- *Red Canary*: `https://redcanary.com/blog`
- *Abuse.ch*: `https://abuse.ch`
- *SANS Internet Storm Center*: `https://isc.sans.edu`

I would like to mention Twitter. It is a tremendous, simply tremendous, place to find truly amazing, novel, and thought-provoking campaign analysis and technique research – but as often as it can be helpful, it can also provide conjecture, incomplete analysis, or even theory purveyed as fact. Find the Twitter accounts that provide information that you're most interested in and carefully curate those that you follow for threat hunting.

In this section, we discussed some common external sources that can assist in honing your hunting techniques as you develop as a practitioner.

Summary

In this chapter, we discussed an IR process and looked at the different phases and examples of how they are leveraged during an incident. We explored how to use the MITRE ATT&CK framework and Lockheed Martin's Cyber Kill Chain model to analyze a supply chain compromise example and inform security priorities. Finally, we discussed several sources for expanding and growing your skills as a threat hunter.

Using the skills we covered in this chapter will make you valuable beyond your ability to find adversaries. While that is crucial in your job as a threat hunter, being able to support the enduring security teams and prioritizations helps the overall posture of the organization.

In the next chapter, we'll discuss enriching events with open source tools, enriching events with third-party tools, and using enrichments to explore additional information.

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. Intrusion grouping and attribution cannot be assessed with a single observed tactic, technique, or sub-technique.

   a. True

   b. False

2. This IR phase is often rushed, but it will frequently lead to reinfection.

   a. Recovery

   b. Preparation

   c. Detection

   d. Eviction

3. A tactic for informing investment priorities can be accomplished by:

   a. Driving the adversary back through the Kill Chain

   b. Making a plan for additional resource requests

   c. Using the Diamond model to describe an adversary

   d. Showing how easy it would be to compromise a sensitive system

4. This IR phase involves validating that steps identified in the eviction phase are carried out.

   a. Preparation

   b. Lessons learned

   c. Containment

   d. Recovery

5. Improving your threat hunting skills requires purchasing training.

   a. True

   b. False

Further reading

To learn more about the subject, I recommend the following source:

- *Incident Response*; *O'REILLY Computer Security Incident Handling Guide*; **National Institute of Standards and Technology** (**NIST**): `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf`

11
Enriching Data to Make Intelligence

In *Chapter 1*, *Introduction to Cyber Threat Intelligence, Analytical Models, and Frameworks*, we discussed the intelligence pipeline and the process of making data into intelligence through analysis, production, context, and enrichment. Enrichment is one of the final steps in transitioning collected data into something that can be actioned for further hunting or defensive considerations by the incident response teams.

In this chapter, you will learn how to use various tools to enrich both local observations and threat information to add contextually relevant information to events in their journey to actionable intelligence.

In this chapter, we're going to cover the following main topics:

- Enhancing analysis with open source tools
- Enriching events with third-party tools
- Enrichments within Elastic

Technical requirements

In this chapter, you will need to have access to the following:

- The Elastic and Windows virtual machines built in *Chapter 4*, *Building Your Hunting Lab – Part 1*
- A modern web browser with a UI

Check out the following video to see the Code in Action:
`https://bit.ly/3xJB5oZ`

Enhancing analysis with open source tools

Throughout this book, we've leaned heavily, if not exclusively, on open source software to achieve our analytical goals. From building our sandbox to analyzing malicious files and network traffic, almost everything we as analysts and hunters do can be derived from the open source community.

When I first started exploring IT security, I was suspicious of open source software. My thought, like many who were new to this discipline, was that if it's open and available, anyone can see how to exploit it. If you had a closed system, those security holes could never be known, and thus never exploited.

If I fast-forward 25 years, I know that was a naïve understanding of the open source community and now realize it as the cornerstone of so many popular and almost required tools for performing analysis.

In the next section, we'll talk about the MITRE ATT&CK Navigator to view and analyze focused TTPs.

MITRE ATT&CK Navigator

We've talked about the MITRE ATT&CK framework multiple times throughout this book, but we're not quite done yet. MITRE has made an open source web application that we can use to view and analyze tactics, techniques, sub-techniques, software, and groups. Not only is this tool helpful for raw research, but we can also use it to identify collections or analysis gaps. This tool is called the ATT&CK Navigator.

> **Important note**
>
> The ATT&CK Navigator provides tremendous value to enhance your analysis by understanding where an observed tactic, technique, or sub-technique may be in the intrusion life cycle. When analyzing events, the Navigator is a visual aid that not only allows you to see where your event may be in a campaign but can also be used as a way of identifying other unknown techniques and sub-techniques that could be related. This is about enhancing your analysis of a campaign instead of enriching an event with contextual metadata.

The Navigator web app, which can be run locally or through MITRE's infrastructure, allows you to add information on top of the ATT&CK matrices in a process called **layering**.

