited to the ones described in this subsection.

4.1.7 Honeypot Detection

To gather useful information, it should be unknown to the attacker that the system is a honeypot. As such, during creation of a high-interaction honeypot, some challenges are faced. In the previous sections where the setup of the honeypot architecture was discussed, some aspects which could aid honeypot detection have already been addressed. For example, the honeypot not having a lot of historic data and having limited resources. This has been solved by creating a story around the honeypot which is reflected in many aspects of the honeypot. The content of the honeypot has also been carefully chosen, to look like the content which you would normally find on a production server. Beside the Splunk Forwarder, no other software has been used to monitor the honeypot. The software has been chosen to monitor the honeypot because it normally is also used to monitor a system, only the goal of the monitoring is different in this case. There is a risk that an attacker stops the Splunk Forwarder, thereby disabling the monitoring of the honeypot. This risk has been mitigated by the Splunk alert which checks if the RDP session does not exceed a 45 minute threshold. The login Windows event is always forwarded to the Splunk server because the attacker is not able to stop the Splunk Forwarder in that stage. So, when an attacker stops the Splunk Forwarder, the honeypot will be taken offline after 45 minutes.

High-interaction honeypots have the disadvantage of being more time consuming in maintenance and deployment compared to honeypots with a lower level of interaction. One of the advantages of using a high-interaction honeypot, is that some honeypot detection mechanism inherently are not applicable. Fingerprinting for example cannot be used to detect a high-interaction honeypot. It can be used for low-interaction honeypots which use predefined static answers [5]. Because a high-interaction honeypot does not use predefined answers, fingerprinting is not possible. Moreover, a high-interaction honeypot is not affected by random delays because of service emulation. Low- and medium-interaction honeypots can be affected by random delays because they emulate their services [8]. These random delays can aid honeypot detection.

As explained by Zakaria and Kiah [12], static honeypots are more likely to be detected by attackers. The created honeypot is static, the same setup is used for all the honeypots and it does not change after redeployment. This static setup has been chosen over a dynamic setup because the goal of the honeypot was to be used temporarily to collect information about attacks. It was not the goal to create a honeypot which was future proof. Static honeypots are easier to deploy because there is no need to update the configuration or the behavior of the honeypot [12].

Detecting honeypots based on their restrictive behavior is a detection mechanism which is impossible to completely mitigate. As already mentioned in the previous section about preventing misuse, a honeypot needs to prevent misuse from an ethical perspective and also to abide by the law. This is a characteristic which can always be used to detect honeypots. The only option is to make the restrictions look as naturally as possible. Some behavior of an attacker has to be stopped immediately to prevent damage to other systems.

4.2 Leaking Credentials

To lure attackers to the honeypot, credentials have been leaked on different platforms. In Section 3.1 popular platforms and credential generation methods have been discussed. Besides leaking credentials on normal websites, previous research has also leaked credentials on the dark web [57] [56]. To gain insights into how credentials are leaked and sold on the dark web, interviews have been conducted with John Fokker, head of Cyber Investigations for McAfee Advanced Threat Research, and with Tycho van Marle, Teamlead of a Research Team at Intel 471. The summaries of the conducted interviews can be found in Appendix C. The outcome of the interviews was that there were two relevant options for spreading RDP credentials on the dark web: sharing credentials on a forum or selling credentials in an RDP shop. The problem with the first option is that dark web forums address a specific subject, thereby attracting users interested in that subject. For example, there are forums where people share knowledge about ransomware and forums where people talk about credit card fraud. By spreading credentials on such kinds of forums, a specific subset of users is addressed. This has influence on the data gathered by the honeypot. When you spread credentials on a ransomware forum, it is more likely that people who use those credentials will install ransomware. So, this platform for spreading credentials is less interesting. The problem with selling credentials in an RDP shop, is that this has ethical concerns. Goods are sold under false pretense and a profit is made from the sale. That is why this option also has not been used for this research.

Based on previous research discussed in Section 3.2 and on the conducted interviews, credentials have been spread on the following platforms:

- **Pastebin**
  A paste was created on Pastebin with the following information:

> Title: RDP
> 52.142.195.56:3389@all-users;new-environment1!

- **Github Gist**
  A public Gist was posted with an embedded RDP connection file. The connection file connected to the honeypot. The following message was posted:

> Because of the recent ransomware attack, a temporary Confluence server has been deployed. Please use the rdp file to connect to the temporary server and the following credentials: temp-env;Welkom123!Temporary

- **Google Docs**
  A Google Docs document was shared publicly containing a public announcement for employees, signed by the CTO of the company:

> Dear employees,
>
> Due to the recent ransomware attack, the server hosting Confluence is unavailable.
> Because of this, a temporary server has been deployed with a recent backup. Unfortunately, we haven't been able to restore all the spaces.
> To access this server via RDP, please use the following information:
> 52.236.128.38:3389@confluence-temporary;TemporaryConfluenceEnv123
> Please use this temporary server until further notice.
>
> Regards,
> Serge Jalbert

- **VirusTotal**
  An RTF file was uploaded to VirusTotal. The file contained an announcement signed by somebody from the imaginary IT department of Manumia:

> NOTICE
>
> Currently the server hosting Confluence is unusable because of the recent ransomware attack.
> Temporarily a backup server will be used.
> This server can be accessed via RDP:
> 52.236.128.192:3389@temp-user;123ConfluenceKnowledge!
>
> We hope to get the original server up and running as fast as possible.
>
> Kind regards,
> Kerman Simon

Besides honeypots deployed for these platforms, another honeypot was deployed with a weak username and password to gather information about brute force attacks. The username *administrator* and the password *password* were used for the Brute Force honeypot. For the leaked credentials, strong usernames and passwords were used to prevent attackers breaking into the system with a brute force attack. On all the platforms, except for VirusTotal, the messages were posted publicly "by accident". Using VirusTotal as a platform to leak credentials came forward during the conducted interviews. Files can be uploaded to VirusTotal to check for malicious content. Files uploaded to VirusTotal can contain confidential information. Uploaded file are accessible via a private API of VirusTotal. To access this private API, you have to contact VirusTotal. An article was written about how companies accidentally leaked sensitive data by uploading it to VirusTotal [80].

Another honeypot has been deployed, while the others had been online for around three weeks, because the leaked credentials did not lead to successful login attempts up till then. For the new honeypot, credentials were spread six times a day on Pastebin. This method for leaking credentials was inspired by the work of Barron and Nikiforakis [50]. The same message was used as for the first Pastebin leak, only with different credentials and a different IP address:

> Title: RDP
> 51.144.52.247:3389@administrator;temporaryserverconf1!

The messages spread on the different platforms all contain the word *RDP* and the default port of RDP *3389* at least once. Moreover, the username, password, IP address and port have been displayed in a format which is commonly used: *0.0.0.0:3389@username;password*. This format was used to clearly show that the messages contain RDP credentials. Web scrapers sometimes search for specific formats or keywords like this.

4.3 Ethics & Law

Ethics and the law are two important aspects to take into account when using a honeypot. As already mentioned, it should be prevented that the honeypot is used to harm other systems. The difficulty with limiting misuse is that limiting the system too much can reveal that the system is a honeypot. As mentioned in Section 4.1.7, this has been used as a honeypot detection mechanism. Finding a proper balance between gathering information and preventing misuse is difficult. This balance needs to be assessed for each new honeypot deployment.

Besides the ethical aspects of using a honeypot, there is also a legal aspect which has to be taken into account. According to article 47, paragraph 1 of The Dutch Criminal Code, the provocation of committing a crime is prohibited. Deliberately targeting people to encourage them to perform a crime is prohibited, while not targeting specific people is not prohibited. The difference is that the attacker already had the intention to attack the system in the latter situation, while that might not have been the case when specific people are actively approached. Using a honeypot and leaking credentials to gain attention is a gray area. The question is whether leaking credentials on a platform falls under deliberately targeting specific people to encourage them to perform a crime.

4.4 Deployment

The five different honeypots have been deployed on the 19th of June 2019 at 14:24. As mentioned in Section 4.2, another honeypot was deployed where its credentials were posted on Pastebin six times a day. This honeypot was deployed on the 9th of July 2019 at 16:36. All the honeypots were taken offline at the 26th of July 2019 at 16:34. During the deployment of the honeypots, problems and setup flaws have been found in the honeypot architecture:

- The Windows event logs contain information about the creation of the honeypot. For example, events about previous RDP connections with the honeypot. The event contains the IP address of the system which connected to the honeypot, which is sensitive information. The Windows event logs could have been used to reveal that the system is a honeypot. This problem is difficult to solve because it is only possible to completely clear a Windows event log. Completely clearing the Windows event logs would have been a bigger indication that the system is a honeypot. So, it was not possible to solve this problem while the honeypots were still deployed. A possible solution would be to clear the Windows event logs when the honeypot creation is finished. After clearing the event logs, let the system run for a few weeks before deploying it. The sensitive information will not be present anymore and the system still has filled event logs.

- When the Splunk alert which checks the bandwidth triggers, it will start the backup and restore process. Because the honeypot is shutdown while the attacker is still logged in, the event of closing the RDP session is not always sent to the Splunk server. Because the Splunk Enterprise instance never registered that the session of the attacker was closed, the Splunk alert which checks the session duration exceeding 45 minutes will trigger, even though nobody has an active connection with the honeypot. The only possible solution is to first disconnect the attacker, wait for a few minutes and then shutdown the honeypot. Unfortunately, this is not possible with the Azure CLI, so there was no solution available for the created honeypot architecture.

- Attackers sometimes only check if credentials are valid or not. This results in RDP sessions of only a single second. This was a problem for the Splunk alert which monitors RDP sessions longer than 90 seconds. Because the login and logout events had the exact same timestamp, Splunk was not able to setup a transaction between those events. Random transactions were made between previous login events and new logout events. This resulted in false positive triggers of the Splunk alert, this happened sporadically. The problem was solved by adding an extra constraint to the query used for the Splunk alert. The duration of the RDP session was not allowed to be longer than 45 minutes. This filtered out the false positive triggers.

- During a session of an attacker, the password of the user account was changed. The attacker did this in less than 90 seconds, so the Splunk alert did not trigger. The attacker did not log into the system for five days. Because the password of the user account was changed, only the attacker was able to login. The honeypot architecture had no solution for this situation. The backup and restore process had to be started manually. This problem can be solved in the future by checking for the Windows events generated by a password change. If a password change happened in a session which was shorter than the threshold value, start the backup and restore process.

- Because there is a delay between an attack closing the RDP session and the Splunk alert starting the backup and restore process, occasionally new attackers logged into the system. Their session was cut short because of the backup and restore process. In rare occasions the delay was long enough for the new attacker to have an RDP session longer than 90 seconds. This triggered the Splunk alert checking for RDP sessions again. This was not a problem because the scenario was taken into account during the creation of the honeypot architecture. The scripts first check if any other script is already running. If that is the case, the script will not start. Nonetheless, this flaw has to be taken into account when analyzing the honeypot image. The second RDP session potentially influenced evidence of the first RDP session.

4.5 Classification

To be able to reason about the gathered honeypot data and to structure its processing, a classification has been created. The goal of this classification is to identify which kinds of attacks have been performed. The classification was also created to improve the efficiency of forensic analysis. The proposed classification can be used to prioritize the analysis of digital artifacts. Effective prioritization can improve the time spent on a forensic analysis. The gathered honeypot data has been used to evaluate the classification. The results of the evaluation are two-fold. It provides feedback on the proposed classification and the performed attacks are identified.

The proposed classification has been created based on previous research and interviews with the members of the Northwave CERT. The proposed classification is a decision tree where the leaf nodes represent attacks, nodes represent attack properties shared by multiple attacks and the edges contain conditions based on the presence of digital artifacts. The method used to construct the decision tree was inspired by the work of Gadelrab et al. [72]. They proposed an attack process model, based on the execution patterns of attacks. The execution pattern of an attack was represented by a sequence of unique attack steps. This way of representing an attack has been used for the creation of the proposed classification. Instead of using attack steps, digital artifacts have been used to create unique patterns which identify attacks. By defining unique patterns for the different attacks, similarities can be found. These similarities have been used to structure the decision tree.

First, a list of relevant attacks was created. The work of Pande et al. [81] contains an extensive list of known cyber attacks. The original source of the list was a website which unfortunately does not exist anymore. Simmons et al. [73] created a cyber attack taxonomy of five dimensions.

Two of those dimensions, *Operational Impact* and *Informational Impact*, contain lists of relevant cyber attacks. Another taxonomy which was used, was the taxonomy proposed by Joshi and Singh [74]. They also propose a five dimensional taxonomy, where the *Method* dimension contains a list of cyber attacks.

For each cyber attack included in the proposed classification, a unique sequence of digital artifacts had to be defined. In previous research, characteristic features of cyber attacks have already been identified. These characteristic features have been used to derive digital artifacts for the unique sequences. The life-cycle of ransomware has, for example, been described by Tailor and Patel [82] and Silva et al. [83] and the attack model of the current generation of Ransomware has been explained by Zimba and Chishimba [84]. Based on this previous research, relevant digital artifacts have been identified for the Ransomware cyber attack. Hoque et al. [85] have described the life cycle of a Botnet, this information has been used to derive digital artifacts for the Distributed Denial-of-Service (DDoS) Bot. A detection algorithm to detect Cryptominers has been proposed by Draghicescu et al. [86]. The algorithm contains a list of indicators which are used to detect Cryptominers. These indicators have been used to define relevant digital artifacts for Crytominers. The work of Yilmaz and Zavrak [87] mentions multiple characteristic features of Adware in their review of existing kinds of Adware. Their work has been used to derive relevant digital artifacts for Adware.

Besides a review of existing literature, interviews have been conducted with employees of the Northwave CERT. The following questions have been asked:

1. What kind of attack families do exist?
2. Which attacks do these attack families consist of?
3. What is the difference between these attacks from a forensics point of view?

After conducting the first interview, it appeared that the order of the questions was not logical. For the remaining interviews, the order was changed. First, the interviewee was asked to create a list of cyber attacks relevant for the domain of the created honeypot. The next question was what uniquely, identifiable characteristics each attack has. Finally, the cyber attacks were grouped by the interviewee, based on the characteristics of the previous question. This order turned out to be more logical. The results of the interviews can be found in Appendix D.

Based on the literature review and the conducted interviews, a classification was created. Cyber attacks which have been mentioned by more than one interviewee have been directly included in the classification. Cyber attacks which have been mentioned by only one interviewee, or which were not mentioned at all during the interviews but were present in the reviewed literature, have been assessed on a per attack basis. Aspects such as the domain of the research setup and the likelihood of the attack, have been used to decide on whether to include the attack in the classification or not. Attacks which currently are performed often, such as defacement, phising and SQL injection, but which cannot be verified by the created honeypot setup, are not included in the proposed classification. For future work, the method used for the creation of the classification could be utilized to add other kinds of attacks to the classification. This would require changing the honeypot setup to be able to verify the added attacks.

The resulting classification has been used on a random sample of twelve images, to filter out general problems in an early stage. As a result of the analysis, every edge was inspected for being too broad or to narrowly defined. This was something that regularly occurred while using the classification. A classification called *Information Gain* was removed. The idea behind the category was to classify reconnaissance, but the problem was that it was not possible to define an edge for the classification which included the behavior entirely. Besides this problem, it was carried out by almost every attacker, ranging from investigating the system properties to inspecting all the user folders thoroughly. Because this behavior was persistently present, the classification was renamed to *Reconnaissance* and was added as a standard action in the classification after logging in. Moreover, a node has been added to the classification called *Pivoting*, which is split into *External* and *Internal*. Both nodes split out into *Single System* and *Multiple Systems* leaf nodes. The *External* node has an extra leaf node called *Hiding Web Activity*. These nodes and leaf nodes have been added because the honeypot was occasionally used as a stepping stone for other malicious behavior, where internal or external systems were targeted.

The proposed classification is displayed in Figure 4.7. All the round nodes represent shared digital artifact sequences, where the shared sequence received a name describing the contents of the sequence. The square leaf nodes are the attack classifications. All the edges between the nodes are numbered. These numbers are tied to an edge description. The numbers are only used for the descriptions, they do not constitute an ordering or a weight.

As previously mentioned, the classification is a decision tree where the edges contain constraints based on the presence of digital artifacts. When the conditions of an edge are met, traversing the edge to the connected node is allowed. This process is reiterated until you reach a leaf node or none of the constrains of any connected edge is met. The classification allows for multiple attacks to be classified. If the constraints of multiple edges are met, including the same edge multiple times, the order of traversing is based on the creation time of the digital artifact. The edge which has the digital artifact with the earliest creation time is traversed first. By reiterating the aforementioned workflow, multiple attacks can be classified. All possible paths are unique sequences of digital artifacts, identifying a cyber attack. After Figure 4.7, a description can be found of all the edges of the classification. The constraints of each edge are numbered. The number is based on the previous amount of constraints which have been met up till then. Subsequently, all the leaf nodes are explained briefly.

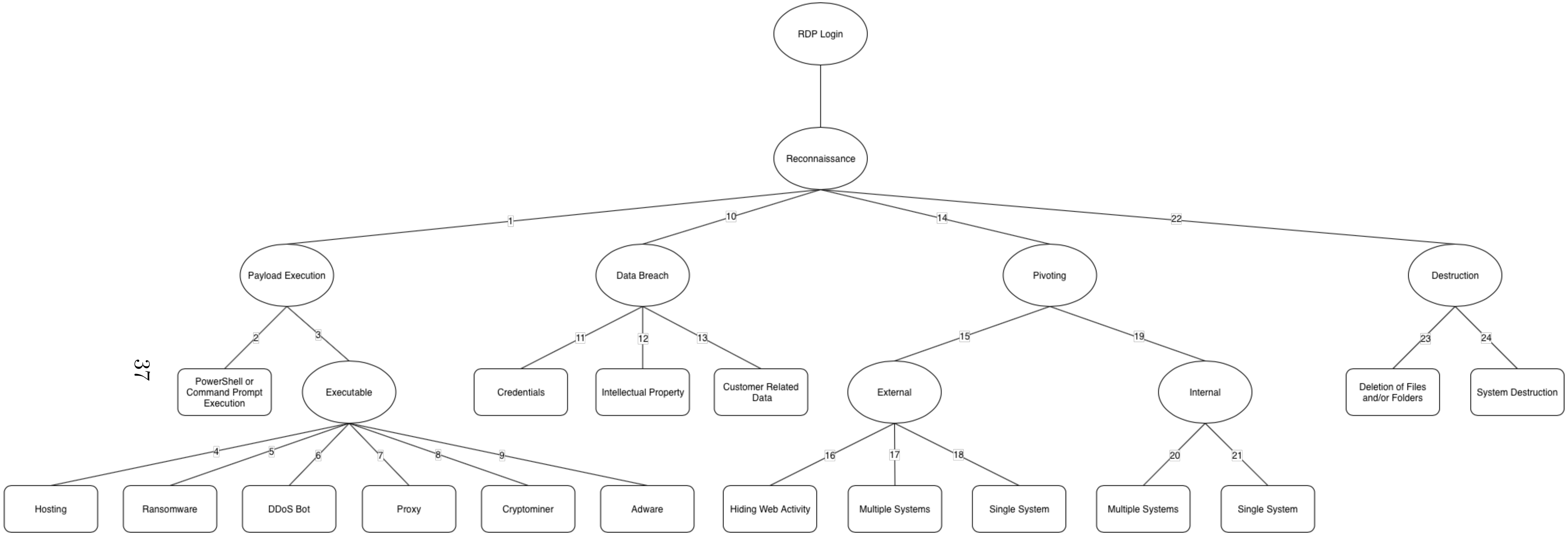


Figure 4.7: Proposed cyber attack classification.

Constraints of the different edges:

1. **Reconnaissance** → **Payload Execution**:
   (1) Execution of an executable, script file or command.

2. **Payload Execution** → PowerShell or Command Prompt Execution:
   (2) PowerShell or Command Prompt process is executed.
   (3) Events in the Windows event logs which contain executed commands in PowerShell or the Command Prompt.
   (4) Optionally, the transfer or creation of a Command Prompt and/or PowerShell script.

3. **Payload Execution** → **Executable**:
   (2) The executed payload is an executable.
   (3) Optionally, the payload is transferred to the system.

4. **Executable** → Hosting:
   (4) Installation of a new service.
   (5) One or more ports are opened in the firewall.
   (6) Incoming and/or outgoing traffic through the opened port/ports.

5. **Executable** → Ransomware:
   (4) Creation of shadow copies is disabled and existing copies are deleted.
   (5) Encryption process which consist of high CPU usage and high disk I/O.
   (6) Encrypted files are created with a different extension.
   (7) Ransomnotes are created.

6. **Executable** → DDoS Bot:
   (4) Contact with a Command and Control server.
   (5) Large volume of outgoing network traffic originating from the executable, to one or more IP addresses.

7. **Executable** → Proxy:
   (4) Port opened in the firewall.
   (5) The executable actively listens on the opened port.
   (6) Incoming and outgoing traffic through the opened port.

8. **Executable** → Cryptominer:

    (4) Outgoing network traffic to a mining address (retrieving information).

    (5) Mining process generating high CPU usage.

9. **Executable** → Adware:

    (4) Installation of one or more browser plugins.

10. **Reconnaissance** → Data Breach:

    (1) Read operations, generating disk I/O, on files which contain information valuable to the owner of the system.

11. **Data Breach** → Credentials:

    (2) Credential related files are accessed.

12. **Data Breach** → Intellectual Property:

    (2) Files containing intellectual property are accessed.

13. **Data Breach** → Customer Related Data:

    (2) Files containing customer related data are accessed.

14. **Reconnaissance** → Pivoting:

    (1) Large volume of network traffic.

15. **Pivoting** → External:

    (2) The outgoing network traffic is directed to one or more IP addresses outside the local network.

16. **External** → Hiding Web Activity:

    (3) A browser is used to visit websites.

17. **External** → Multiple Systems:

    (3) The outgoing network traffic is directed to multiple different IP addresses, one or more different ports can be accessed per IP address.

18. **External** → Single System:

    (3) The outgoing network traffic is directed to a single IP address, accessing multiple ports.

19. **Pivoting** → Internal:

    (2) The outgoing network traffic is directed to one or more IP addresses inside the local network.

20. **Internal** → Multiple Systems:

    (3) The outgoing network traffic is directed to multiple different IP addresses, one or more different ports can be accessed per IP address.

21. **Internal** → Single System:

    (3) The outgoing network traffic is directed to a single IP address, one or more different ports can be accessed.

22. **Reconnaissance** → Destruction:

    (1) A series of delete operations on important files.

23. **Destruction** → Deletion of Files and/or Folders:

    (2) Files and/or folders are deleted which are important to the owner of the system.

24. **Destruction** → System Destruction:

    (2) System critical files are deleted, making the system unusable.

Classifications:

- **Hosting:** Service hosting to facilitate other malicious activities, for example, hosting a phishing website.
- **Ransomware:** Malware which encrypts files, thereby blocking access. The victim is asked for ransom to regain access to the encrypted files.
- **DDoS Bot:** Malware which allows an attacker to use the system in a Distributed Denial-of-Service attack.
- **Proxy:** Software which uses the system as a proxy server.
- **Cryptominer:** Malware which uses system resources to mine cryptocurrencies.
- **Adware:** Malware showing advertisements in the user interface of the system.
- **Hiding Web Activity:** Hiding web activity by using the browser of the system. This includes misusing active sessions in the browser.
- **External Pivoting - Multiple Systems:** Interaction with multiple external IP addresses.
- **External Pivoting - Single System:** Interaction with a single external IP address.
- **Internal Pivoting - Multiple Systems:** Interaction with multiple internal IP addresses.
- **Internal Pivoting - Single System:** Interaction with a single internal IP address.
- **Deletion of Files and/or Folders:** Files and/or folders are removed from the system.
- **System Destruction:** The system is destroyed up to a point where it is not usable anymore.

Checking the constraints of the edges requires knowledge about the location of relevant digital artifacts. Where digital artifacts are located, depends on the operating system used. This research will focus on the Windows operating system, which is used for the created honeypot. The SANS Institute offers a training on Windows Forensic Analysis [88]. For the training, a poster was created which contains a summary of digital artifacts on Windows. The summary includes what the digital artifacts mean and their location [89]. An overview of the most used digital artifacts used during the evaluation can be found after this paragraph. The digital artifacts are grouped based on the classification of the SANS poster. The descriptions originate from the aforementioned SANS poster. Additionally, relevant Windows event IDs have been added to the overview. The description of the event IDs originate from the Windows documentation [90].

- **Program Execution:**
  - *UserAssist:* GUI-based programs launched from the desktop are tracked in the launcher on a Windows System.
  - *Shimcache:* Windows Application Compatibility Database is used by Windows to identify possible application compatibility challenges with executables. Tracks the executables file name, file size, last modified time, and in Windows XP the last update time.
  - *Amcache.hve:* ProgramDataUpdater (a task associated with the Application Experience Service) uses the registry file Amcache.hve to store data during process creation
  - *Last-Visited MRU:* Tracks the specific executable used by an application to open the files documented in the OpenSaveMRU key. In addition, each value also tracks the directory location for the last file that was accessed by that application.
  - *Jump Lists:* The Windows 7 task bar (Jump List) is engineered to allow users to “jump” or access items they have frequently or recently used quickly and easily. This functionality cannot only include recent media files; it must also include recent tasks. The data stored in the AutomaticDestinations folder will each have a unique file prepended with the AppID of the associated application.
  - *Event ID 4688:* A new process has been created.
- **Deleted File or File Knowledge**
  - *IE/Edge file://:* A little-known fact about the IE History is that the information stored in the history files is not just related to Internet browsing. The history also records local and remote (via

network shares) file access, giving us an excellent means for determining which files and applications were accessed on the system, day by day.

  - *Win7/8/10 Recycle Bin*: The recycle bin is a very important location on a Windows file system to understand. It can help you when accomplishing a forensic investigation, as every file that is deleted from a Windows recycle bin aware program is generally first put in the recycle bin.
  - *Last-Visited MRU*: Tracks the specific executable used by an application to open the files documented in the OpenSaveMRU key. In addition, each value also tracks the directory location for the last file that was accessed by that application.
  - *Event ID 4659:* A handle to an object was requested with intent to delete.
  - *Event ID 4660:* An object was deleted.

