Master Thesis

Development and design of a honney pot

Mathieu TROUSSEU

Academic Year 2018–2019

Final year internship done in partnership with

Excellium Service S.A.

in preparation for the engineering diploma of TELECOM Nancy

Internship supervisor: Mathieu BAEUMLER

Academic supervisor: Isabelle CHRISMENT

Déclaration sur l'honneur de non-plagiat

Je soussigné(e),

Nom, prénom : Trousseu, Mathieu

Élève-ingénieur(e) régulièrement inscrit(e) en 3^{e} année à TELECOM Nancy

Numéro de carte de l'étudiant(e) : 0407034827K

Année universitaire : 2018–2019

Auteur(e) du document, mémoire, rapport ou code informatique intitulé :

Development and design of a honney pot

Par la présente, je déclare m'être informé(e) sur les différentes formes de plagiat existantes et sur les techniques et normes de citation et référence.

Je déclare en outre que le travail rendu est un travail original, issu de ma réflexion personnelle, et qu'il a été rédigé entièrement par mes soins. J'affirme n'avoir ni contrefait, ni falsifié, ni copié tout ou partie de l'œuvre d'autrui, en particulier texte ou code informatique, dans le but de me l'accaparer.

Je certifie donc que toutes formulations, idées, recherches, raisonnements, analyses, programmes, schémas ou autre créations, figurant dans le document et empruntés à un tiers, sont clairement signalés comme tels, selon les usages en vigueur.

Je suis conscient(e) que le fait de ne pas citer une source ou de ne pas la citer clairement et complètement est constitutif de plagiat, que le plagiat est considéré comme une faute grave au sein de l'Université, et qu'en cas de manquement aux règles en la matière, j'encourrais des poursuites non seulement devant la commission de discipline de l'établissement mais également devant les tribunaux de la République Française.

Fait à Contern, le September 2, 2019

Signature :

Master Thesis

Development and design of a honney pot

Mathieu TROUSSEU

Academic Year 2018–2019

Final year internship done in partnership with

Excellium Service S.A.

in preparation for the engineering diploma of TELECOM Nancy

Mathieu TROUSSEU
2, Impasse des jardins
57100, Manom
06 88 63 30 76
mathieu.trousseu@telecomnancy.eu

TELECOM Nancy
193 avenue Paul Muller,
CS 90172, VILLERS-LÈS-NANCY
+33 (0)3 83 68 26 00
contact@telecomnancy.eu

Excellium Service S.A.
5, rue goell
5326, Contern
+352 26 20 39 64

Internship supervisor: Mathieu BAEUMLER

Academic supervisor: Isabelle CHRISMENT

Acknowledgements

I would like to thank all the people that in one form or another participated to my internship.

My first thoughts go to M. Xavier Vincens and M. Cristophe Bianco who are the two managing partners of Excellium. They make substantial efforts to open their company to new people by opening opportunity of internships and (for Christophe) by delivering very interesting courses at Telecom Nancy.

Being a part of the CSIRT team for the past six months has been a great experience, thanks to all its members, namely: Paul JUNG, Céline MASSOMPIERRE, Mathieu BAEUMLER, Guenaëlle DE-JULIS, Yoann CHEVALIER, Valentin GIANNINI, and Baptiste CHOCOT.

But all of this would also not have been possible without the immeasurable help of my tutor, Mathieu BAEUMLER. He took time to lead and help me during this internship, so thank you Mathieu.

And last but not least, I wish to thank my academic supervisor Isabelle Chrisment, who followed my internship with attention and who watched out that everything went well during these 6 months.

I would like to add a personal thanks to all the Excellium workers and all the internees that I have meet during my internship, it has been a pleasure to see your smiles every morning.

Contents

1 Introduction

The increasing demand for IT services in enterprise goes in pair with the need to protect these services. This is one of the reasons why companies invest more and more into Cyber Security. And this is an area which evolves constantly, with new threats emerging every day, and new defenses to counteract them.
Moreover, the attack surface inside a company's infrastructure can be very broad: services, servers and workstations, network and security equipment, but also personal devices, managed or not by the company. The vulnerabilities affecting these devices can vary with the protocols in use, the OS or the software version installed. That means the teams in charge of the company IT security has to be aware of all the latest attacks and evolution in the IT field, and learn to detect threat, protect against them or mitigate them when a breach occurs.

Centralizing, analyzing and applying remediation to every alert generated by all the components might seem tantamount to an impossible task. And this is were a Security Operation Center (SOC) comes in handy for any security aware company. Members of the blue team (defense) of the company, the analysts working in the SOC gather every log and alerts into a System Information and Event Management (SIEM). A SIEM without specific Use Cases would not be really useful, therefore they also develop these correlation scenarios which help them perform the necessary triage to quickly pinpoint real threats on the network.

Another way to detect ongoing attacks is to use a Honeypot (HP). Honeypots are nodes in the network that simulate various services, or even servers and workstations. From an unaware point of view, they reproduce the normal behavior of the simulated service or equipment, but internally they monitor and keep track of every activity, collecting information and alerting the operator. By coupling these HP with a SIEM, it is possible to achieve a very fast and precise way to detect attacks. Indeed, each interaction that the honeypot detects and forward to the SIEM is potentially a real attack, without false positive, as honeypots do not have a real use and no one is supposed to interact with them.

We can thus wonder how a deception system could be craft ?

2 Excellium Services S.A.

I've done my internship in the headquarters of Excellium Services S.A. (Excellium, for short). Excellium is a young cybersecurity company which provide to its clients a wide range of IT services. Excellium covers many fields such as infrastructure, audit, monitoring, support and training.

Excellium is a limited company located at the 5 rue Goell, L-5326 Contern, Luxembourg. Founded in 2012 by Xavier Vincens and Christophe Bianco, Excellium has grown very fast since the beginning, doubling the sales every years. Lately, Sonae Investment Management (Sonae IM) became the majority shareholder of Excellium, creating one of the largest independent European cybersecurity group.

2.1 Detailed presentation

Very proud of its Luxembourgian roots, Excellium want to grow and share its vision of the cybersecurity. That is why Excellium want to be present in every French speaking country. Today, Excellium has a presence in seven french speaking countries: Belgium, France, Luxembourg, Morocco, Tunisia, Senegal, and the Ivory Coast in order to help more than 150 clients. In 2020, the plan is to penetrate the Swiss market and to activate the french market.

Today more than a hundred of persons are working for Excellium. Most of them are cybersecurity engineers with a broad base of knowledge in IT security fields allowing Excellium to offer services in all area of cybersecurity to the clients. With this strength and the Eyeguard cybersecurity platform (a 24/7 platform with skilled cyber hunter analyzing log of Excellium client), Excellium is a major actor of cybersecurity in Luxembourg and a good challenger in the French-speaking Countries.

With the gross sales doubling each year and shareholders with a lot of cash flow, all the indicators all good for the future of Excellium.

2.2 Departments

The skills and know-how of Excellium's people are divided across several multi-disciplinary teams. From governance to vulnerability exploitation, from defining secure architectures to setting up and validating information systems, the range competence is very broad and deep. Here, I'm going to present you all the Department of Excellium Services.

2.2.1 Intrusion Test - Red Team

The "pentest" team is called when a company want an intrusion test to be performed against its assets. In other words, this team will act as white hats (ethical hackers) and will look for security breaches in order to evaluate the security posture of an organization.

2.2.2 Infrastructure security services

This team installs, configures and manages the best-of-breed technologies in order to protect the client assets. Excellium also provides these technologies via its own private cloud which offers Security as a Service.

2.2.3 Information Security Governance

The ISG team provides the clients with the expertise in matter of cybersecurity risks management. In short, this department identifies risks, evaluates them and gives security recommendations to mitigate or nullify them.

2.2.4 Cybersecurity Operation Center

The department has its own in-house developed platform called Eyeguard, which provides a 24-7-365 visibility on clients' assets, securing them round the clock against threats. The analysts investigate any malicious activity, provide first reaction support, and when needed, escalate any issue to CERT-XLM.

2.2.5 CERT-XLM

Excellium's Cyber Security Incident Response Team (CSIRT), also called CERT-XLM (for Computer Emergency Response Team) is the team where I did my internship. The main goal of the team is to manage intrusion and security incidents of our clients. The CSIRT is called whenever a breach is detected, to help the company to quickly return safely to production. CERT-XLM will evaluate the extent of the compromise, mitigate it, find its point of origin, clean it and make sure it never happens again. The CSIRT main services are: breach analysis, server take down, CVE publication, generic incident response, document analysis, forensic investigation, doubt removal and malware analysis.

Tools used in CERT

An important work for the CERT is to regroup, organize and seek information about cyber threat. This discipline is called threat intelligence and the CSIRT has developped its own tools:

- Eye Deep: The name of this tool comes from the DeepWeb that is the part of internet that is not indexed by classical web browsers. The Eye Deep tool will seek in all the internet (including the DeepWeeb) for domains and keywords of excellium clients.
- Eye Tld: DNS is the service providing domain name. It is used everyday and could be under attack (for example typographic squatting). This tool offers a monitoring on client domain names.
- Eye Notify: Every day new vulnerabilities are discovered and published in the web. A vulnerability could not be impacting you, even thought you should be investagating it. This tool will filter all the vulnerabilities that are published and just warn the client when he is impacted.

Discover of the team usual work

During my internship, the CSIRT introduce me to the tasks they perform on a daily basis, when no incident is ongoing:

EyeNews: The daily survey of news about cybersecurity, ongoing phishing campaigns, DDoS, or exploitation of known vulnerabilities in the wild. Innovations in both attack and defense fields are covered as well. In case of something of interest, CERT-XLM notify the clients that could be impacted. In order to do perform this task, the CSIRT team has a Twitter account that follows all the most important cybersecurity publishers of Twitter, as well as other CSIRT teams. Another thing that has to be checked is the report that we receive on the mail account of the cert, including daily report of the Austrian Cert [9].

EyeNotify: CERT-XLM has developed its own aggregator of published Common Vulnerabilities and Exposures (CVE). Each CVE has a Common Vulnerability Scoring System (CVSS) score and is linked to specific version of products, the Common Platform Enumeration (CPE). CERT-XLM can notify its customers whenever there are critical vulnerabilities, such as those allowing remote code execution, as well as those impacting their assets to protect their infrastructure.

Use Case: The Excellium CSIRT work in pair with the Excellium SOC. As mentioned earlier in the section 2.2.4, the Excellium SOC develops detection use-cases to detect cyber threats. In CSIRT we help them to do that in creating in our private confluence a list of cyber threat and the way to detect them. With my colleague Baptiste Chocot, we did two use cases that detect unwanted access to a specified folder and the modification of a specified file in a specified folder.

Eye Change: As presented before, the CERT provide three eye products. Sometime some change have to be done on this product as changing the client mail.

Tap Dance: An internal joke to the CSIRT, referring to when one or several members of the CSIRT visit a client to promote all the services provided by the CSIRT.

Take down: A 'take down' in cybersecurity is when you shut down a malicious or compromised website. In a CSIRT, this often has to be done when there is a phishing website targeting you or your subsidiaries and you want to close it. The way to do it is to contact the domain provider of the phishing website and ask him to close it, by presenting relevant evidences of the malicious activity. There is however no foolproof method, as some provider will simply refuse to acknowledge something is happening on their network. Escalating to national CSIRT is sometimes needed.

Phishing: When there are phishing websites that are identified, the CSIRT will try to take down these websites, but this take time. In order to protect the client before the website is closed, phishing websites are reported on some sharing platformes like phishtank. These websites are used by antivirus to warn users that this website is potentialy a fishing one.

Incident preparation: Some Excellium clients have subscribed to a forensic insurance. That means that if they have a security incident, they can call the CSIRT to help them respond to the threat. To ensure both the clients and the CSIRT will be ready to intervene, a preparation phase is needed beforehand. The client is asked to answer a detailed survey about its infrastructure, from the border security perimeter to the internal assets. Availability of logs, their rotation period and the actual data they contain is also covered. This gives the insurance that each party has an in-depth knowledge of the environment. Furthermore, a gap-analysis report is produced at the end of the preparation phase to help mitigate any weakness in the security in place.

Newsletter: Each month, one of the teams of Excellium is selected to write down a paper on a cyber topic, which is then sent to the clients. After a given period of time, those papers are published freely on the Excellium blog [15].

With my colleague Baptiste Chocot, we were brought to test a big part of all thoses task.

2.2.6 Organizational chart

The CSIRT organizational chart can be found figure 2.1 that was provided by Excellium upon our arrival. For this organizational chart to be exact, we will have to add Guenaëlle DE-JULIS as a consultant (she arrived in February 2019) and also add Baptiste Chocot and me as trainees.