We can begin by browsing to the Navigator web page, `https://mitre-attack.github.io/attack-navigator`, where we can start by selecting a blank layer by clicking on **Create New Layer**:

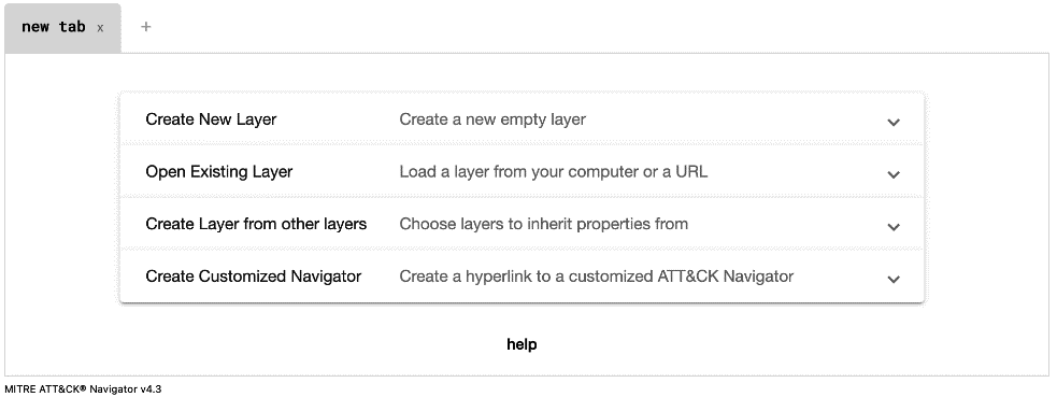


Figure 11.1 – The ATT&CK Navigator opening page

Click on **More Options** and then select **ATT&CK v9** for the version and **Enterprise** for the domain, as we can see in the following screenshot:

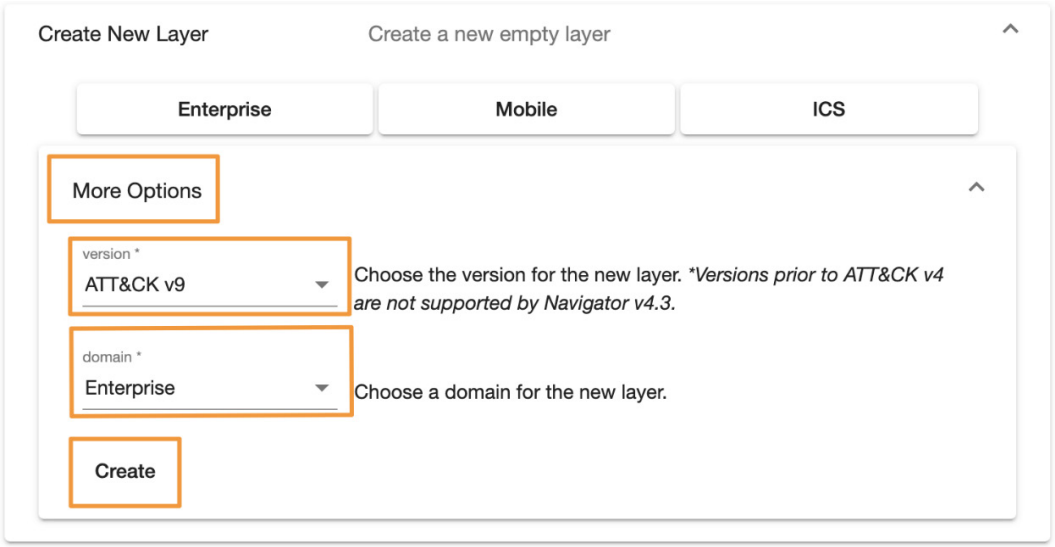


Figure 11.2 – ATT&CK Navigator creating a new layer

Now we can see a few of the different tactics and techniques in the Enterprise domain of (in our case) version 9 of the ATT&CK matrix:

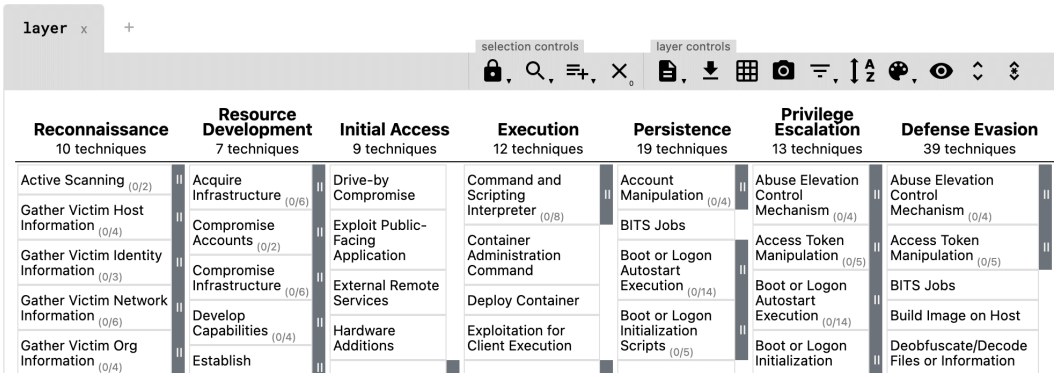


Figure 11.3 – The ATT&CK Navigator's default layer

From here, we can expand the techniques and sub-techniques by clicking on || to review them, as we're doing with this example of the techniques and sub-techniques for the **Scheduled Task/Job** technique within the **Execution** tactic:

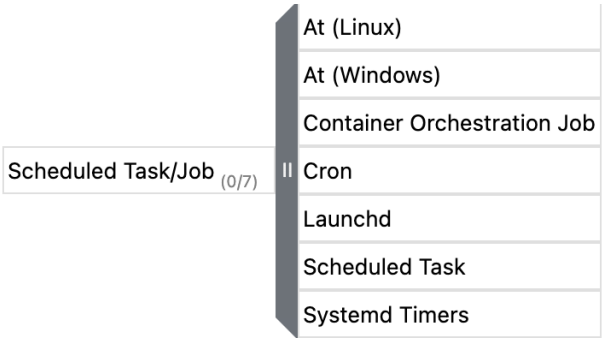


Figure 11.4 – The ATT&CK Navigator's expanded technique menu

While this is a different view than what is available on the main ATT&CK website, this isn't terribly novel.

So, what MITRE did was allow us to apply filters to the data to identify things that are associated that may be of particular interest. As an example, if our organization stored its most critical intellectual property on a set of Linux systems, we could apply filters here in the Navigator to only show techniques and sub-techniques that have been observed as targeting Linux systems:

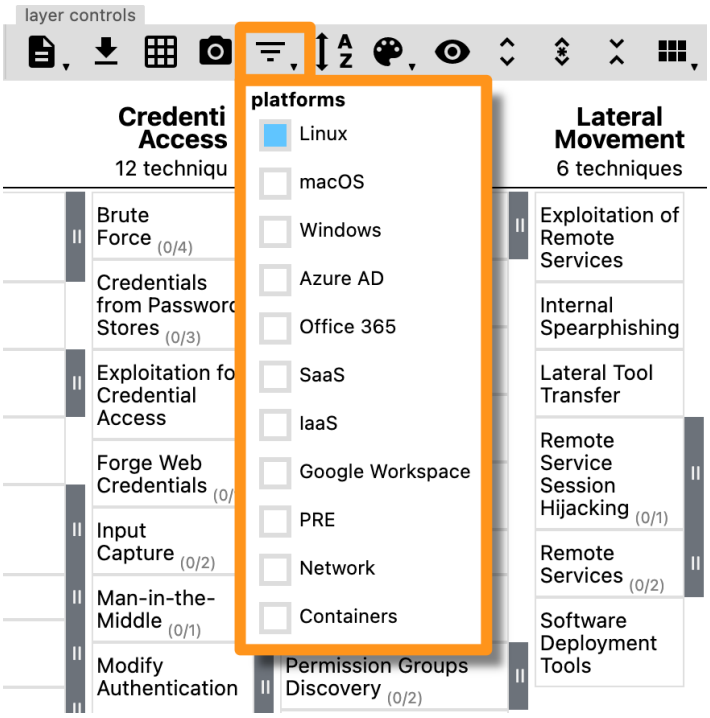


Figure 11.5 – The ATT&CK Navigator's filtering capabilities

Furthermore, if we were considering implementing three new enhancements to the security posture of an organization, we could use the Navigator to find out what tactics, techniques, and sub-techniques would be impacted by those changes. In the following example, I selected **Antivirus/Antimalware** as a mitigation and the Navigator highlighted the tactics, techniques, and sub-techniques that would be impacted by those mitigating steps:

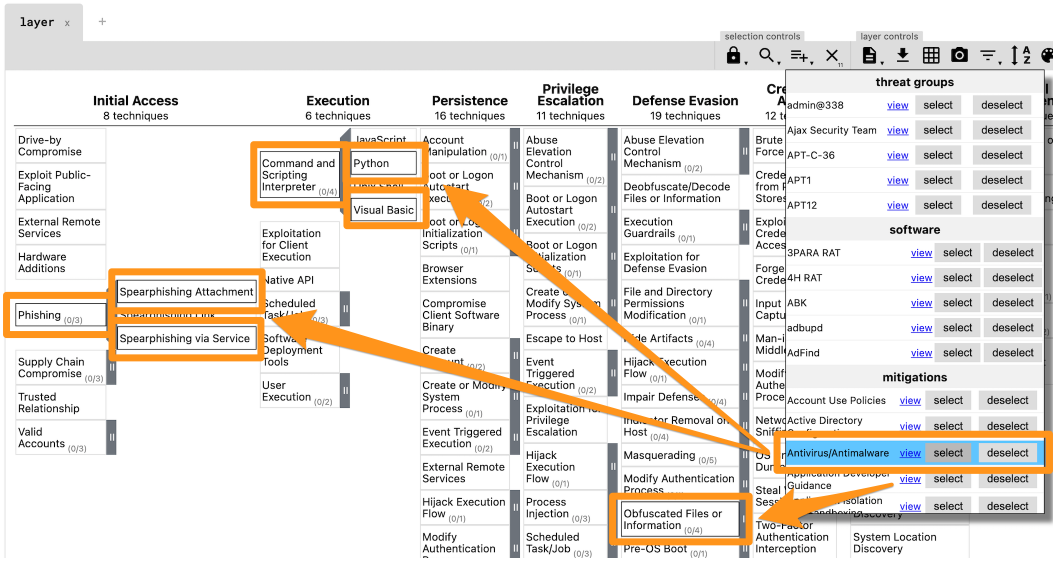


Figure 11.6 – The ATT&CK Navigator selecting Antivirus/Antimalware mitigations

This is even more helpful when you have three possible changes to the security program, but only enough resources for one. As we discussed in *Chapter 10*, *Leveraging Hunting to Inform Operations*, this can help prioritize resources to make the biggest impact on an adversary campaign.