- **Network Activity/Physical Location:**
  - *Cookies:* Cookies give insight into what websites have been visited and what activities may have taken place there.
  - *Browser Search Terms:* Records websites visited by date and time. Details stored for each local user account. Records number of times visited (frequency). Also tracks access of local system files. This will also include the website history of search terms in search engines.
  - *Event ID 4946:* A change has been made to Windows Firewall exception list. A rule was added.
  - *Event ID 4957:* Windows Firewall did not apply the following rule.
  - *Event ID 5152:* The Windows Filtering Platform blocked a packet.
  - *Event ID 5156:* The Windows Filtering Platform has permitted a connection.
  - *Event ID 5158:* The Windows Filtering Platform has permitted a bind to a local port.

- **File/Folder Opening:**
  - *Open/Save MRU:* In the simplest terms, this key tracks files that have been opened or saved within a Windows shell dialog box. This happens to be a big data set, not only including web browsers like Internet Explorer and Firefox, but also a majority of commonly used applications.

  - *Recent Files:* Registry Key that will track the last files and folders opened and is used to populate data in "Recent" menus of the Start menu.
  - *Jump Lists:* The Windows 7 task bar (Jump List) is engineered to allow users to "jump" or access items have frequently or recently used quickly and easily. This functionality cannot only include recent media files; it must also include recent tasks. The data stored in the AutomaticDestinations folder will each have a unique file prepended with the AppID of the association application and embedded with LNK files in each stream.
  - *Shell Bags:* Which folders were accessed on the local machine, the network, and/or removable devices. Evidence of previously existing folders after deletion/overwrite. When certain folders were accessed.
  - *Shortcut (LNK) Files:* Shortcut Files automatically created by Windows: recent items and opening local and remote data files and documents will generate a shortcut file (.lnk).
  - *Last-Visited MRU:* Tracks the specific executable used by an application to open the files documented in the OpenSaveMRU key. In addition, each value also tracks the directory location for the last file that was accessed by that application.
  - *IE/Edge file://:* A little-known fact about the IE History is that the information stored in the history files is not just related to Internet browsing. The history also records local and remote (via network shares) file access, giving us an excellent means for determining which files and applications were accessed on the system, day by day.
  - *Event ID 4656:* A handle to an object was requested.
  - *Event ID 4663:* An attempt was made to access an object.

- **Account Usage:**

  - *Last Login:* Lists the local accounts of the system and their equivalent security identifiers.
  - *Last Password Change:* Lists the last time the password of a specific local user has been changed.
  - *Logon Types:* Logon Events can give us very specific information regarding the nature of account authorizations on a system if we know where to look and how to decipher the data that we find. In addition to telling us the date, time, username, hostname, and success/failure status of a logon, Logon Events also enables us to determine by exactly what means a logon was attempted.

  - *Authentication Events:* Authentication mechanisms.
  - *Success/Fail Logons :* Determine which accounts have been used for attempted logons. Track account usage for known compromised accounts
  - *Event ID 4624:* An account was successfully logged on.
  - *Event ID 4625:* An account failed to log on.
  - *Event ID 4634:* An account was logged off.
  - *Event ID 4720:* A user account was created.
  - *Event ID 4724:* An attempt was made to reset an account's password.
  - *Event ID 4735:* A security-enabled local group was changed.
  - *Event ID 4738:* A user account was changed.
  - *Event ID 4776:* The computer attempted to validate the credentials for an account.

- **Browser Usage:**
  - *History:* Records websites visited by date and time. Details stored for each local user account. Records number of times visited (frequency). Also tracks access of local system files.
  - *Cookies:* Cookies give insight into what websites have been visited and what activities may have taken place there.
  - *Cache:* The cache is where web page components can be stored locally to speed up subsequent visits. Gives the investigator a "snapshot in time" of what a user was looking at online.

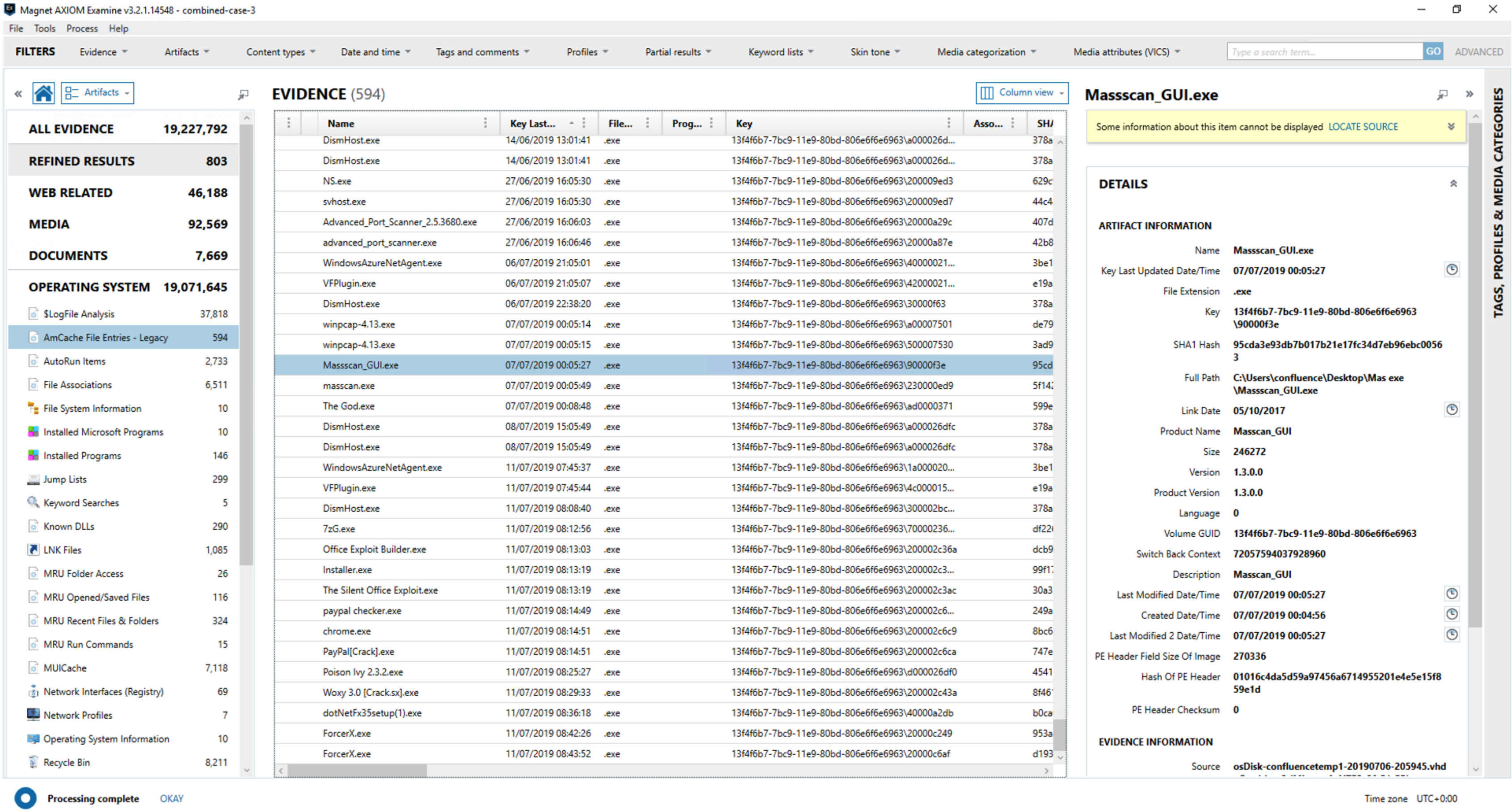
4.6 Evaluation Steps

The evaluation serves two different purposes: evaluating the performance of the proposed classification and discovering which cyber attacks have been performed. For each different image, the analysis starts with finding the date and time of the RDP session which triggered the backup and restore process. After this session has been found, all activity within the time frame of the RDP session is classified by the proposed classification. The result of the classification sets up a hypothesis concerning which attacks have been performed. Next, a more thorough analysis is performed to test the hypothesis. It checks whether the classifications given by the proposed classification were correct. To structure the more thorough analysis, the following three questions are answered:

1. Are all the given classifications correct?
2. Was the attack classified with too few classifications?
3. Did the classification miss information which was important to determine what the attacker did?

The goal of these questions are to verify if the classifications are correct, to check if no attacks have been missed and to check if any relevant artifacts have been missed. The answers to these questions are compared to the given classifications, validating the hypothesis. The validation provides feedback on the performance of the proposed classification. It also provides information about which attacks have been performed. This is information which is necessary for answering the research question.

Searching manually for the different digital artifacts takes a lot of time. To decrease the analysis time per image, Magnet Axiom has been used to extract digital artifacts [91]. It is a piece of software which extracts digital artifacts from a broad variety of platforms. Figure 4.8 shows the graphical user interface of Magnet Axiom. The figure shows how information is displayed about extracted digital artifacts. Besides Magnet Axiom, Splunk [79] has been used to analyze the Windows event logs. Splunk already indexed the forwarded Windows event logs of the different honeypots because it was used to actively monitor the honeypots. Splunk allows to search through the Windows events by formulating search queries. It has an extensive set of features to facilitate searching through large amounts of data. Using these two pieces of software reduced the analysis time per image.



Magnet AXIOM Examine v3.2.1.14548 - combined-case-3

ALL EVIDENCE	19,227,792
REFINED RESULTS	803
WEB RELATED	46,188
MEDIA	92,569
DOCUMENTS	7,669
OPERATING SYSTEM	19,071,645
$LogFile Analysis	37,818
AmCache File Entries - Legacy	594
AutoRun Items	2,733
File Associations	6,511
File System Information	10
Installed Microsoft Programs	10
Installed Programs	146
Jump Lists	299
Keyword Searches	5
Known DLLs	290
LNK Files	1,085
MRU Folder Access	26
MRU Opened/Saved Files	116
MRU Recent Files & Folders	324
MRU Run Commands	15
MUICache	7,118
Network Interfaces (Registry)	69
Network Profiles	7
Operating System Information	10
Recycle Bin	8,211

EVIDENCE (594)

Name	Key Last...	File...	Prog...	Key	Asso...	SH/
DismHost.exe	14/06/2019 13:01:41	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\a000026d...		378a
DismHost.exe	14/06/2019 13:01:41	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\a000026d...		378a
NS.exe	27/06/2019 16:05:30	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\200009ed3		629c
svhost.exe	27/06/2019 16:05:30	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\200009ed7		44c4
Advanced_Port_Scanner_2.5.3680.exe	27/06/2019 16:06:03	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\20000a29c		407d
advanced_port_scanner.exe	27/06/2019 16:06:46	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\20000a87e		42b8
WindowsAzureNetAgent.exe	06/07/2019 21:05:01	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\40000021...		3be1
VFPlugin.exe	06/07/2019 21:05:07	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\42000021...		e19a
DismHost.exe	06/07/2019 22:38:20	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\30000f63		378a
winpcap-4.13.exe	07/07/2019 00:05:14	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\a00007501		de79
winpcap-4.13.exe	07/07/2019 00:05:15	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\500007530		3ad9
Massscan_GUI.exe	07/07/2019 00:05:27	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\90000f3e		95cd
masscan.exe	07/07/2019 00:05:49	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\230000ed9		5f14
The God.exe	07/07/2019 00:08:48	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\ad0000371		599e
DismHost.exe	08/07/2019 15:05:49	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\a000026dfc		378a
DismHost.exe	08/07/2019 15:05:49	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\a000026dfc		378a
WindowsAzureNetAgent.exe	11/07/2019 07:45:37	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\1a000020...		3be1
VFPlugin.exe	11/07/2019 07:45:44	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\4c000015...		e19a
DismHost.exe	11/07/2019 08:08:40	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\300002bc...		378a
7zG.exe	11/07/2019 08:12:56	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\70000236...		df22
Office Exploit Builder.exe	11/07/2019 08:13:03	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\200002c36a		dcb9
Installer.exe	11/07/2019 08:13:19	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\200002c3...		99f1
The Silent Office Exploit.exe	11/07/2019 08:13:19	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\200002c3ac		30a3
paypal checker.exe	11/07/2019 08:14:49	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\200002c6...		249a
chrome.exe	11/07/2019 08:14:51	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\200002c6c9		8bc6
PayPal[Crack].exe	11/07/2019 08:14:51	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\200002c6ca		747e
Poison Ivy 2.3.2.exe	11/07/2019 08:25:27	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\d000026df0		4541
Woxy 3.0 [Crack.sx].exe	11/07/2019 08:29:33	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\200002c43a		8f46
dotNetFx35setup(1).exe	11/07/2019 08:36:18	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\40000a2db		b0ca
ForcerX.exe	11/07/2019 08:42:26	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\20000c249		953a
ForcerX.exe	11/07/2019 08:43:52	.exe		13f4f6b7-7bc9-11e9-80bd-806e6f6e6963\20000c6af		d193

Massscan_GUI.exe

Some information about this item cannot be displayed LOCATE SOURCE

DETAILS

ARTIFACT INFORMATION

Name	Massscan_GUI.exe
Key Last Updated Date/Time	07/07/2019 00:05:27
File Extension	.exe
Key	13f4f6b7-7bc9-11e9-80bd-806e6f6e6963 \90000f3e
SHA1 Hash	95cda3e93db7b017b21e17fc34d7eb96ebc0056 3
Full Path	C:\Users\confluence\Desktop\Mas exe \Massscan_GUI.exe
Link Date	05/10/2017
Product Name	Masscan_GUI
Size	246272
Version	1.3.0.0
Product Version	1.3.0.0
Language	0
Volume GUID	13f4f6b7-7bc9-11e9-80bd-806e6f6e6963
Switch Back Context	72057594037928960
Description	Masscan_GUI
Last Modified Date/Time	07/07/2019 00:05:27
Created Date/Time	07/07/2019 00:04:56
Last Modified 2 Date/Time	07/07/2019 00:05:27
PE Header Field Size Of Image	270336
Hash Of PE Header	01016c4da5d59a97456a6714955201e4e5e15f8 59e1d
PE Header Checksum	0

EVIDENCE INFORMATION

Source	osDisk-confluencetemp1-20190706-205945.vhd

Processing complete OKAY

Time zone UTC+0:00

Figure 4.8: The graphical user interface of Magnet Axiom.

Chapter 5

Results

The results of the evaluation will be discussed in the coming sections. First, the results of the evaluation of the proposed classification will be elaborated on. This will be explained first because the results address aspects which possibly influenced the performed cyber attacks, thereby possibly influencing the results which are discussed in Section 5.2. This section will provide statistics about the honeypots and the classified cyber attackers will be elaborated on.

5.1 Classification Evaluation

In total, 35 different RDP sessions have been analyzed. Of those sessions, 22 come from the Brute Force honeypot and 13 come from the Pastebin honeypot. The honeypots have been named after their way of acquiring credentials: *Brute Force*, *Google Spreadsheet*, *Github Gist*, *VirusTotal*, *Pastebin (single)* and *Pastebin*. A comprehensive description of the analysis of all the sessions can be found in Appendix E. Basic information about the image is mentioned first, such as why the backup process was triggered, the IP address of the attacker and the duration of the RDP session. Next, the classification is used to classify the actions taken by the attacker in this session. All relevant digital artifacts which were used by the classification are mentioned in detail. Lastly, the questions mentioned in Section 4.6 are answered by a more thorough analysis. The images used for the evaluation are a sample of the period the honeypots were online. Table 5.1 shows a timeline of the evaluated images. The date and time represent when the backup and restore process started. The following issues were identified by the evaluation of the classification:

- In multiple RDP sessions, the password of the user account was changed or a new account was added. This behavior is not included in the proposed classification. It will only be classified when the aforementioned actions are performed via PowerShell or a Command Prompt. In the RDP session of the *osDisk-confluencetemp1-20190625-113600* image, for example, a binary was used to add an extra user account. This activity was not classified.

- The proposed classification occasionally did not classify an attack, because it was not executed yet. For example, in the session of the *osDisk-confluencetemp1-20190714-143238* image, a cryptominer was installed. A task was scheduled to start the cryptominer but because the attacker logged out, the backup and restore process started. This happened before the cryptominer was executed. The origin of this issue lies in the honeypot architecture, not in the proposed classification. When the backup and restore process would not have started directly after the attacker logged out, the attack would have been classified correctly by the proposed classification. This issue with the backup and restore process also prevents attackers from reconnecting with the honeypot. In the session of the *osDisk-confluencetemp1-20190714-190330* image, a brute force tool was transferred to the desktop. After the transfer, the attacker logged out and almost directly logged in again. Because the first session was longer than 90 seconds, the backup and restore process started, terminating the second session of the attacker.

- In the RDP session of the *osDisk-confluencetemp6-20190712-132216* image, the attacker used the Windows *Add Roles and Features Wizard* to install new components related to the creation of a web server. The installed features have not been used because the installation exceeded the 45 minutes session threshold. If the attacker would have been able to continue his activities and possibly hosted malicious content on the honeypot, this would not have been classified by the proposed classification. This scenario should be classified as *Hosting* but the constraint on the first edge leading to the *Hosting* leaf node states that an executable has to be executed. In this case, not an executable but a Windows component was used to install the necessities.

- During the sessions of the *osDisk-confluencetemp6-20190726-113406* image and the *osDisk-confluencetemp1-20190706-205945* image, port scans have been executed by only sending SYN packets. The TCP connections were not completed, so they were not logged in the Windows event logs. Therefore, the attacks were not classified. During the thorough analysis, it was discovered that the firewall blocked a large amount of incoming traffic from different IP addresses, but all from the same port. This is a strong indication that a port scan has been performed. The problem is that this piece of information depends on the configuration of the firewall.

- An inherent problem with spreading credentials to lure attackers, is that there is a possibility that an attacker terminates the session of another attacker by also logging into the system. This renders the collected information useless because the activities of the first attacker are cut short and the session of the second attacker does not start with a clean image. The aforementioned scenario has happened occasionally.

- As the clean honeypot image becomes older, the number of available software updates increases. If a software update is available, it generally first needs to be installed before the software can be used. This was the case for multiple pieces of software installed on the honeypot. Software updates create a lot of digital artifacts. This increases the analysis time of the honeypot image because more digital artifacts have to be inspected.

- The *Hiding Web Activity* classification includes a broad variety of activities. In multiple sessions, attackers went to payment websites which have an auto login feature such as `https://www.paypal.com/`. The browser has also been used to download files to facilitate other malicious behavior. Moreover, the browser has been used to check credentials on various websites. The leaf node includes too much activity. The *Large volume of network traffic* constraint which needs to be met to reach the *Hiding Web Activity* leaf node is too vague in the context of browser usage. In one of the sessions Chrome was installed. The installation showed browser activity and outgoing network connections. It was classified as *Hiding Web Activity* while this obviously was not the case.

Some of the aforementioned issues possibly influenced the performed cyber attacks. The possible influence will be elaborated on in the Discussion chapter. An overview of the classified attacks and if they were correct or not, can be found in Table 5.2. The *Classification* column contains the results of the proposed classification and the *Thorough Analysis* column contains the results of the manual analysis. Unfortunately, it was not possible to cover all the different paths of the decision tree. This would require analyzing more images. Writing a master thesis forces to work under time constraints, it was not possible to analyze more images. Nonetheless, the evaluation shows that the proposed classification works for different kinds of attacks. It also shows that the method used for the evaluation is effective in providing feedback on the classification.

Date and Time	**Honeypot**	**Image Name**
2019-06-22 09:40:33	Brute Force honeypot	osDisk-confluencetemp1-20190622-085704
2019-06-25 18:35:01	Brute Force honeypot	osDisk-confluencetemp1-20190625-113600
2019-06-25 20:45:08	Brute Force honeypot	osDisk-confluencetemp1-20190625-183650
2019-06-26 07:35:34	Brute Force honeypot	osDisk-confluencetemp1-20190626-034704
2019-06-27 18:35:01	Brute Force honeypot	osDisk-confluencetemp1-20190627-161038
2019-06-28 12:05:01	Brute Force honeypot	osDisk-confluencetemp1-20190628-102113
2019-06-29 03:28:31	Brute Force honeypot	osDisk-confluencetemp1-20190628-155626
2019-06-29 15:01:41	Brute Force honeypot	osDisk-confluencetemp1-20190629-041201
2019-06-30 13:29:15	Brute Force honeypot	osDisk-confluencetemp1-20190630-125301
2019-07-03 07:35:02	Brute Force honeypot	osDisk-confluencetemp1-20190703-054200
2019-07-07 00:09:01	Brute Force honeypot	osDisk-confluencetemp1-20190706-205945
2019-07-09 05:22:46	Brute Force honeypot	osDisk-confluencetemp1-20190708-211449
2019-07-11 06:09:07	Brute Force honeypot	osDisk-confluencetemp1-20190711-042216
2019-07-11 15:01:58	Pastebin honeypot	osDisk-confluencetemp6-20190711-130651
2019-07-12 13:43:46	Pastebin honeypot	osDisk-confluencetemp6-20190712-132216
2019-07-12 14:15:08	Pastebin honeypot	osDisk-confluencetemp6-20190712-140509
2019-07-14 15:38:48	Pastebin honeypot	osDisk-confluencetemp6-20190714-121244
2019-07-14 19:01:33	Brute Force honeypot	osDisk-confluencetemp1-20190714-143238
2019-07-15 04:35:36	Brute Force honeypot	osDisk-confluencetemp1-20190714-190330
2019-07-15 17:51:18	Pastebin honeypot	osDisk-confluencetemp6-20190715-173400
2019-07-15 23:53:57	Brute Force honeypot	osDisk-confluencetemp1-20190715-113155
2019-07-16 13:35:06	Brute Force honeypot	osDisk-confluencetemp1-20190716-131759
2019-07-17 07:14:48	Brute Force honeypot	osDisk-confluencetemp1-20190717-041135
2019-07-18 09:40:36	Pastebin honeypot	osDisk-confluencetemp6-20190718-091249
2019-07-20 17:48:06	Pastebin honeypot	osDisk-confluencetemp6-20190720-063730
2019-07-22 10:55:01	Brute Force honeypot	osDisk-confluencetemp1-20190722-080211
2019-07-22 13:43:49	Pastebin honeypot	osDisk-confluencetemp6-20190722-132356
2019-07-23 04:51:00	Pastebin honeypot	osDisk-confluencetemp6-20190723-044035
2019-07-23 22:21:14	Pastebin honeypot	osDisk-confluencetemp6-20190723-203232
2019-07-24 19:02:04	Brute Force honeypot	osDisk-confluencetemp1-20190724-111535
2019-07-25 01:58:36	Pastebin honeypot	osDisk-confluencetemp6-20190724-170610
2019-07-25 23:05:01	Pastebin honeypot	osDisk-confluencetemp6-20190725-194521
2019-07-26 12:25:06	Pastebin honeypot	osDisk-confluencetemp6-20190726-113406

Table 5.1: Timeline of the images used for the evaluation.

Honeypot	Date and Time	Image Name	Classification	Thorough Analysis
Brute Force honeypot	2019-06-26 07:35:34	osDisk-confluencetemp1-20190626-034704	PowerShell or Command Prompt Execution (2x)	PowerShell or Command Prompt Execution (2x)
Brute Force honeypot	2019-07-14 19:01:33	osDisk-confluencetemp1-20190714-143238	None	Cryptominer
Brute Force honeypot	2019-07-24 19:02:04	osDisk-confluencetemp1-20190724-111535	PowerShell or Command Prompt Execution	PowerShell or Command Prompt Execution
Pastebin honeypot	2019-07-26 12:25:06	osDisk-confluencetemp6-20190726-113406	PowerShell or Command Prompt Execution (3x)	PowerShell or Command Prompt Execution (3x) and External Pivoting Multiple Systems
Brute Force honeypot	2019-06-22 09:40:33	osDisk-confluencetemp1-20190622-085704	External Pivoting Multiple Systems	External Pivoting Multiple Systems
Brute Force honeypot	2019-06-29 15:01:41	osDisk-confluencetemp1-20190629-041201	Hiding Web Activity	Hiding Web Activity
Brute Force honeypot	2019-06-29 15:01:41	osDisk-confluencetemp1-20190629-041201	PowerShell or Command Prompt Execution (2x) and Cryptominer	PowerShell or Command Prompt Execution (2x) and Cryptominer
Brute Force honeypot	2019-06-25 18:35:01	osDisk-confluencetemp1-20190625-113600	Powershell or Command Prompt Execution	Powershell or Command Prompt Execution
Brute Force honeypot	2019-06-25 18:35:01	osDisk-confluencetemp1-20190625-113600	Powershell or Command Prompt Execution (2x) and External Pivoting Single System	Powershell or Command Prompt Execution (2x), External Pivoting Single System and Unknown Binary Execution

Table 5.2: Evaluation results.

Brute Force honeypot	2019-06-29 03:28:31	osDisk-confluencetemp1-20190628-155626	None	None
Brute Force honeypot	2019-07-15 04:35:36	osDisk-confluencetemp1-20190714-190330	None	None
Brute Force honeypot	2019-07-15 23:53:57	osDisk-confluencetemp1-20190715-113155	Deletion of Files and/or Folders	Deletion of Files and/or Folders
Pastebin honeypot	2019-07-12 14:15:08	osDisk-confluencetemp6-20190712-140509	Hiding Web Activity	Hiding Web Activity
Pastebin honeypot	2019-07-15 17:51:18	osDisk-confluencetemp6-20190715-173400	Hiding Web Activity	Hiding Web Activity
Brute Force honeypot	2019-06-27 18:35:01	osDisk-confluencetemp1-20190627-161038	PowerShell or Command Prompt Execution and External Pivoting Multiple Systems	PowerShell or Command Prompt Execution and External Pivoting Multiple Systems
Brute Force honeypot	2019-06-30 13:29:15	osDisk-confluencetemp1-20190630-125301	Hiding Web Activity	Hiding Web Activity
Pastebin honeypot	2019-07-14 15:38:48	osDisk-confluencetemp6-20190714-121244	Hiding Web Activity	Hiding Web Activity
Pastebin honeypot	2019-07-23 22:21:14	osDisk-confluencetemp6-20190723-203232	Hiding Web Activity	Hiding Web Activity
Brute Force honeypot	2019-06-25 20:45:08	osDisk-confluencetemp1-20190625-183650	Hiding Web Activity	Hiding Web Activity

Table 5.2 Continued: Evaluation results.