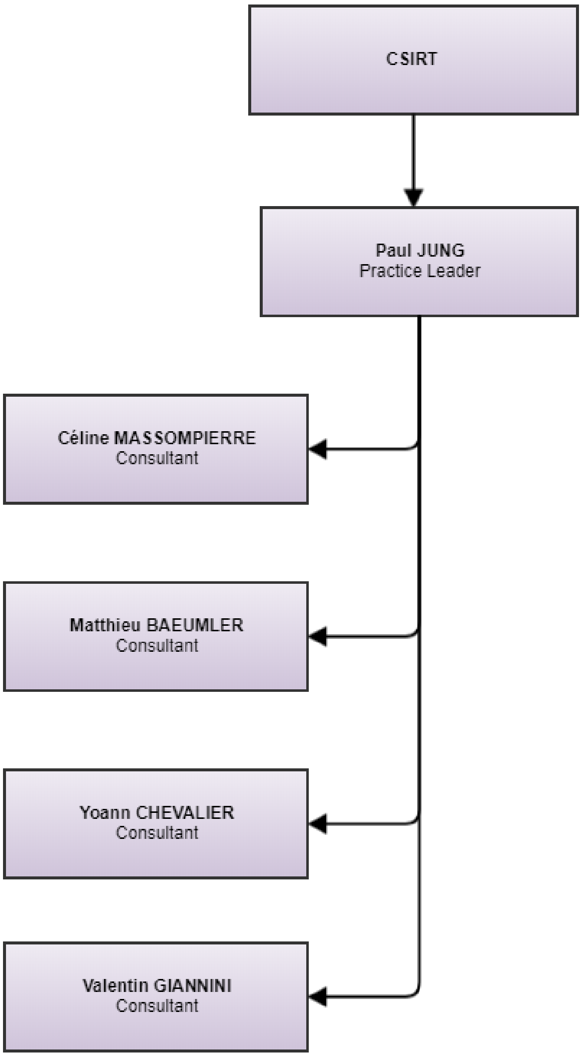


Figure 2.1: Excellium CSIRT organigram

3 Presentation of the project

3.1 Genesis of the project

3.1.1 Deception Technology: an emerging defense

The term **Deception technology** might be unknown or obscure to you, and that is perfectly normal considering it refers to one of the latest trends in the cybersecurity field. Its concepts are however quite intuitive and easy to understand, and based on other well-known technologies such as Honeypots. To put it simply, imagine you left something that will seem attractive in your enterprise network to any malicious actor, but booby-trapped in some way. Now hide it a little so only people that are tech-savvy and specifically looking for interesting or out-of-place data can find it. For example, a file with a name like *DomainPasswords.xlsx*, containing erroneous credentials. Keep the persons on charge of the IT of this enterprise informed of the presence of this decoy. Then, just wait for somebody to trigger it. And voila! When the trap is activated, the probability of a false positive should be very low, and you most likely caught a person with malicious intentions.

The definition Gartner[11], one of the most famous advisory firms, gives of deception platforms is:*[. . .] centrally managed systems for organizations to create, distribute and manage an entire deceptive environment and its related architectural elements. These decoy workstations, servers, devices, applications, services, protocols, data elements or users are often virtualized, essentially indistinguishable from real assets and identities, and are used as lures to entice, engage and detect an attacker.* Deception has been listed on the top ten of technologies for information in 2016 by Gartner[13][16]. They suggested that before 2018, around 10% of enterprises will use deception tools and tactics and *By 2022, 25% of all threat detection and response projects will include deception features and functionality, either embedded in their current vendor's threat detection technology stack or through pure-play deception platforms, up from 5% today.*

The main goal of the deception technology is to increase the Threat Detection and Response (TDR) level of a company, by providing low rate of false positive, very accurate data and valuable telemetry. This approach is not the same as the modern cybersecurity that holds on big data

analysis via SIEM and data mining tools. Indeed, this technology favors a 'right data' approach rather than a 'big data' one. This is even more true for actors such as the Excellium SOC, which deals with hundreds of GB of logs every day. At this point, the need for more accurate and thus less numerous indicators becomes more understandable.
When a deception system is deployed inside an infrastructure, the blueteam gains the ability to hunt for anyone or anything interacting with deception items. It requires a time of setup and tuning in the beginning, to identify what are the legitimate interactions (for example a vulnerability scanner such as Qualys …), and then you will see if there is a nasty agent which is already there scanning your network. But keep in mind that the deception technology is just another tool in the Defense in Depth (DiD) approach, and should always be used in conjunction with classic cybersecurity technology. Indeed, deception is based on decoys, simulated services or fake documents. If an attacker manages to compromise the enterprise without interacting with any decoys, the only way to detect it is by traditional methods.
Deception technology also allows seeing if there is any suspicious activity inside your network.

3.1.2 Deception Technology: a detailed presentation

Deception Items

Deception technology regroups all assets that can be considered as fake. We can identify these assets as follow:

- Fake servers also called honeypots,
- Fake data also called as honey-tokens,
- Data that lead to honeypots also called breadcrumbs.

Honeypots are servers seemingly part of infrastructure, but isolated and closely monitored. The literature also refers to honeypots as decoys. They can take several aspects. Their operation systems can vary, as the services shown to the attacker. These honeypots can provide either low or high interaction. A low interaction honeypot will just emulate legitimate services that a given server is expected to run, and only accept the most basic commands from the. This makes the honeypot more secure, but it also reduces the information that can be gathered form the attackers. In the other hand, a high interaction honeypot will run the full-fledged service, recording all the attacker actions to give the security analysts detailed data on an ongoing attack. These high-interaction services can comes with a higher security risk and maintenance cost.

Honey-tokens are conspicuous data dropped in sensitive area of a network. One particularity is that they are stand-alone decoys, and requires little to no backend at all to work. That attackers, by exploiting these data, are tricked into sending information about themselves to the logs management system. These honey-tokens often take form of office or PDF files on network shares or workstations, with for example erroneous credentials for content, or crafted metadata that will make the office application generate a network connection to a monitored service.

The breadcrumbs are data deployed on real assets and that will lead an inquiring attacker to the honeypots. For example, simple DNS names hinting to high-value targets but whose records actually point to the honeypot. Or fake admin credentials or password hashes injected into the LSASS process, which when discovered and used by the attackers will generate an alert. These are called honey credentials. Connection shortcuts (RDP) can also be used for the same purpose.

Deploying deception technology

The deployment process depends on what solution is chosen but we can summarize it with the most common methods.
The first part of the deployment is to gather information on the existing infrastructure, that means getting Proxy, DNS, NTP, SIEM IP, company information, IP for honeypot, access for dropping honey-tokens. The network design of the company is also important in order to know where the deception should be put to be at the most relevant place.

The second part is to deploy all the deceptions tools, once all the company information has been gathered by the cybersecurity team. During the installation part, there is still need to have interaction with the network administrators of the company where the deception technology is being installed. For example, if you need some DNS records, AD accounts or anything else, you'll have to ask them.

Once the second part is over, you need to define the Use Cases for the company. Use cases define basically when the log deception manager has to raise an alert. This will define what the person in charge of managing these alerts will see when they arise, but also how to test the scenario, what have to done when there is a match, how to investigate the event and verify if it is a false positive, and how to escalate the incident when it appear to be a real threat.

Finally, the fourth part is the testing part where security analysts test the product in the client network infrastructure and whitelist all the legitimate users or engines that trigger the deception items.

How does it work

Once it is installed. The workflow will be divided in several parts:

- Installation of the deception item,
- An attacker managed to breach in the company network,
- The attacker triggers a deception tool,
- The analyst receives an alert,
- The analyst checks the alert and realizes it is an attacker,
- The attacker is discovered.

We can resume the deception work-flow with the figure 3.1.

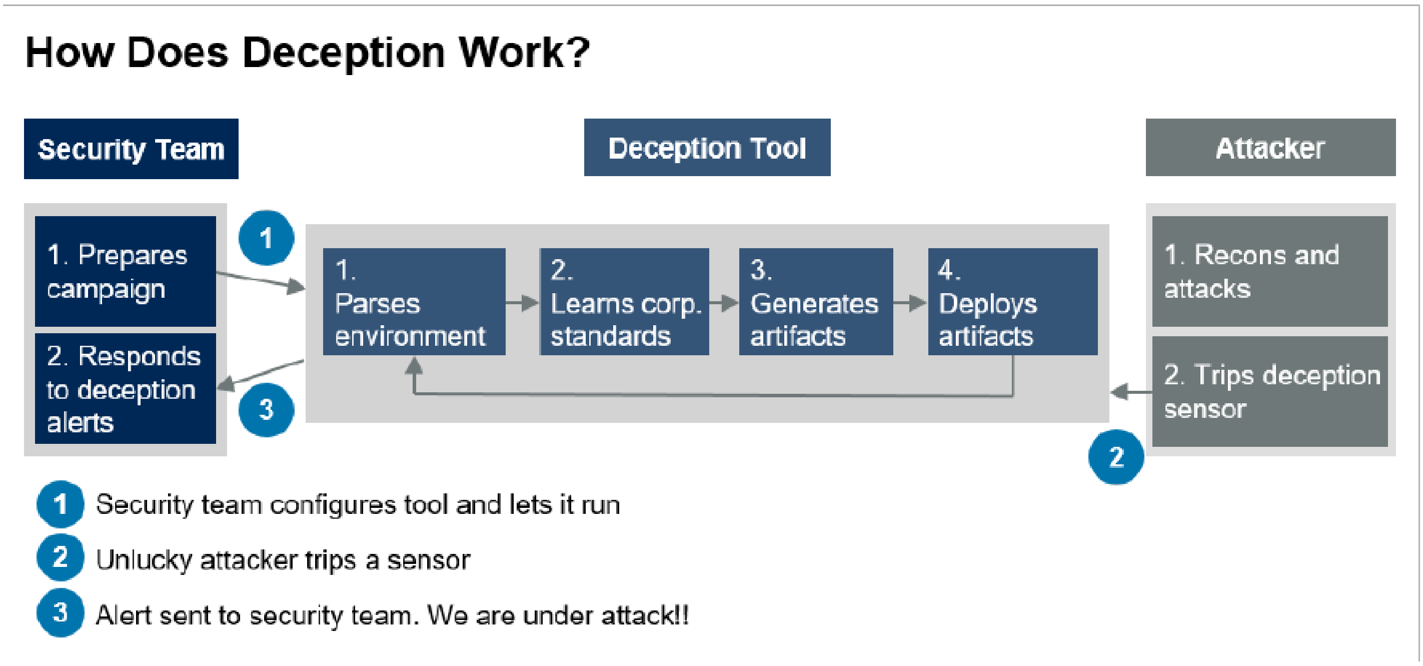

How Does Deception Work?
Security Team
Deception Tool
Attacker
1. Prepares campaign
2. Responds to deception alerts
1. Parses environment
2. Learns corp. standards
3. Generates artifacts
4. Deploys artifacts
1. Recons and attacks
2. Trips deception sensor
1
2
3
1 Security team configures tool and lets it run
2 Unlucky attacker trips a sensor
3 Alert sent to security team. We are under attack!!

Figure 3.1: How Does Deception Work? [16]

As we can see, the first action is to configure and install the deception system on the client network. Once it is done, the security analysts can just wait that someone triggers one artifact generated by the deception tool while doing their other day-to-day tasks. Once the artifact is triggered, the analysts follow what have been defined in the use case to determine if other actions are required.

3.1.3 Deception Technology: the variable part in the deployment

Every network infrastructure has its own particularity. Each company has its own way of writing documents, a specific log management policy and a different view on what have to be considered as critical for its business. That’s why deception technology cannot be set up twice the same way. For example, the naming policy is very fluctuating, as the IP attribution is. One advantage of the deception technology is the ability to look like a real asset. That why a preparation phase has to be done prior to any deployment.

3.1.4 Deception Technology: pro and con

Let’s do the pro and con for deception technology.The main advantage of deception technology is that the false positive rate of deception is low. Indeed, by the design of how deception works, the simple connection to one deception item is not supposed to happen.
Another advantage is that it enables to a security engineer to use his knowledge of his infrastructure for its own defense.
However, deception also has drawbacks. For example, this is not something that will be triggered all the time. An attack could be successful and undetected if the attacker does not trigger deception tokens. Moreover, some tools that detect some kind of deception are already available so a pentester aware could dodge some traps. Furthermore, what it is complicated in deception is that testing is quite complicated because attacks are always new.