Expanding on the mitigating steps, you can apply selections for specific groups to highlight the known TTPs or software titles to identify observed capabilities and behaviors.

In this section, we introduced the ATT&CK Navigator, an extremely powerful tool to *theorycraft* your existing and future security posture, as well as assisting you in hunting, defensive prioritization, or incident response. Unfortunately, this was just an introduction, but there are resources in the *Further reading* section of this chapter that provide information on additional capabilities of the tool.

In the next section, we'll use third-party tools to enrich events and observations.

Enriching events with third-party tools

In the previous section, we worked a bit with MITRE's ATT&CK Navigator, a powerful tool that will allow you to shape and prioritize defensive considerations. Next, we're going to look at tools that can be used to enrich technical data observed during a hunt or incident response.

IPinfo

IPinfo is a website that can be used for free to gain insights into IP addresses, such as where they are located geographically, who owns them, and the hostname assigned to the IP.

This information can be collected from their website or using their exposed API endpoint, which is faster and can be done anywhere you have a command prompt.

To start, you can browse to `https://ipinfo.io` and either create an account or see basic information without registering. While that can be helpful, let's query the API to get information about IP addresses we may identify during a hunt. To do that, open a command prompt and use the cURL program (this is built into all modern OSes) to run the following command:

```
$ curl ipinfo.io
```

This will return your public IP address and some basic information about it. Now, if we want to use this to get additional information from an IP address during a hunt, we can simply add that to the `curl` command. I'm going to use an IP address from a public threat feed:

```
$ curl ipinfo.io/64.225.18.241
{
  "ip": "64.225.18.241",
  "city": "Clifton",
  "region": "New Jersey",
  "country": "US",
  "loc": "40.8344,-74.1377",
  "org": "AS14061 DigitalOcean, LLC",
  "postal": "07014",
  "timezone": "America/New_York",
  "readme": "https://ipinfo.io/missingauth"
}
```

The preceding code shows the output from the API endpoint of the IPinfo service.

> **Note**
>
> The `missingauth` output is because we're not registered for an account. If you register for a free account, you'll get an authentication token (I recommend this method):
>
> `$ curl ipinfo.io/64.225.18.241?token=1234567`

In this section, we discussed using IPinfo as a way to gain additional information about IP addresses.

In the next section, we'll use an open source tool to get additional information about files.

Abuse.ch's ThreatFox

Abuse.ch is a platform that publishes threat information based on their own research as well as samples submitted by the information security community. ThreatFox was released in March 2021 as a way to research and share indicators.

As with most tools, it leverages a powerful API. There is a lot that can be done with their API; we'll look at an example of querying a malware sample and then you can continue to explore all the facets on your own as an exercise left to you.

Again, using cURL, we'll query the API for ThreatFox to get some information on a credential stealer called `ArkeiStealer`.

As this command is a bit more complex than what we've done previously, let me walk through what we're doing:

- `curl`: This is the binary that we're running to make a web request.
- `-X POST`: This changes the default HTTP request from GET to POST, because we're going to be making a query.
- `https://threatfox-api...`: This is the URL that we're going to be querying.
- `-d`: This is telling cURL the data that we're going to send as a JSON object.
- `{ "query": "search_hash"`: This is the JSON object that we're sending; for ThreatFox, it means that we're making a query, we're going to search a hash, and the hash is as follows.
- `"hash": "5b7e82e051ade4b14d163eea2a17bf8b"`: This is the hash that we're going to be searching for.

When we put that all together, it looks like this:

```
$ curl -X POST https://threatfox-api.abuse.ch/
api/v1/ -d '{ "query": "search_hash", "hash":
"5b7e82e051ade4b14d163eea2a17bf8b" }'
```

This will return the following information, which gives us a lot of useful data that can show us what other researchers have observed:

```
{
    "query_status": "ok",
    "data": [
        {
            "id": "4594",
            "ioc": "http:\/\/choohchooh.com\/",
            "threat_type": "botnet_cc",
            "threat_type_desc": "Indicator that identifies a
botnet command&control server (C&C)",
            "ioc_type": "url",
            "ioc_type_desc": "URL that is used for botnet
Command&control (C&C)",
            "malware": "win.arkei_stealer",
            "malware_printable": "Arkei Stealer",
            "malware_alias": "ArkeiStealer",
            "malware_malpedia": "https:\/\/malpedia.caad.fkie.
fraunhofer.de\/details\/win.arkei_stealer",
            "confidence_level": 100,
            "first_seen": "2021-03-23 08:18:05 UTC",
            "last_seen": null,
            "reference": null,
            "reporter": "abuse_ch",
            "tags": [
                "ArkeiStealer"
            ]
        }
    ]
}
```

As we can see, there is a tremendous amount of information that we're provided on this malware sample. Some elements of note that are useful are when it was first observed and additional resources to check out to learn more (`malpedia`, in this case – which is a great resource).