Brute Force honeypot	2019-07-16 13:35:06	osDisk-confluencetemp1-20190716-131759	PowerShell or Command Prompt Execution and Internal Pivoting to Multiple Systems	PowerShell or Command Prompt Execution and Internal Pivoting to Multiple Systems
Pastebin honeypot	2019-07-11 15:01:58	osDisk-confluencetemp6-20190711-130651	Hiding Web Activity	None
Pastebin honeypot	2019-07-18 09:40:36	osDisk-confluencetemp6-20190718-091249	None	None
Pastebin honeypot	2019-07-23 04:51:00	osDisk-confluencetemp6-20190723-044035	None	None
Brute Force honeypot	2019-06-28 12:05:01	osDisk-confluencetemp1-20190628-102113	PowerShell or Command Prompt Execution and External Pivoting Multiple Systems	PowerShell or Command Prompt Execution and External Pivoting Multiple Systems
Brute Force honeypot	2019-07-11 06:09:07	osDisk-confluencetemp1-20190711-042216	None	None
Pastebin honeypot	2019-07-12 13:43:46	osDisk-confluencetemp6-20190712-132216	PowerShell or Command Prompt Execution	PowerShell or Command Prompt Execution
Pastebin honeypot	2019-07-22 13:43:49	osDisk-confluencetemp6-20190722-132356	None	None
Pastebin honeypot	2019-07-25 01:58:36	osDisk-confluencetemp6-20190724-170610	None	None
Pastebin honeypot	2019-07-20 17:48:06	osDisk-confluencetemp6-20190720-063730	None	None

Table 5.2 Continued: Evaluation results.

Pastebin honeypot	2019-07-25 23:05:01	osDisk-confluencetemp6-20190725-194521	Hiding Web Activity	Hiding Web Activity
Brute Force honeypot	2019-07-09 05:22:46	osDisk-confluencetemp1-20190708-211449	None	None
Brute Force honeypot	2019-07-22 10:55:01	osDisk-confluencetemp1-20190722-080211	None	None
Brute Force honeypot	2019-07-03 07:35:02	osDisk-confluencetemp1-20190703-054200	PowerShell or Command Prompt Execution	PowerShell or Command Prompt Execution
Brute Force honeypot	2019-07-07 00:09:01	osDisk-confluencetemp1-20190706-205945	Hiding Web Activity	Hiding Web Activity and External Pivoting Multiple Systems
Brute Force honeypot	2019-07-17 07:14:48	osDisk-confluencetemp1-20190717-041135	None	None

Table 5.2 Continued: Evaluation results.

5.2 Cyber Attacks on the Honeypots

In total, six honeypots have been deployed. Five of those honeypots have been online from the 19th of June 2019 till the 26th of July 2019. The sixth honeypot was deployed on the 9th of July 2019 and was also taken offline on the 26th of July 2019. While the honeypots were deployed, 351 images have been collected, where an image can possibly contain multiple RDP sessions. Images have only been collected from the Brute Force honeypot and the Pastebin honeypot where credentials were leaked multiple times a day. The other honeypots did not receive any successful login attempts. Table 5.3 shows the distribution of the triggers of the backup process per honeypot.

Trigger	**Brute Force honeypot**	**Pastebin honeypot**	**Total**
RDP session between 90 seconds and 45 minutes	122	148	270
RDP session longer than 45 minutes	45	33	78
Bandwidth threshold exceeded	3	0	3
Total	**170**	**181**	**351**

Table 5.3: Trigger distribution of collected images.

Unfortunately, only two of the six honeypots have collected information about cyber attacks. While the credentials of the other honeypots were publicly accessible, they were not indexed by search engines such as Google and Bing. The only exception were the credentials on Github Gist, which were indexed by Google. The access to the VirusTotal leak could not be monitored because access to a private API is required.

From the moment the first five honeypots were deployed and the credentials were leaked, it took 19 minutes and 26 seconds before the first connection with the RDP port of a honeypot was made. The first authentication attempt occurred 2 hours 4 minutes and 53 seconds after the deployment. The first successful authentication event was 3 hours, 25 minutes and 1 second after the deployment. For the honeypot, which was deployed after the initial five, the first connection was made after 15 seconds, the first authentication attempt after 17 seconds and the first successful authentication attempt was after 7 minutes and 21 seconds. These numbers show how dangerous a credential leak can be. The previously mentioned information can be found per honeypot in Table 5.4.

Honeypot	**First Connection Attempt**	**First Authentication Attempt**	**First Successful Authentication Attempt**
Brute Force	2019-06-19 15:35:29	2019-06-19 16:54:50	2019-06-19 17:49:04
Google Spreadsheet	2019-06-19 16:28:53	2019-06-19 16:28:53	-
Github Gist	2019-06-19 16:44:36	2019-06-19 16:54:40	-
VirusTotal	2019-06-19 16:43:03	2019-06-19 16:43:04	-
Pastebin (single)	2019-06-19 14:43:29	2019-06-19 16:54:14	-
Pastebin	2019-07-09 16:35:38	2019-07-09 16:35:40	2019-07-10 07:42:44

Table 5.4: First connection attempt information per honeypot.

While the honeypots were online, a total of 3.224.406 connections were made to the RDP ports of the honeypots. These connections were made by 4791 unique IP addresses. Of those connections, 2.427.196 attempts have tried a username and password combination. In total, 833 authentication attempts were successful. The numbers per honeypot can be found in Table 5.5. These numbers show how dangerous it can be to directly allow RDP access from the internet. The difference between the number of connections and the number of authentication attempts shows how often it was checked whether the RDP port was accessible or not.

An IP address can be used to trace from which country a connection comes from. Attackers often use VPN services to hide their own IP address and location. Nonetheless, it is still interesting to see whether there is a trend between the honeypots where the connection attempts originate from. The top 10 countries where connection attempts originate from per honeypot can be found in Tables 5.6 till 5.11. Table 5.12 contains the previously mentioned information for all the honeypots together. The most connection attempts have been made from South-Korea for five of the six honeypots. Moreover, The United States is in the top three of each honeypot. Besides South-Korea and The United States, also Russia, Germany, The Netherlands and South Africa are in the top 10 of all the honeypots.

Honeypot	**Connection Attempts**	**Authentication Attempts**	**Successful Authentication Attempts**
Brute Force	223520	216628	500
Google Spreadsheet	723167	545647	0
Github Gist	789722	619390	0
VirusTotal	681597	501386	0
Pastebin (single)	634814	458128	0
Pastebin	171586	86017	333
Total	**3224406**	**2427196**	**833**

Table 5.5: Connection attempt information per honeypot.

When looking at the top 10 of most connection attempts per IP address, a flaw in the configured Azure firewall of all the honeypots was found. As can be seen in Tables 5.13 till 5.18, for five of the six honeypots, the IP address with the most connection attempts is another honeypot. In the firewall of the honeypots, all incoming connections from the local network were denied. The problem was that the rule allowing connections on the default RDP port has a higher priority than the previously mentioned rule. So, it was still possible to connect to the RDP port of another honeypot within the local network. From the Pastebin honeypot, a large volume of connection attempts have been made to the RDP port of the other honeypots. Only the Brute Force honeypot was not included. The strange thing about this attack is that it also targeted itself as can be seen in Table 5.18. The top 10 IP addresses which made the most connection attempts to all the honeypots can be found in Table 5.19. Multiple IP addresses mentioned in Table 5.19 made connections to more than one honeypot. Even though the tables describe connection attempts and not authentication attempts, it is fair to assume that the vast majority of the attempts are authentication attempts. As shown in Table 5.5, roughly 75 percent of all the connection attempts include an authentication attempt. The amount of connection attempts shown in the previously mentioned tables show how aggressive attackers can be in gaining access.

As already mentioned, only two of the six honeypots collected information about cyber attacks. The top 10 of most successful authentication attempts per IP address can be found for the Brute Force honeypot in Table 5.20 and for the Pastebin honeypot in Table 5.21. It is curious that multiple IP addresses have quite a large number of successful authentication attempts. The 59 successful authentication attempts of IP address *141.98.81.191* on the Brute Force honeypot span almost the whole period the Brute Force honeypot was online. There only was no successful authentication attempt during the first three days. All the attempts had a maximum session duration of three seconds. The 60 successful login attempts of IP address *36.73.137.67* on the Pastebin honeypot span only five days. In total, 37 sessions are longer than 90 seconds. That means that a single IP address triggered the backup and restore process 37 times. One of those sessions exceeded the 45 minute threshold, so the session of the attacker was terminated. This did not stop the attacker for returning for another 45 sessions.

The distribution of successful authentication attempts over time for both honeypots can be found in Figure 5.1 for the Brute Force honeypot and in Figure 5.2 for the Pastebin honeypot. The gap in the amount of successful authentication attempts, shown in Figure 5.1, can be explained by an attacker changing the password of the only user account in a session which was too short to trigger the backup and restore process. Because of the aforementioned situation, no other attacker was able to log into the honeypot. This explains the gap where no successful authentication attempts were performed. Both figures show that the number of successful authentication attempts can vary per day, no clear trend can be discovered.

Weak RDP credentials were used for the Brute Force honeypot. The top 10 most used usernames for unsuccessful login attempts can be found in Table 5.22. It is clear which username is used most often. It is almost used ten times more often compared to the second most used username. Windows usernames are case insensitive. Figure 5.23 contains the top 10 most used usernames for unsuccessful authentication attempts, where the usernames are all converted to lowercase. The Spanish translation *administrador* and the French translation *administrateur* for *administrator* have both been used often for authentication. This is interesting because the honeypot was hosted in The Netherlands.

Country	Connection Attempts
South Korea	52744
United States	26855
Russia	26270
Germany	17856
Netherlands	16145
South Africa	11681
China	7937
Ukraine	6077
Canada	3513
India	3261

Table 5.6: Top 10 most connection attempts made to the Brute Force honeypot per country.

Country	Connection Attempts
South Korea	85846
United States	59910
Finland	52737
Russia	48626
South Africa	33853
United Kingdom	30821
Netherlands	30474
France	28506
Germany	28404
Canada	21461

Table 5.7: Top 10 most connection attempts made to the Google Spreadsheet honeypot per country.

Country	Connection Attempts
South Korea	118857
United States	99851
Netherlands	79078
Russia	63595
Germany	50504
South Africa	37263
France	28922
Canada	15182
Brazil	15009
Thailand	13171

Table 5.8: Top 10 most connection attempts made to the Github Gist honeypot per country.

Country	Connection Attempts
South Korea	82819
Russia	68140
United States	57826
Netherlands	46771
Canada	43234
South Africa	33868
Germany	32425
France	29144
Brazil	13724
China	12298

Table 5.9: Top 10 most connection attempts made to the VirusTotal honeypot per country.

Country	**Connection Attempts**
Russia	83626
United States	66339
Netherlands	58822
South Korea	42159
South Africa	37156
Germany	22632
France	19281
Brazil	15327
Canada	12766
Thailand	11089

Table 5.10: Top 10 most connection attempts made to the Pastebin (single) honeypot per country.

Country	**Connection Attempts**
South Korea	18933
United States	18434
Netherlands	14967
Russia	9920
France	7719
China	5676
Germany	3838
Republic of Moldova	2283
Vietnam	2255
South Africa	2163

Table 5.11: Top 10 most connection attempts made to the Pastebin honeypot per country.

Country	**Connection Attempts**
South Korea	401358
United States	329215
Russia	300177
Netherlands	246257
South Africa	155984
Germany	155659
France	116686
Canada	98249
Finland	65678
Brazil	62652

Table 5.12: Top 10 most connection attempts made to all the honeypots per country.

IP Address	Country	Connection Attempts
118.45.230.251	South Korea	38543
41.216.186.99	South Africa	9169
121.144.195.164	South Korea	9146
185.156.177.99	Russia	2881
185.125.124.165	Poland	2420
78.31.71.81	Germany	2209
85.14.245.154	Germany	2204
85.14.245.156	Germany	2204
78.31.71.93	Germany	2163
80.82.77.139	Seychelles	2163

Table 5.13: Top 10 most connection attempts made to the Brute Force honeypot per IP address.

IP Address	Country	Connection Attempts
10.0.0.10		173815
95.217.58.110	Finland	49292
118.47.63.229	South Korea	39754
41.216.186.99	South Africa	31374
51.38.83.215	United Kingdom	27497
91.121.81.64	France	19140
138.122.71.235	Brazil	11619
118.37.130.5	South Korea	11238
119.196.66.197	South Korea	9621
1.179.154.108	Thailand	9103

Table 5.14: Top 10 most connection attempts made to the Google Spreadsheet honeypot per IP address.

IP Address	Country	Connection Attempts
10.0.0.10		164257
118.45.230.251	South Korea	91913
176.67.84.125	Netherlands	39979
41.216.186.99	South Africa	34574
52.171.212.92	United States	27998
195.154.150.58	France	18016
185.220.70.140	Germany	17328
96.76.194.78	United States	13378
121.144.195.164	South Korea	11910
138.122.71.235	Brazil	11619

Table 5.15: Top 10 most connection attempts made to the Github Gist honeypot per IP address.

IP Address	Country	Connection Attempts
10.0.0.10		170256
118.47.63.229	South Korea	54407
41.216.186.99	South Africa	31398
167.114.113.193	Canada	27498
91.121.81.64	France	19140
118.44.31.238	South Korea	18731
185.156.177.203	Russia	18311
212.92.120.198	Netherlands	16707
138.122.71.235	Brazil	11620
81.171.81.198	Netherlands	10000

Table 5.16: Top 10 most connection attempts made to the VirusTotal honeypot per IP address.

IP Address	Country	Connection Attempts
10.0.0.10		170847
41.216.186.99	South Africa	34576
185.107.45.91	Netherlands	20424
185.156.177.215	Russia	13023
118.45.230.251	South Korea	11909
121.144.195.164	South Korea	11909
138.122.71.235	Brazil	11619
176.67.84.158	Netherlands	10000
91.121.81.64	France	9570
96.125.135.204	Canada	9169

Table 5.17: Top 10 most connection attempts made to the Pastebin (single) honeypot per IP address.

IP Address	Country	Connection Attempts
10.0.0.10		57821
59.25.55.154	South Korea	14581
212.32.230.203	Netherlands	4635
163.172.7.162	France	4065
173.208.248.2	United States	3506
185.162.235.238	Netherlands	2533
212.92.116.86	Netherlands	2346
69.30.241.194	United States	2256
41.216.186.99	South Africa	1905
51.68.80.58	France	1872

Table 5.18: Top 10 most connection attempts made to the Pastebin honeypot per IP address.

IP Address	Country	Connection Attempts
10.0.0.10		736996
41.216.186.99	South Africa	142996
118.45.230.251	South Korea	142365
118.47.63.229	South Korea	94161
95.217.58.110	Finland	49292
91.121.81.64	France	47850
138.122.71.235	Brazil	46581
176.67.84.125	Netherlands	39979
96.125.135.204	Canada	36832
185.156.177.203	Russia	33018

Table 5.19: Top 10 most connection attempts made to all the honeypot per IP address.

IP Address	Country	Successful Connection Attempts
141.98.81.191	Panama	59
129.144.27.16	United States	43
139.28.218.220	Romania	23
157.230.94.198	United States	17
89.238.154.166	United Kingdom	15
89.238.178.214	Spain	13
93.116.235.14	Republic of Moldova	10
220.188.176.165	China	8
172.94.15.22	Germany	6
185.148.73.93	Slovenia	6

Table 5.20: Top 10 most successful connection attempts made to the Brute Force honeypot per IP address.

IP Address	Country	Successful Connection Attempts
36.73.137.67	Indonesia	60
106.206.41.230	India	29
129.205.112.203	Nigeria	13
223.186.73.40	India	12
180.246.83.144	Indonesia	11
223.186.137.151	India	10
223.186.122.11	India	9
223.186.41.164	India	7
223.186.67.176	India	7
185.245.84.100	Denmark	6

Table 5.21: Top 10 most successful connection attempts made to the Pastebin honeypot per IP address.

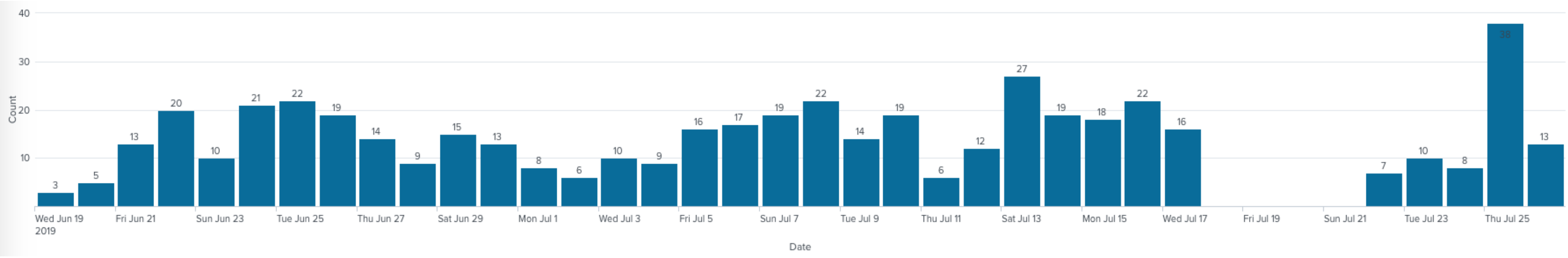


Figure 5.1: Bar chart showing the successful login count per day of the Brute Force honeypot.

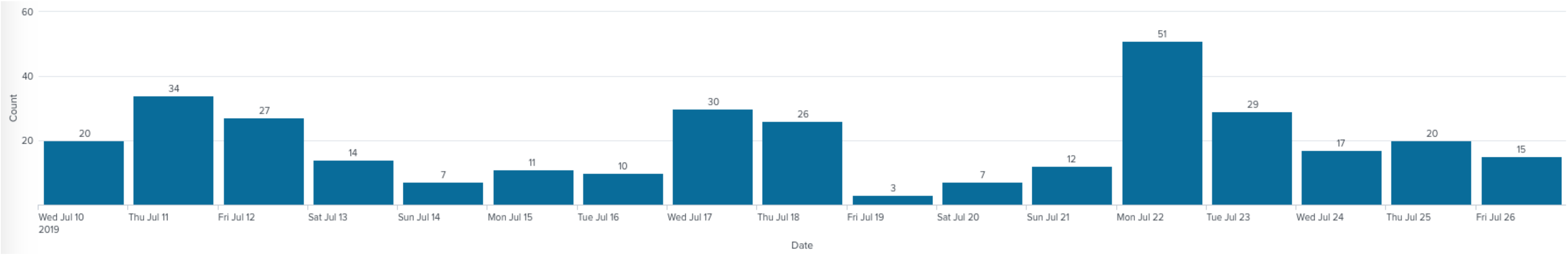


Figure 5.2: Bar chart showing the successful login count per day of the Pastebin honeypot.

Username	Count
ADMINISTRATOR	1092169
ADMIN	150351
USER	57452
administrator	26834
Administrator	22380
SERVER	21102
TEST	20806
BACKUP	14362
ADMINISTRADOR	12752
ADM	11864

Table 5.22: Top 10 most used usernames for unsuccessful connection attempts made to the Brute Force honeypot.

Username	Count
administrator	1141384
admin	160577
user	59976
test	25169
server	21386
administrador	18286
backup	15149
adm	12038
administrateur	11895
support	11113

Table 5.23: Top 10 most used usernames for unsuccessful connection attempts made to the Brute Force honeypot.

As mentioned before, 35 RDP sessions have been analyzed during the evaluation. Of those sessions, 22 come from the Brute Force honeypot and 13 come from the Pastebin honeypot. Based on the results of the evaluation, which can be found in Table 5.2, the distribution of classified attacks can be found of the Brute Force honeypot in Figure 5.3 and of the Pastebin honeypot in Figure 5.4. A single RDP session can consist of multiple attacks. Due to the limited amount of sessions which have been classified, it would not be appropriate to label any of the attacks as the most common one for the way of acquiring credentials. To be able to do this, more RDP sessions should be analyzed.

The attack which was classified most often in the sessions of the Brute Force honeypot, is the *PowerShell or Command Prompt Execution* classification. The evaluation showed that PowerShell or the Command Prompt is regularly used to change the password of the current user or to start the execution of an executable created by the attacker. The usage of scripts reduces the amount of manual labor, instead of clicking through multiple user interfaces to change the password of a user account, it can be changed by executing a single script file once.

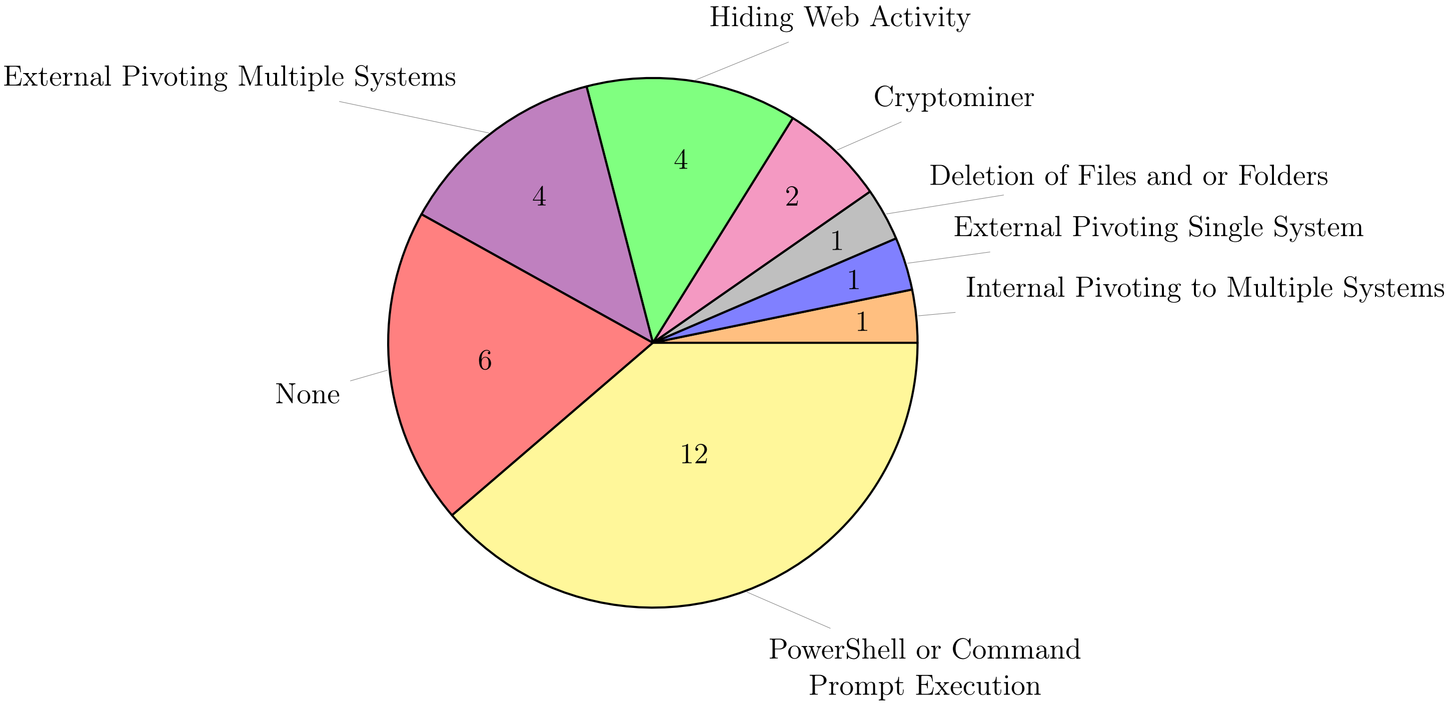


Figure 5.3: Pie chart of the classified attacks of the Brute Force honeypot.

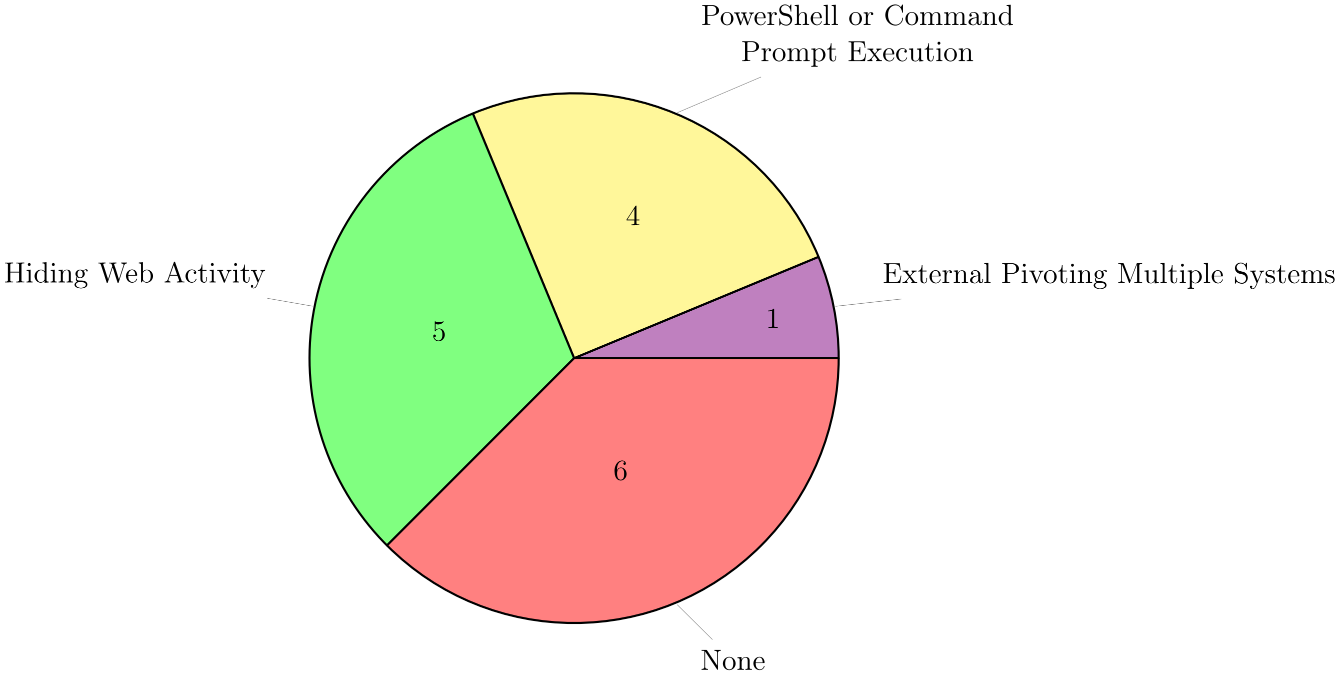


Figure 5.4: Pie chart of the classified attacks of the Pastebin honeypot.

Even though the Brute Force honeypot did not have a large amount of resources in terms of hardware, cryptominers have still been installed in two separate sessions. Systems with more resources are better suited for mining crypto currencies. During one of the RDP sessions, the attacker deleted everything which was placed on the Desktop and nothing else. This is strange behavior because the attacker does not gain anything from deleting the files, folders and shortcuts. The desktop also did not contain any sensitive files. In 6 of the 22 RDP sessions, the attacker did not perform any malicious activity. This is almost a third of all the analyzed RDP sessions of the Brute Force honeypot. The activities in these sessions varied from opening folders and checking the systems specifications to doing nothing at all.