3.2 Goal of the project

The main goal of this project is to be able to produce a working Proof of Concept (POC) of deception technologies. A definite plus for this project would be to deploy this POC on client infrastructures without a lot of effort for the person who will be in charge of the solution installation.

The beginning of the project will be to get familiar with all information dealing with deception technology. That means reading the available literature and documentation about deception, but also broadening this with related topics such as how to make a deception network safe. The result of this first step is a state of the art document to be uploaded on the Excellium Confluence. The main parts of the state of the art document are:

- Deception System,
- Open Source Honeypot,
- Open Source Deception Tools,
- IPS/IDS Open Source.

The second part of the project is to establish the design of the application. In a simple way, this objective was to draw the design solution at a network level and software level (see part 5). This gives the solution a body and help to imagine how will the final product will look like, helping to think about the entire solution before even doing it.

The third part consists in building the solution on what we worked before. This took place on the virtualized environment of Excellium (VMware ESXi) in order to see if the solution is something that we are capable of doing. It also helps to face the potential problems that someone could encounter during a client installation, but in this case it is easier because we have 'master' over the network.

The last part is to install the solution that have been tried at Excellium in a client infrastructure. This part is the part where the project is truly evaluated.

3.3 Final deliverable

The final deliverable has to be as simple as possible. It will be divided in several parts. The first big part will be a documentation one. In this part will be presented the state of the Art described previously, but also the design solution (network and application). It will also contains all the documentation on how the solution is working, and how someone can install it in a client infrastructure. Another deliverable is the templates of the virtual workstations that was produced during the solution elaboration.f

The documentation part is available in the Excellium Confluence under the section: Computer Security Incident Response Team Home - CSIRT TRAININGS - Internal deception device. The templates are available on the ESXi in the trainee folder.

4 State of the Art

This project regrouped a lot of different technologies. I had to increase my knowledge, or learn, about network infrastructure, network protocols, pentester attack plans and of course all that was a part of my internee subject. That is why the main parts of my state of the art are:

- Deception System,
- Deception Tools,
- Network Security Tool,
- Orchestration,
- Kill chain.

Another version of this state of the art was made with Microsoft Word in the beginning of my internship and is a little shorter.

4.1 Deception System

Here is the part where I'll be talking about the most relevant technologies in place within the deception system. The amount of information on each product differs because most deception systems are black boxes, and the information that I gathered are only the results of the information found on Internet. Deception Systems are cybersecurity systems that are incorporated in a company network to fool potential attackers. What makes them useful is that there are very accurate, and easy to deploy. Every deception System works approximately the same way: first, they scan the network looking for some patterns in the network (they often use machine learning), after that, they spread their endpoints and deception items over the networks.

I have done my state of the art of deception on the products presented in the table 4.1

Vendor	Name of the Solution
Acalvio	Shadowplex[8]
Attivo	ThreatDefend Platform[1]
CounterCraft	Cyber Deception Platform[10]
Cymmetria	MazeRunner[3]
Deceptive Bytes	Active Endpoint Cyber Defense
Fidelis Security	Fidelis Security Deception
Illusive Networks	Deception Management System
PacketViper	PacketViper
Smokescreen	IllusionBLACK
TrapX	DeceptionGrid

Table 4.1: List of Deception Systems

4.1.1 General offer

The general offer is what is clasic to see in deception system. All of them provide main installation via virtualisation. Deception systems are based on five main pillars: Manageability, Discoverability, Undectability, Interoperability and Cost-effectiveness. They all provide all of decoys. They often works with a first network scan for knowing what they have to do and continue doing it continuously for having a constant knowing of network state. This allows having an automated deployment and an optimal deployment.
All these systems are managed by a platform that allows the client to follow time to time what alerts are triggered and what action have to be taken.

4.1.2 Specific offer

Every system doesn't provide exactly the same kind of deception service. Here we are going to see what are the main differences. Every system uses virtualisation, but the hypervisor used is not the same, some use KVM, a linux open-source hypervisor (cymmetria for example) and some the vmware hypervisor: ESX.
Some provide API for easier implementation.
Some oter offer cloud tokens for the three main cloud platform: AWS (Amazon cloud), Google cloud and Azure (Microsoft cloud).
CounterCraft offer a unique deception which is called wifi deception and use a physical item to fake a wifi endpoint. Finaly, the more advence system provide active response. That means that theses solutions enable security teams to scripts and automate workflows for active response and investigation (Fidelis Security Deception for example).

4.1.3 Exploitation Model

There are two kind of exploitation model presented in all the deception system that I have study. The first one is the model that is told as DaaS (Deception as a Service). This model is an installation that use the example of the cloud. Service and software are based on distant server.
The other model is the classical model where all the solution is installed on the client infrastructure.
The first one is instresting because the client doesn't have a lot to take care off. The provider of the solution deal with the management of what is installed. But the second one is more intresting regarding security issues. Indeed with the solution in intern, the security is handle by the client and the client only have to trust its own infrastructure.

4.1.4 Pricing

There is two kind of pricing in the field of deception system. A global pricing which turns between twenty thousand dollars (Acalvio) and fivety hundred dollars. And a pricing by sensor which is around five hundred and one thousand dollars by sensor. The last payment offer is by user, for example Guardicore offers its solution for 60 dollars by user.

4.2 Deception Tools

4.2.1 Deploy-Deception [6]

Deploy-Deception is a powershell module that is able create active directory decoy objects. This module have to be run on a Domain Controller with a domain admin account. This decoys are based on domain object flags and attributes. For example the flag 'PasswordNeverExpires' is a flag that interest attacker because those password could be very old and often easy to crack. There are five different kind of active directory objects that can be deployed:

- UserDeception: that creates a user account with some attractive property and flags and monitor any access in read to its property,
- SlaveDeception: that creates a slave and a master which have 'GenericAll' right over the slave: any interaction to the master or the slave is monitored,
- PrivilegedUserDeception: All is in the name, that create a privileged user, often member of the domain admin group but on which nobody can log on. Any interaction with this user is monitored,

- ComputerDeception: that create a computer object with 'Unconstrained Delegation' property enable. This is an interesting property for having Ticket Granting Tickets, a ticket which allow you to access to some services. Any interaction with this computer object is monitored.
- GroupDeception: that create a group which is member of the dnsadmin group. Any interaction with this group object is monitored.

Any monitor action is done via the event viewer. The principal events that are monitored are:

- the 4662: 'An operation was performed on an object',
- the 4768: 'A Kerberos authentication ticket (TGT) was requested'.

Those event are triggered when the objects are accessed or requested. It could be interesting, but the inconvenient is that it is mandatory to be domain admin to run that, and that in a client infrastructure, it is hard to have this kind of access.
All the code is available in Github and totally free.

4.2.2 Canary Token [7]

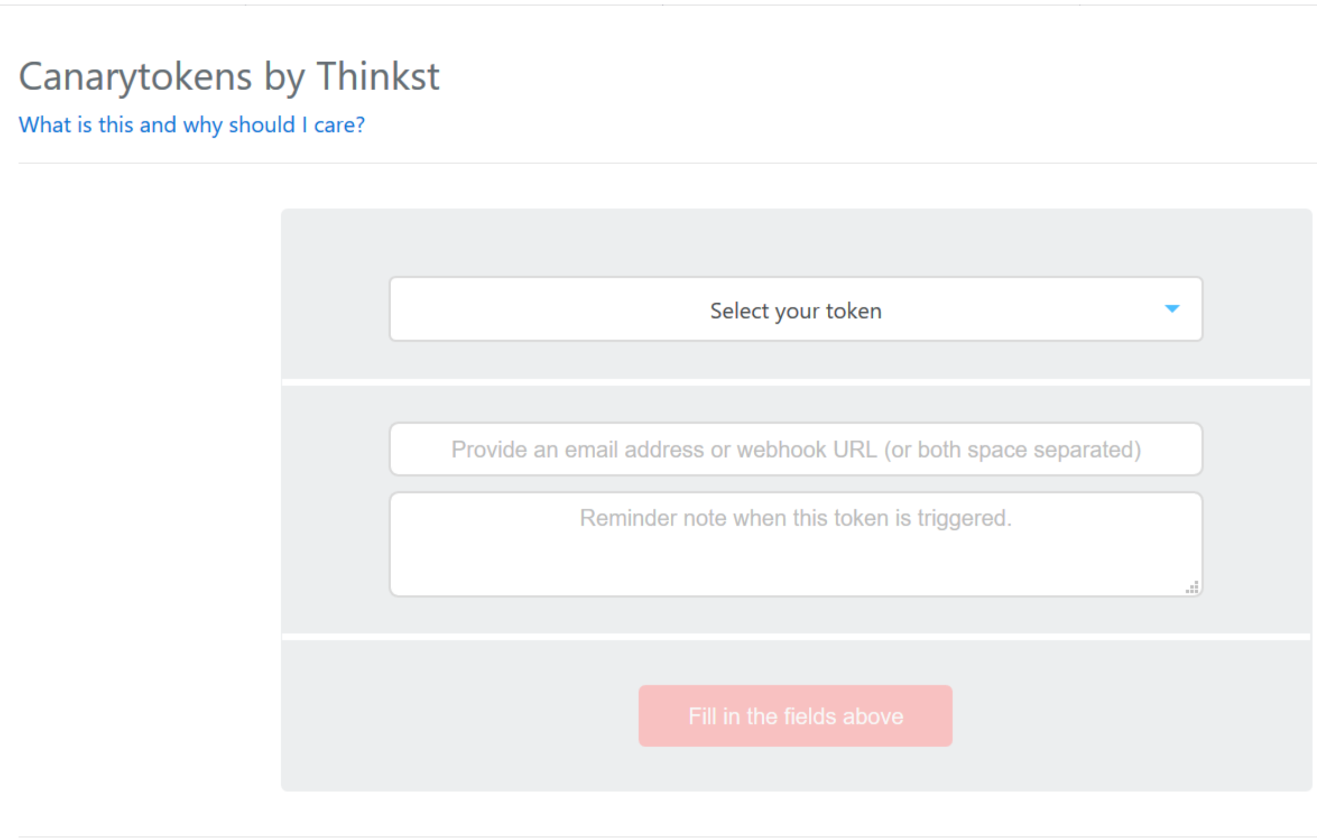
What is it ?

Created by Thinkst, this technology is based on a tool that can create files called deceptions tokens or honey tokens, that are triggered during the token opening. These tokens are often linked to a webserver that is contacted every time a honey token is triggered. What is called token could be different kind of files like URL, DNS, email, word, PDF, windows Folder, dll, exe, docx...

A basic user has access to a website: ***https://canarytokens.org/generate***. This website ask the user to give:

- The kind of canary token that he want,
- A webhook url or an email address; this is where the alerts will be send,
- A reminder note to send when the canary token is triggered.

Once these informations are given, the user download a web-generated file that correspond to the kind of file that the user specified before. The user will drop this file in a place that need to be monitored. And finally, when the token will be triggered, the user will be notified either by mail or by the webhook URL. The interface looks like the figure 4.1.



Canarytokens by Thinkst

What is this and why should I care?

Select your token
Provide an email address or webhook URL (or both space separated)
Reminder note when this token is triggered.
Fill in the fields above

Figure 4.1: Canary Token Web Interface [18]

For a company, Thinkst provide a github where there are python scripts and a docker image that allow a quick installation of the token generator website. It provide also more advance informations when one canary token is triggered.

The most relevant are PDF, docx and URL. These tokens only are triggered when they are open by classical applications, for example if an html is created with a URL pointing to the server is opened by a cat command, the token will not be triggered.

Pros and Cons

The pros for this solution is that can be easily spread on the network, this is not something heavy to drop on the network and when canary tokens are triggered we can choose by what means the alert can be send. Furthermore, the kind of notification we receive is quite complete : there are information about the IP which triggered the document, the note that have been writing down when the creation of the token mentioned, and finally there is the timestamp. The cons for this solution is that we need at least one server to create the canary tokens. Moreover, some tokens need permissions to be triggered (pdf, docx . . .).

4.2.3 Domain Controller Enticing Password Tripwire (DCEPT)[5] [12]

This tool is based on the pass-the-hash attack which consist on extracting hashes credential stored when an LM or NTLM authentication have be done. The DCEPT agent inserts fake data on the cache in order to deceive the attacker once he is local admin on the endpoint where the DCEPT agent is installed.

This is a tool that works with three endpoints:

- One agent written on C# which caches honey tokens,
- One server that generate, issues and manage the honey tokens,
- One component that listen logon attempts.