In this section, we learned how to query the ThreatFox API to learn information about a suspicious file.

In the next section, we'll use VirusTotal as a one-stop shop for research.

VirusTotal

VirusTotal is often, and rightfully so, referred to as a one-stop shop for research. If you want to search IP addresses, you can do it here; if you want to search domains and URLs, you can do it here; if you want to search files, you guessed it, you can do it here.

VirusTotal has an API, but this is one of those rare cases where the website has so much great information and a very impressive user experience, so it makes sense to use the website.

Like ThreatFox, VirusTotal has a huge amount of information that I encourage you to explore on your own, but as an example, we'll search for the same file hash we used previously.

To start, browse to `https://virustotal.com` and enter an IP address, domain, or file hash (MD5, SHA1, SHA256) into the search bar:

Figure 11.7 – VirusTotal file hash search

When you perform this search, you'll receive information about the file, which security vendors detect the file as malicious, what they call it, relationships between other malware samples, basic metadata about the file, and even contributions from the community:

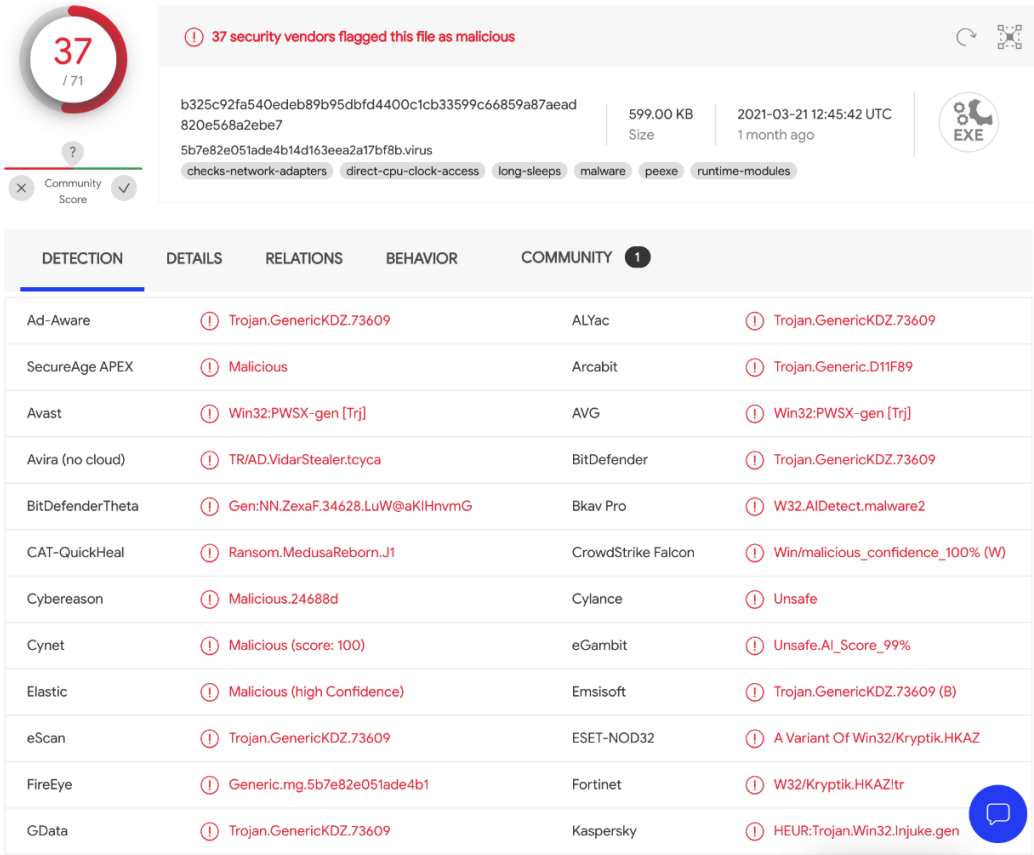
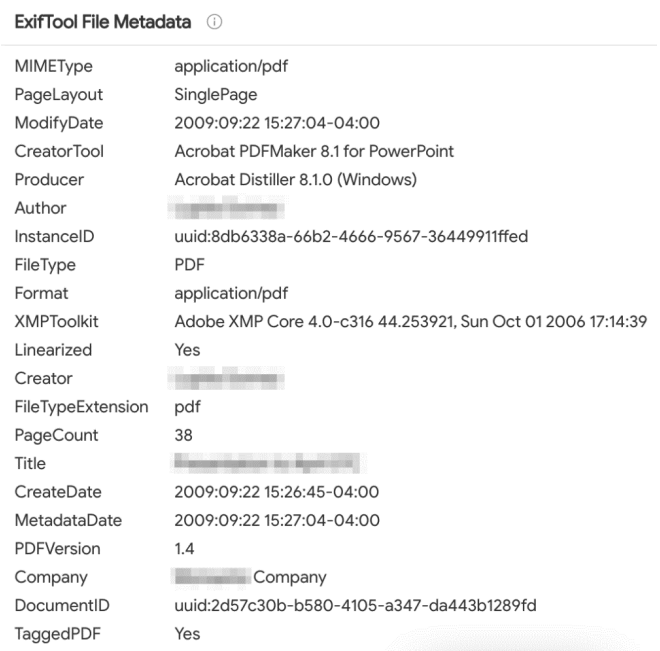