Scanning software has been used in 6 of the analyzed RDP sessions of the Brute Force honeypot. The targets varied from systems inside the local network to external IP address. The vast majority of the scans targeted a specific kind of protocol. Only one of the sessions targeted multiple ports of a single IP address. The browser of the system has been used to access Paypal, checking if the website would automatically log in. Upon discovery that this was not the case, the attacker logged out. In one of the sessions, the browser was also used to access the Airbnb website. The attacker logged into an account and performed no other activity while logged in. A possible reason for this behavior could be that the attacker validated the credentials of an Airbnb account. In the remaining sessions, the browser was used to gain information about the system. For example, a website was accessed which provides information related to the IP address of the system, such as the country where the system is located.

For the Pastebin honeypot, the variety of attacks performed is lower compared to the Brute Force honeypot. A lower number of sessions have been analyzed for the Pastebin honeypot, so that could explain the lower variety. Again, the browser was used to check websites of payment providers such as Paypal and Cash App. In one of the sessions, the browser was also used to access a website containing pornographic content. Before going to that website, a Google search was performed with an email address, followed by a colon and then something which looked like a password. On the pornographic website, the login page was accessed. Possibly, the aforementioned email address and password combination has been used to login to the website. After logging in, no other activity has been performed. Just as with the Brute Force honeypot, there were RDP sessions where the attackers did not perform any malicious activity. In 6 of the 13 sessions, no malicious activity was performed. PowerShell and the Command Prompt have been used to perform the same kind of actions as with the Brute Force honeypot.

Comparing the results of the different honeypots shows that all the different kinds of attacks which have been performed on the Pastebin honeypot, have also been performed on the Brute Force honeypot. On the Brute Force honeypot a wider variety of attacks have been performed, but this can be due to the larger number of RDP sessions which have been analyzed. The percentage of attacks falling under *Pivoting* is quite different between the honeypots, it is higher for the Brute Force honeypot. Moreover, *Hiding Web Activity* has a far greater share on the Pastebin honeypot compared to the Brute Force honeypot. The shares of *PowerShell or Command Prompt Execution* and *None* are roughly the same for both honeypots. The previously mentioned observations are based on the results of the evaluation. To actually be able to compare the attack distributions of the different honeypots in detail, more RDP sessions have to be analyzed.

Chapter 6

Discussion

The collection of information to answer the research questions, was done by deploying multiple honeypots. The setup of the honeypot can possibly influence the activities performed by an attacker. The decision was made to use a story for the honeypot, stating that is was only a temporary server, hence the spreading of credentials and the low amount of resources. The low amount of resources could potentially discourage attackers to perform activities which require a large amount of resources, which normally would have been present on a production system. The temporary server story was visible in some of the leaked messages and in the Confluence installation. For the Pastebin honeypot, the story was not present in the leaked message. For the Brute Force honeypot, there was no information leaked at all. The evaluation showed that none of the attackers have accessed the Confluence pages, so the story probably has not been clear to the attackers of the honeypots. Without the context of the story, the server might have been suspicious to the attacker.

The backup process of the honeypot architecture has had an influence on the data collection. The backup process of the honeypot always started directly after an attacker logged out. This prevented attackers from reconnecting to the system. The evaluation showed that in multiple situations everything was setup to perform malicious activity, but it was not started yet. An attacker never had the chance to start the malicious activity in a second session. This influenced the collection of information, cutting the activities of an attacker short. By making the backup strategy less strict, more useful information could be collected.

A threshold value was used to terminate the session of an attacker to prevent outgoing malicious behavior. Choosing the threshold value was making a trade-off between information collection and ethical considerations. From an information collection point of view, it was preferable to not use such a threshold at all. From an ethical point of view, harm to other systems facilitated by the honeypot should be prevented as much as possible. So, from the ethical point of view the threshold should be as low as possible. The evaluation has showed that attacker sessions have been cut short but also that outgoing malicious behavior perhaps could have been prevented even more. The balance which was chosen between these two conflicting interests has had influence on the gathered data. The influence of the honeypot architecture on the activities of an attacker and the collection of information, both potentially influence answering the research question. The impact on the answer of the research question can only be verified by future research by performing the experiment again with a different honeypot and a different backup strategy.

The same honeypot was used for the different ways of acquiring credentials. While this is required to be able to compare the results of the different ways of acquiring credentials, it is less optimal for the evaluation of the classification. The proposed classification has been evaluated with data from a single source. It would be better to use different sources of information for the evaluation, not only testing the classification with scenarios where the classification has been created for.

The evaluation showed that the proposed classification has a few issues. The alteration of a password of a user account and the creation of a new user account is not included in the proposed classification. This could be solved by adding a node called *Account Related Activity* which splits into *Account Creation* and *Account Alteration* leaf nodes. The *Hosting* leaf node can never be reached when Windows components are used to install necessities for hosting malicious content. The constraints of a binary being executed should be made optional. The *Hiding Web Activity* classification includes a broad variety of activities. This leaf node should be split into multiple leaf nodes representing different kinds of activity. The *Large volume of network traffic* constraint on the path to the *Pivoting* node should be more explicitly defined. Besides collecting information from the honeypot, collection of additional sources could improve the performance of the proposed classification. For example, port scans performed by only sending SYN packets could be classified when firewall logs are also analyzed during the forensic analysis.

Besides structuring the forensic analysis, the proposed classification also aims to improve the efficiency of forensic analysis on the Windows operating system. During the evaluation, RDP sessions have been analyzed with the proposed classification and a thorough analysis was performed. Estimating based on the evaluation, analysis with the proposed classification takes roughly 1 hour and the thorough analysis takes around 1 hour and 20 minutes. These estimations are for an RDP sessions where an attacker performed a moderate amount of activity. Even though using the proposed classification reduced the analysis time, it is based on the evaluation of a single source of information. Moreover, in a real forensic analysis more than only a single RDP session is analyzed. Whether the classification improves the efficiency of forensic analysis, should be tested by using it in a real forensic analysis.

From the six different honeypots, only two collected information about cyber attacks. The Github Gist credentials were indexed by a search engine but the credentials on the other platforms were not indexed. The access to the VirusTotal leak could not be monitored because access to a private API is required. Credentials not being indexed by search engines could be an explanation for why there have not been any successful login attempts. The reason for the VirusTotal and Github Gist leaks not being used, could be that attackers are not actively searching for credentials on these platforms. However, this cannot be concluded based on leaking credentials once.

Of the collected images, a total of 35 RDP sessions have been analyzed due to time constraints. This limited number of results makes it difficult to answer the research question. It will be answered based on the evaluated data, but the applicability in general can only be revealed by future research. Moreover, comparing only two different ways of acquiring credentials limits the conclusions which can be made regarding the research question.

Chapter 7

Future Work

The created honeypot has shown to be effective in the collection of information about cyber attacks. Nonetheless, as mentioned in the Section 4.4 and in Section 5.1, improvements can be made to the architecture to improve the collection of images. Changing when the backup and restore process starts would already have a major impact on the collection of images. By postponing the process and monitoring for the same attackers logging in, more information can be gathered about the activities of the attacker. Moreover, keeping the honeypot up to date reduces the amount of unnecessary additional digital evidence. Furthermore, monitoring password changes can improve the continuity of the honeypot. It can also be expanded to gather information about other kinds of activity. For example, by adding a local network consisting of multiple other systems to the honeypot, information can also be gathered about lateral movement. Another possibility is using a different valuable asset on the honeypot. This might influence the behavior of attackers. Credentials were leaked to lure attackers to the honeypots. Besides the platforms used in this research, there are other platforms which potentially are interesting to gather information from.

During the evaluation of the collected information, it was not possible to evaluate all the different paths in the proposed classification. Moreover, some paths have been evaluated only once. Future research could evaluate these paths to check if they are correct. Besides evaluating existing attacks, new attacks could be added to the proposed classification. The created honeypot setup only allowed a subset of possible cyber attacks to be performed on the system. By using different sources of information, new attacks could potentially be added to the classification. The method used for the creation and evaluation of the proposed classification could be used to create classifications for other domains. The methodology which was used is not tied to a specific domain.

Analyzing more images could provide better insights into what kind of cyber attacks are performed. Based on intuition, around 50 RDP sessions should be analyzed per honeypot to be able to draw conclusions regarding the research question. Analyzing more images would also allow to compare the different ways of acquiring credentials in more detail. Furthermore, by spreading credentials on other platforms, a broader understanding of cyber attacks originating from leaked credentials can be acquired.

Chapter 8

Conclusion

This research focused on what kind of attacks are performed via RDP by attackers which acquire credentials in different ways. The goal was to determine whether it would be possible to make a distinction between ways of acquiring RDP credentials, based on the most common cyber attacks. To be able to answer this question, information needed to be collected about cyber attacks. This has been done by creating a honeypot and generating attention by leaking credentials and using a weak username and password. A fake company was created to increase the credibility of the honeypots and the leaked credential messages. Within five and a half weeks, 351 images have been collected. An image includes the complete hard drive of a honeypot. The next step was to determine which cyber attacks have been performed on these images. To extract this information from an acquired image, a classification has been proposed, based on the presence of digital artifacts. Besides classifying cyber attacks, the classification also provides a prioritization model for analyzing digital artifacts, aiding forensic analysis. The evaluation of the images was two-fold, providing feedback on the proposed classification and extract what kind of cyber attacks have been performed.

While the honeypots were online, only two of the six different honeypots collected information about cyber attacks. The other four honeypots did not receive any successful login attempts. Nonetheless, all honeypots have contributed to providing insights about connection attempts to open RDP ports. A total of 3.224.406 connections have been made to the RDP ports of the honeypots. This shows how dangerous it can be to directly allow RDP access from the internet. Moreover, when RDP is accessible from the internet and credentials are leaked, it does not take long before the credentials will be used. The Pastebin honeypot showed that it took 7 minutes and 21 seconds before the first successful authentication attempt was performed with the leaked credentials. This shows how dangerous a credential leak can be.

Because of time constraints, only 35 different RDP sessions have been analyzed. Due the limited number of analyzed images and the limited number of honeypots which collected information about cyber attacks, it is not possible to fully answer the research question, only an approximation can be made. The most common cyber attack performed on the Brute Force honeypot is *PowerShell or Command Prompt Execution* and the most common cyber attack performed on the Pastebin honeypot is *Hiding Web Activity*. As described in Section 5.2, there is no considerable difference between the attacks performed on both honeypots. All the attacks which have been performed on the Pastebin honeypot were also performed on the Brute Force honeypot. A wider variety of attacks was performed on the Brute Force honeypot compared to the Pastebin honeypot. A possible explanation is that more images of the Brute Force honeypot have been analyzed. Furthermore, the percentage of attacks falling under *Pivoting* is higher for the Brute Force honeypot compared to the Pastebin honeypot. As for *Hiding Web Activity*, the percentage is lower for the Brute Force honeypot compared to the Pastebin honeypot. There are differences in the occurrence of the remaining attack classifications, but these differences are small. Based on the limited analyzed results, no distinction can be made between how attackers acquired credentials, based on the most common cyber attacks. Future research could potentially provide a conclusive answer to the research question, by analyzing a greater number of images, originating from a larger variety of ways to acquire credentials.

Bibliography

[1] Federal Bureau of Investigation, "Cyber Actors Increasingly Exploit the Remote Desktop Protocol to Conduct Malicious Activity." `https://www.ic3.gov/media/2018/180927.aspx`, 2018. Accessed: 2019-09-04.

[2] D. Shackleford, "SANS 2019 Cloud Security Survey." `https://www.sans.org/reading-room/whitepapers/cloud/paper/38940`, 2019. Accessed: 2019-09-24.

[3] C. Stoll, *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage.* Doubleday, 1989.

[4] L. Spitzner, *Honeypots: Tracking Hackers.* Addison-Wesley, 2002.

[5] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A Survey on Honeypot Software and Data Analysis," *arXiv:1608.06249*, 2016.

[6] Software Engineering Institute - Carnegie Mellon University, "2002 CERT Advisories - CA-2002-01: Exploitation of Vulnerability in CDE Subprocess Control Service." `https://www.cert.org/historical/advisories/CA-2002-01.cfm`, 2002. Accessed: 2019-09-03.

[7] "The Honeynet Project." `https://www.honeynet.org/`. Accessed: 2019-09-03.

[8] M. Tsikerdekis, S. Zeadally, A. Schlesener, and N. Sklavos, "Approaches for Preventing Honeypot Detection and Compromise," in *Global Information Infrastructure and Networking Symposium (GIIS)*, 2018.

[9] M. Winn, M. Rice, S. Dunlap, J. Lopez, and B. Mullins, "Constructing cost-effective and targetable industrial control system honeypots for production networks," *International Journal of Critical Infrastructure Protection (IJCIP)*, vol. 10, pp. 47–58, 2015.

[10] L. Spitzner, "Knowy Your Enemy: Honeynets." `http://old.honeynet.org/papers/honeynet/`, 2005. Accessed: 2019-09-03.

[11] L. Spitzner, “Honeypot Farms.” https://www.symantec.com/connect/articles/honeypot-farms, 2003. Accessed: 2019-09-03.

[12] W. Z. A. Zakaria and M. L. M. Kiah, “A review of dynamic and intelligent honeypots,” *ScienceAsia*, vol. 39, pp. 1–5, 2013.

[13] W. Fan, Z. Du, D. Fernandez, and V. A. Villagra, “Enabling an Anatomic View to Investigate Honeypot Systems: A Survey,” *IEEE Systems Journal*, vol. 12, no. 4, pp. 3906–3919, 2018.

[14] N. Provos, “Developments of the Honeyd Virtual Honeypot.” http://www.honeyd.org/. Accessed: 2019-04-15.

[15] MushMush Foundation, “Glastopf.” https://github.com/mushorg/glastopf. Accessed: 2019-04-15.

[16] DinTools, “Dionaea.” https://github.com/DinoTools/dionaea. Accessed: 2019-04-15.

[17] U. Tamminen, “Kippo.” https://github.com/desaster/kippo. Accessed: 2019-04-15.

[18] Cowrie Project, “Cowrie.” https://github.com/cowrie/cowrie. Accessed: 2019-04-15.

[19] A. Agnaou, A. A. El Kalam, and A. A. Ouahman, “Towards a collaborative architecture of Honeypots,” *ACS/IEEE 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1299–1305, 2017.

[20] “Argos - An emulator for capturing zero-day attacks.” https://www.few.vu.nl/argos/. Accessed: 2019-04-15.

[21] P. Fanfara, M. Dufala, and J. Radušovský, “Autonomous Hybrid Honeypot as the Future of Distributed Computer Systems Security,” *Acta Polytechnica Hungarica*, vol. 10, no. 6, pp. 25–42, 2013.

[22] J. Guarnizo, A. Tambe, S. S. Bhunia, M. Ochoa, N. O. Tippenhauer, A. Shabtai, and Y. Elovici, “SIPHON: Towards Scalable High-Interaction Physical Honeypots,” *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security (CPSS)*, pp. 57–68, 2017.

[23] Microsoft, “Remote Desktop Protocol.” https://docs.microsoft.com/en-us/windows/win32/termserv/remote-desktop-protocol. Accessed: 2019-09-04.

[24] LastPass, “The Password Exposé.” https://www.lastpass.com/nl/business/articles/password-expose-report, 2017. Accessed: 2019-09-09.

[25] E. Stobert and R. Biddle, "The Password Life Cycle: User Behaviour in Managing Passwords," *10th Symposium On Usable Privacy and Security (SOUPS)*, pp. 243–255, 2014.

[26] NIST, "Special Publication 800-63B - Digital Identity Guidelines." `https://pages.nist.gov/800-63-3/sp800-63b.html`, 2017. Accessed: 2019-09-09.

[27] K. Leswing, "A password for the Hawaii emergency agency was hiding in a public photo, written on a Post-it note." `https://www.businessinsider.nl/hawaii-emergency-agency-password-discovered-in-photo-sparks/-security-criticism-2018-1?international=true&r=US`, 2018. Accessed: 2019-09-09.

[28] S. Chatterjee, X. Gao, S. Sarkar, and C. Uzmanoglu, "Reacting to the scope of a data breach: The differential role of fear and anger," *Journal of Business Research (JBR)*, vol. 101, pp. 183–193, 2019.

[29] Encyclopedia Britannica, "Modus operandi - Criminology." `https://www.britannica.com/topic/modus-operandi`. Accessed: 2019-09-05.

[30] NIST, "Special Publication 800-61 Rev. 2 - Computer Security Incident Handling Guide." `https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final`, 2012. Accessed: 2019-10-01.

[31] M. Olivier, "On a Scientific Theory of Digital Forensics," in *12th IFIP International Conference on Digital Forensics (DF)*, pp. 3–24, 2016.

[32] Y. Yusoff, R. Ismail, and Z. Hassan, "Common Phases of Computer Forensics Investigation Models," *International Journal of Computer Science and Information Technology (IJCSIT)*, vol. 3, no. 3, pp. 17–31, 2011.

[33] E. A. Vincze, "Challenges in digital forensics," *s*, vol. 17, no. 2, pp. 183–194, 2016.

[34] X. Li, X. Liang, R. Lu, X. Shen, X. Lin, and H. Zhu, "Securing smart grid: Cyber attacks, countermeasures, and challenges," *e*, vol. 50, no. 8, pp. 38–45, 2012.

[35] D. Papp, Z. Ma, and L. Buttyan, "Embedded Systems Security: Threats, Vulnerabilities, and Attack Taxonomy," in *13th Annual Conference on Privacy, Security and Trust (PST)*, pp. 145–152, 2015.

[36] M. Yampolskiy, P. Horvath, X. D. Koutsoukos, Y. Xue, and J. Sztipanovits, "Taxonomy for Description of Cross-Domain Attacks on CPS Mark," in *Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems*, pp. 135–142, 2013.

[37] B. Zhu, A. Joseph, and S. Sastry, "A Taxonomy of Cyber Attacks on SCADA Systems," in *International Conference on Internet of Things (iThings) and 4th International Conference on Cyber, Physical and Social Computing (CPSCom)*, pp. 380–388, 2011.

[38] R. Koch, M. Golling, and G. D. Rodosek, "A Revised Attack Taxonomy for a New Generation of Smart Attacks," *Computer and Information Science*, vol. 7, no. 3, pp. 18–30, 2014.

[39] K. Ahmed, S. Verma, N. Kumar, and J. Shekbar, "Classification Of Internet Security Attacks," *Proceedings of the 5th National Conference on Computing for Nation Development (INDIACom)*, 2011.

[40] C. Harry and N. Gallagher, "Classifying Cyber Events: A Proposed Taxonomy," *Journal of Information Warfare (JIW)*, vol. 17, no. 3, 2018.

[41] H. Šemić and S. Mrdovic, "IoT honeypot: A multi-component solution for handling manual and Mirai-based attacks," in *25th Telecommunications Forum (TELFOR)*, IEEE, nov 2017.

[42] U. D. Gandhi, P. M. Kumar, R. Varatharajan, G. Manogaran, R. Sundarasekar, and S. Kadu, "HIoTPOT: Surveillance on IoT Devices against Recent Threats," *Wireless Personal Communications*, vol. 103, no. 2, pp. 1179–1194, 2018.

[43] S. Dowling, M. Schukat, and H. Melvin, "A ZigBee honeypot to assess IoT cyberattack behaviour," in *28th Irish Signals and Systems Conference (ISSC)*, IEEE, 2017.

[44] Ó. Navarro, S. A. J. Balbastre, and S. Beyer, "Gathering Intelligence Through Realistic Industrial Control System Honeypots," in *13th International Conference on Critical Information Infrastructures Security (CRITIS)*, pp. 143–153, 2018.

[45] J. Lee, J. Jeon, C. Lee, J. Lee, and J. Cho, "An implementation of log visualization system combined SCADA Honeypot," *International Conference on Advanced Communication Technology (ICACT)*, 2016.

[46] J. Nazario, "PhoneyC: A Virtual Client Honeypot," *Proceedings of the 2nd USENIX Conference on Large-scale Exploits and Emergent Threats (LEET)*, 2009.

[47] D. Canali and D. Balzarotti, "Behind the Scenes of Online Attacks: an Analysis of Exploitation Behaviors on the Web," in *Proceedings of the 20th Annual Network & Distributed System Security Symposium (NDSS)*, pp. 44–62, 2013.

[48] M. Wählisch, A. Vorbach, C. Keil, J. Schönfelder, T. C. Schmidt, and J. H. Schiller, "Design, Implementation, and Operation of a Mobile Honeypot," *arXiv:1301.7257*, 2013.

[49] C. Irvene, D. Formby, S. Litchfield, and R. Beyah, "HoneyBot: A Honeypot for Robotic Systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 61–70, 2018.

[50] T. Barron and N. Nikiforakis, "Picky Attackers: Quantifying the Role of System Properties on Intruder Behavior," in *Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC)*, pp. 387–398, 2017.

[51] C. Pohl, A. Zugenmaier, M. Meier, and H.-J. Hof, "B.Hive: A Zero Configuration Forms Honeypot for Productive Web Applications," in *IFIP International Information Security and Privacy Conference*, pp. 267–280, 2015.

[52] H. Wang and Q. Chen, "Dynamic Deploying Distributed Low-interaction Honeynet," *Journal of Computers (JOC)*, vol. 7, no. 3, pp. 692–698, 2012.

[53] X. Jiang and D. Xu, "BAIT-TRAP: a Catering Honeypot Framework," Department of Computer Sciences Purdue University, 2004.

[54] G. Wagener, R. State, T. Engel, and A. Dulaunoy, "Adaptive and Self-Configurable Honeypots," in *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 345–352, IEEE, may 2011.

[55] J. Onaolapo, E. Mariconti, and G. Stringhini, "What Happens After You Are Pwnd: Understanding The Use Of Leaked Account Credentials In The Wild," *Proceedings of the ACM SIGCOMM Conference on Internet Measurement Conference (IMC)*, pp. 65–79, 2016.

[56] D. A. Bermudez Villalva, J. Onaolapo, G. Stringhini, and M. Musolesi, "Under and over the surface: a comparison of the use of leaked account credentials in the Dark and Surface Web," *Crime Science*, vol. 7, 2018.

[57] D. Fraunholz, D. Schneider, J. Zemitis, and H. D. Schotten, "Hack My Company: An Empirical Assessment of Post-exploitation Behavior and Lateral Movement in Cloud Environments," in *Proceedings of the Central European Cybersecurity Conference (CECC)*, 2018.

[58] M. Lazarov, J. Onaolapo, and G. Stringhini, "Honey Sheets: What Happens to Leaked Google Spreadsheets?," *USENIX Workshop on Cyber Security Experimentation and Test (CSET)*, vol. 9, 2016.

[59] M. Akiyama, T. Yagi, T. Hariu, and Y. Kadobayashi, "HoneyCirculator: distributing credential honeytoken for introspection of web-based attack cycle," *International Journal of Information Security*, vol. 17, pp. 135–151, 2018.

[60] A. Valjarevic and H. S. Venter, "A Comprehensive and Harmonized Digital Forensic Investigation Process Model," *Journal of Forensic Sciences (JFS)*, vol. 60, no. 6, pp. 1467–1483, 2015.

[61] B. Yang, N. Li, and J. Jiang, "A New Triage Process Model for Digital Investigations," in *Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 712–717, 2016.

[62] A. Agarwal, M. Gupta, S. Gupta, and S. C. Gupta, "Systematic Digital Forensic Investigation Model," *International Journal of Computer Science and Security (IJCSS)*, vol. 5, no. 1, pp. 118–131, 2011.

[63] I. O. Ademu, C. O. Imafidon, and D. S. Preston, "A New Approach of Digital Forensic Model for Digital Forensic Investigation," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 2, no. 12, pp. 175–178, 2011.

[64] A. Tanner and D. Dampier, "Concept Mapping for Digital Forensic Investigations," in *Advances in Digital Forensics V*, vol. 306, pp. 291–300, 2009.

[65] X. Du, N.-A. Le-Khac, and M. Scanlon, "Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service," in *Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS)*, pp. 573–581, 2017.

[66] R. Schlegel, A. Hristova, and S. Obermeier, "A Framework for Incident Response in Industrial Control Systems," in *12th International Joint Conference on e-Business and Telecommunications (ICETE)*, pp. 178–185, 2015.

[67] O. Brady, R. Overill, and J. Keppens, "Addressing the Increasing Volume and Variety of Digital Evidence Using an Ontology," in *IEEE Joint Intelligence and Security Informatics Conference (JISIC)*, pp. 176–183, 2014.

[68] K.-S. Lim, S. Lee, and S. Lee, "Applying a Stepwise Forensic Approach to Incident Response and Computer Usage Analysis," in *Proceedings of thep 2nd International Conference on Computer Science and Its Applications (CSA)*, 2009.

[69] N. L. Beebe, J. G. Clark, G. B. Dietrich, M. S. Ko, and D. Ko, "Post-retrieval search hit clustering to improve information retrieval effectiveness: Two digital forensics case studies," *Decision Support Systems (DSS)*, vol. 51, pp. 732–744, nov 2011.

[70] C. Hargreaves and J. Patterson, "An automated timeline reconstruction approach for digital forensic investigations," in *Proceedings of the Digital Forensic Research Conference (DFRWS)*, pp. 69–79, Digital Forensic Research Workshop, 2012.

[71] M. Kjaerland, "A taxonomy and comparison of computer security incidents from the commercial and government sectors," *a*, vol. 25, pp. 522–538, oct 2006.

[72] M. Gadelrab, A. A. El Kalam, and Y. Deswarte, "Execution Patterns in Automatic Malware and Human-Centric Attacks," in *Proceedings of the 7th IEEE International Symposium on Networking Computing and Applications (NCA)*, pp. 29–36, 2008.

[73] C. B. Simmons, S. G. Shiva, H. Bedi, and D. Dasgupta, "AVOIDIT: A Cyber Attack Taxonomy," in *9th Annual Symposium on Information Assurance (ASIA)*, pp. 2–12, 2014.

[74] C. Joshi and U. Kumar Singh, "ADMIT- A Five Dimensional Approach towards Standardization of Network and Computer Attack Taxonomies," *International Journal of Computer Applications (IJCA)*, vol. 100, no. 5, pp. 30–36, 2014.

[75] Atlassian, "Confluence." `https://www.atlassian.com/nl/software/confluence`. Accessed: 2019-09-17.

[76] Microsoft, "Azure." `https://azure.microsoft.com/`. Accessed: 2019-09-17.