The pros for this solution is that it is a very tricky one and that the code available. But the cons it is that the solution use many resources, that the solution has a bad detection ratio (pentester get there late, they rather find creds first), and this solution needs to use nonAD credentials (for detection, because there is no sniffer).

4.3 HoneyPot

One of my main interest was about HoneyPot, the central point of my subject. I found a great GitHub [12] that listed nearly all the most interesting links to open-source honeypots and deception tools.

Before presenting the most interesting honeypot that I found, I will present the principals characteristics that define a honeypot. The term honeypot first refers to the pot containing the sweet product that bears love unconditionally. But it also refers to the cybersecurity assets that are set to fool attacker by simulating real IT assets. One of the question that you have to ask yourself when you first install a honeypot is how well it is secured. An attacker must be flagged when he contacts an honeypot but he also must not be able to gain real access of the honey otherwise he could do more damage, especially if this is a honeypot integrated to a real production environment. According to that, two categories of honeypot can be defined, the high level interaction honeypots and the medium level interaction honeypots. The level of interaction is based on what an attacker is able to do on the honeypot. While a high level honeypot is interesting to study the kind of attack a hacker can do, and is therefore more valuable for cyber research purposes, a low interaction level is more interesting for generating alert and making a company aware that someone is probably targeting it. We will focus our analysis on the last category because it corresponds to the kind of honeypot I was looking for.

I present in the figure 4.2 the most interesting honeypots that I found. There is a lot of available open-source honeypot on github or other platforms. Some choice have to be made. The CSIRT is used to use python so I choose to focus mainly on the technologies that use python (Python 3 is better, but there are still a lot of project in 2.7). One other thing very interesting is to see what services are monitored. SSH, HTTP/HTTPS and data base are the three main services emulated because there are frequently under attack and often exposed. Moreover, what was put on the figure 4.2 is the years of the last commit on the project. Indeed, a project that have a recent commit shows that it have a community behind and that it is maintained. This is a good thing because protocoles can evolve and it is good to have a product that evolve at the same speed. Finaly, OS on where honeypots could run are also important because if the honeypot run on licensed OS, that should be taken into account so as the OS company park.

4.4 Network Security Tool

As mentioned earlier, the security of the environment is at stake each time a new equipment is deployed. In order to ensure a high level of protection, I've done a comparative study on the IPS/IDS (Intrusion Prevention/Detection System) technology. The two main targets of my state of the art was Suricata and snort. The choose of the study of the two of them was done because they are both free, open-source and well documented. I also study the functioning of iptables for natting purpose.

4.4.1 Snort

Snort is a free open-source networks base IPS/IDS. Available since 1998, this is a tool that have a big group of user. He is using a wildly know language, the C. One good point of this tool is its scalability and its flexibility. Indeed, it can be deployed in multiple network environment and on several OS. Snort is rule-based IPS/IDS. That means that Snort will log / block only the traffic that is told by the rules to be logged / blocked. When you are using Snort, four choose are open for having rules:

- Using the basic rule set providing by snort,
- Using rules shared by the snort community,
- Using rules that you bought, for example Talos ones,
- Creating your own rules.

The ruling syntax of snort is quite basic, you can see an example of one in the picture **??**. You can define the ports, the ip addresses and the protocols to monitor. This way, the rule defined on figure **??** will make an alert saying "ICMP Attempt Attack" whenever an ICMP packet which have a source ip address 192.168.1.10.

The main drawback of Snort is that this is not a solution "application aware". That means that snort is not able to detect a traffic HTTP, it could only detect a TCP traffic on port 80 for example. A notable and famous thing about snort is his big pig as a logo.

4.4.2 Suricata

Suricata is also a free open-source networks base IPS/IDS. Its first version came out in 2009 and was developed by the Open Information Security Foundation (OISF). The use of Suricata and Snort is quite the same. Indeed Suricata is also rule-based. He also have its own ruling language but also support most of the rules written for snort (the Talos rules works). The rules query are like presented in the figure 4.4.

In the example figure 4.4, red is the action, green is the header and blue are the options. It is really close to snort language.
The main advantage of Suricata over Snort is that Suricata is younger. Thus, Suricata functionality widely used nowadays like multithreading and application detection.

4.4.3 Iptables

Iptables is free software embedded with Linux kernel. The purpose of iptables is to manage packet and it is mainly used to manage and configure the Linux kernel firewall. Iptables has been developed in the C language and is a part of the Netfilter framework. Iptables is used in command line. These commands are tables-based. There are four table:

- Nat table: used for address translation,
- Filter: used for packet filtering,
- Raw: used for configuring exemptions from connection,
- Mangle: used for special purpose.

A table is a set of rules divided into chains. According to routing decision, every couple table-chain crossed will have his rules triggered. This process can be illustrated by the picture 4.5. In the figure 4.5 the first routing decision is to determine if the packet has to destination the endpoint where iptables is installed or if the packet has been forwarded. According to this, different rules will be triggered following the path of the figure 4.5.

The rules are organized in list, and the rules are executed in order. For example if there is a rule that drop all packet from an ip X and a other rule that accept all packet from an IP address X is set, the one that will define the becoming of the packet will be the one on the top of the list.

A classical iptables query is:

```
sudo iptables -A -i <interface> -p <protocol (tcp/udp) > -s <source> --dport <port no.> -j <target>
```

For adding rules in iptables the user should have by default root rights. The -A option mean adding the rule in the last position of rule set. The other option i, p, s and dport specify the targeted packet, other property can be selected as sport for example. The j option specify what to do if the options match, for example the target LOGGING will send the packet to the kernel log. Thus, thus rules will send to the target -j the packet comming to -i on port −dport with the proctocole -p comming from -s.
Iptables is a very usefull tool on ubuntu because it is available from scratch and have a lot of tutorials online.

Saving rules

The working process of Iptables is to destroy all rules after each reboot. As a workaround, a solution is to install the package 'iptables-persistent'.

4.5 Orchestration

Thomas Erl define the orchestration as *[. . .] the automated configuration, coordination, and management of computer systems and software* [14].For maintaining my network infrastructure, the subject of the orchestration was something to think about. I had several chose to make between the ease and the practical. I'll show you the three options that I though for this part.

4.5.1 Puppet

Puppet an open-core software configuration management tool. It permits to deploy new configuration through easily. It is based on a client-server architecture. The server is called master, the workstations are called slaves. The slaves often communicate with their master and ask if there is a new configuration to apply. When there is one the server managed to give the slaves their new configuration. It is then easy to deploy a new update just by pushing something to the master, which will be, after, in charge of pushing it back to the slave concerned. The advantage is the capability of a quick and centralized deployment of configuration. But the drawback is that a server is needed to install the puppet master and the slave also have to be configured.

4.5.2 Debian Package

Debian package is well known and used by developers. Debian packages integrate very well into the entire system. One advantage is that thanks to the fame of the packaging, there are a lot

The packaging work flow is usually like this:

- Step 1: Rename the upstream tarball,
- Step 2: Unpack the upstream tarball,
- Step 3: Add the Debian packaging files,
- Step 4: Build the package,
- Step 5: Install the package.

This is a good way of installing and deploying a craft package because it doesn't need that much user interaction and user are used to use it.

4.5.3 Bash script

A bash script is the simplest way of installing and configuring a software. The advantage is that an advanced user can see what is really happening, and that this is quite simple to do. A bash script is a script beginning by #!/bin/sh, it is basically a plain text that are composed by many commands.

4.6 Kill Chain

The Cyber kill chain in cybersecurity, originally developed by Lockheed Martin[17], comes from the military kill chain which consists of an end-to-end process. In cybersecurity, this term designates all the steps followed by the attackers to finally achieve the goal of his mission. This often means be administrator of the domain. This chain is divided into seven main parts that are:

- Reconnaissance,
- Weaponization,
- Delivery,
- Exploitation,
- Installation,
- Command and Control,
- Actions on Objective.

This is a framework well known in the cybersecurity industry that everybody in this field should be aware of, because when you build a cybersecurity tool, you have to know in which part of the chain will you potentially catch the attacker.
The kill chain can be represented by the figure 4.6.

This figure 4.6 is a representation of all the possible steps that are done by an attacker and where each part can go into a part presented above.

4.6.1 Reconnaissance

The reconnaissance part is the part when the attackers regroup information about the target. They will gather information that you can have easily or not. For example they will take the public IP available via DNS records, or look for open port and IP potentially vulnerable via scanner like Shodan[19]. They will also gather company information, such as the basic email patern : 'firstname.lastname@company.com' for example.
In this part there is nothing we can't do with actual deception. We could try to spread fake information through internet but this could have a bad impact on the society and this is therefore not advised.

4.6.2 Weaponization

The weaponization part is when the attackers prepare something that will allow them to have a first foothold in the company. What is most frequent is ether a malware that will be dropped on a company, a phishing website that is customized for the company targeted, a leak of passwords collected on the darknet …In this part, there is also nothing that we can do, at a deception level.

4.6.3 Delivery

The delivery part is the part when the attackers deliver the ’weapon’ that they collected during the previous part. If the weapon is a malware, the delivering process could be sending a mail to the victim with the malware embedded as an attachment. In this part, there is also nothing that we can do, at a deception level.

4.6.4 Exploitation

The exploitation part is the part when the attackers have dropped their payload and it has been activated. For example, if a malware has been downloaded as an executable file and that the targeted victim execute it. Here, a deception tools could be triggered. For example, there is the worm malware category that regroups malware using software vulnerability to spread over the network. If this malware tries to spread, we could know.

4.6.5 Installation

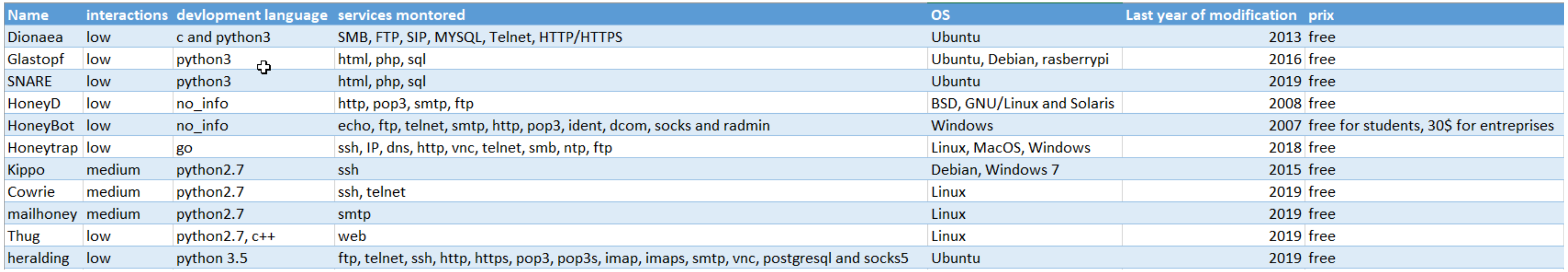
The installation part is the part when you install the malware backdoor on the victim asset. This part is not a part of all attacks but it occurs more often than not. A backdoor is a remote access that gives a persistent access for the attacker. This could be seen if the attacker tries to do it on a deception asset.

4.6.6 Command and Control

The Command and Control (C2 or C&C) part are the part when the attackers make an easy access to the targeted network from a server. The name of the subsection comes from the name that are given to these kind of servers.

4.6.7 Actions on Objective

The action on objective part is the part when the attackers have the access and the ability to do what for they came for. That means steal juicy data, compromising a service / network, spying…Here deception is also very interesting. A text file *’domainpassword.txt’* in an admin folder of a shared file could be very attractive to an attacker.