Figure 11.8 – VirusTotal file search results

Beyond querying, you can also submit malware to VirusTotal to analyze. This should be used with caution as when you submit a sample, it becomes public. Anything that is in the malware is now public, and if your organization was specifically targeted, you could be exposing sensitive information:



ExifTool File Metadata

MIMEType	application/pdf
PageLayout	SinglePage
ModifyDate	2009:09:22 15:27:04-04:00
CreatorTool	Acrobat PDFMaker 8.1 for PowerPoint
Producer	Acrobat Distiller 8.1.0 (Windows)
Author	[redacted]
InstanceID	uuid:8db6338a-66b2-4666-9567-36449911ffed
FileType	PDF
Format	application/pdf
XMPToolkit	Adobe XMP Core 4.0-c316 44.253921, Sun Oct 01 2006 17:14:39
Linearized	Yes
Creator	[redacted]
FileTypeExtension	pdf
PageCount	38
Title	[redacted]
CreateDate	2009:09:22 15:26:45-04:00
MetadataDate	2009:09:22 15:27:04-04:00
PDFVersion	1.4
Company	[redacted] Company
DocumentID	uuid:2d57c30b-b580-4105-a347-da443b1289fd
TaggedPDF	Yes

Figure 11.9 – Information on VirusTotal with company-specific information

Additionally, if the adversary is watching VirusTotal (and I guarantee they are), when you upload their malware for analysis, they will know you're onto them and they need to change tactics. Uploading files to VirusTotal is not bad, it is very common, but it should be something you are doing intentionally and with a purpose rather than just uploading everything that looks suspicious before performing any analysis. More specifically, querying a hash value is important and can certainly provide value; however, if a hash value returns no results, a sample may be uploaded as a last resort. Queries on VirusTotal are not publicly trackable:

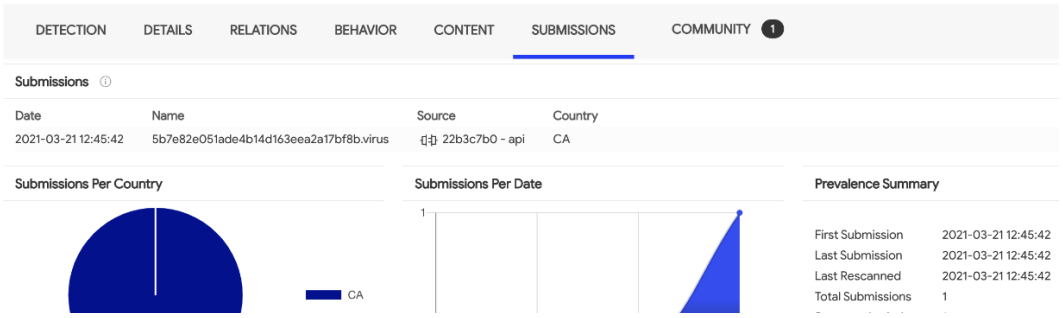


Figure 11.10 – VirusTotal malware submission tracking

As we can see, VirusTotal certainly can be used as a one-stop shop for indicator research and enrichment.

In this section, we introduced enriching indicators with third-party tools such as IPinfo for IP addresses, ThreatFox for files, and VirusTotal for indicators of all sizes and shapes. Enriching indicators is paramount to a hunter's success, and these are some tools to get you down the right path.

Enrichments within Elastic

The **Elastic Security app** currently has IP reputation links that can be used to gain additional information about threat detections. To use these, simply click on an IP address of interest from within a timeline to be sent to either VirusTotal or Talos Intelligence and automatically perform a search for the IP address. Additional indicator types will hopefully be added in the future:

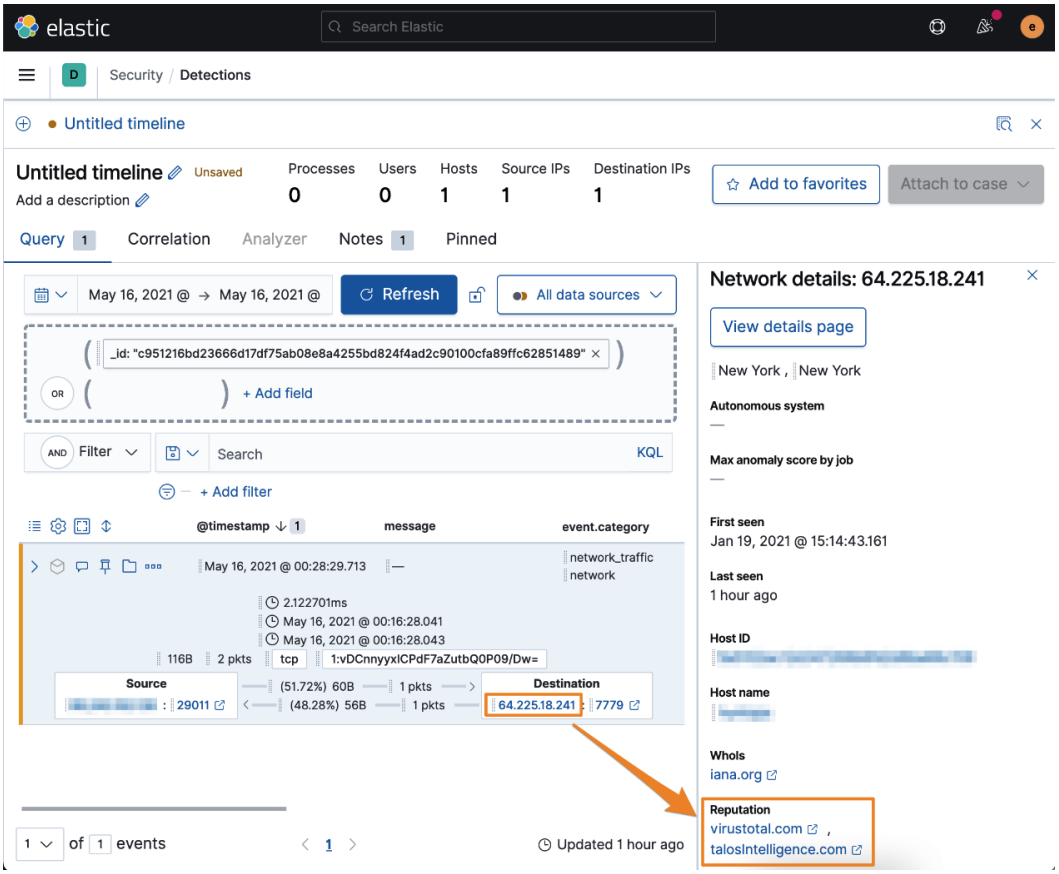


Figure 11.11 – IP reputation checking from within Elastic

In the preceding screenshot, you can see the IP address `64.225.18.241` has been identified in a timeline. From here we can click on the IP address and the network details flyout pane has hyperlinks that we can click on for VirusTotal and Talos Intelligence that will provide us with additional enrichments on this IP address.

In this section, we saw how we can use the timeline feature of the Elastic Security app to perform enrichments for IP addresses.

Summary

In this chapter, we explored a powerful tool by MITRE, the ATT&CK Navigator, to research adversary tactics, techniques, and sub-techniques. We also used tools and platforms to gain additional information about indicators that we've collected during a hunt.

If you're interested in malware reverse engineering, there is a tremendous book on the topic referenced in the *Further reading* section, *Practical Malware Analysis*.

In the next chapter, we'll discuss information sharing and analysis. Additionally, we'll explore how to get standardized data into and out of the Elastic Stack for sharing.

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. MITRE's ATT&CK Navigator allows you to research what?

   a. File hashes

   b. IP addresses

   c. Tactics, techniques, and sub-techniques

   d. Domains

2. A command-line tool used to interact with APIs is called what?

   a. cURL

   b. Vi

   c. Nano

   d. Chrome

3. What allows you to authenticate to IPinfo's API?

   a. Key

   b. Username and password

   c. Cookie

   d. Token

4. Currently, you can perform enrichments from within an Elastic timeline for which indicator types?

   a. Domains

   b. IP addresses

   c. Registry keys

   d. File hashes

5. When uploading a file to VirusTotal, what is a risk?

   a. The adversary could know they've been detected.

   b. You could infect yourself.

   c. The results may not be accurate.

   d. You cannot upload malware to VirusTotal.

Further reading

To learn more about the subject, check out the following:

- *Comparing Layers in Navigator*, MITRE: `https://attack.mitre.org/docs/Comparing_Layers_in_Navigator.pdf`
- *Introduction to ATT&CK Navigator*, MITRE: `https://www.youtube.com/watch?v=pcclNdwG8Vs`
- IPinfo documentation, IPinfo.io: `https://ipinfo.io/developers`
- ThreatFox API, Abuse.ch: `https://threatfox.abuse.ch/api`
- Malpedia, Fraunhofer FKIE: `https://malpedia.caad.fkie.fraunhofer.de`
- *Practical Malware Analysis*, No Starch Press: `https://nostarch.com/malware`

12
Sharing Information and Analysis

Being an amazing threat hunter is something to be proud of, there's no doubt about it. An adversary carrying out a delicate dance across network protocols, dipping and ducking in and through legitimate network traffic, only to be observed and recorded by an analyst with a keen eye is impressive. Monitoring and recording processes that have been started, stopped, or modified, collecting or compiling tools locally, and attempting to exfiltrate sensitive data is the nirvana for any threat actor – but the talented hunter and responder tracks and blocks all their tricks. This is the *arms race* of threat hunting, incident response, and information security as a whole.

All that said, no one can do all of this alone. It takes a team, both locally and at your fingertips, to enable the threat hunter to frustrate the adversary into failure. Rest assured: they have a team, and so should we. We can do this by sharing curated, contextual, and relevant information.