[77] Microsoft, "Sysmon v10.41." `https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon`. Accessed: 2019-09-17.

[78] SwiftOnSecurity, "Sysmon configuration file template with default high-quality event tracing." `https://github.com/SwiftOnSecurity/sysmon-config`. Accessed: 2019-09-17.

[79] "Splunk." `https://www.splunk.com/`. Accessed: 2019-09-17.

[80] S. Khandelwal, "How Top Companies Accidentally Leaking Terabytes of Sensitive Data Online." `https://thehackernews.com/2017/08/fortune-1000-data-leak.html`. Accessed: 2019-09-18.

[81] J. Pande, R. Goswami, S. Sharma, and C. S. Chawla, "Cyber Attacks and Counter Measures: User Perspective," 2016.

[82] J. P. Tailor and A. D. Patel, "A Comprehensive Survey: Ransomware Attacks Prevention, Monitoring and Damage Control," *International Journal of Research and Scientific Innovation (IJRSI)*, vol. 4, no. VIS, pp. 116–121, 2017.

[83] J. A. Herrera Silva, L. I. Barona López, Á. L. Valdivieso Caraguay, and M. Hernández-Álvarez, "A Survey on Situational Awareness of Ransomware Attacks - Detection and Prevention Parameters," *Remote Sensing*, vol. 11, no. 10, 2019.

[84] A. Zimba and M. Chishimba, "Understanding the Evolution of Ransomware: Paradigm Shifts in Attack Structures," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 11, no. 1, pp. 26–39, 2019.

[85] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Botnet in DDoS Attacks: Trends and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2242–2270, 2015.

[86] D. Draghicescu, A. Caranica, A. Vulpe, and O. Fratu, "Crypto-Mining Application Fingerprinting Method," in *International Conference on Communications (COMM)*, pp. 543–546, 2018.

[87] S. Yilmaz and S. Zavrak, "Adware: A Review," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 6, no. 6, pp. 5599–5604, 2015.

[88] SANS Institute, "FOR500: Windows Forensic Analysis." `https://www.sans.org/course/windows-forensic-analysis`. Accessed: 2019-09-23.

[89] SANS Institute, "Windows Forensic Analysis." `https://www.sans.org/security-resources/posters/dfir/windows-forensic-analysis-170`. Accessed: 2019-09-23.

[90] Microsoft, "Microsoft Docs." `https://docs.microsoft.com/en-us/`. Accessed: 2019-09-23.

[91] Magnet Forensics, "Magnet Axiom." `https://www.magnetforensics.com/products/magnet-axiom/`. Accessed: 2019-09-23.

[92] Thinkst Applied Research, "Canarytokens by Thinkst." `http://canarytokens.org/`. Accessed: 2019-10-02.

Appendix A

Honeypot Development Lessons Learned

During the development of the honeypot architecture, the following lessons were learned:

- Think critically about what kind of data needs to be collected with the honeypot. This can have an influence on the setup of the honeypot. For example, collecting data about a specific attack might be more successful when a specific kind of operating system is used. Moreover, it is easy to gather a large amount of data with a honeypot, but it can be difficult to do something useful with that large amount of data.

- Take the time to search through existing literature about honeypots. They have been used often by previous research. Use the existing information to prevent making beginner mistakes and to improve the quality of the honeypot.

- When it is clear what kind of information needs to be collected, pick the kind of honeypots which suits the purpose best. Using a high-interaction honeypot will provide a large amount of interesting data, but using it comes at a cost. High-interaction honeypots are more difficult to develop and maintain. Maybe using a honeypot with a lower level of interaction would already be sufficient.

- Take scalability into account. What might work for a single honeypot, might not work when multiple honeypots are deployed. Test everything for the actual number of honeypots which are going to be deployed.

- Using virtualization makes it easier to redeploy the honeypot. It also improves the scalability of the honeypot setup.

- Take into account that creating the honeypot leaves traces of the creator on the honeypot. For example, Windows logs the IP address of users logging in via RDP. The browser is also used often during the creation of the honeypot. Do not forget to clean the personal information from the honeypot or minimize the amount of personal information stored as much as possible.

- Automation can spare a lot of time. It can be difficult and time consuming to implement, but when the number of honeypots grows and their time online increases, it can spare a lot of manual labor.

- Monitoring is an important part of the honeypot. It provides insights into the data collection without interfering with the data collection. It also allows to spot problems in an early stage. Monitoring can be used to prevent misuse of the honeypot. Excessive misuse can be detected and acted upon when necessary. Monitoring can be implemented in different ways. The best solution depends on the domain and implementation of the honeypot.

Appendix B

Feedback on the Honeypot

During the development of the honeypot, members of the Northwave Red team have been asked to provide feedback on the content of the honeypot. Their feedback is summarized and can be found in the coming paragraphs.

The following feedback was provided by Luc van den Ackerveken:

- The honeypot is pretty empty. There are not a lot of files on the Desktop or in the User Folders.
- The history of Firefox needs to be cleaned, it contains information about the creation of the honeypot.
- The files and Windows events are all quite new.

The following feedback was provided by Tijme Gommers:

- The boot time of the system is quite recent.
- The first Windows event logs are quite recent.
- The installation date of the system is quite recent.
- There is a shortcut to PowerShell on the Desktop but there is no PowerShell history.
- The history of Firefox shows the manual creation of all the Confluence pages and that an evaluation license is requested.
- The server is very slow.

The following feedback was provided by Alex Rommelse:

- The language of Firefox should be changed to English.
- Reduce the number of shortcuts to tools which are used by system administrators. They often do not use shortcuts because they know their way around the system.
- Remove the *Office365* item from the *TODO.txt* file in the Documents folder.
- Try to hide or rename the Splunk forwarder.

Appendix C

Dark Web Credential Spreading Interviews

To gain information about how RDP credentials are spread on the dark web, interviews have been conducted with John Fokker, head of Cyber Investigations for McAfee Advanced Threat Research, and with Tycho van Marle, Teamlead of a Research Team at Intel 471. The lessons learned from these interviews are discussed in the coming paragraphs.

The interview with John Fokker provided the following lessons learned:

- On the dark web there are so-called *RDP Shops* where RDP credentials are sold. The credentials sold on these kinds of shops are generally weak username and password combinations.
- It can take a long time before credential leaks are accessed. To gain attention to the credential leaks, it can be considered to post the URLs of the credential leaks on hacking forums.
- Directly publishing the IP address and credentials of a honeypot can be too obvious.
- Generally, there are two different kinds of RDP attackers: attackers which harvest RDP credentials to sell them and attackers which perform targeted attacks to infiltrate and sometimes destroy the infrastructure of a company. This research tries to attract the second kind of attackers, this has to be taken into account for the development of the honeypot. The honeypot needs to be attractive enough for an attacker to target the machine.
- Known hacking forums:
  - `hackforums.net`

  - `nulled.io`
  - `blackpass.bz`

- Currently, there is a trend for Russian RDP shops, moving away from TOR because of a lack of trust. They shift to surface web websites with a strict access policy.
- Shodan is used often to search for IP addresses with an accessible RDP port.
- *UAS Shop* is currently one of the biggest RDP shops on the dark web.
- *Canarytokens* [92] can be used to monitor file access on the honeypot.

The interview with Tycho van Marle provided the following lessons learned:

- Consider whether the honeypot should be restored after a single RDP session. It would be interesting to gather information about attackers which login multiple times. Restoring the honeypot can also lead to honeypot detection.
- There are three interesting setups possible for the honeypot:
  - *Use the subdomain of an existing company:* this setup can provide insights into what kind of attacks are performed on the servers of the company. It increases the credibility of the honeypot. The honeypot should be properly separated from the real company network. This would not only be useful for the research, but also the company can benefit from the collected information. Depending on what kind of company would be interested in this scenario, if it is a well-known company its RDP credentials are more valuable on the dark web.
  - *Build an extensive honeypot:* provide the honeypot with content which is interesting to an attacker. For example, store fake vital company assets or provide the honeypot with an extensive local network. The assets should persuade the attacker to attack the system itself, instead of using it as a stepping stone to perform other malicious behavior. The valuable asset on the honeypot could be used as a selling point for the RDP credentials on the dark web.
  - *Build a simple honeypot:* keep the setup of the honeypot simple and spread on a larger number of platforms. This will increase chance of the honeypot being used as a stepping stone. Moreover, selling credentials of such a honeypot is less interesting. The price of the credentials will be low because the honeypot does not have valuable assets.

- Credentials are occasionally leaked by accident on software development version control websites such as Github, Gitlab and Bitbucket.
- Skype automatically scans hyperlinks. It could be interesting to send URLs with credentials included.
- Telegram can be used to spread or sell credentials.

Appendix D

Interviews Members CERT

Interviews have been conducted with employees of the Northwave CERT, for the creation of the classification. The results of the interviews are described in the coming sections per employee.

D.1 Peter Wagenaar

In the coming subsections, a summary of the interview with Peter Wagenaar of the Northwave CERT will be given.

D.1.1 Reconnaissance

Goal of reconnaissance is to gain knowledge about the system. The hardware of the system and/or the information the server contains will generally determine what will be done with the system by an attacker. Questions which are important in this phase are:

- What are the hardware specifications of the system?
- What kind of information does the system contain?
- Is the system critical to the owner of the system?

Reconnaissance can be done in multiple ways:

- Usage of tools.
- Manually search through the files and folders.
- Use code/commands in CMD/PowerShell.

Not only the system itself is searched, also the local network and connected devices will generally be checked.

After the reconnaissance phase, an attacker will decide on whether persistent access should be gained or not. Gaining persistent access is beneficial to the attacker but there is a possibility that it increases the risk of being detected. It is also the question whether the time put into gaining persistent access is worth the effort.

Examples of how persistent access can be gained are:

- Usage of malware.
- Credential grabbing, acquire as much credentials of different accounts as possible.
- Create one or more new user accounts.

It occurs that more than one method is used to ensure the persistent access.

Besides gaining persistent access, also lateral movement is an optional phase. This depends on whether the system is connected to other systems. The following are examples of activities which can be performed as part of lateral movement:

- Port scan on the local system.
- Port scan on other systems in the local network.
- Accessing systems using remote administration tools (e.g. RDP, PowerShell Remoting, Windows Management Instrumentation (WMI), Virtual Network Computing (VNC)).
- Accessing network shares.
- Credential guessing.
- Credential grabbing.
- Usage of known exploits.

Using legitimate tools like PsExec and legitimate services such as PowerShell Remoting are preferred because it decreases the chance of detection.

D.1.2 Malicious Behavior

An attacker will strive to maximize monetization.

- Root → Proxy: 1. Installation of proxy software or use of existing services like Netcat, 2. Network traffic being received on a single listening port from one or more unknown IP addresses, incoming network traffic is answered with outgoing network traffic.

- Root → Payload Execution: 1. Payload transfer, 2. Payload execution, optionally 3. Persistent execution (ensuring payload execution after a reboot). All further steps basically require payload analysis but nonetheless distinguishing events are still present.
  - Payload Execution → DDoS Bot: 4. Outgoing network traffic of the same kind related to amplification techniques.
  - Payload Execution → Ransomware: 4. A lot of file access, 5. The files accessed are encrypted, 6. Ransomware notes are placed.
  - Payload Execution → PoS Malware: 4. Payload executes continuously to harvest credentials, 5. Storage of harvested credentials or network traffic generated by sending the credentials to an unknown IP address.
  - Payload Execution → Cryptominer: 4. High CPU usage, 5. Regular network traffic to an unknown IP address.
  - Payload Execution → Bulk Credential Harvesting: 4. Large amount of network traffic to an unknown IP address.
  - Payload Execution → Specific Credential Harvesting : 4. Accessing specific files which contain credentials, 5. Network traffic to an unknown IP address.
  - Payload Execution → Resource Misuse: 4. High Disk I/O, Memory and/or CPU usage.
    - Resource Misuse → Spam: 5. High volume of SMTP traffic.
- Root → Data Breach: 1. High disk I/O/high volume of file/folder access, 2. High volume of network traffic to a single unknown IP address, depending on the context 3. Database transaction logs containing information about read credentials.
- Root → Hosting Services: 1. Optionally new software will be installed, but there are always new files created to facilitate the hosted service, 2. Changes to the firewall.
  - Hosting Services → Phishing: 3. Port 80 and/or 443 opened, 4. Website related files stored.
  - Hosting Services → DDoS Controller: 3. Administrative files concerning DDoS bots are stored.
  - Hosting Services → Seedbox: 3. Files concerning torrents are stored.

D.2 Marinus Boekelo

The following decision tree is the result of the interview with Marinus Boekelo:

- Root → Payload Execution: 1. Payload transfer, 2. Payload execution.
  - Payload Execution → Persistency: 3. Gain persistent access.
    - Persistency → Cryptominer: 4. Mining process results in a high CPU usage, 5. Content containing information about the mining pool is stored.
    - Persistency → DDoS Bot: 4. Network traffic to a Command and Control (C&C) server, 5. Outgoing network traffic relating to denial of service traffic (amplification methods).
    - Persistency → Remote Access Tool (RAT): 4. Network traffic to a Command and Control (C&C) server, 5. Proxy traffic, 6. RDP or VNC traffic.
    - Persistency → Banking Malware: 4. Network driver(s) are installed or DLL hooking is enabled, 5. Network traffic to a Command and Control server.
    - Persistency → Keylogger: 4. Installation of driver(s) or DLL hooking is enabled, 5. Regular contact with a Command and Control server.
  - Payload Execution → Ransomware: 3. Disable Shadow Copies, 4. A large volume of read and write operations, 5. A large volume of new files are created, 6. Network traffic to a C&C server, 7. Ransomnotes are created possibly containing a Bitcoin and/or Onion address.
  - Payload Execution → Adware: 3. One or more browser plugins are installed, 4. Network traffic to ad platforms.
  - Payload Execution → Credential Stealing: 3. Read operations on files which contain credentials, 4. Transfer of the credentials generating network traffic.
  - Payload Execution → Dropper: 3. Retrieve payload, 4. Execute retrieved payload.
- Root → Destruction: 1. Files, users or configuration files are deleted, thereby destroying the system.
- Root → Data Breach: 1. Large volume of outgoing network traffic to a single IP address, 2. Large volume of disk I/O.
  - Data Breach → Credential Stealing: 3. Files containing credentials are read.

  - Data Breach → Database: 4. One or more databases are accessed in a large volume.
  - Data Breach → Personal Files: 5. Personal files are accessed.

D.3 Martijn Hoogesteger

The following decision tree is the result of the interview with Martijn Hoogesteger:

- RDP Login → Payload Execution: 1. Payload delivery, 2. Payload execution.
  - Payload Execution → PowerShell/Command Prompt Script: 3. PowerShell/Command Prompt process is created, 4. Script and/or commands are executed which could be logged.
  - Payload Execution → Binary: 3. The payload is a binary.
    - Binary → Hosting: 4. A new service is installed, 5. Ports are opened in the firewall, 6. Incoming and outgoing traffic though the opened port.
    - Binary → Cryptominer: 4. Connection is made with a mining pool, retrieving mining information, resulting in incoming and outgoing network traffic, 5. Mining starts thereby generating a high CPU usage.
    - Binary → Adware: 4. Installation of one or more browser plugins.
    - Binary → Ransomware: 4. Encryption process generating a high CPU usage and high volume of disk I/O, 5. Ransomnotes are created.
    - Binary → Hacktools: 4. Known hashes of hacktools.
- RDP Login → Data Theft: 1. Files accessed containing private information, 2. Transfer of the data generates network traffic to a single IP address.
  - Data Theft → Credentials: 3. Files accessed containing credentials.
  - Data Theft → Company Data: 3. Files accessed containing company data.
  - Data Theft → Private Information: 3. Files accessed containing confidential information, like customer information.
- RDP Login → Information Gain: 1. System properties are viewed, 2. Connected shares are accessed.

- Information Gain → System: 3. Optionally volume information is accessed, 4. Optionally the register is accessed.
- Information Gain → Local Network: 3. Network traffic generated by accessing the local network.

Appendix E

Classification Evaluation

A detailed description of the analyzed images will be given in this Appendix. When Firefox connections are made, only the connections to port 80 (HTTP) and port 443 (HTTPS) are counted. All timestamps are in Coordinated Universal Time (UTC) + 0. There are three different triggers for the backup and restore process: RDP session between 90 seconds and 45 minutes, RDP session longer than 45 minutes and bandwidth threshold exceeded.

Image Name: osDisk-confluencetemp1-20190626-034704
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 26/06/2019 07:32:33 - 26/06/2019 07:34:26
IP Address: 94.156.133.60
Proposed Classification

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:
  - 26/06/2019 07:34:07: UserAssist shows the date and time of when *C:\hide.bat* was last executed.
  - 26/06/2019 07:34:07: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.ex* process. The executable was provided with the following arguments *C:\windows\system32\cmd.exe /c ""C:\hide.bat" "*.
  - 26/06/2019 07:34:08: Event ID 4688 shows the creation of the *C:\Windows\System32\reg.ex* process. The executable was provided with the following arguments *REG QUERY "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v Hidden*. The process was started by *hide.bat*, created at 07:34:07.

  - 26/06/2019 07:34:08: Event ID 4688 shows the creation of the *C:\Windows\System32\find.ex* process. The executable was provided with the following arguments *Find "0x2"*. The process was started by *hide.bat*, created at 07:34:07.
  - 26/06/2019 07:34:08: Event ID 4688 shows the creation of the *C:\Windows\System32\reg.ex* process. The executable was provided with the following arguments *REG ADD "HKCU\Software\ Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v Hidden /t REG_DWORD /d 1 /f* (hidden files are now shown). The process was started by *hide.bat*, created at 07:34:07.

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:
  - 26/06/2019 07:34:09: UserAssist shows the date and time of when *C:\system.bat* was last executed.
  - 26/06/2019 07:34:09: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.ex* process. The executable was provided with the following arguments *C:\windows\system32\cmd.exe /c ""C:\system.bat" "*.
  - 26/06/2019 07:34:09: Event ID 4688 shows the creation of the *C:\Windows\System32\reg.ex* process. The executable was provided with the following arguments *REG QUERY "HKCU\Software\ Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v ShowSuperHidden*. The process was started by *system.bat*, created at 07:34:09.
  - 26/06/2019 07:34:09: Event ID 4688 shows the creation of the *C:\Windows\System32\find.ex* process. The executable was provided with the following arguments *Find "0x0"*. The process was started by *system.bat*, created at 07:34:09.
  - 26/06/2019 07:34:10: Event ID 4688 shows the creation of the *C:\Windows\System32\reg.ex* process. The executable was provided with the following arguments *REG ADD "HKCU\Software\ Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v ShowSuperHidden /t REG_DWORD /d 1 /f* (super-hidden/system files are now shown). The process was started by *system.bat*, created at 07:34:09.

Classifications: PowerShell or Command Prompt Execution (2x)
Thorough Search

- *Are all the given classifications correct?*
  Yes, two different bat scripts were executed.

- *Was the attack classified with too few classifications?*
  Besides the script execution, no other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190714-143238
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 14/07/2019 18:58:51 - 14/07/2019 19:00:24
IP Address: 185.189.114.11
Proposed Classification

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:

  - 14/07/2019 18:59:53: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\load.exe*.
  - 14/07/2019 19:00:15: UserAssist shows the date and time of when *C:\Users\confluence\Desktop\load.exe* was last executed.
  - 14/07/2019 19:00:15: Event ID 4688 shows the creation of the *C:\Users\confluence\Desktop\load.exe* process.
  - 14/07/2019 19:00:15: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.ex* process. The executable was provided with the following arguments *"C:\Windows\System32\cmd.exe" /k ping -n 2 localhost < nul & del /F /Q "C:\Users\confluence\Desktop\load.exe"*. The process was started by *load.exe*, created at 19:00:15.
  - 14/07/2019 19:00:15: Event ID 4688 shows the creation of the *C:\Windows\System32\PING.EX* process. The executable was provided with the following arguments *ping -n 2 localhost*. The process was started by *cmd.exe*, created at 19:00:15.
  - 14/07/2019 19:00:15: AmCache shows that *C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\My\CTLs\Adobe\weniter.exe was executed. The SHA1 hash of the executable is 6e891ff0285b53592b92e393b9bd3a7b4bfb2e34.*
  - *14/07/2019 19:00:15: Event ID 4688 shows the creation of the C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\*

My\CTLs\Adobe\weniter.exe process. The process was started by load.exe, created at 19:00:15.

- *14/07/2019 19:00:15: Event ID 4688 shows the creation of the C:\Windows\System32\cmd.ex process. The executable was provided with the following arguments C:\windows\system32\cmd.exe /c SchTasks /create /SC MINUTE /TN AdobeUpdate /TR C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\My\CTLs\Adobe\taskhost.exe. The process was started by weniter.exe, created at 19:00:15.*
- *14/07/2019 19:00:15: Event ID 4688 shows the creation of the C:\Windows\System32\cmd.ex process. The executable was provided with the following arguments C:\windows\system32\cmd.exe /c SchTasks /create /SC HOURLY /TN AdobeServis /TR C:\ProgramData\Flash\MSACuiL.exe. The process was started by weniter.exe, created at 19:00:15.*
- *14/07/2019 19:00:15: Event ID 4688 shows the creation of the C:\Windows\System32\schtasks.exe process. because of the following Command Line execution SchTasks /create /SC HOURLY /TN AdobeServis /TR C:\ProgramData\Flash\MSACuiL.exe".* The process was started by *cmd.exe*, created at 19:00:15.
- 14/07/2019 19:00:15: Event ID 4688 shows the creation of the *C:\Windows\System32\schtasks.ex* process. The executable was provided with the following arguments *SchTasks /create /SC MINUTE /TN AdobeUpdate /TR C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\My\CTLs\Adobe\taskhost.exe.* The process was started by *cmd.exe*, created at 19:00:15.
- 14/07/2019 19:00:16: Event ID 4659 shows that *C:\Users\confluence\Desktop\load.exe* was deleted.
- 14/07/2019 19:00:17: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\weniter.exe* to *87.236.19.238:20664*.
- 14/07/2019 19:00:17: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\weniter.exe* to *87.236.19.238:21*.

- 14/07/2019 19:00:18: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\weniter.exe* to *87.236.19.238:20942*.
- 14/07/2019 19:00:18: Edge/IE was used to access `ftp://kinirwy6.beget.tech//taskhost.dll` to download *taskhost.dll*.
- 14/07/2019 19:00:21: Shimcache shows when the key of *C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\My\CTLs\Adobe\taskhost.exe* was last updated in the Shimcache. The executed flag of the executable is *True*.
- 14/07/2019 19:00:22: Event ID 4688 shows the creation of the *C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\My\CTLs\Adobe\taskhost.exe* process. The process was started by *weniter.exe*, created at 19:00:15.
- 14/07/2019 19:00:22: AmCache shows that *C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\My\CTLs\Adobe\taskhost.exe* was executed. The SHA1 hash of the executable is *4dccfde8b36e26c5d4a07ae79e9431b6a775cfbe*.
- 14/07/2019 19:00:23: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\taskhost.exe* to *87.236.19.238:20769*.
- 14/07/2019 19:00:23: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\taskhost.exe* to *87.236.19.238:21*.
- 14/07/2019 19:00:24: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\taskhost.exe* to *87.236.19.238:20639*.
- 14/07/2019 19:00:24: Edge/IE was used to access `ftp://kinirwy6.beget.tech/MSASCuiL.dll` to download *MSASCuiL.dll*.
- 14/07/2019 19:00:27: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\*

roaming\microsoft\systemcertificates\my\ctls\adobe\taskhost.exe to *87.236.19.238:20894*.

- 14/07/2019 19:00:27: Edge/IE was used to access `ftp://kinirwy6.beget.tech/intel.dll` to download *intel.dll*.
- 14/07/2019 19:00:58: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\weniter.exe* to *87.236.19.238:21*.
- 14/07/2019 19:00:59: Edge/IE was used to access `ftp://kinirwy6.beget.tech//taskhost.dll` to download *taskhost.dll*.
- 14/07/2019 19:00:59: Event ID 4688 shows the creation of the *C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\My\CTLs\Adobe\taskhost.exe* process. The process was started by *weniter.exe*, created at 19:00:15.
- 14/07/2019 19:01:01: Event ID 4688 shows the creation of the *C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\My\CTLs\Adobe\taskhost.exe* process.
- 14/07/2019 19:01:30: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\weniter.exe* to *151.139.128.14:80*.
- 14/07/2019 19:01:30: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\weniter.exe* to *88.99.66.31:443*.
- 14/07/2019 19:01:31: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\appdata\roaming\microsoft\systemcertificates\my\ctls\adobe\weniter.exe* to *151.139.128.14:80*.
- 14/07/2019 19:01:31: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process. The executable was provided with the following arguments *"C:\Windows\System32\cmd.exe" /k ping -n 2 localhost < nul & del /F /Q "C:\Users\confluence\AppData\Roaming\Microsoft\SystemCertificates\My\CTLs\Adobe\weniter.exe"*. The process was started by *weniter.exe*, created at 19:00:15.
- 14/07/2019 19:01:31: Event ID 4688 shows the creation of the *C:\Windows\System32\PING.EX* process. The executable was

provided with the following arguments *ping -n 2 localhost*. The process was started by *cmd.exe*, created at 19:01:31.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.