Name	interactions	devlopment language	services montored	OS	Last year of modification	prix
Dionaea	low	c and python3	SMB, FTP, SIP, MYSQL, Telnet, HTTP/HTTPS	Ubuntu	2013	free
Glastopf	low	python3	html, php, sql	Ubuntu, Debian, rasberrypi	2016	free
SNARE	low	python3	html, php, sql	Ubuntu	2019	free
HoneyD	low	no_info	http, pop3, smtp, ftp	BSD, GNU/Linux and Solaris	2008	free
HoneyBot	low	no_info	echo, ftp, telnet, smtp, http, pop3, ident, dcom, socks and radmin	Windows	2007	free for students, 30$ for entreprises
Honeytrap	low	go	ssh, IP, dns, http, vnc, telnet, smb, ntp, ftp	Linux, MacOS, Windows	2018	free
Kippo	medium	python2.7	ssh	Debian, Windows 7	2015	free
Cowrie	medium	python2.7	ssh, telnet	Linux	2019	free
mailhoney	medium	python2.7	smtp	Linux	2019	free
Thug	low	python2.7, c++	web	Linux	2019	free
heralding	low	python 3.5	ftp, telnet, ssh, http, https, pop3, pop3s, imap, imaps, smtp, vnc, postgresql and socks5	Ubuntu	2019	free

Figure 4.2: State of the art honeypots

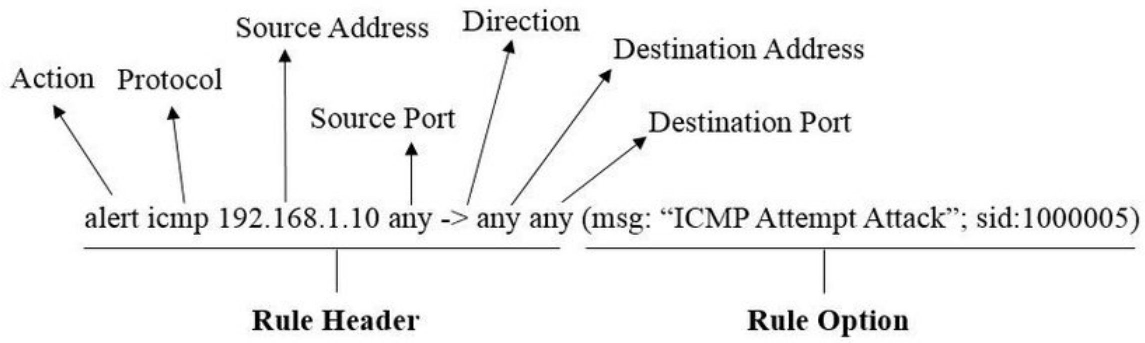
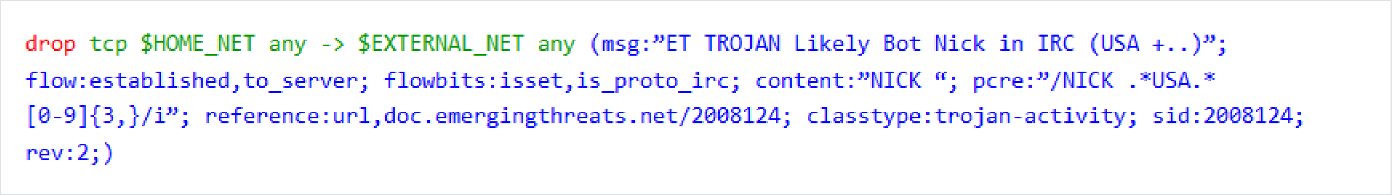


Figure 4.3: Snort rule [18]



```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”;
flow:established,to_server; flowbits:isset,is_proto_irc; content:”NICK “; pcre:”/NICK .*USA.*
[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124;
rev:2;)
```

Figure 4.4: Suricata Rules

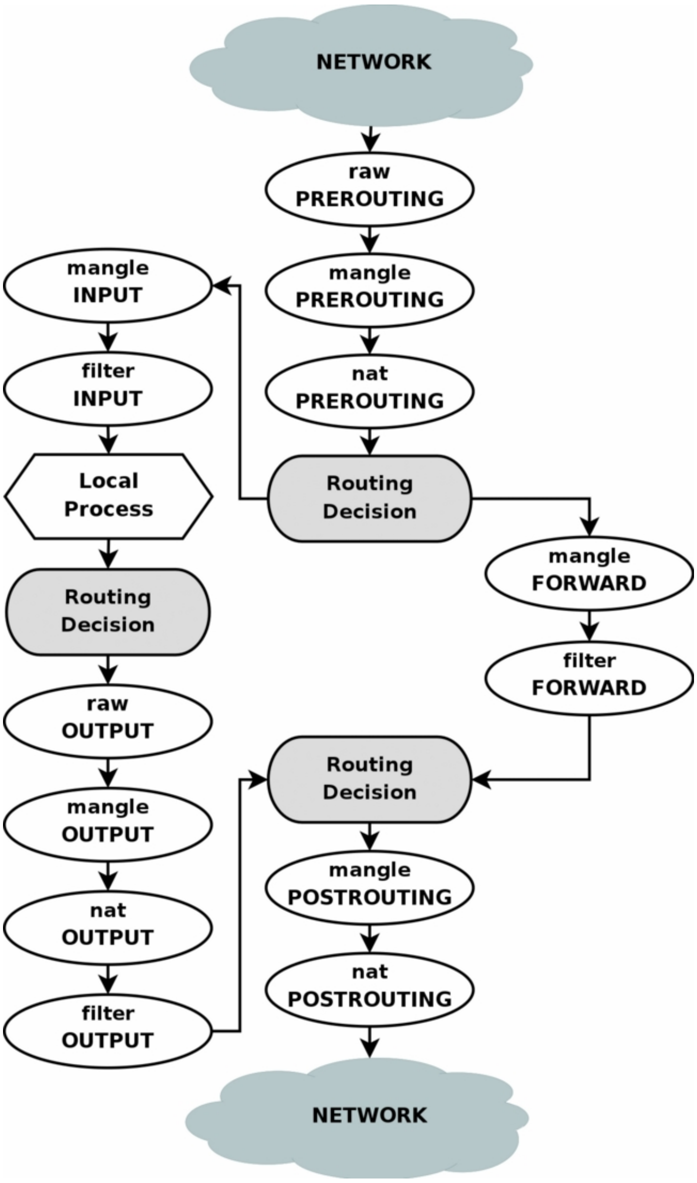


Figure 4.5: Iptables workflow [2]

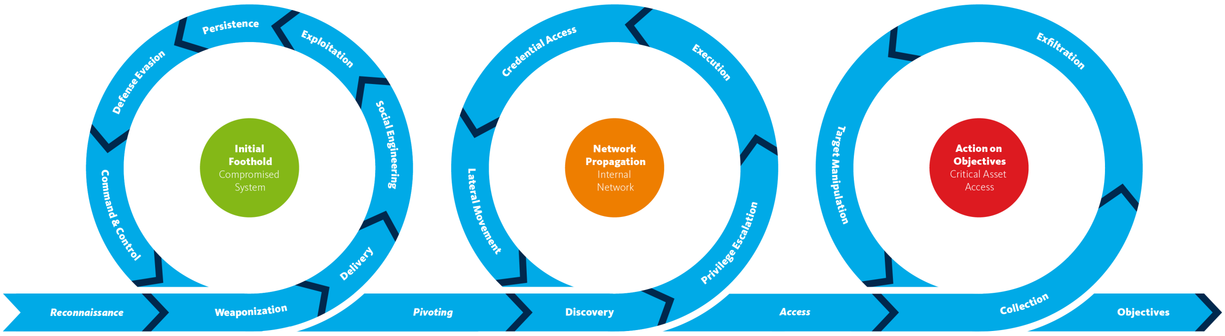


Figure 4.6: Kill Chain [4]

5 Design of the solution

Some people like to just jump into developing without writing down and without thinking about the project in a global form. That's not what I have done. I chose to take time to think about the global design of my future project. Indeed it is better to think about modularity, maintainability, performance, usability, tracability and deployment before the beginning of the project than think about it at the end.

5.1 Network Configuration

The application will be deployed via the ESXi. The ESXi deploys three workstations: two Ubuntu and one Windows workstation.

The choice of the ESXi was made because, first, it is easy to deploy workstations of any OS that already has anything needed pre-installed on them. This is done via the help of VMware templates. The choice of ESXi was also done because of monitoring information that are provided via the vSphere interface.

The first Windows workstation will be created with an attractive AD user name, a RDP emulation and a folder shared with anyone (configuration bait). It will only need one IP address and a DNS record.

The first Ubuntu server emulates **n** (number to define) services and have **n** IP addresses, one for each services. The number of IPs could be less than **n** because it is not surprising for an IPs to have several services that are running.

The second Ubuntu server is there to act as a listener and filtering network flow between the outside (client's network) and the VMs decoys. This Ubuntu server is also charged to detect and stop targeted malicious activities on the other VMs with the help of an IDS. It will NAT all the traffic which has for destination the decoys.

Some fake data will be uploaded on the Share Server of the company and the opening of those documents will trigger alerts to the first Ubuntu server. This functionality will only be possible if the customer allows us to drop fake data on sensitive network emplacement. If the customer want this functionality, the fake documents will be dropped on a shared ftp server for example.

The log report will be done via the SIEM which already has a route to the SOC of Excellium Services. All the logs will be sent to the SIEM via the second Ubuntu server. This way all the traffic coming and leaving from the decoys will pass through the IDS. This is done to have the best security flow.

The access to the ESXi for the installation and for the admin processes is done via the route already configured for the SIEM installation and monitoring. If there is no route, one need to be configured between Excellium and the customers.
Access to the Internet will be granted by the link between the ESXi and Excellium (or if possible, by the Internal Company Proxy).

The schema of this architecture can be seen in the figure 5.1. This is a layer 3 architecture and so we don't see the layer 2 architecture. But the architecture of the decoys will be divided in two separated VLANs. An administration one where only administrators will have access, and a more open one.

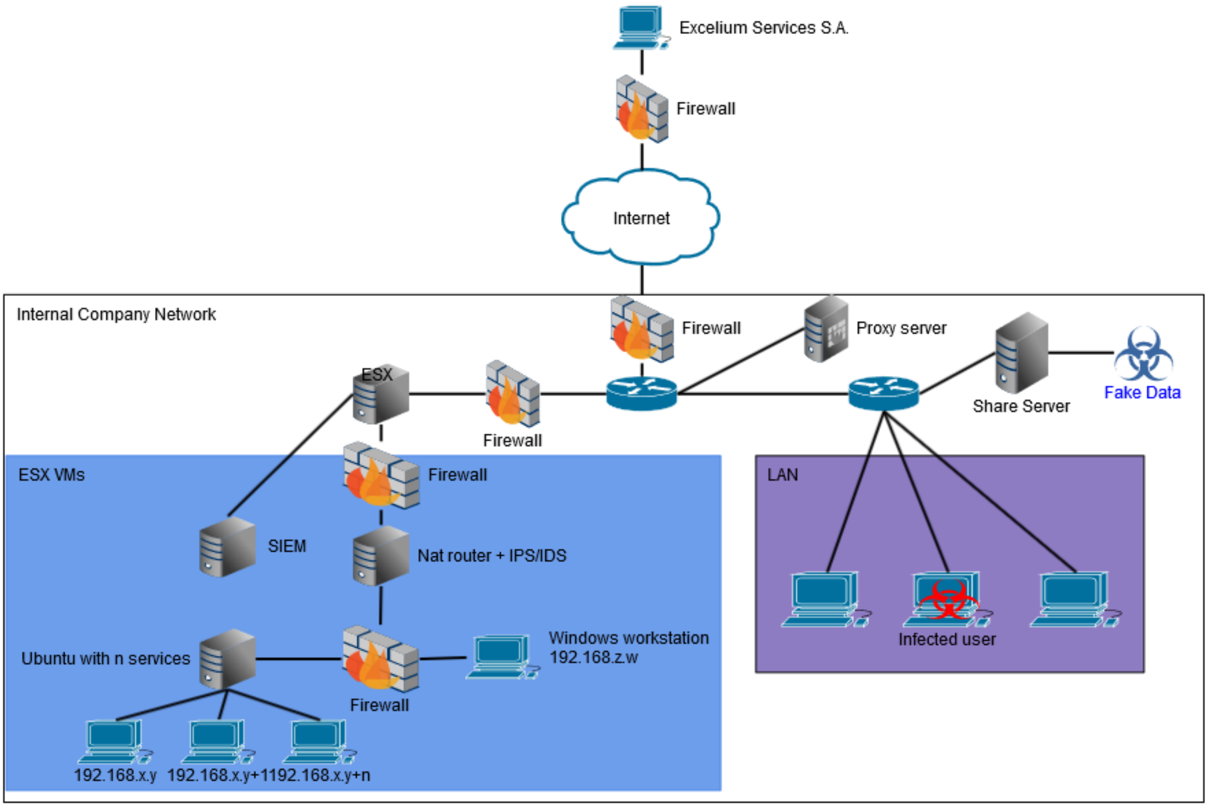


Figure 5.1: Design of the deception network

5.2 Application selected

In this part, we will discuss about the software and the OS that we deploy on the three VM's that we install on the customer ESXi.