- *Was the attack classified with too few classifications?*
  Yes, a cryptominer (*weniter.exe*) was installed. The cryptominer did not start before the backup and restore process started. Because of this, there never was high CPU usage. That is why the attack was missed by the classification.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190724-111535
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 24/07/2019 18:59:24 - 24/07/2019 19:01:01
IP Address: 89.238.154.166
Proposed Classification

- 1. Reconnaissance → Payload Execution & & 2. Payload Execution → PowerShell or Command Prompt Execution:
  - 24/07/2019 19:00:28: MRU Run Commands show when the *cmd.exe* command was used in the Run utility of Windows.
  - 24/07/2019 19:00:28: UserAssist shows the date and time of when *cmd.exe* was last executed.
  - 24/07/2019 19:00:28: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process.
  - 24/07/2019 19:00:50: Event ID 4688 shows the creation of the *C:\Windows\System32\WindowsPowerShell\v1.0\powershell.ex* process. The executable was provided with the following arguments *PowerShell -command "$cmd = (New-Object System.Net.WebClient) .DownloadString('http://104.248.167.144:8333/blockchainBitcoin'); Invoke-Expression $cmd; Stop-Process -Force -processname cmd;";*. The process was started by *cmd.exe*, created at 19:00:28.
  - 24/07/2019 19:00:51: Event ID 5156 shows an outbound connection being made by *C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"* to *104.248.167.144:8333.*

Classifications: PowerShell or Command Prompt Execution
Thorough Search

- *Are all the given classifications correct?*
  Yes, a Command Prompt was used to execute a command in PowerShell.
- *Was the attack classified with too few classifications?*
  No other activity has been found. Even though with the downloaded PowerShell script a cryptominer could be downloaded and installed, it did not start. The script script first checks if the user is an administrator, if that is the case, the installation will not start. The user which executes the script is an administrator in the research setup, so the script does not continue.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190726-113406
Trigger: RDP session longer than 45 minutes
Login Time Frame: 26/07/2019 11:37:56 - 26/07/2019 12:25:30
IP Address: 108.62.5.136
Proposed Classification

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:
  - 26/07/2019 11:38:48: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 26/07/2019 11:40:29: Firefox was used to access `http://www.mediafire.com/file/xxj5bz1tndcaf5a/DUbrute_2.1_%252B_Scanner_IP_%255BNmap%255D.rar/file` to download *DUbrute 2.1 + Scanner IP [Nmap]*.
  - 26/07/2019 11:40:44: Jump Lists show when *C:\Users\confluence\Downloads\DUbrute 2.1 + Scanner IP [Nmap].rar* was last accessed.
  - 26/07/2019 11:43:22: Event ID 4688 shows the creation of the *C:\Program Files\7-Zip\7zG.ex* process. The executable was provided with the following arguments *"C:\Program Files\7-Zip\7zG.exe" x -o"C:\Users\confluence\Downloads\" -an -ai#7zMap27551:134:7zEvent15396.*
  - 26/07/2019 11:43:29: UserAssist shows the date and time of when *C:\Users\confluence\Downloads\DUbrute + Scanner IP\nmap-5.51-setup.exe* was last executed.

- 26/07/2019 11:43:29: AmCache shows that *C:\Users\confluence\Downloads\DUbrute + Scanner IP\nmap-5.51-setup.exe* was executed. The SHA1 hash of the executable is *af9383ed82388f86b5ab8381be2a78dabf932950*.
- 26/07/2019 11:43:29: Event ID 4688 shows the creation of the *C:\Users\confluence\Downloads\DUbrute + Scanner IP\nmap-5.51-setup.exe* process.
- 26/07/2019 11:43:53: Event ID 4688 shows the creation of the *C:\Program Files (x86)\Nmap\vcredist_x86.exe* process. The process was started by *nmap-5.51-setup.exe*, created at 11:43:29.
- 26/07/2019 11:43:53: AmCache shows that *C:\Program Files (x86)\Nmap\vcredist_x86.exe* was executed. The SHA1 hash of the executable is *372d9c1670343d3fb252209ba210d4dc4d67d358*.
- 26/07/2019 11:43:55: Event ID 4688 shows the creation of the *C:\b052af2d6fb4152324\Setup.exe* process. The process was started by *vcredist_x86.exe*, created at 11:43:53.
- 26/07/2019 11:45:05: AmCache shows that *C:\Program Files (x86)\Nmap\winpcap-nmap-4.12.exe* was executed. The SHA1 hash of the executable is *467e53ec841638a7b1ef5b979415c513c109a7a1*.
- 26/07/2019 11:47:48: AmCache shows that *C:\Program Files (x86)\Nmap\vcredist2008_x86.exe* was executed. The SHA1 hash of the executable is *bd18409cfe75b88c2a9432d36d96f4bf125a3237*.
- 26/07/2019 11:48:12: AmCache shows that *C:\Program Files (x86)\Nmap\nmap.exe* was executed. The SHA1 hash of the executable is *4e3fc623510bf3a81e726c5a4fdb0b0df9bb7c59*.
- 26/07/2019 11:48:26: Jump Lists show when *C:\Users\confluence\Downloads\DUbrute + Scanner IP* was last accessed.
- 26/07/2019 11:48:26: Jump Lists show when *C:\Users\confluence\Downloads\DUbrute + Scanner IP\results.txt* was last accessed.
- 26/07/2019 11:45:05: Event ID 4688 shows the creation of the *C:\Program Files (x86)\Nmap\winpcap-nmap-4.12.exe* process. The process was started by *nmap-5.51-setup.exe*, created at 11:43:29.
- 26/07/2019 11:47:43: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\net.ex* process. The executable was

provided with the following arguments *net stop npf.* The process was started by *winpcap-nmap-4.12.exe*, created at 11:45:05.

  - 26/07/2019 11:47:43: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 stop npf.* The process was started by *net.exe*, created at 11:47:43.
  - 26/07/2019 11:47:47: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\net.ex* process. The executable was provided with the following arguments *net start npf.* The process was started by *winpcap-nmap-4.12.exe*, created at 11:45:05.
  - 26/07/2019 11:47:47: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 start npf.* The process was started by *net.exe*, created at 11:47:47.
  - 26/07/2019 11:47:48: Event ID 4688 shows the creation of the *C:\Program Files (x86)\Nmap\vcredist2008_x86.exe* process. The process was started by *nmap-5.51-setup.exe*, created at 11:43:29.
  - 26/07/2019 11:47:48: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\regedt32.ex* process. The executable was provided with the following arguments *regedt32 /S "C:\Program Files (x86)\Nmap\nmap_performance.reg".* The process was started by *nmap-5.51-setup.exe*, created at 11:43:29.
  - 26/07/2019 11:47:49: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\regedit.ex* process. The executable was provided with the following arguments *"C:\windows\regedit.exe" /S "C:\Program Files (x86)\Nmap\nmap_performance.reg".* The process was started by *regedt32.exe*, created at 11:47:48.
  - 26/07/2019 11:47:50: Event ID 4688 shows the creation of the *\4a8c772743e7975ae86798\install.exe* process. The process was started by *vcredist2008_x86.exe*, created at 11:47:48.

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:

  - 26/07/2019 11:38:48: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 26/07/2019 11:40:57: Firefox was used to access `https://gulfupload.com/c3jgi5t3w3jo` to download تحميل *user and pass rar.*

- 26/07/2019 11:41:54: Firefox was used to access `http://s1.gulfupload.com:8080/d/esciivzgrktdjfvgtgl42yvcenij7doxekcjtkzn6rx5wqwen\7zkrjbpx43bdbco6dxbat7t/user%20and%20pass.rar` to download *user and pass.rar*.
- 26/07/2019 11:41:52: Jump Lists show when *C:\Users\confluence\Downloads\user and pass.rar* was last accessed.
- 26/07/2019 11:41:55: MRU Recent Files & Folders shows when *user and pass.rar* was last opened or saved.
- 26/07/2019 11:42:24: Event ID 4688 shows the creation of the *C:\Program Files\7-Zip\7zG.ex* process. The executable was provided with the following arguments *"C:\Program Files\7-Zip\7zG.exe" x -o"C:\Users\confluence\Downloads\" -an -ai#7zMap201:98:7zEvent5119*.
- 26/07/2019 11:42:24: AmCache shows that *C:\Program Files\7-Zip\7zG.exe* was executed. The SHA1 hash of the executable is *df22612647e9404a515d48ebad490349685250de*.
- 26/07/2019 11:42:25: Jump Lists show when *C:\Users\confluence\Downloads\passward.txt* was last accessed.
- 26/07/2019 11:42:25: Jump Lists show when *C:\Users\confluence\Downloads\user.txt* was last accessed.
- 26/07/2019 11:42:29: Event ID 4688 shows the creation of the *C:\Windows\System32\notepad.exe* process. The executable was provided with the following arguments *"C:\windows\system32\NOTEPAD.EXE" C:\Users\confluence\Downloads\user.txt*.
- 26/07/2019 11:42:35: Event ID 4688 shows the creation of the *C:\Windows\System32\notepad.exe* process. The executable was provided with the following arguments *"C:\windows\system32\NOTEPAD.EXE" C:\Users\confluence\Downloads\passward.txt*.
- 26/07/2019 11:42:48: Event ID 4688 shows the creation of the *C:\Windows\System32\notepad.exe* process. The executable was provided with the following arguments *"C:\windows\system32\NOTEPAD.EXE" C:\Users\confluence\Downloads\user.txt*.
- 26/07/2019 11:44:47: Event ID 4688 shows the creation of the *C:\Windows\System32\notepad.ex* process. The executable was provided with the following arguments *"C:\windows\system32\NOTEPAD.EXE" C:\Users\confluence\Downloads\passward.txt*.

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:
  - 26/07/2019 11:38:48: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 26/07/2019 12:04:03: Firefox was used to access `https://codeload.github.com/mdhinkle038/Arduino_RDP_Exploit/zip/master` to download *Arduino_ RDP_ Exploit-master.zip*.
  - 26/07/2019 12:04:31: Jump Lists show when *C:\Users\confluence\Downloads\Arduino_ RDP_ Exploit-master\Arduino_ RDP_ Exploit-master* was last accessed.
  - 26/07/2019 12:04:40: MRU Recent Files & Folders shows when *RDP.ino* was last opened or saved.
- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:
  - 26/07/2019 11:38:48: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 26/07/2019 12:03:50: Event ID 4688 shows the creation of the *\Program Files\Mozilla Firefox\firefox.exe* process.
  - 26/07/2019 12:13:57: Event ID 4688 shows the creation of the *\Program Files\Mozilla Firefox\firefox.exe* process.
  - 26/07/2019 12:16:23: Shimcache shows when the key of *SYSVOL\Users\confluence\Downloads\kube-forwarder.exe* was last updated in the Shimcache. The execution flag of the executable is *True*.
  - 26/07/2019 12:16:33: Event ID 4688 shows the creation of the *\Users\confluence\Downloads\kube-forwarder.exe* process.
  - 26/07/2019 12:16:34: AmCache shows that *C:\Users\confluence\Downloads\kube-forwarder.exe* was executed. The SHA1 hash of the executable is *c2616b6f4569d00b22e00af0c1a71e9e292c437e*.
  - 26/07/2019 12:18:56: UserAssist shows the date and time of when *C:\Users\confluence\Downloads\kube-forwarder.exe* was last accessed.
  - 26/07/2019 12:18:56: Event ID 4688 shows the creation of the *\Users\confluence\Downloads\kube-forwarder.exe* process.
  - 26/07/2019 12:18:58: Event ID 4688 shows the creation of the *\Users\confluence\AppData\Local\Temp\2\nso75DF.tmp\old-uninstaller.ex* process. The executable was provided with the following arguments

"C:\Users\confluence\AppData\Local\Temp\2\nso75DF.tmp\old-uninstaller.exe" /S /KEEP_APP_DATA /currentuser –keep-shortcuts –updated _?=C:\Users\confluence\AppData\Local\Programs\kube-forwarder. The process was started by *kube-forwarder.exe*, created at 12:18:56.

  - 26/07/2019 12:18:59: AmCache shows that *C:\Users\confluence\AppData\Local\Temp\2\nso75DF.tmp\old-uninstaller.exe* was executed. The SHA1 hash of the executable is *c38093fd0afc96c3c52dffd3c175b983cae0fe41.*
  - 26/07/2019 12:20:33: Jump Lists show when *C:\Users\confluence\Downloads\kube-forwarder-1.3.1.zip* was last accessed.
  - 26/07/2019 12:20:35: Firefox was used to access `https://codeload.github.com/pixel-point/kube-forwarder/zip/v1.3.1` to download *kube-forwarder-1.3.1.zip.*
  - 26/07/2019 12:20:36: MRU Recent Files & Folders shows when *kube-forwarder-1.3.1.zip* was last opened or saved.

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:

  - 26/07/2019 11:48:12: Event ID 4688 shows the creation of the *\Windows\System32\cmd.ex* process. The executable was provided with the following arguments *C:\windows\system32\cmd.exe /c ""C:\Users\confluence\Downloads\DUbrute + Scanner IP\Scanner.bat" ".*
  - 26/07/2019 11:48:12: Event ID 4688 shows the creation of the *\Program Files (x86)\Nmap\nmap.ex* process. The executable was provided with the following arguments *nmap -n -Pn -p T:3389 -T5 –script rdp.nse -iR 0.* The process was started by *Scanner.bat*, created at 11:48:12.
  - 26/07/2019 11:48:12: Event ID 4688 shows the creation of the *\Program Files (x86)\Nmap\nmap.ex* process. The executable was provided with the following arguments *nmap -n -Pn -p T:3389 -T5 –script rdp.nse -iR 0.* The process was started by *Scanner.bat*, created at 11:48:12.

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:

  - 26/07/2019 11:49:06: Event ID 4688 shows the creation of the *\Windows\System32\cmd.ex* process. The executable was provided with the following arguments *C:\windows\system32\cmd.exe /c ""C:\Users\confluence\Downloads\DUbrute + Scanner IP\Scanner.bat" "*.

  - 26/07/2019 11:49:06: Event ID 4688 shows the creation of the *\Program Files (x86)\Nmap\nmap.ex* process. The executable was provided with the following arguments *nmap -n -Pn -p T:3389 -T5 –script rdp.nse -iR 0.* The process was started by *Scanner.bat*, created at 11:49:06.
  - 26/07/2019 11:49:06: Event ID 4688 shows the creation of the *\Program Files (x86)\Nmap\nmap.ex* process. The executable was provided with the following arguments *nmap -n -Pn -p T:3389 -T5 –script rdp.nse -iR 0.* The process was started by *Scanner.bat*, created at 11:49:06.

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:
  - 26/07/2019 11:49:09: Event ID 4688 shows the creation of the *\Windows\System32\cmd.ex* process. The executable was provided with the following arguments *C:\windows\system32\cmd.exe /c ""C:\Users\confluence\Downloads\DUbrute + Scanner IP\Scanner.bat" "*.
  - 26/07/2019 11:49:09: Event ID 4688 shows the creation of the *\Program Files (x86)\Nmap\nmap.ex* process. The executable was provided with the following arguments *nmap -n -Pn -p T:3389 -T5 –script rdp.nse -iR 0.* The process was started by *Scanner.bat*, created at 11:49:09.
  - 26/07/2019 11:49:09: Event ID 4688 shows the creation of the *\Program Files (x86)\Nmap\nmap.ex* process. The executable was provided with the following arguments *nmap -n -Pn -p T:3389 -T5 –script rdp.nse -iR 0.* The process was started by *Scanner.bat*, created at 11:49:09.

Classifications: PowerShell or Command Prompt Execution (3x)
Thorough Search

- *Are all the given classifications correct?*
  Yes, the same bat script was executed three times. The bat script started an Nmap scan with the help of a script.
- *Was the attack classified with too few classifications?*
  Yes, the Nmap scan was not classified properly. The default scan

method used by Nmap only sends a SYN packet and does not setup a full TCP connection. These SYN packets are not logged by Windows. That is the reason why the Nmap scan was not properly classified. The Windows Events Logs contained events about packets blocked by the Windows Filtering Platform, originating from the IP addresses which were probed by Nmap. It is unknown what kind of packets were received.

- *Did the classification miss information which was important to determine what the attacker did?*
  Yes, as already mentioned the Nmap scan was missed. The only way in which the Nmap scan could have been detected in this case, is by logging SYN packets which are sent or received. Event though you can see the responses originating from the IP addresses which were probed, you cannot determine with what purpose the packets were sent to the host. Windows Packet capture (WinPcap) driver called NPF is used by Nmap. This could be an indication for the usage of malicious software, but the driver is also used by other applications such as Wireshark. So it is not a direct indication.

Image Name: osDisk-confluencetemp1-20190622-085704
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 22/06/2019 09:02:08 - 22/06/2019 09:39:27
IP Address: 41.203.72.245
Proposed Classification

- 1. Reconnaissance → Payload Execution:
  - 22/06/2019 09:24:35: Event ID 4688 shows the creation of the *\Windows\System32\cmd.exe* process.
  - 22/06/2019 09:38:19: Event ID 4688 shows the creation of the *\Windows\System32\cmd.exe* process.
  - 22/06/2019 09:38:24: Event ID 4688 shows the creation of the *\Windows\System32\cmd.exe* process.
  - 22/06/2019 09:38:31: Event ID 4688 shows the creation of the *\Windows\System32\cmd.exe* process.
  - 22/06/2019 09:38:36: Event ID 4688 shows the creation of the *\Windows\System32\cmd.exe* process.
  - 22/06/2019 09:38:36: UserAssist shows the date and time of when *cmd.exe* was last executed from the taskbar.
- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:

- 22/06/2019 09:47:32: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\bz2.pyd.
- 22/06/2019 09:49:03: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\emails.txt.
- 22/06/2019 09:49:08: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\help.txt.
- 22/06/2019 09:49:18: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\library.zip.
- 22/06/2019 09:49:21: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\passwords.txt.
- 22/06/2019 09:49:32: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\python27.dll.
- 22/06/2019 09:49:34: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\select.pyd.
- 22/06/2019 09:49:37: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\unicodedata.pyd.
- 22/06/2019 09:49:38: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\valid.txt.
- 22/06/2019 09:49:39: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\w9xpopen.exe.
- 22/06/2019 09:49:40: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER\winlogon.exe.
- 22/06/2019 09:49:43: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER_socket.pyd.
- 22/06/2019 09:49:43: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER_hashlib.pyd.

- 22/06/2019 09:49:48: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\BRUTER\BRUTER_ssl.pyd.
- 22/06/2019 09:49:49: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\connection_fix.reg.
- 22/06/2019 09:49:50: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\ip.txt.
- 22/06/2019 09:49:51: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\log-SMTP-Server.txt.
- 22/06/2019 09:49:51: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\keys.txt.
- 22/06/2019 09:49:52: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\READ ME.txt.
- 22/06/2019 09:49:52: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\log-valid.txt.
- 22/06/2019 09:49:53: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\scan.txt.
- 22/06/2019 09:49:54: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\settings.ini.
- 22/06/2019 09:49:55: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\SMTP-Error-Log.txt.
- 22/06/2019 09:49:57: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \tex-

titC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\Smtp.exe.

  - 22/06/2019 09:49:58: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\thebuzis@xmpp.jp.txt.
  - 22/06/2019 09:49:59: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\wordlist.txt.
  - 22/06/2019 09:50:17: Event ID 4688 shows the creation of the *C:\Users\confluence\Desktop\RMGBrute\Windows SMTP Bruteforce Scanner 2015 - Ar3s\Smtp.exe* process.

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 17. External → Multiple Systems:
  - 27/06/2019 17:49:53: Event ID 5156 shows that *\device\harddiskvolume2\users\confluence\desktop\rmgbrute\windows smtp bruteforce scanner 2015 - ar3s\smtp.exe* creates 2967 outbound connections to port 25. Every connection is to a different IP address.

-

Classifications: External Pivoting Multiple Systems
Thorough Search

- *Are all the given classifications correct?*
  Yes, an executable created connections to the standard SMTP port of multiple external IP addresses.
- *Was the attack classified with too few classifications?*
  No other activity has been found besides the SMTP scan. Multiple Command Prompts have been executed but no related activities have been found.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

There were two different sessions captured in the same image. The first one was too short to trigger the automated backup and restore process. Because the first session has not influenced the second session, they both have been added to the evaluation.

Image Name: osDisk-confluencetemp1-20190629-041201
Trigger: None
Login Time Frame: 29/06/2019 06:00:29 - 29/06/2019 06:01:55 & 29/06/2019 06:03:10 - 29/06/2019 06:04:07
IP Address: 220.188.176.165
Proposed Classification

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 16. External → Hiding Web Activity:
  - 29/06/2019 06:00:48: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 29/06/2019 06:00:54: Event ID 5156 shows that Firefox creates 62 outbound connections starting from this point in time.
  - 29/06/2019 06:01:17: Firefox was used to access `https://www.google.com/search?client=firefox-b-d&q=paypal` (Google search for "paypal").
  - 29/06/2019 06:01:28: Firefox was used to access `https://www.paypal.com/nl/signin`.
  - 29/06/2019 06:03:40: Firefox was used to access `https://www.paypal.com/authflow/safe?returnUri=signin&country.x=NL&locale.x=nl_NL`.
  - 29/06/2019 06:03:41: Firefox was used to access `https://www.paypal.com/authflow/safe/?returnUri=signin&country.x=NL&locale.x=nl_NL`.
  - 29/06/2019 06:03:48: Firefox was used to access `https://www.paypal.com/authflow/entry/?clientInstanceId=9eecb522-\08e0-48bb-b33c-88e3927d8072&redirectUri=%2Fsignin&country.x=NL&locale.x=nl_NL`.

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  Yes, the browser of the system was used to access the login page of Paypal. Paypal is a payment system, where it is possible to automatically login.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190629-041201
Trigger: RDP session longer than 45 minutes
Login Time Frame: 29/06/2019 14:14:35 - 29/06/2019 15:01:56
IP Address: 89.39.107.205
Proposed Classification

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:
  - 29/06/2019 14:15:29: UserAssist shows the date and time of when *powershell.exe* was last executed from the taskbar.
  - 29/06/2019 14:15:29: Event ID 4688 shows the creation of the *C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe* process.
  - 29/06/2019 14:15:29: UserAssist shows the date and time of when *PowerShell.exe"* was last executed from the taskbar.
  - 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *"C:\windows\system32\net.exe" user Administrator اصنzor@#2536fkuj*. The process was started by *powershell.exe*, created at 14:15:29.
  - 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user Administrator اصنzor@#2536fkuj*. The process was started by *net.exe*, created at 14:15:32.
  - 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *"C:\windows\system32\net.exe" user Administrator اصنzor@#2536fkuj*. The process was started by *powershell.exe*, created at 14:15:29.
  - 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user Administrator اصنzor@#2536fkuj*. The process was started by *net.exe*, created at 14:15:32.
  - 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *"C:\windows\system32\net.exe" user Administrator*

اصنzor@#2536fkuj. The process was started by *powershell.exe*, created at 14:15:29.

- 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user Administrator اصنzor@#2536fkuj*. The process was started by *net.exe*, created at 14:15:32.
- 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *"C:\windows\system32\net.exe" user Administrator اصنzor@#2536fkuj*. The process was started by *powershell.exe*, created at 14:15:29.
- 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user Administrator اصنzor@#2536fkuj*. The process was started by *net.exe*, created at 14:15:32.
- 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *"C:\windows\system32\net.exe" user Administrator اصنzor@#2536fkuj*. The process was started by *powershell.exe*, created at 14:15:29.
- 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user Administrator اصنzor@#2536fkuj*. The process was started by *net.exe*, created at 14:15:32.
- 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *"C:\windows\system32\net.exe" user Administrator اصنzor@#2536fkuj*. The process was started by *powershell.exe*, created at 14:15:29.
- 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user Administrator اصنzor@#2536fkuj*. The process was started by *net.exe*, created at 14:15:32.
- 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process.

The executable was provided with the following arguments *"C:\windows\system32\net.exe" user Administrator اصنzor@#2536fkuj*. The process was started by *powershell.exe*, created at 14:15:29.

  - 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user Administrator اصنzor@#2536fkuj*. The process was started by *net.exe*, created at 14:15:32.
  - 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *"C:\windows\system32\net.exe" user Administrator اصنzor@#2536fkuj*. The process was started by *powershell.exe*, created at 14:15:29.
  - 29/06/2019 14:15:32: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user Administrator اصنzor@#2536fkuj*. The process was started by *net.exe*, created at 14:15:32.
  - 29/06/2019 14:15:32: The password of the administrator account was changed.

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:

  - 29/06/2019 14:17:46: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.ex* process. The executable was provided with the following arguments *C:\windows\system32\cmd.exe /c ""C:\Users\confluence\Desktop\mi\start.cmd" "*.

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable & 8. Executable → Cryptominer:

  - 29/06/2019 14:17:46: Event ID 4688 shows the creation of the *C:\Users\confluence\Desktop\mi\xmrig.ex* process. The executable was provided with the following arguments *xmrig.exe*. The process was started by *cmd.exe*, created at 14:17:46.
  - 29/06/2019 14:17:46: UserAssist shows the date and time of when *C:\Users\confluence\Desktop\mi\start.cmd* was last executed.
  - 29/06/2019 14:17:47: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\desktop\mi\xmrig.exe* to *94.23.247.226:5555*.

  - 29/06/2019 14:18:03: The *Processor Time Percentage* performance monitor increases to a hundred percent. It will stay a hundred percent until the system is shutdown because of the backup and restore process.

Classifications: PowerShell or Command Prompt Execution (2x) and Cryptominer
Thorough Search

- *Are all the given classifications correct?*
  Yes, a Command Prompt was used to change the password of the administrator account. A Command Prompt script was also used to start the process creation of an executable. The executable which was started by the cmd script was a Cryptominer. It made an outgoing connection and after the process started, the processor time percentage increased to a hundred percent and stayed a hundred percent.
- *Was the attack classified with too few classifications?*
  No other activity has been found.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

There were two separate sessions where different attackers were active. The backup and restore process was not triggered for this session because of the faulty Splunk query. This session did not influence the other sessions so both have been analyzed. The sessions originate from different IP addresses but they both perform exactly the same steps.

Image Name: osDisk-confluencetemp1-20190625-113600
Trigger: None
Login Time Frame: 25/06/2019 16:37:05 - 25/06/2019 17:00:49
IP Address: 89.238.185.222
Proposed Classification

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:
  - 25/06/2019 16:37:26: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process.
  - 25/06/2019 16:38:29: Event ID 4688 shows the creation of the *C:\Windows\System32\reg.exe* process. The executable was provided with the following arguments *REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v Debugger /t REG_SZ /d*

"C:\windows\system32\cmd.exe". The execution changes the "Sticky Keys" to a Command Prompt with elevated privileges, in other words by pressing the Shift key 5 times at the Windows Login Screen will start a Command Prompt with elevated privileges.
  - 25/06/2019 16:52:36: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe.*

Classifications: Powershell or Command Prompt Execution
Thorough Search

- *Are all the given classifications correct?*
  Yes, the Command Prompt was used to add a key to the registry.
- *Was the attack classified with too few classifications?*
  No other activity has been found.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190625-113600
Trigger: RDP session longer than 45 minutes
Login Time Frame: 25/06/2019 17:48:35 - 25/06/2019 17:49:55 & 25/06/2019 17:54:02 - 25/06/2019 17:55:45 & 25/06/2019 17:59:19 - 25/06/2019 18:35:12
IP Address: 88.99.67.34
Proposed Classification

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:
  - 25/06/2019 17:54:08: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process.
  - 25/06/2019 17:54:09: Event ID 4688 shows the creation of the *C:\Windows\System32\reg.exe* process. The executable was provided with the following arguments *REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v Debugger /t REG_SZ /d "C:\windows\system32\cmd.exe"*. The execution changes the "Sticky Keys" to a Command Prompt with elevated privileges, in other words by pressing the Shift key 5 times at the Windows Login Screen will start a Command Prompt with elevated privileges.
- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:

- 25/06/2019 17:54:23: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process.
- 25/06/2019 17:54:29: Event ID 4688 shows the creation of the *\Device\Mup\tsclient\D\Prog\RDP38\info\inf.exe* process. The executable was provided with the following arguments *\\tsclient\D\Prog\RDP38\info\inf.exe 865165397 2 132773 o1 3 0 195.208.10.114:31856 lynn:Kassword3* 1:1 52.142.195.42 20856878168 0 0 0 RD10-C6F8A96D-19-53-39-1B-2F-A0-1F-C0.* The executable is stored on a shared drive.

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:

  - 25/06/2019 17:54:29: Event ID 4688 shows the creation of the *\Device\Mup\tsclient\D\Prog\RDP38\info\inf.exe* process. The executable was provided with the following arguments *\\tsclient\D\Prog\RDP38\info\inf.exe 865165397 2 132773 o1 3 0 195.208.10.114:31856 lynn:Kassword3* 1:1 52.142.195.42 20856878168 0 0 0 RD10-C6F8A96D-19-53-39-1B-2F-A0-1F-C0.* The executable is stored on a shared drive.
  - 25/06/2019 17:55:12: Event ID 5156 shows an outbound connection being made by *\device\mup\tsclient\d\prog\rdp38\info\inf.exe* to *195.208.10.114:31856.*
  - 25/06/2019 17:55:17: A user account "lynn" was created. The user account was added to the following groups: Administrators and Remote Desktop Users
  - 25/06/2019 17:55:39: Event ID 5156 shows an outbound connection being made by *\device\mup\tsclient\d\prog\rdp38\info\inf.exe* to *104.73.129.194:80.*
  - 25/06/2019 17:55:40: Event ID 5156 shows an outbound connection being made by *\device\mup\tsclient\d\prog\rdp38\info\inf.exe* to *104.73.152.80:80.*

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 18. External → Single System:

  -

- 25/06/2019 18:21:46: Event ID 4688 shows the creation of the *C:\Windows\System32\mstsc.exe* process.
- 25/06/2019 18:21:54: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\windows\system32\mstsc.exe* to *95.158.37.200:54654*.
- 25/06/2019 18:21:58: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\windows\system32\mstsc.exe* to *95.158.37.200:54654*.
- 25/06/2019 18:21:59: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\windows\system32\mstsc.exe* to *95.158.37.200:54654*.
- 25/06/2019 18:21:59: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\windows\system32\mstsc.exe* to *95.158.37.200:54654*.
- 25/06/2019 18:22:00: Jump Lists show when *C:\Windows \System32*
  mstsc.exe was last accessed. The executable was provided with the following arguments */v:"95.158.37.200:54654*.

Classifications: Powershell or Command Prompt Execution (2x) and External Pivoting Single System
Thorough Search

- *Are all the given classifications correct?*
  Yes, a Command Prompt was used to change the Sticky Keys to a command prompt. Another Command Prompt is used to start an executable. The system also has been used as a pivot to make an RDP connection to an external system.

- *Was the attack classified with too few classifications?*
  Yes, the "inf.exe" binary was not classified. The binary created a user account and has made outbound connections to specific IP addresses. The proposed classification does not have a classification available for these kinds of binaries.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other activity has been found.

Image Name: osDisk-confluencetemp1-20190628-155626
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 29/06/2019 03:16:18 - 29/06/2019 03:27:27
IP Address: 5.121.104.73
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.
- *Was the attack classified with too few classifications?*
  No other activity has been found.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190714-190330
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 15/07/2019 04:30:19 - 15/07/2019 04:34:32 & 15/07/2019 04:34:41 - 15/07/2019 04:35:48 (session was terminated because of the backup and restore process)
IP Address: 89.238.178.214
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.
- *Was the attack classified with too few classifications?*
  No, only a folder containing files was transferred to the Desktop of the system. No further activities have been performed. After the transfer, the attacker disconnected the session. Shortly after the user disconnected, the attacker reconnected but the first session triggered the automated backup and restore process. Because of this, the second session was terminated. There is a possibility that the attacker had planes with the transferred files but this cannot be confirmed because the honeypot was shutdown.
- *Did the classification miss information which was important to determine what the attacker did?*
  Besides the transfer of files, no other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190715-113155
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 15/07/2019 23:31:04 - 15/07/2019 23:52:51
IP Address: 5.121.110.249
Proposed Classification

- 22. Reconnaissance → Destruction & 23. System Destruction → Deletion of Files and/or Folders:
  - 16/07/2019 23:32:19: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\config - backup.xml* was deleted.
  - 16/07/2019 23:32:19: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\7-Zip File Manager.lnk* was deleted.
  - 16/07/2019 23:32:19: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\maintenance scripts* was deleted.
  - 16/07/2019 23:32:19: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\maintenance scripts\remove-old-printjobs.ps1* was deleted.
  - 16/07/2019 23:32:19: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\maintenance scripts\find-errors.bat* was deleted.
  - 16/07/2019 23:32:19: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\maintenance scripts\clear-temp.bat* was deleted.
  - 16/07/2019 23:32:19: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\maintenance scripts\boot.bat* was deleted.
  - 16/07/2019 23:32:20: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\config - Copy (2).xml* was deleted.
  - 16/07/2019 23:32:21: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\Confluence.url* was deleted.
  - 16/07/2019 23:32:21: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\config.xml* was deleted.
  - 16/07/2019 23:32:21: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\WP-fifty-critical-alerts-for-win-servers.pdf* was deleted.
  - 16/07/2019 23:32:21: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\Windows Server Backup.lnk* was deleted.
  - 16/07/2019 23:32:21: Event ID 4663 with "DELETE" access shows that *C:\Users\confluence\Desktop\WinDirStat.lnk* was deleted.

- 16/07/2019 23:32:21: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\perftuningguideserver2012r2.pdf* was deleted.
- 16/07/2019 23:32:21: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\Notepad++.lnk* was deleted.
- 16/07/2019 23:32:21: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\Event Viewer.lnk* was deleted.
- 16/07/2019 23:32:21: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\dfrgui.lnk* was deleted.
- 16/07/2019 23:32:21: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\Confluence_5-9-1_CompleteGuide_PDF.pdf* was deleted.
- 16/07/2019 23:32:24: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\maintenance scripts* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\Windows Server Backup.lnk* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\WinDirStat.lnk* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\perftuningguideserver2012r2.pdf* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\Notepad++.lnk* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\Event Viewer.lnk* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\dfrgui.lnk* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\Confluence_5-9-1_CompleteGuide_PDF.pdf* was deleted.

- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\Confluence.url* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\config.xml* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\config - Copy (2).xml* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\config - backup.xml* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\7-Zip File Manager.lnk* was deleted.
- 16/07/2019 23:32:25: Event ID 4663 with “DELETE” access shows that *C:\Users\confluence\Desktop\WP-fifty-critical-alerts-for-win-servers.pdf* was deleted.
- 16/07/2019 23:32:19: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\config - backup.xml.*
- 16/07/2019 23:32:19: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\7-Zip File Manager.lnk.*
- 16/07/2019 23:32:19: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\maintenance scripts.*
- 16/07/2019 23:32:19: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\maintenance scripts\remove-old-printjobs.ps1.*
- 16/07/2019 23:32:19: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\maintenance scripts\find-errors.bat.*
- 16/07/2019 23:32:19: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\maintenance scripts\clear-temp.bat.*
- 16/07/2019 23:32:19: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\maintenance scripts\boot.bat.*
- 16/07/2019 23:32:20: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\config - Copy (2).xml.*
- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Confluence.url.*
- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\config.xml.*

- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\WP-fifty-critical-alerts-for-win-servers.pdf.*
- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Windows Server Backup.lnk.*
- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\WinDirStat.lnk.*
- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\perftuningguideserver2012r2.pdf.*
- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Notepad++.lnk.*
- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Event Viewer.lnk.*
- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\dfrgui.lnk.*
- 16/07/2019 23:32:21: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Confluence_ 5-9-1_ CompleteGuide _PDF.pdf.*
- 16/07/2019 23:32:24: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\maintenance scripts.*
- 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Windows Server Backup.lnk.*
- 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\WinDirStat.lnk.*
- 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\perftuningguideserver2012r2.pdf.*
- 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Notepad++.lnk.*
- 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Event Viewer.lnk.*
- 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\dfrgui.lnk.*
- 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Confluence_ 5-9-1_ CompleteGuide _PDF.pdf.*

  - 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\Confluence.url*.
  - 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\config.xml*.
  - 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\config - Copy (2).xml*.
  - 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\config - backup.xml*.
  - 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\7-Zip File Manager.lnk*.
  - 16/07/2019 23:32:25: The following file was moved to the Recycle bin *C:\Users\confluence\Desktop\WP-fifty-critical-alerts-for-win-servers.pdf*.

Classifications: Deletion of Files and/or Folders
Thorough Search

- *Are all the given classifications correct?*
  Yes, all files and folders which were on the Desktop were moved to the Recycle Bin.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190712-140509
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 12/07/2019 14:11:29 - 12/07/2019 14:14:06
IP Address: 106.206.41.230
Proposed Classification

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 16. External → Hiding Web Activity:

  - 12/07/2019 14:12:45: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 12/07/2019 14:12:46: UserAssist shows the date and time of when *Firefox.exe* was last executed from the taskbar.
  - 12/07/2019 14:13:29: Event ID 5156 shows an outbound connection being made by *Firefox* to *88.221.144.26:80*.
  - 12/07/2019 14:13:34: Event ID 5156 shows an outbound connection being made by *Firefox* to *67.199.248.11:80*.

- 12/07/2019 14:13:36: Event ID 5156 shows an outbound connection being made by *Firefox* to *93.184.220.29:80*.
- 12/07/2019 14:13:42: Event ID 5156 shows an outbound connection being made by *Firefox* to *172.217.17.131:80*.
- 12/07/2019 14:13:43: Event ID 5156 shows an outbound connection being made by *Firefox* to *67.199.248.11:80*.
- 12/07/2019 14:13:44: Firefox was used to access `http://bit.ly/32hMRrE`.
- 12/07/2019 14:13:46: Event ID 5156 shows an outbound connection being made by *Firefox* to *151.139.128.14:80*.
- 12/07/2019 14:13:47: Firefox was used to access `https://mirr.re/d/11k6`.
- 12/07/2019 14:13:48: Firefox was used to access `https://multifilemirror.com/d/11k6`.
- 12/07/2019 14:13:49: Event ID 5156 shows an outbound connection being made by *Firefox* to *93.184.220.29:80*.
- 12/07/2019 14:13:50: Firefox was used to access `https://multifilemirror.com/k0ljpcu467uf` to download *SpyMAX rar*.
- 12/07/2019 14:13:51: Event ID 5156 shows an outbound connection being made by *Firefox* to *172.217.17.131:80*.
- 12/07/2019 14:13:51: Event ID 5156 shows an outbound connection being made by *Firefox* to *151.139.128.14:80*.
- 12/07/2019 14:13:51: Event ID 5156 shows an outbound connection being made by *Firefox* to *172.217.17.131:80*.
- 12/07/2019 14:13:52: Event ID 5156 shows an outbound connection being made by *Firefox* to *80.239.137.121:80*.
- 12/07/2019 14:13:54: Event ID 5156 shows an outbound connection being made by *Firefox* to *151.139.128.14:80*.
- 12/07/2019 14:13:54: Event ID 5156 shows an outbound connection being made by *Firefox* to *151.139.128.14:80*.
- 12/07/2019 14:13:54: Event ID 5156 shows an outbound connection being made by *Firefox* to *151.139.128.14:80*.
- 12/07/2019 14:14:11: Event ID 5156 shows an outbound connection being made by *Firefox* to *80.239.137.121:80*.
- 12/07/2019 14:14:11: Event ID 5156 shows an outbound connection being made by *Firefox* to *80.239.137.121:80*.
- 12/07/2019 14:14:27: Event ID 5156 shows an outbound connection being made by *Firefox* to *88.221.144.26:80*.

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  Yes, the browser of the system was used to access a website where it was possible to download an Android RAT.

- *Was the attack classified with too few classifications?*
  No other activity has been found. The attackers' session was disconnected because another person connected to the system. The attacker did not download the file.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190715-173400
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 15/07/2019 17:44:43 - 15/07/2019 17:48:49 (disconnected by another user)
IP Address: 106.206.47.125
Proposed Classification

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 16. External → Hiding Web Activity:
  - 15/07/2019 17:45:05: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 15/07/2019 17:45:06: UserAssist shows the date and time of when *Firefox.exe* was last executed from the taskbar.
  - 15/07/2019 17:45:22: Event ID 5156 shows that Firefox creates 148 outbound connections starting from this point in time.
  - 15/07/2019 17:45:56: Firefox was used to access `https://www.google.com/search?client=firefox-b-d&q=two_hands31%40hotmail.com%3ACreative2321` (Google search for "two_hands31@hotmail.com:Creative2321).
  - 15/07/2019 17:46:11: Firefox was used to access `https://www.google.com/search?client=firefox-b-d&q=pornhub+live+login+us` (Google search for "pornhub live login us).
  - 15/07/2019 17:46:19: Firefox was used to access a pornographic website named Pornhub.
  - 15/07/2019 17:47:06: Firefox was used to access the account creation page of the pornographic website.
  - 15/07/2019 17:47:24: Firefox was used to access the login page of the pornographic website.

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  Yes, the browser of the system was used to access multiple URL's.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190627-161038
Trigger: RDP session longer than 45 minutes
Login Time Frame: 27/06/2019 17:45:00 - 27/06/2019 18:35:26
IP Address: 5.121.148.132
Proposed Classification

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:

  - 27/06/2019 17:45:19: UserAssist shows the date and time of when *cmd.exe* was last executed from the taskbar.
  - 27/06/2019 17:45:19: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process.
  - 27/06/2019 17:45:37: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *net user administrator der@17342*. The process was started by *cmd.exe*, created at 17:45:19.
  - 27/06/2019 17:45:37: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user administrator der@17342*. The process was started by *net.exe*, created at 17:45:37.
  - 27/06/2019 17:45:37: Last password change of the administrator account

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:

  - 27/06/2019 17:49:35: UserAssist shows the date and time of when *C:\Users\confluence\Desktop\NL\.exe* was last executed.
  - 27/06/2019 17:49:35: Event ID 4688 shows the creation of the *C:\Users\confluence\Desktop\NL\.exe* process.

- 27/06/2019 17:46:21: LNK Files show when a Windows shortcut file was created for file *C:\Users\confluence\Desktop\NL*.
- 27/06/2019 17:46:21: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\.exe*.
- 27/06/2019 17:47:11: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\newpasssh.txt*.
- 27/06/2019 17:47:11: LNK Files show when a Windows shortcut file was created for file *C:\Users\confluence\Desktop\NL\newpasssh.txt*.
- 27/06/2019 19:47:11: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\newpasssh.txt*.
- 27/06/2019 17:47:16: Jump Lists show when *C:\Users\confluence\Desktop\NL\user.txt* was last accessed.
- 27/06/2019 17:47:16: LNK Files show when a Windows shortcut file was created for file *C:\Users\confluence\Desktop\NL\user.txt*.
- 27/06/2019: Shell bags show when *NL* was last accessed.
- 27/06/2019 19:47:16: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\user.txt*.
- 27/06/2019 19:47:16: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\settings.ini*.
- 27/06/2019 17:49:01: LNK Files show when a Windows shortcut file was created for file *C:\Users\confluence\Desktop\NL\colom.txt*.
- 27/06/2019 19:49:01: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\IPs.txt*.
- 27/06/2019 19:49:22: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to

  C:\Users\confluence\Desktop\NL\user.txt.
  - 27/06/2019 19:49:53: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\settings.ini.*
  - 27/06/2019 19:49:53: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\servers.txt.*
  - 27/06/2019 19:49:53: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\credentials.txt.*
  - 27/06/2019 19:49:57: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\NL\settings.ini.*

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 17. External → :
  - 27/06/2019 17:49:53: Event ID 5156 shows that *\device\harddiskvolume2\users\confluence\desktop\nl\.exe* creates 5202 outbound connections to port 3389. Every connection is to a different IP address.

Classifications: PowerShell or Command Prompt Execution and External Pivoting Multiple Systems

Thorough Search

- *Are all the given classifications correct?*
  Yes, an executable created connections to the standard RDP port of other IP addresses. Based on the created files, it can be assumed that login attempts are made. Username and password possibilities are stored in the same folder as the executable. The system is used as a pivot to hide where the login attempts originate from. The Command Prompt was used to change the password of the administrator account.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190630-125301
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 30/06/2019 13:21:02 - 30/06/2019 13:28:12
IP Address: 220.188.176.165
Proposed Classification

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 16. External → Hiding Web Activity:
  - 30/06/2019 13:21:53: UserAssist shows the date and time of when *Firefox.exe* was last executed from the taskbar.
  - 30/06/2019 13:21:54: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 30/06/2019 13:21:58: Event ID 5156 shows that Firefox creates 116 outbound connections starting from this point in time.
  - 30/06/2019 13:22:54: Firefox was used to access `https://www.google.com/search?client=firefox-b-d&q=airbnb` (Google search for "airbnb).
  - 30/06/2019 13:23:08: Firefox was used to access `https://www.airbnb.nl/s/homes?allow_override%5B%5D=&ib=true&s_tag=xXA0C5PT&af=43720035&c=.pi0.pk5008905443_345318717115_c_12026464216&sem_position=1t1&sem_target=kwd-12026464216&location_of_interest=&location_physical=1010543&ghost=true&gclid=EAIaIQobChMIvJmdxqaR4wIVzZ13Ch3h6QvlEAAYASABEgKsi_D_BwE`.
  - 30/06/2019 13:23:16: Firefox was used to access `https://www.airbnb.nl/s/homes?allow_override%5B%5D=&s_tag=xXA0C5PT&af=43720035&c=.pi0.pk5008905443_345318717115_c_12026464216&sem_position=1t1&sem_target=kwd-12026464216&location_of_interest=&location_physical=1010543&ghost=true&gclid=EAIaIQobChMIvJmdxqaR4wIVzZ13Ch3h6QvlEAAYASABEgKsi_D_BwE&ib=true&refinement_paths%5B%5D=%2Fhomes`.
  - 30/06/2019 13:27:58: Firefox was used to access `https://www.airbnb.nl/account`.
  - 30/06/2019 13:27:58: Firefox was used to access `https://www.airbnb.nl/users/notifications`.
  - 30/06/2019 13:28:06: Firefox was used to access `https://www.airbnb.nl/airlock?al_id=4166996063`.

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  Yes, the system was used to access multiple URL's.

- *Was the attack classified with too few classifications?*
  No, besides the browser activty no other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190714-121244
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 14/07/2019 15:36:07 - 14/07/2019 15:37:46
IP Address: 157.51.64.64
Proposed Classification

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 16. External → Hiding Web Activity:
  - 14/07/2019 15:36:46: UserAssist shows the date and time of when *Firefox.exe* was last executed from the taskbar.
  - 14/07/2019 15:36:46: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 14/07/2019 15:36:50: Event ID 5156 shows that Firefox creates 64 outbound connections starting from this point in time.
  - 14/07/2019 15:37:15: Firefox was used to access `https://www.google.com/`.
  - 14/07/2019 15:37:29: Firefox was used to access `https://whatleaks.com`.
  - 14/07/2019 15:37:24: Firefox was used to access `https://www.google.com/search?source=hp&ei=KkwrXbr4OMGzkwXxzLjoCw&q=whatleaks&oq=whatleaks&gs_I=psy-ab.3..0j0i30.4550.6997..7947...0.0..0.55.407.9......0....1..gws-wiz.....0..0i131j0i10j0i10i30.oT9KZQ4_A8w` (Google search for "whatleaks).

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  Yes, the browser of the system was used to access multiple URL's.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190723-203232
Trigger: RDP session longer than 45 minutes
Login Time Frame: 23/07/2019 21:30:57 - 23/07/2019 22:21:25
IP Address: 192.96.203.152
Proposed Classification

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 16. External → Hiding Web Activity:
  - 23/07/2019 21:31:06: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 23/07/2019 21:31:07: UserAssist shows the date and time of when *Firefox.exe* was last executed from the taskbar.
  - 23/07/2019 21:36:26: Event ID 5156 shows that Firefox creates 151 outbound connections starting from this point in time.
  - 23/07/2019 21:32:19: Firefox was used to access `https://artstudiorental.com/wp-content/plugins/sdf.php`.
  - 23/07/2019 21:32:38: Firefox was used to access `https://colortexperu.com.pe/wp-content/uploads/2019/-7/sdf.php`.
  - 23/07/2019 21:33:13: Firefox was used to access `https://www.google.com/`.
  - 23/07/2019 21:33:14: Firefox was used to access `https://www.google.com/#cns=1`.
  - 23/07/2019 21:33:56: Firefox was used to access `https://www.google.com/#cns=0`.
  - 23/07/2019 21:34:14: Firefox was used to access `https://www.google.com/search?source=hp&ei=GX03XdzyL5HSkgXDiam4BA&q=cashapp&oq=cashapp&gs_l=psy-ab.3..0l2j0i10j0j0i10j0j0i10j0l3.56087.58208..59684...0.0..0.50.263.7......0....1..gws-wiz.....0..0i131.5AXJsQzllhk&ved=0ahUKEwic44qF_8vjAhURqaQKHcNECkcQ4dUDCAU&uact=5` (Google search for "cashapp).
  - 23/07/2019 21:34:27: Firefox was used to access `https://cash.app/`.
  - 23/07/2019 21:34:43: Firefox was used to access `https://cash.app/account`.

  - 23/07/2019 21:34:47: Firefox was used to access `https://cash.app/login?return_to=account.index`.

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  Yes, the browser of the system was used to access multiple websites, including a payment service.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190625-183650
Trigger: RDP session between 90 seconds and 45 minutes
Login Time Frame: 25/06/2019 20:37:56 - 25/06/2019 20:44:03 & 25/06/2019 20:44:57 - 25/06/2019 20:45:19 (session terminated because of the backup and restore process)
IP Address: 220.188.251.16
Proposed Classification

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 16. External → Hiding Web Activity:
  - 25/06/2019 20:38:51: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 25/06/2019 20:40:03: UserAssist shows the date and time of when *C:\Public\Desktop\Firefox.lnk* was last executed.
  - 25/06/2019 20:40:09: Event ID 5156 shows that Firefox creates 73 outbound connections starting from this point in time.
  - 25/06/2019 20:41:03: Firefox was used to access `https://www.google.com/search?client=firefox-b-d&q=paypal` (Google search for "paypal).
  - 25/06/2019 20:41:09: Firefox was used to access `https://www.paypal.com/nl/signin`.
  - 25/06/2019 20:41:54: Firefox was used to access `https://www.paypal.com/authflow/safe?returnUri=signin&country.x=NL&locale.x=nl_NL`.
  - 25/06/2019 20:41:55: Firefox was used to access `https://www.paypal.com/authflow/safe/?returnUri=signin&country.x=NL&locale.x=nl_NL`.

  - 25/06/2019 20:42:41: Firefox was used to access `https://www.paypal.com/authflow/entry/?clientInstanceId=115f87de-71f7-4aed-a5fd-6869e48aeabe&redirectUri=%2Fsignin&country.x=NL&locale.x=nl_NL`.
  - 25/06/2019 20:43:11: Firefox was used to access `https://www.paypal.com/authflow/challenges/securityQuestions/?clientInstanceId=115f87de-71f7-4aed-a5fd-6869e48aeabe&country.x=NL&locale.x=nl_NL&redirectUri=%2Fsignin`.

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  Yes, the browser of the system was used to access multiple websites, including a payment service.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190716-131759
Trigger: Standard RDP
Login Time Frame: 16/07/2019 13:22:38 - 16/07/2019 13:34:03
IP Address: 188.159.100.184
Proposed Classification

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:

  - 16/07/2019 13:23:31: Event ID 4688 shows the creation of the *C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe* process.
  - 16/07/2019 13:23:32: UserAssist shows the date and time of when *PowerShell* was last executed.
  - 16/07/2019 13:25:12: User ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *"C:\windows\system32\net.exe" user administrator saeed@1q2*. The process was started by *powershell.exe*, created at 13:23:31.
  - 16/07/2019 13:25:12: User ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user administrator saeed@1q2*. The process was started by *net.exe*, created at 13:25:12.

  - 16/07/2019 13:25:13: The password of the administrator account was changed.

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:
  - 16/07/2019 13:32:18: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to *C:\Users\confluence\Downloads\5-NS new.exe.*
  - 16/07/2019 13:32:30: Event ID 4688 shows the creation of the *C:\Users\confluence\Downloads\5-NS new.exe* process.
  - 16/07/2019 13:32:30: UserAssist shows the date and time of when *C:\Users\confluence\Downloads\5-NS new.exe* was last executed.
  - 16/07/2019 13:32:30: AmCache shows that *C:\Users\confluence\Downloads\5-NS new.exe* was executed. The SHA1 hash of the executable is *bd5354cd813130687c66ecc132d50770d48417cf.*
  - 16/07/2019 13:32:34: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\cmd.exe* process. The executable was provided with the following arguments *C:\windows\system32\cmd.exe /c cls.* The process was started by *5-NS new.exe*, created at 13:32:30.

- 14. Reconnaissance → Pivoting & 19. Pivoting → Internal & 20. Internal → Multiple Systems:
  - 16/07/2019 13:32:39: Event ID 5152 shows 151 outbound connections to the local IP range 10.0.0.1 up till 10.0.0.152 (excluding its own IP address 10.0.0.4) originating from the *System* application.