As it can be seen in the diagram below, we use several apps. First of all we use Python 3 which is the main scripting language that we use for this application. With the help of python 3 we use Pysyslog and Heralding which are python 3 packages. The choice of Heralding was done by the need of the CSIRT to have a global scripting language: python 3. Moreover, Heralding is a very good honeypot that is easy to download and run. It provides good logging information, and this is really important for helping SOC engineers to determine if received alerts are real positive. The Ubuntu honeypot will also have a webhook server which is linked with decoy data installed on the share Server. Once these fake data will be triggered, the webhook will collect information and forward them back to the SIEM. The webhook selected is the one provided with the package webhook of ubuntu. The choice was made based on the ease of installing.

We use Cacti, an open-source web based network monitoring tool on the Ubuntu controller machine to prevent our application from saturating the network and specially not to overflow the bandwidth. The web based application needs a server web to work, that's why we also use Apache on this server.

We use Suricata (an IDS/IPS open-source) to forbid malicious activities on our VMs. The choice of Suricata over Snort was made because of the experience I had on the two IDS, on the performance on the two solutions and on the rulesets available. Moreover, Suricata logs are already parsed by Excellium SOC, so it will be easy to parse logs of the clients SIEM.

Sysmon was also something that was installed on the Windows workstation. Sysmon is a Windows service that is useful in cyberhunting because it brings more detailed logs about system activity. For example, network connection, process creation are well logged with Sysmon.

For the logging part, the module rsyslog was chosen. It appear to be a logic chose because this is the one that is provided with the installation of ubuntu. It is also one that have a lot of documentation online and my tutor knew the most.

A recap of all application used is done in the figure 5.2.

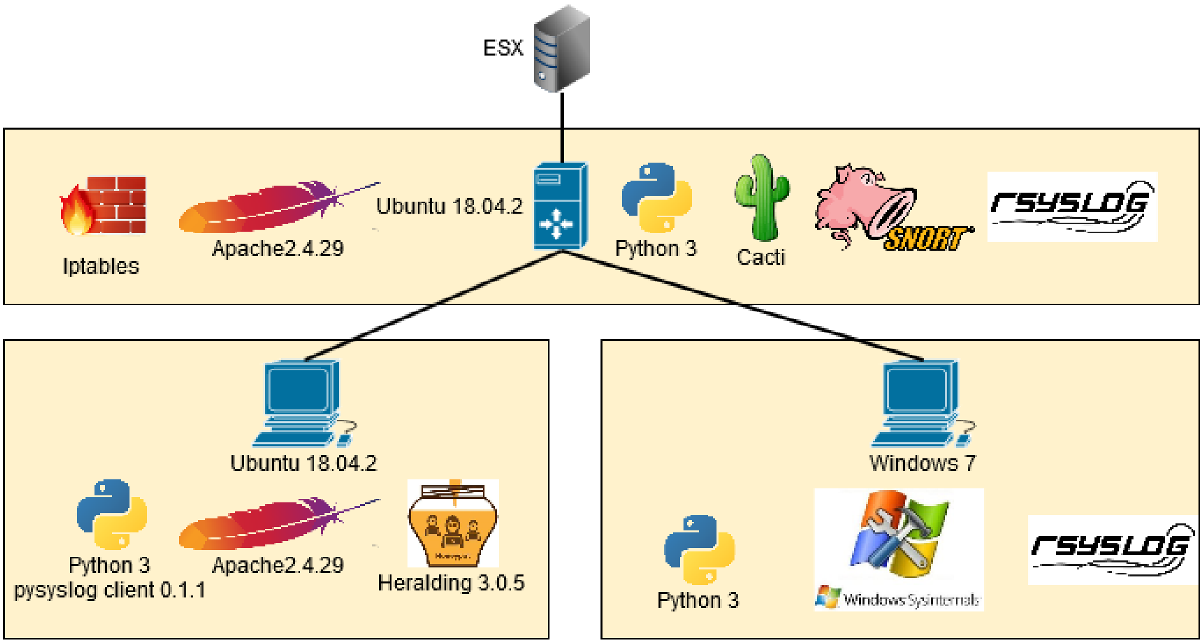


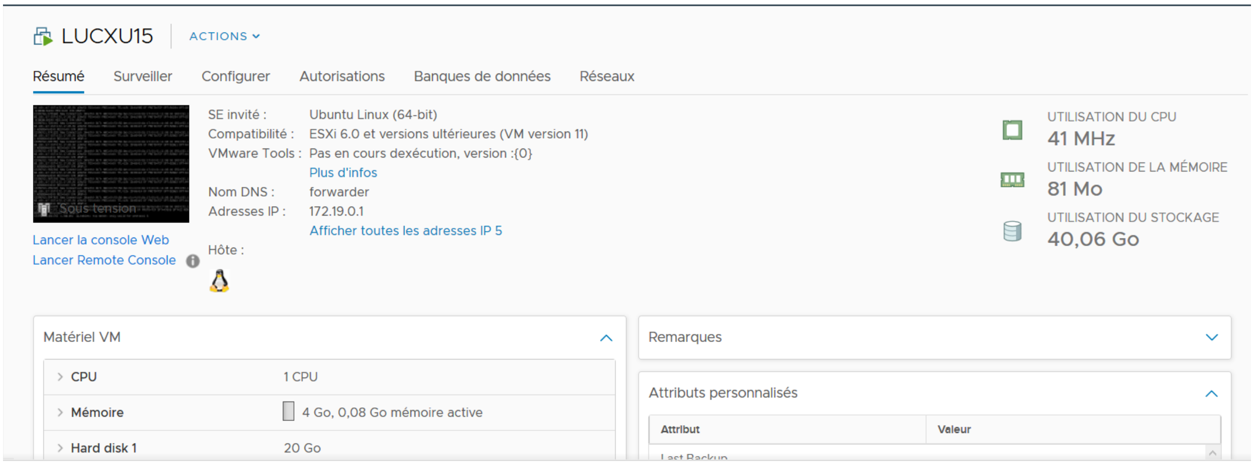
Figure 5.2: Design of the deception applications

6 UAT deployment

User Acceptance Testing consists of a process that verify if a solution works for the final user. Thus, the environment need to match as close as possible the final environment where the final solution will be implemented. The use case of the solution should also be relevant to simulate a real utilization. A good (UAT) deployment can give a first look to the client and make him see what the product looks like. (UAT)

6.1 UAT envrionment

The UAT environment for Excellium is his own ESX platform. ESX stands for the VMware ESXi product which is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers.
Virtualization has a lot of advantage. For the project, it will help by deploying easily the deception infrastructure. Indeed, in some click, an ESX administrator can deploy an OS on the network via the graphic web interface: vSphere. This interface is helpful because it allows you to have a quick look on the VM. 6.1



SE invité : Ubuntu Linux (64-bit)
Compatibilité : ESXi 6.0 et versions ultérieures (VM version 11)
VMware Tools : Pas en cours dexécution, version :{0}
Nom DNS : forwarder
Adresses IP : 172.19.0.1
Hôte :

UTILISATION DU CPU
41 MHz
UTILISATION DE LA MÉMOIRE
81 Mo
UTILISATION DU STOCKAGE
40,06 Go

CPU	1 CPU
Mémoire	4 Go, 0,08 Go mémoire active
Hard disk 1	20 Go

Attribut	Valeur
Last Backup	

Figure 6.1: vSphere interface

The interface of vSphere also allow to have a quick look of the physical element of the vm. On the figure 6.1, on the right, there are the cpu, memory and storage usage value that are shown.

This allow a quick monitoring on the VM.

Templating is also available in ESX. By simply converting a virtual machine into a template you can whenever you want to deploy the exact same virtual machine on whatever ESX you want. A template cannot be edited and is not compressed, so it is quite eavy. An alternative is to create an OVA.

6.2 Asking for the Virtual environment

6.2.1 The process

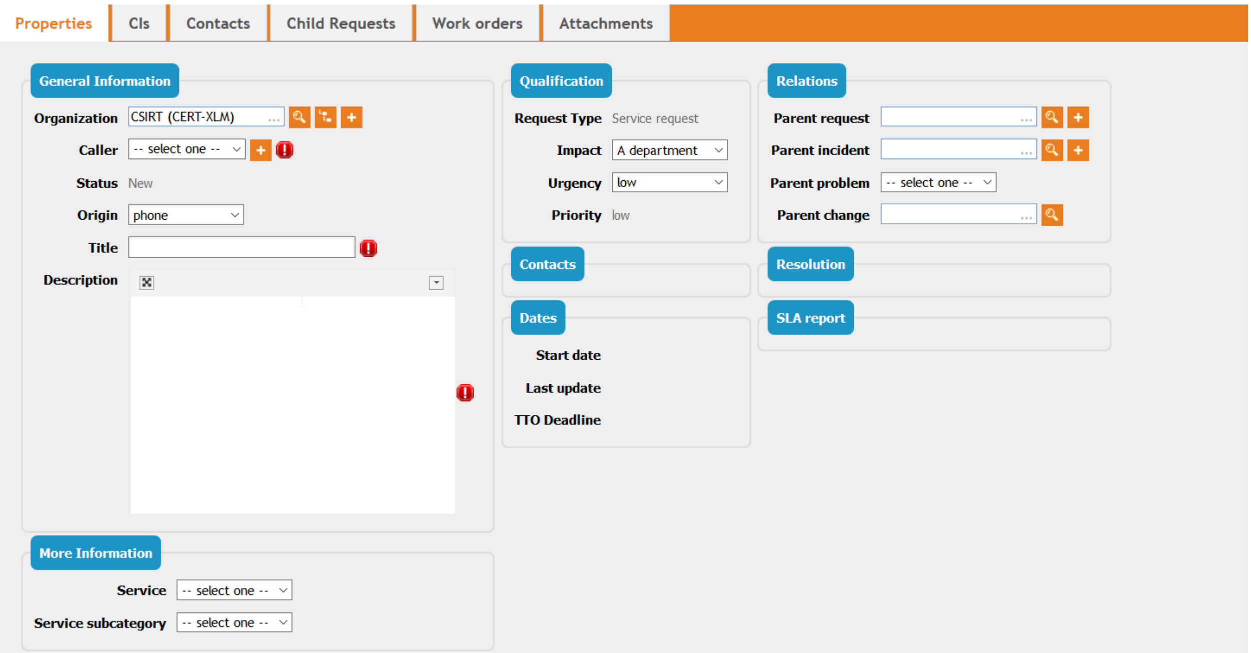
In Excellium, the process for asking for virtual assets is to ask to the IT service via the Itop interface. The figure below shows you the web interface of Excellium Itop. As you can see, you have to mention the group where you belong, the reason of the ticket creation, the importance of the urgency of the request.
Once it is done, you just have to wait for the IT response that tell you that the ticket has been resolved.
For the UAT deployment, it has been asked then to the IT:

- three Virtual hosts in the ESX,
- for these hosts: two ubuntu and one windows image,
- one Windows keys,
- access to two VLANs.

Once this was done, I had to pick some IP on the Itop interface.
Indeed, in order to map the Excellium network, in Itop, every time someone takes an IP address on production environment or in the UAT they note it in a section in Itop (see in figure 6.2). Then your IP address is reserved and the mapping done.

6.2.2 Network environment

I had to test the solution I wanted to implement in Excellium infrastructure first. So, I did a request to the Excellium IT team to ask for 3 Virtual Machines on the ESX, and an access to Excellium vSphere. They gave me what I asked and then I establish my Lab. All the computer in my lab was generated by the XLM ESX.



Organization	CSIRT (CERT-XLM)
Caller	-- select one --
Status	New
Origin	phone
Title	
Description	

Request Type	Service request
Impact	A department
Urgency	low
Priority	low

Parent request	
Parent incident	
Parent problem	-- select one --
Parent change	

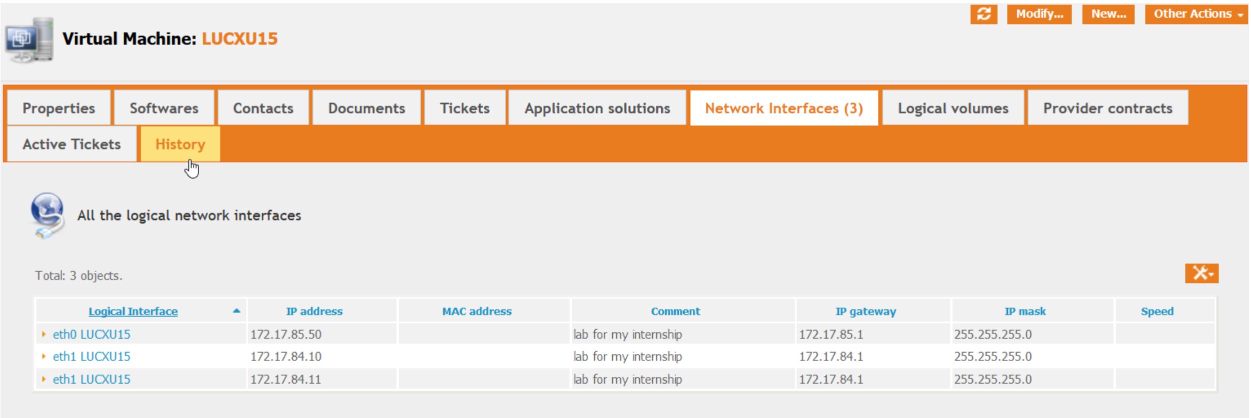
Start date

Last update

TTO Deadline

Service	-- select one --
Service subcategory	-- select one --

Figure 6.2: Itop interface for ticketing



Virtual Machine: LUCXU15

All the logical network interfaces

Total: 3 objects.