Classifications: PowerShell or Command Prompt Execution and Internal Pivoting to Multiple Systems
Thorough Search

- *Are all the given classifications correct?*
  Yes, PowerShell was used to change the password of the administrator account. In a time window of 1 minute and 14 seconds, 151 outbound connections have been made to unique local IP addresses. The outbound connections shortly started after the execution of an executable created by the attacker. The outbound connections stopped when the process of the executable exited.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190711-130651
Trigger: RDP session longer than 45 minutes
Login Time Frame: 11/07/2019 14:13:17 - 11/07/2019 15:02:08
IP Address: 41.249.26.16
Proposed Classification

- 14. Reconnaissance → Pivoting & 19. Pivoting → Internal & 20. Internal → Hiding Web Activity:
  - 11/07/2019 14:16:28: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 11/07/2019 14:16:28: UserAssist shows the date and time of when *Firefox.exe* was last executed from the taskbar.
  - 11/07/2019 14:16:33: Event ID 5156 shows that Firefox creates 65 outbound connections starting from this point in time.
- 14. Reconnaissance → Pivoting & 19. Pivoting → Internal & 20. Internal → Hiding Web Activity:
  - 11/07/2019 14:18:27: Event ID 4688 shows the creation of the *C:\Program Files (x86)\Google\Chrome\Application\chrome.exe* process.
  - 11/07/2019 14:18:28: Event ID 5156 shows that Chrome creates 74 outbound connections starting from this point in time

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  No, Firefox was used to download the setup for the Chrome browser. After the file was downloaded, Chrome was installed as the default browser. This whole process met all the requirements for the classification but in the end, besides the installation of Chrome, no web browsing has been done.
- *Was the attack classified with too few classifications?*
  No other activity has been found.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190718-091249
Trigger: Standard RDP
Login Time Frame: 18/07/2019 09:37:16 - 18/07/2019 09:39:33
IP Address: 37.237.77.10
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.
- *Was the attack classified with too few classifications?*
  Besides opening PuTTY once, no other activity has been found..
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190723-044035
Trigger: Standard RDP
Login Time Frame: 23/07/2019 04:46:15 - 23/07/2019 04:49:57 & 23/07/2019 04:50:05 - 23/07/2019 04:51:22 (session was terminated because of the backup and restore process)
IP Address: 36.73.137.67
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.
- *Was the attack classified with too few classifications?*
  No other activity has been found.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190628-102113
Trigger: RDP session longer than 45 minutes
Login Time Frame: 28/06/2019 11:19:01 - 28/06/2019 12:04:34
IP Address: 188.201.8.170
Proposed Classification

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:
  - 28/06/2019 11:19:55: MRU Run Commands show when the *cmd* command was used in the Run utility of Windows.
  - 28/06/2019 11:19:55: UserAssist shows the date and time of when *cmd.exe* was last executed.
  - 28/06/2019 11:19:55: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process.
  - 28/06/2019 11:20:19: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *net user administrator der@17342*. The process was started by *cmd.exe*, created at 11:19:55.
  - 28/06/2019 11:20:19: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user administrator der@17342*. The process was started by *net.exe*, created at 11:20:19.
  - 28/06/2019 11:20:19: The password of the administrator account was changed.
- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:
  - 28/06/2019 11:20:41: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to *C:\Users\confluence\Desktop\*.
  - 28/06/2019 11:21:34: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to *C:\Users\confluence\Desktop\NL\newpasssh.txt*.
  - 28/06/2019 11:21:38: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to *C:\Users\confluence\Desktop\NL\settings.ini*.
  - 28/06/2019 11:21:39: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to *C:\Users\confluence\Desktop\NL\user.txt*.
  - 28/06/2019 11:22:59: Shell bags show when *C:\Users\confluence\Desktop\NL* was last accessed.

  - 28/06/2019 11:23:29: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to *C:\Users\confluence\Desktop\NL\Part7.txt*.
  - 28/06/2019 11:23:57: Event ID 4688 shows the creation of the *C:\Users\confluence\Desktop\NL\.exe* process.
  - 28/06/2019 11:24:37: UserAssist shows the date and time of when *C:\Users\confluence\Desktop\NL\.exe* was last executed.
  - 28/06/2019 11:24:50: Jump Lists show when *C:\Users\confluence\Desktop\NL" was last accessed* was last accessed.
  - 28/06/2019 11:24:50: LNK Files show when a Windows shortcut file was created for file *C:\Users\confluence\Desktop\NL\as.txt*.
  - 28/06/2019 11:24:53: LNK Files show when a Windows shortcut file was created for file *C:\Users\confluence\Desktop\NL\user.txt*.
  - 28/06/2019 11:24:56: LNK Files show when a Windows shortcut file was created for file *C:\Users\confluence\Desktop\NL*.
  - 28/06/2019 11:24:56: MRU Recent Files & Folders shows when *C:\Users\confluence\Desktop\NL* was last opened or saved.
  - 28/06/2019 11:24:56: LNK Files show when a Windows shortcut file was created for file *C:\Users\confluence\Desktop\NL\newpasssh.txt*.
  - 28/06/2019 11:24:56: MRU Recent Files & Folders shows when *C:\Users\confluence\Desktop\NL\newpasssh.txt* was last opened or saved.
  - 28/06/2019 11:24:50: Edge/IE shows that the following file was accessed `file:///C:/Users/confluence/Desktop/NL/as.txt`.
  - 28/06/2019 11:24:53: Edge/IE shows that the following file was accessed `file:///C:/Users/confluence/Desktop/NL/user.txt`.
  - 28/06/2019 11:24:56: Edge/IE shows that the following file was accessed `file:///C:/Users/confluence/Desktop/NL/newpasssh.txt`.

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 17. External → Multiple Systems:

  - 28/06/2019 11:25:06: Event ID 5156 shows that *C:\Users\confluence\Desktop\NL\.exe* creates 38979 outbound connections to port 3389. Every connection is to a different IP address.

Classifications: PowerShell or Command Prompt Execution and External Pivoting Multiple Systems
Thorough Search

- *Are all the given classifications correct?*
  Yes, a Command Prompt was used to change the password of the administrator account. An executable was used to make connections to multiple external systems to the standard Remote Desktop port.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190711-042216
Trigger: Standard RDP
Login Time Frame: 11/07/2019 05:42:48 - 11/07/2019 06:08:04 & 11/07/2019 06:08:53 - 11/07/2019 06:09:27 (session was terminated because of the backup and restore process)
IP Address: 89.39.107.205
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190712-132216
Trigger: Standard RDP
Login Time Frame: 12/07/2019 13:26:12 - 12/07/2019 13:42:47
IP Address: 106.206.41.230
Proposed Classification

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:

  - 12/07/2019 13:26:53: UserAssist shows the date and time of when *Firefox.lnk* was last executed from the taskbar.
  - 12/07/2019 13:26:53: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 12/07/2019 13:27:50: UserAssist shows the date and time of when *cmd.exe* was last executed from the taskbar.
  - 12/07/2019 13:27:50: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process.
  - 12/07/2019 13:28:00: Event ID 4688 shows the creation of the *C:\Windows\System32\net.ex* process. The executable was provided with the following arguments *net user administrator 9066.* The process was started by *cmd.exe*, created at 13:27:50.
  - 12/07/2019 13:28:00: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.ex* process. The executable was provided with the following arguments *C:\windows\system32\net1 user administrator 9066.* The process was started by *net.exe*, created at 13:28:00.
  - 12/07/2019 13:28:00: The password of the administrator account was changed.

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:

  - 12/07/2019 13:28:03: Firefox was used to access `https://github-production-release-asset-2e65be.s3.amazonaws.com/152125527/c2744080-86d7-11e9-92e5-5bebf1ad9d46?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20190712%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190712T132322Z&X-Amz-Expires=300&X-Amz-Signature=4632e99822b0ef505e6f3616935d9d44366eb199a1e37a986ddc04801a95823d&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3DWatson_Net45.exe&response-content-type=application%2Foctet-stream` (page title "Watson_Net45.exe") ("Watson is a .NET tool designed to enumerate missing KBs and suggest exploits for useful Privilege Escalation vulnerabilities.
  - 12/07/2019 13:28:04: Shimcache shows when the key of *SYSVOL\Users\confluence\Downloads\Watson_Net45.exe* was last updated in the Shimcache. The execution flag of the executable is *False.*

- 12/07/2019 13:28:10: Firefox was used to acces `https://github-production-release-asset-2e65be.s3.amazonaws.com/152125527/c30cd700-86d7-11e9-9a90-f0ba423e9def?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20190712%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190712T132318Z&X-Amz-Expires=300&X-Amz-Signature=54f228a0994bc36495bc59b57400b196b340f5912d0c16e3fd12f0d21adffdd6&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3DWatson_Net35.exe&response-content-type=application%2Foctet-stream` (page title "Watson_Net35.exe") ("Watson is a .NET tool designed to enumerate missing KBs and suggest exploits for useful Privilege Escalation vulnerabilities.
- 12/07/2019 13:28:10: Shimcache shows when the key of *SYSVOL\Users\confluence\Downloads\Watson_Net35.exe* was last updated in the Shimcache. The execution flag of the executable is *True*.
- 12/07/2019 13:28:12: Shell bags show when *My Computer:C:\Users\confluence\Downloads\* was last accessed.
- 12/07/2019 13:29:00: Event ID 4688 shows the creation of the *C:\Users\confluence\Downloads\Watson_Net35.exe* process.
- 12/07/2019 13:29:00: Event ID 4688 shows the creation of the *C:\Windows\System32\Fondue.ex* process. The executable was provided with the following arguments *"C:\windows\system32\fondue.exe" /enable-feature:NetFx3 /caller-name:mscoreei.dll*. The process was started by *Watson_Net35.exe*, created at 13:29:00.
- 12/07/2019 13:30:12: Event ID 4688 shows the creation of the *C:\Users\confluence\Downloads\Watson_Net35.exe* process.
- 12/07/2019 13:30:12: Event ID 4688 shows the creation of the *C:\Windows\System32\Fondue.ex* process. The executable was provided with the following arguments *"C:\windows\system32\fondue.exe" /enable-feature:NetFx3 /caller-name:mscoreei.dll*. The process was started by *Watson_Net35.exe*, created at 13:30:12.
- 12/07/2019 13:38:15: Event ID 4688 shows the creation of the *C:\Users\confluence\Downloads\Watson_Net35.exe* process.
- 12/07/2019 13:38:15: Event ID 4688 shows the creation of the *C:\Windows\System32\Fondue.ex* process. The

executable was provided with the following arguments *"C:\windows\system32\fondue.exe" /enable-feature:NetFx3 /caller-name:mscoreei.dll.* The process was started by *Watson_Net35.exe*, created at 13:38:15.

Classifications: PowerShell or Command Prompt Execution
Thorough Search

- *Are all the given classifications correct?*
  Yes, a Command Prompt was used to change the password of the administrator account.

- *Was the attack classified with too few classifications?*
  No, but that is because the server was shutdown by the backup and restore process before any other activites could have been performed. The attacker installed multiple services on the system with the help of the "Add Roles and Features Wizard". Multiple services from multiple categories have been installed. Examples of categores where services were installed from are: "Application Server", "Remote Access", ".NET Framework 3.5 Features", "NET Framework 4.5 Features" and "Web Server". Because the server was shutdown before the installation was complete, it is unknown what the purpose of the installation of the services was.

- *Did the classification miss information which was important to determine what the attacker did?*
  Yes, even though it is unknown what the purpose of the newly installed services was, it shows that standard Windows functionality can be used to facilitate malicious behaviour. This is not clearly mentioned in the proposed classification. Standard Windows processes and file creation of these processes should be taken into account.

Image Name: osDisk-confluencetemp6-20190722-132356
Trigger: Standard RDP
Login Time Frame: 22/07/2019 13:39:38 - 22/07/2019 13:42:45
IP Address: 36.73.137.67
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190724-170610
Trigger: Standard RDP
Login Time Frame: 25/07/2019 01:55:46 - 25/07/2019 01:58:52
IP Address: 36.73.137.67
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.
- *Was the attack classified with too few classifications?*
  No other activity has been found.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190720-063730
Trigger: Standard RDP
Login Time Frame: 20/07/2019 17:44:44 - 20/07/2019 17:47:04
IP Address: 223.186.137.151
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.
- *Was the attack classified with too few classifications?*
  No other activity has been found.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp6-20190725-194521
Trigger: RDP session longer than 45 minutes
Login Time Frame: 25/07/2019 22:15:21 - 25/07/2019 23:05:15
IP Address: 129.205.112.227
Proposed Classification

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 16. External → Hiding Web Activity:
  - 25/07/2019 22:15:42: UserAssist shows the date and time of when *Firefox.exe* was last executed from the taskbar.
  - 25/07/2019 22:15:42: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 25/07/2019 22:15:57: Event ID 5156 shows that Firefox creates 150 outbound connections starting from this point in time.
  - 25/07/2019 22:16:41: Firefox was used to access `http://www.godaddy.com/`.
  - 25/07/2019 22:16:42: Firefox was used to access `https://nl.godaddy.com/`.
  - 25/07/2019 22:16:55: Firefox was used to access `https://sso.godaddy.com/account/create?realm=idp&path=%2Fproducts&app=account`.
  - 25/07/2019 22:17:07: Firefox was used to access `https://sso.godaddy.com/?realm=idp&path=%2Fproducts&app=account`.
  - 25/07/2019 22:17:17: Firefox was used to access `https://account.godaddy.com/products`.
  - 25/07/2019 22:17:18: Firefox was used to access `https://account.godaddy.com/products/`.
  - 25/07/2019 22:17:40: Firefox was used to access `http://shortener.godaddy.com/`.
  - 25/07/2019 22:17:41: Firefox was used to access `https://shortener.godaddy.com/`.
  - 25/07/2019 22:17:57: Firefox was used to access `https://nl.godaddy.com/`.
  - 25/07/2019 22:17:57: Firefox was used to access `https://sso.godaddy.com/logout?realm=idp&app=shortener&page=%2F`.
  - 25/07/2019 22:17:57: Firefox was used to access `https://www.godaddy.com/`.

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  Yes, the browser of the system was used to access the website of Go Daddy. The attacker logged into an account and accessed multiple areas of the website.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190708-211449
Trigger: Standard RDP
Login Time Frame: 09/07/2019 04:38:49 - 09/07/2019 05:21:44
IP Address: 139.28.218.220
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190722-080211
Trigger: RDP session longer than 45 minutes
Login Time Frame: 22/07/2019 10:05:18 - 22/07/2019 10:55:12
IP Address: 41.203.78.64
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.

- *Was the attack classified with too few classifications?*
  No other activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190703-054200
Trigger: RDP session longer than 45 minutes
Login Time Frame: 03/07/2019 06:46:26 - 03/07/2019 07:35:13
IP Address: 181.143.153.74
Proposed Classification

- 1. Reconnaissance → Payload Execution & 2. Payload Execution → PowerShell or Command Prompt Execution:
  - 03/07/2019 06:47:08: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process.
  - 03/07/2019 06:47:23: Event ID 4688 shows the creation of the *C:\Windows\System32\cmd.exe* process.
  - 03/07/2019 06:47:23: UserAssist shows the date and time of when *cmd.exe* was last executed from the taskbar.
  - 03/07/2019 06:47:37: Event ID 4688 shows the creation of the *C:\Windows\System32\net.exe* process. The executable was provided with the following arguments *net user administrator Amane5745452*. The process was started by *cmd.exe*, created at 06:47:23.
  - 03/07/2019 06:47:37: Event ID 4688 shows the creation of the *C:\Windows\System32\net1.exe* process. The executable was provided with the following arguments *C:\windows\system32\net1 user administrator Amane5745452*. The process was started by *net.exe*, created at 06:47:37.
- 03/07/2019 06:47:08

Classifications: PowerShell or Command Prompt Execution
Thorough Search

- *Are all the given classifications correct?*
  Yes, a Command Prompt was used to change the password of the administrator account.
- *Was the attack classified with too few classifications?*
  No other activity has been found.
- *Did the classification miss information which was important to determine what the attacker did?*
  No other relevant artifacts have been found.

Image Name: osDisk-confluencetemp1-20190706-205945
Trigger: Bandwidth threshold exceeded
Login Time Frame: 06/07/2019 23:38:03 - 07/07/2019 00:09:11
IP Address: 5.121.129.241
Proposed Classification

- 14. Reconnaissance → Pivoting & 15. Pivoting → External & 16. External → Hiding Web Activity:
  - 06/07/2019 23:38:34: Event ID 4688 shows the creation of the *C:\Program Files\Mozilla Firefox\firefox.exe* process.
  - 06/07/2019 23:38:35: UserAssist shows the date and time of when *C:\Users\Public\Desktop\Firefox.lnk* was last executed.
  - 06/07/2019 23:38:50: Event ID 5156 shows that Firefox creates 197 outbound connections starting from this point in time.
  - 06/07/2019 23:40:07: Firefox was used to access `https://www.google.com/search?client=firefox-b-d&q=myips.msc` (Google search for "myips.msc").
  - 06/07/2019 23:40:29: Firefox was used to access `https://myip.ms/view/ip_addresses/3413360760/203.115.192.120_203.115.192.255`.
  - 06/07/2019 23:40:49: Firefox was used to access `https://www.google.com/search?client=firefox-b-d&q=myips.msc#cns=1` (Google search for "myips.msc").
  - 06/07/2019 23:40:58: Firefox was used to access `https://myip.ms/view/ip_addresses/203.115.197.0`.
  - 06/07/2019 23:41:02: Firefox was used to access `https://try.wrike.com/resource-management/?targetID=&ga_campaign=(ROI)+GDN+Standard+PSA+-+NA+-+CA&ga_adgroup=Custom+Affinity&ga_keyword=myip.ms&gclid=EAIaIQobChMIr4qB8ruh4wIVEweLCh08MQaGEAEYASAAEgLRAPD_BwE`.
  - 06/07/2019 23:41:26: Firefox was used to access `https://www.google.com/search?client=firefox-b-d&q=myips.msc#cns=0` (Google search for "myips.msc").
  - 06/07/2019 23:41:26: Firefox was used to access `https://consent.google.com/done3?continue=https://www.google.com/search?client%3Dfirefox-b-d%26q%3Dmyips.msc%23cns%3D1&origin=https://www.google.com&gl=NL&pc=s`.
- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:
  - 07/07/2019 00:04:54: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\Mas exe\masscan.exe.
  - 07/07/2019 00:04:54: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\Mas exe\input.txt.

- 07/07/2019 00:04:56: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\Mas exe\Massscan_GUI.exe.
- 07/07/2019 00:04:57: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\Mas exe\msvcr100.dll.
- 07/07/2019 00:04:59: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\Mas exe\Output.txt.
- 07/07/2019 00:05:04: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\Mas exe\winpcap-4.13.exe.
- 07/07/2019 00:05:04: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\Mas exe\Packet.dll.
- 07/07/2019 00:05:06: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\Mas exe\wpcap.dll.
- 07/07/2019 00:05:07: Event ID 4663 with “WriteData (or AddFile)” access shows that data is written to \textitC:\Users\confluence\Desktop\Mas exe_config.ini.
- 07/07/2019 00:05:10: Shell bags show when *Mas exe*\ was last accessed.
- 07/07/2019 00:05:14: UserAssist shows the date and time of when *C:\Users\confluence\Desktop\Mas exe\winpcap-4.13.exe* was last executed.
- 07/07/2019 00:05:14: Event ID 4688 shows the creation of the *C:\Users\confluence\Desktop\Mas exe\winpcap-4.13.exe* process.
- 07/07/2019 00:05:14: AmCache shows that *C:\Users\confluence\Desktop\Mas exe\winpcap-4.13.exe* was executed. The SHA1 hash of the executable is *de7993f716ed7f8a38f0340c32ec7b13e3cf86ed.*
- 07/07/2019 00:05:15: Shimcache shows when the key of *SYSVOL\Windows\svchost.com* was last updated in the Shimcache. The execution flag of the executable is *True.*
- 07/07/2019 00:05:15: Shimcache shows when the key of *SYSVOL\Users\confluence\AppData\Local\Temp\2\3582-490\winpcap-4.13.exe* was last updated in the Shimcache. The execution flag of the executable is *True.*

  - 07/07/2019 00:05:15: Event ID 4688 shows the creation of the *C:\Users\confluence\AppData\Local\Temp\2\3582-490\winpcap-4.13.exe* process. The process was started by *winpcap-4.13.exe*, created at 00:05:14.
  - 07/07/2019 00:05:15: AmCache shows that *C:\Users\confluence\AppData\Local\Temp\2\3582-490\winpcap-4.13.exe* was executed. The SHA1 hash of the executable is *3ad99ec2bf6cc4f947bb09be627c91f82a898aa8*.
  - 07/07/2019 00:05:17: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\net.exe* process. The executable was provided with the following arguments *net stop npf*. The process was started by *winpcap-4.13.exe*, created at 00:05:15.
  - 07/07/2019 00:05:18: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\net1.exe* process. The executable was provided with the following arguments *C:\windows\system32\net1 stop npf*. The process was started by *net.exe*, created at 00:05:17.
  - 07/07/2019 00:05:22: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\net.exe* process. The executable was provided with the following arguments *net start npf*. The process was started by *winpcap-4.13.exe*, created at 00:05:15.
  - 07/07/2019 00:05:22: Event ID 4688 shows the creation of the *C:\Windows\SysWOW64\net1.exe* process. The executable was provided with the following arguments *C:\windows\system32\net1 start npf*. The process was started by *net.exe*, created at 00:05:22.

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:

  - 07/07/2019 00:05:27: Shimcache shows when the key of *SYSVOL\Users\confluence\Desktop\Mas exe\Massscan_GUI.exe* was last updated in the Shimcache. The execution flag of the executable is *True*.
  - 07/07/2019 00:05:27: AmCache shows that *C:\Users\confluence\Desktop\Mas exe\Massscan_GUI.exe* was executed. The SHA1 hash of the executable is *95cda3e93db7b017b21e17fc34d7eb96ebc00563*.
  - 07/07/2019 00:05:27: Event ID 4688 shows the creation of the *C:\Windows\svchost.com* process. The executable was provided with the following arguments *"C:\windows\svchost.com" "C:\Users\confluence\Desktop\Mas exe\Massscan_GUI.exe"*.

  - 07/07/2019 00:05:27: Event ID 4688 shows the creation of the *C:\Users\confluence\Desktop\Mas exe\Massscan_GUI.exe* process. The process was started by *svchost.com*, created at 00:05:27.
  - 07/07/2019 00:05:31: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\desktop\mas exe\massscan_gui.exe* to *192.241.240.22:80.*
  - 07/07/2019 00:05:43: Event ID 5156 shows an outbound connection being made by *\device\harddiskvolume2\users\confluence\desktop\mas exe\massscan_gui.exe* to *192.241.240.22:80.*
  - 07/07/2019 00:05:49: Shimcache shows when the key of *SYSVOL\Users\confluence\Desktop\Mas exe\masscan.exe* was last updated in the Shimcache. The execution flag of the executable is *True*.
  - 07/07/2019 00:05:49: AmCache shows that *C:\Users\confluence\Desktop\Mas exe\masscan.exe* was executed. The SHA1 hash of the executable is *5f14241aea174608a7c85127fdad042d7382277d.*
  - 07/07/2019 00:05:49: Event ID 4688 shows the creation of the *C:\Windows\svchost.com* process. The executable was provided with the following arguments *"C:\windows\svchost.com" "C:\Users\confluence\Desktop\Mas exe\masscan.exe" -iL Input.txt -oL Output.txt –open –rate 100000000 -p3389 –exclude 255.255.255.255 –open-only.* The process was started by *Massscan_GUI.exe*, created at 00:05:27.
  - 07/07/2019 00:05:49: Event ID 4688 shows the creation of the *C:\Users\confluence\Desktop\Mas exe\masscan.exe* process. The executable was provided with the following arguments *"C:\Users\confluence\Desktop\Mas exe\masscan.exe" -iL Input.txt -oL Output.txt –open –rate 100000000 -p3389 –exclude 255.255.255.255 –open-only.* The process was started by *svchost.com*, created at 00:05:49.
  - 07/07/2019 00:05:51: Event ID 4663 with "WriteData (or AddFile)" access shows that data is written to *C:\Users\confluence\Desktop\Mas exe\Output.txt.*

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:

  - 07/07/2019 00:07:11: Event ID 4688 shows the creation of the *C:\Windows\svchost.com* process. The executable was provided with the following arguments *"C:\windows\svchost.com" "C:\windows\system32\cmd.exe"*.

  - 07/07/2019 00:07:11: UserAssist shows the date and time of when *cmd.exe* was last executed from the taskbar.

- 1. Reconnaissance → Payload Execution & 3. Payload Execution → Executable:

  - 07/07/2019 00:08:48: Shimcache shows when the key of *SYSVOL\Users\confluence\Desktop\The God.exe* was last updated in the Shimcache. The execution flag of the executable is *True*.
  - 07/07/2019 00:08:48: AmCache shows that *C:\Users\confluence\Desktop\The God.exe* was executed. The SHA1 hash of the executable is *599e199273484d944ab76725e3ca6fbdcba42087*.

Classifications: Hiding Web Activity
Thorough Search

- *Are all the given classifications correct?*
  Yes, the browser of the system was used to access multiple URL's. No activity related to the second execution of *svchost.com* was found. Activity related to the execution of *The God.exe* also was not found, but this was probably because of the honeypot being shutdown.

- *Was the attack classified with too few classifications?*
  Yes, WinPcap is installed to be able to use the executable *masscan.exe*. The installation of the software and the execution of the executable were not classified properly. The executable is a port scanner which only sends SYN packets. These SYN packets are not logged by Windows. That is the reason why the scanner was not properly classified. The Windows Events Logs contained events about packets blocked by the Windows Filtering Platform, originating from the IP addresses which were probed by the port scanner. It is unknown what kind of packets were received.

- *Did the classification miss information which was important to determine what the attacker did?*
  Yes, as already mentioned the port scan was missed. The only way in which the port scan could have been detected in this case, is by logging SYN packets which are sent or received. Event though you can see the responses originating from the IP addresses which were probed, you cannot determine with what purpose the packets were sent to the host. That all the blocked traffic originates from port 3389, the default

RDP port, could be a lead for determining the purpose of the network packets. Windows Packet capture (WinPcap) driver called NPF is used by this executable. This could be an indication for the usage of malicious software, but the driver is also used by other applications such as Wireshark. So it is not a direct indication.

Image Name: osDisk-confluencetemp1-20190717-041135
Trigger: Standard RDP
Login Time Frame: 17/07/2019 07:11:09 - 17/07/2019 07:15:27 & 17/07/2019 07:13:46 - 17/07/2019 07:15:33
IP Address: 197.210.55.143
Proposed Classification

- No artifacts have been found.

Classifications: None
Thorough Search

- *Are all the given classifications correct?*
  No classification has been given.

- *Was the attack classified with too few classifications?*
  Yes, a new account was added to the system called *sys*. This user was added to the *Users* and the *Administrators* group. The attacker logged into this account, that is why a second login time frame is stated above. Besides logging into the created account and executing some desktop accessories, no other malicious activity has been found.

- *Did the classification miss information which was important to determine what the attacker did?*
  Yes, the creation of a new account. Besides seeing this in the list of users, Windows events 4738 (user account password change), 4732 (a member was added to a security-enabled local group), 4720 (a user account was created), 4722 (a user account was enabled) and 4728 (a member was added to a security-enabled global group) could have been used to detect the account creation. Windows events started to originate from an unknown user. A large volume of LNK files were created for the Desktop shortcuts, when the attacker logged into the created user.