Logical Interface	IP address	MAC address	Comment	IP gateway	IP mask	Speed
eth0 LUCXU15	172.17.85.50		lab for my internship	172.17.85.1	255.255.255.0	
eth1 LUCXU15	172.17.84.10		lab for my internship	172.17.84.1	255.255.255.0	
eth1 LUCXU15	172.17.84.11		lab for my internship	172.17.84.1	255.255.255.0	

Figure 6.3: Itop interface for interfaces / IPs

6.3 Installation from scratch

Before doing the templating process, I had to make a clean installation. For the ubuntu workstation, a process was available on confluence.

6.3.1 Ubuntu Installation

NTPD

The time of the computer are defined by the NTPD server of the UAT. This is important to have the same NTPD on each workstation because in case of an accident, log timing have to be right to be able to investigate and correlate.

DNS

The DNS servers are the one of the UAT. Dns is important during the UAT deployment for accessing to the installation repository of package that have to be installed.

Proxy

The Proxy server is the one of the UAT. For the same reason that explained above, proxy access need to be set up for the installation process. It is also important to have an internet connection for updating process. Indeed, if the solution is installed on the client infrastructure and a vulnerability of the OS or in some service occur, update should be installed.

Ipv6

IPv6 is disable on grub configuration to avoid trouble that can occure with ipv6.

SSH

SSH server was installed to make the installation and have a easy access of the virtual machines. A strong password was generated and stored in a Keypass.

Log forwarding

All the emergency logs are, by default, send to the Qradar. All the logs are managed by the Rsyslog service.

6.3.2 Forwarder Installation

Looking back to the design of the deception network as shown on figure 5.1 page 32 we see that all the traffic will be passing through a NAT router. We will call him forwarder for the rest of the section.

Iptables

As told before, forwarder is a NAT router. To do this, iptables was the good chose because we can do it easily on a Ubuntu workstation in combination with an IPS. Iptables is provided with a basic ubuntu installation, so there is no need to install the package.
All that need to be done is configuration. In our case, all the traffic that comes to the interfaces belonging to the 'bait' vlan will be forwarded to the decoys servers. Before this, theses packets will pass through the IPS for security issue. For passing through the IPS, the packet pass via the table NFQUEUE. NFQUEUE table which delegate the decision on packets to a userspace software. Here this is Suricata which will deal with the packet.

Suricata

Suricata was put in IPS mode. The rules set use is the basic one provided. Some rules are bloquant and other just in warning mode. All the log comming from the suricata are sent to the Qradar.

Canary Token

An instance of Canary Token server is running on the port 80 of the forwarder IP : 172.17.85.50. It is here to provide informations about canary tokens which are triggered. It is also here to produce all the documents that will be spread in the network.

6.3.3 Heralding Installation

Heralding

Here we are talkig about heralding python module. It was installed with pip3 which is the package management system of Python. All the services are active and logging is also active. All the logs are forwarded to the Qradar via the forwarder endpoint.

Webhook

A webhook is installed on the heralding workstation. A webhook is just an http / https server that listen on a dicted port. Once someone POST some data on this webhook, an or many actions are defined to be executed. He is here to gather all the event triggered by the canaryTokens. Once it receives something, it just send the raw data to the Qradar via Syslog. Of course all the traffic go first by the workstation forwarder.
On the UAT, the webhook is reachable by the address : http://172.17.84.10:9000/hooks/canary.

6.3.4 Windows 7 Install

ntpd

The time of the computer are defined by the NTPD server of the UAT.

DNS

The DNS servers are the one of the UAT.

Proxy

The Proxy server is the one of the UAT.

Ipv6

IPv6 is also disable by default.

RDP

An audited RDP is also configured on the windows workstation. He is protected by a strong password which is registered in a Keypass. It is used to see any logon to the RDP service but also for administrative tasks. Indeed the instalation was done via a RDP connection.

Sysmon

Sysmon is a tool available on Windows that give you more logs with more information that security logs gives. The sysmon is installed and configure to send log to the Qradar.

Share

A windows share is build in this workstation. It is shared with anyone so anyone in the network can access it. When a person try to log into this shared folder, an alert is triggered. Canary token are dropped in the shared folder. Any use of the credentials included in the canary token or any oppening of the canary tokens will also triggered alerts.

6.4 Network installation

This installation in a layer two view can be seen in the figure 6.4. In the excellium UAT, two Vlans are attributed to the project. The Vlan 605 vaw supposed to be the administration one and the Vlan 604 the public one. For a easy use, The LUCXU15 workstation had two IPs address on the VLAN 604. On the VLAN 605, all the workstation have one IP address. The workstation LUCXU15 is worksing as a NAT router forwarding the traffic comming on VLAN 604 to LUCXU16

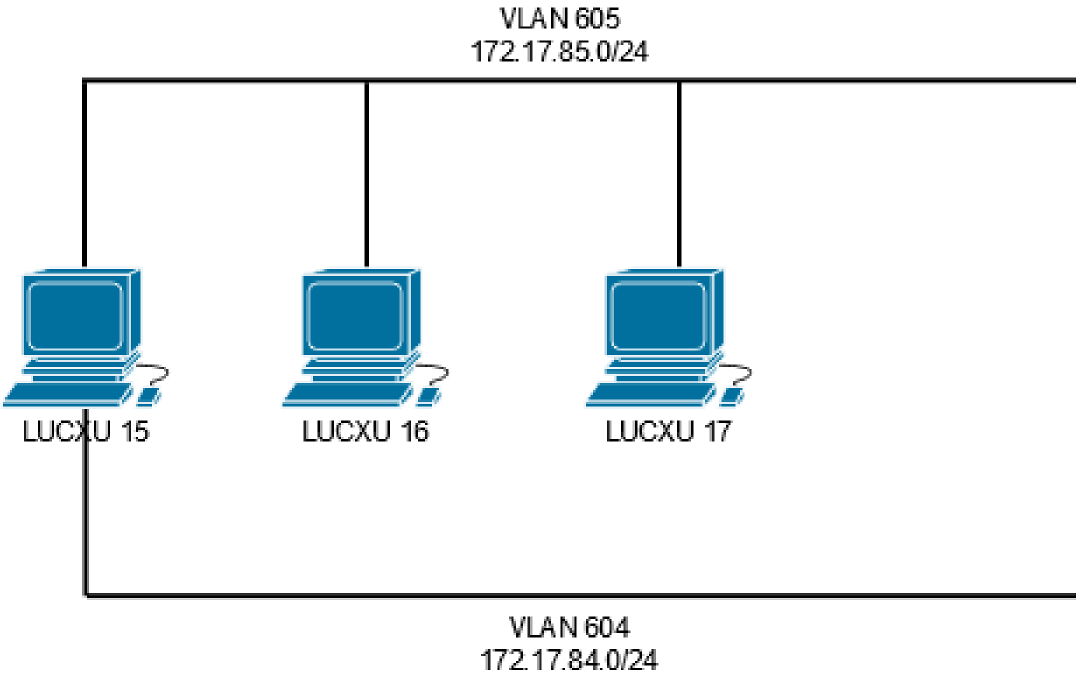


Figure 6.4: Network schema

and LUCXU17.

6.5 Documentation

During the UAT process, one important thing is to keep a trace to everything that is made. In order to do this, confluence is used. Confluence is a wiki webserver used as a co working software. A trainee section was reserved for the project and each part of the project is documented on it.

For an easy reuse, a document presenting an installation was done with a lot of screen-shot and command lines.

6.6 Log Management

One problem that is present is how bring clear, synthetic and understandable log to the SIEM. This was not that simple because according to the network environment as shown on figure 6.4. Indeed, relevant logs that was coming from LUCXU16 and LUCXU17 have as IP source LUCXU15 (because of natting).

For example, an example of a log is :

```
<156>Aug 20 15:41:47 heralding heralding {"timestamp": "2019-08-20 13:41:44.925163", "duration": 0, "session_id": "9c56927d-97a0-4039-b796-e9401cd145cd", "source_ip": "172.17.85.50", "source_port": 49316, "destination_ip": "172.17.85.51", "destination_port": 443, "protocol": "https", "num_auth_attempts": 1, "auth_attempts" : [{"timestamp": "2019-08-20 13:41:44.925968", "username": "nasty_user", "password": "nasty_password"}], "session_ended": true, "auxiliary_data": {"Host": "172.17.84.10", "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0", "Accept": "text/html, application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "Accept-Language": "fr, en-US ;q=0.7,en; q=0.3", "Accept-Encoding": "gzip, deflate, br", "Connection": "keep-alive", "Upgrade-Insecure-Requests": "1", "Authorization": "Basic YTph"}}
```

This is quite a long log but this is quite easy to parse so it is not such a problem. The context of this log is, for a testing purpose, I tried (IP address 192.168.180.117) to join the server web on the IP address 172.17.84.10 with the credential nasty_user:nasty_password. The LUXCU15 workstation NAT the address to the honeypot server hosted on the LUXCU16 workstation. But the problem is here, if a SOC analyst sees this log, he will understand that someone is trying to connect to the honeypot but will not be able to investigate because the source IP address is the one of the LUCXU15 workstation (the nat router).
This is why what was done to make this problem not that important was to also generate nat log

on the LUCXU15 workstation. Fortunately, iptables that is used to NAT allow some log. Here as the log in question:

```
<4>Aug 20 15:41:34 forwarder kernel: [4056859.198406] New Connection IN=eth1 OUT=eth0
MAC=00:50:56:be:3b:2a:00:1b:17:00:01:11:08:00 SRC=192.168.180.117 DST=172.17.84.10 LEN=52 TOS
= 0x00 PREC=0x00 TTL=126 ID=51039 DF PROTO=TCP SPT=49314 DPT=443 WINDOW=64240
RES=0x00 SYN URGP=0
```

We can see that there are the IP origin and the IP destination here. The correlation of the two previous log are now good for an analyst to investigate.

6.7 Accepting Test

For approuving the solution, several tests are done. Two of them are presented here. The first test that seems to be relevant was the most obvious one. Let's begging with Nmap scanning. Nmap is a free open-source software used to identify open ports, operation systems and get information about services running on a specified address IP. We can see below the response of nmap scanning on the two exposed IP on the VLAN 604.

```
Nmap scan report for lucxu15.int.excellium.lu (172.17.84.10)
Host is up (0.015s latency).
Not shown: 984 closed ports
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
23/tcp open telnet
25/tcp open smtp
80/tcp open http
110/tcp open pop3
143/tcp open imap
443/tcp open https
514/tcp open shell
993/tcp open imaps
995/tcp open pop3s
1080/tcp open socks
3306/tcp open mysql
5432/tcp open postgresql
5900/tcp open vnc
9000/tcp open cslistener
```

This is working, all services are on one single port but in production, there will be on several IPs addresses. Looking to Qradar after this is showing a lot of alert showing that the system was scanned.

The Canary tokens also have to be tested. For an easy test, a docx file was created with the name secret_password.docx. After having successfully created this document. The file was dropped on the Windows Share. When someone opened the file, this log was send to the SIEM :

```
<164>Aug 20 15:32:26 heralding webhook "additional_data":"location":null,"referer" : null,"src_ip"
: "192.168.180.117","useragent": "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64;
x64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR 3.0.30729; .NET CLR 3.5.30729;
ms-office; MSOffice 16)","channel": "HTTP", "manage_url" : "http://172.17.84.10/manage?token=[...]
auth=62f02306e397af03283b66a606c23a7a" ,"memo":"Test Meeting Mathieu","time":"2019-08-20 13:32:18"
```

This is a very good log because in one log an investigation could be made. The IP address of the computer that triggered the document is here, there is also the remember message and the link of the server where there is more document available.

7 Client deployment

7.1 Preparation

Before coming into the client, a preparation sheet should be written. In the following of UAT implementation, two documents have been written. The first one is a document dealing with the installation process and the second one is a document that needs to be filled by a client network administrator.

The first document is written with the help of all the note takken during the UAT devlopment. It defines all the processes that need to be done. First this is how to set up an installation from a template, then this is what script or command to run with what argument.

The second document is sent to the client. This is a document that is sent in order to gather private informations that will be needed for the deception system installation. The list of information ask is:

- DNS IPs,
- Fixe IP address,
- Vlans accessible,
- Email of the supervisor,
- Phone number of supervisor.

7.2 Deployment

Unfortunately, at a time where the memory is wrote, the deployment is not yet finished.

8 Review of the Project

This was a project that last six months. After a time like this, it is mandatory to do a recap of what was done, was could have been better and what was good. It is also important to see where the project goes and what are its impacts.

8.1 Axes of improvement

Unavailable, after six months errors occur and lessons have to be taken.
The first thing that could have been improved is my research process. I think that I should have maybe spent less time at the beggining, do some design and then do again some research. This could have help me to face some technical issues. I also think that I should have been more precise during the creation of the Gantt because during the middle of the internship, I lost myself doing some research. This could be avoided if every task as a beggining and an ending phase.

8.2 Project becoming

The project deliverable is available via confluence access to the trainee folder. The project, after what I've done will be taken by the members of the CSIRT. There will aslso have some training sessions to SOC engenieers and analysts for explaining them what is the project about and to what correspond the alerts.

8.3 Excellium Impact

This part will be dealing about the impacts that the project had on Excellium and the CSIRT. Firstly, the project allows Excellium-services to beef up their portfolio of services that they can provide. That means that if some client of Excellium heard about deception or honeypot technologie, Excellium can provide them my project. One other impact is that the global knowledge of Excellium is growing. Indead, the work I did about state of the art and the deployment is steal here and can help for other projects. The POC that I did will be a good start for anything looking like that.

9 Project Management

The project assigned to me was a complete one. I needed first to understand what was the technologies already in place, what design I wanted for the project, I also needed to implement a test of this solution in Excellium UAT and finally I had to implement it in a client infrastructure.

With this in mind, several parts of the projects (see picture 9.1) came up:

- State of the Art: the part of gathering information,
- Design: the part of designing the solution,
- Deployment UAT: the part of the first deployment and testing the solution,
- Deployment Production: the part where the solution is test on a client infrastructure.

With this in mind, and some discussions with my industrial supervisor, we came up with a previsional Gantt which is shown on the top of the figure 9.1. The beginning of the project was perfectly respected. The Deployment on the UAT of Excellium took more time that expected because of some trouble that I had technically speaking and also because I had a car accident that put me in labour disruption. As a result, as it is shown in figure 9.1, three more weeks was spent on the UAT deployment.

Some other tasks was obviously present during the project and planned on the Gantt but they are not shown in the figure 9.1 with a view to making things clearer. There are the detail presentation of theses tasks not mentioned on the Gantt:

- Project Management: to have a better view of the project, the plan was to have a weekly review on the Friday with my industrial manager. Those weekly review have two main goal, the first is to see if the project is going well and if we have to change something in the Gantt. The second goal is to see if I have any question and giving me path of reflections.
- Reporting: a part of my working hours was dedicated to the writing of my master thesis. Four hours was dedicated to the writing of my report every week.

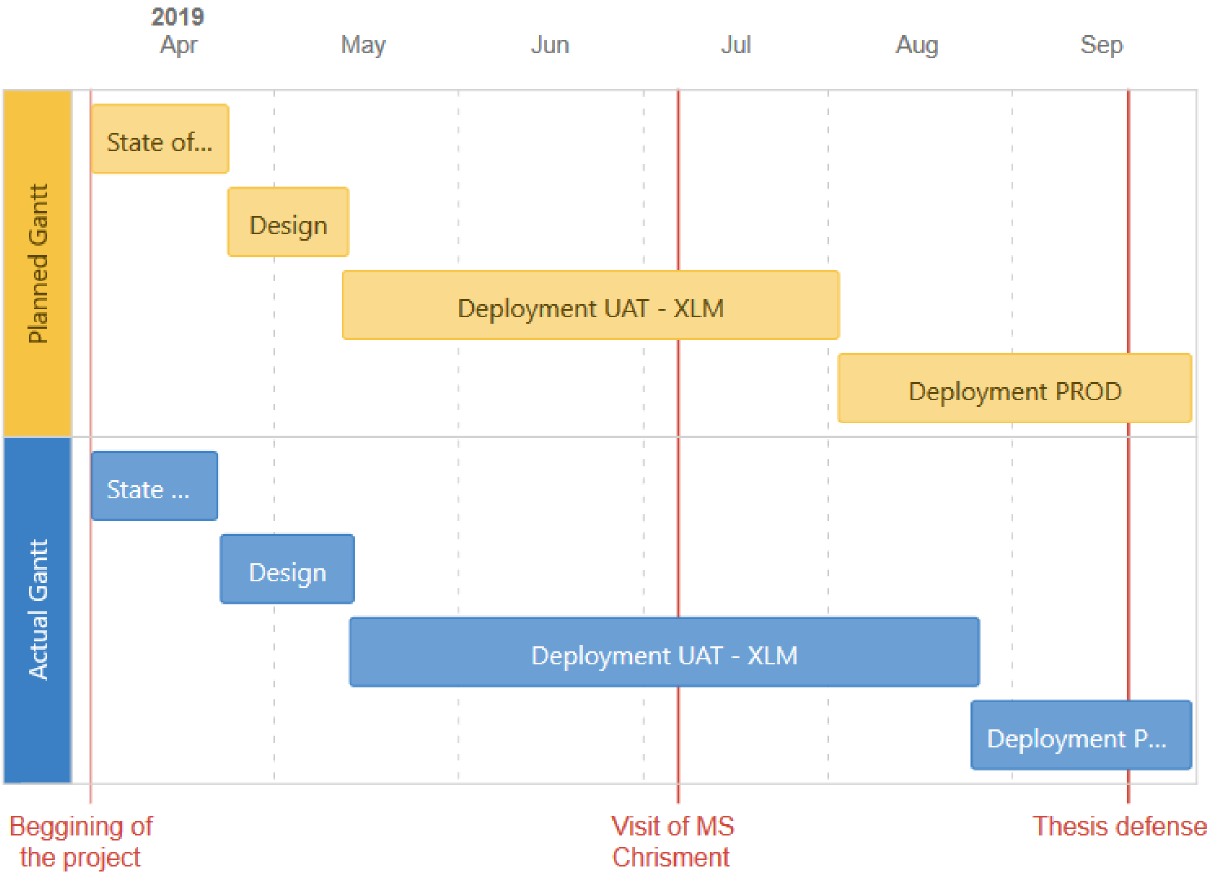


Figure 9.1: Previsional Gantt vs actual

- Documentation: this task goes in pair with the previous one. When the previous was for academical purpose, this one was for industrial purpose. Indeed, knowing the fact that the project will maybe be taken by someone else after my internship, all that I did need to be documented for a quick understanding. The time allowed to this was share with the reporting one because, a lot of time, what was put on my documentation section was also in my report, and vis versa.

- Training: Some time, some training was given in the company and I had the chance to attend them. The time dedicated to this was very punctual, I spent twenty hours at the most.

- CERT XML duty: As explained in the section 2.2.5, I had the chance to see and do some actual daily CSIRT's job. The time spent for each activity was very variant. The news lasts one hour when the Newsletter lasts one complete day.

10 Conclusion

The project objective was to build a deception system from A to Z. I first learn a lot about deception technology as an all and wrote a state of it. Then, I had to think about how doing it and how to present it. After, I built a first version of what I designed and test it. Finally I tried it on a client environment. We could say that the objective that was fixed in the begging was followed.

I learned a lot during this project regarding many topics. Of course about cybersecurity but also about networking that is a mandatory topic in computer science. I also learned how to lead a project from its begging to a production and all the project management that is behind. This is also a great feeling to apply what I study during my years in Telecom Nancy. I could see that something that I thought meaningless could be very important and also what I thought important could not be. But most importantly I think that this internship was a human journey. It makes a bridge between student life and the adult working life that I will certainly have for the forty next year of my life.

The project strengthens my wish of working in the cybersecurity area and helping build a safer numeric world.

Bibliographie / Webographie

[1] https://attivonetworks.com/. Attivonetworks Website. 16

[2] https://commons.wikimedia.org/wiki/file:tables_traverse.jpg. Image iptables. 29, 55

[3] https://cymmetria.com/. Cymmetria Website. 16

[4] https://en.wikipedia.org/wiki/kill_chain. Kill Chain article in Wikipedia. 29, 55

[5] https://github.com/paralax/awesome-honeypots. An awesome list of honeypot resources. viii, 20

[6] https://github.com/samratashok/deploy-deception. Deploy-Deception github. viii, 17

[7] https://github.com/thinkst/canarytokens. Canary Tokens github. viii, 18

[8] https://www.acalvio.com/product/. Alcavio Website. 16

[9] https://www.cert.at/. Austrian Cert Website. 6

[10] https://www.crunchbase.com/organization/countercraft#section-current-team. CounterCraft Website. 16

[11] https://www.gartner.com/en. Gartner website. 9

[12] https://www.secureworks.com/blog/dcept. DCEPT: An Open-Source Honeytoken Tripwire. viii, 20

[13] Anna Belak Augusto Barros, Anton Chuvakin. Applying deception technologies and techniques to improve threat detection and response, 2019. Gartner Report. 9

[14] Thomas Erl. Service-oriented architecture: Concepts, technology & design. about orchestration. 23

[15] Excellium-Services. https://excellium-services.com/blog/, 2019. Excellium Blog. 7

[16] Rajpreet Kaur Gorka Sadowski. Improve your threat detection function with deception technologies, 2019. Gartner Report. 9, 12, 55

[17] Lockheed Martin. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html. Cyber Kill Chain. 24

[18] Saiyan Saiyod Nattawat Khamphakdee, Nunnapus Benjamas. Improving intrusion detection system based on snort rules for network probe attacks detection with association rules technique of data mining, 2016. 19, 28, 55

[19] Shodan. https://www.shodan.io/. Search engine for Internet-connected devices. 25

List of Figures

List of Tables

Glossary

C2 Command and Control, also refered as C&C. 28

CERT Computer Ermergency Response Team. Equivalent to CSIRT. vii, 4–6

CPE Common Platform Enumeration. 6

CSIRT Cyber Security Incident Response Team. v, 5–8, 14, 31, 48, 53

CVE Common Vulnerabilities and Exposures. 5, 6

CVSS Common Vulnerability Scoring System. 6

DiD Defense in Depth. 10

DNS Domain Name System. 11

Excellium Excellium Services S.A.. v, ix, 3–8, 10, 13, 14, 30, 33, 34, 45, 47, 53

IDS Intrusion Detection System. 13, 21, 22, 29–31

IPS Intrusion Prevention System. 13, 21, 22, 31, 37

NTP Network Time Protocol. 11

POC Proof of Concept. 13

RDP Remote Desktop Protocol. 11

SIEM System Information and Event Management. 1, 10, 11, 30, 31

SOC Security Operations Center. 1, 6, 10, 30, 31

TDR Threat Detection and Response. 9

UAT User Acceptance Testing. viii, 33, 34, 36, 38, 40, 47

Résumé

Ce mémoire d'ingénieur décrit le projet que j'ai mené lors de mon stage de fin d'étude dans l'entreprise Excellium au Luxembourg. Lors de ce stage, j'ai été amener à élaborer et à mettre en place un système de deception. La deception est un concept qui reprend le concept plus connu des 'honneypots' ou pots de miels qui ont pour principe de leurrer un potentiel attaquant en imitant un faux serveur. Le principe de deception est un peu plus étendu que celui des pots des miels et ne se limite pas qu'à la création de faux serveurs. La solution que j'ai élaborée est détaillé et chaque choix effectué est justifié. Une partie de gestion de projet et un bilan du projet sont également disponible.

Mots-clés: Deception, Honeypot

Abstract

This master thesis describ the project that I realized during my final intership in the company Excellium in Luxembourg. During this internship, I was led to develop and set up a deception system. Deception is a concept that takes up the well-known concept of honeypot which principle is tricking a potential attacker mimicking a fake server. The principle of deception is a little more extensive than that of honeypots because is not limited to the creation of fake servers. The solution I developed is detailed and each choice made is justified. Project management and a review of project are also available.

Keywords: Deception, Honeypot