In this chapter, you will learn how to use a common data language to share relevant threat hunting objects with your peers, how to consume information provided by your peers in the Elastic Stack, and how to contribute to the Elastic security community.

In this chapter, we're going to cover the following main topics:

- The Elastic Common Schema
- Importing and exporting Kibana saved objects
- Contributing detection logic to the community

Technical requirements

In this chapter, you will need to have access to the following:

- The Elastic and Windows virtual machines you built in *Chapter 4*, *Building Your Hunting Lab – Part 1*
- A modern web browser with a UI

The code for the examples in this chapter can be found at the following GitHub link: `https://github.com/PacktPublishing/Threat-Hunting-with-Elastic-Stack/tree/main/chapter_12_sharing_information_and_analysis`.

Check out the following video to see the Code in Action:
`https://bit.ly/3klx1aN`

The Elastic Common Schema

In the previous chapters, most notably in *Chapter 7*, *Using Kibana to Explore and Visualize Data*, we discussed that the **Elastic Common Schema** (**ECS**) is a data model, developed by Elastic and their community, to describe common fields that are used when storing data in Elasticsearch. ECS defines specific field names, organizations, and data types for each field that is stored in Elasticsearch. While ECS is an open source model and is frequently contributed to by the Elastic community, it is maintained by Elastic.

Later, we'll see why ECS is strongly encouraged but not mandatory for storing data in Elasticsearch. When data cannot be stored in ECS, data providers can use general ECS guidelines (Elastic, `https://www.elastic.co/guide/en/ecs/current/ecs-guidelines.html`) to name and structure custom fields. This helps uniformly structure fields that are not in ECS.

While ECS is a data model, it is also an ideology that data should be stored uniformly so that it can be used for multiple use cases and, in our situation, shared across teams and organizations so that there are many eyes looking at data in the same way, which helps them be more informed of adversary activities.

Describing data uniformly

As we alluded to previously, describing data uniformly is paramount to sharing information. This mythology has been approached in other security platforms, such as writing endpoint rules with YARA (*Chapter 2*, *Hunting Concepts, Methodologies, and Techniques*), network rules with Snort (an open source, network-based intrusion detection system), or log events rules with the Sigma project (Sigma, `https://github.com/SigmaHQ/sigma`).

As a brief example, when talking about the source IP address of a network event, if you refer to it as `s.ip`, someone else refers to the source IP address as `src.ip`, and I refer to the source IP address as `source-ipaddr`, we'll either miss an event someone was trying to share or we'll spend a tremendous amount of time converting shared detection logic to meet our environment. A uniformly described data model is the answer to this.

Collecting non-ECS data

While the goal is to natively collect ECS-compliant data using Beats modules, any of the preconfigured Logstash input plugins we introduced in *Chapter 3*, *Introduction to the Elastic Stack*, can be used to convert our data into an ECS-compliant format.

As an example, we can use Logstash to convert network metadata provided by the open source network monitoring tool known as Zeek (`https://www.zeek.org`) into ECS.

The following is a snippet of a Logstash configuration file that converts a network event from the Zeek-described **Secure Socket Layer** (**SSL**) dataset to the ECS-compliant **Transport Layer Security** (**TLS**) dataset and then renames a TLS field from `client_issuer` to the ECS-compliant field `client.issuer`:

```
...
      mutate {
          rename => { "[ssl]" => "[tls]" }
          update => { "[event][dataset]" => "tls" }
      }

      mutate {
        rename => {
          "[tls][client_issuer]" => "[tls][client][issuer]"
...
```

This is just one example of how you could convert non-ECS compliant fields into ECS using the `mutate` Logstash filter plugin.

ECS is a large and continually maturing data model that is the future of describing data in the Elastic Stack. It is the solution to rapidly sharing information across organizations.

In this section, we discussed the Elastic Common Schema, described the value of uniformly described data, and provided an example of how to convert non-ECS compliant data into ECS using native Elastic tools.

In the next section, we'll discuss how to import and export Kibana saved objects.

Importing and exporting Kibana saved objects

Now that we've discussed ECS and why it is important from a theoretical standpoint, let's discuss how we can apply that to information sharing by exporting (and importing) our ECS-compliant Kibana saved objects.

Kibana saved objects are used to store data that you intend to use (or share) elsewhere. This includes saved searches, tags, visualizations, dashboards, index patterns, and more. As you may recall from *Chapter 7*, *Using Kibana to Explore and Visualize Data*, we created several saved searches, visualizations, dashboards, and tags.

To review our saved objects, we must log into Kibana and then go to **Stack Management**. We can do this by either clicking on the **Manage** button on the Kibana **Home** screen, or by going to the bottom of the menu bar on the left-hand side of the screen. If you need a reminder on how to get to **Stack Management**, you can review the *Adding index patterns* section of *Chapter 3*, *Introduction to the Elastic Stack*.

Once you have accessed the **Stack Management** section, you can view your Kibana saved objects by clicking on the **Saved Object** section under **Kibana**. From here, we can see all our saved objects, filter them by type, import or export them, or even remove them. In the following screenshot, you can see I have 1,094 saved objects, but you may have slightly more or less, depending on the version of or the modifications you may have made to your Elastic Stack:

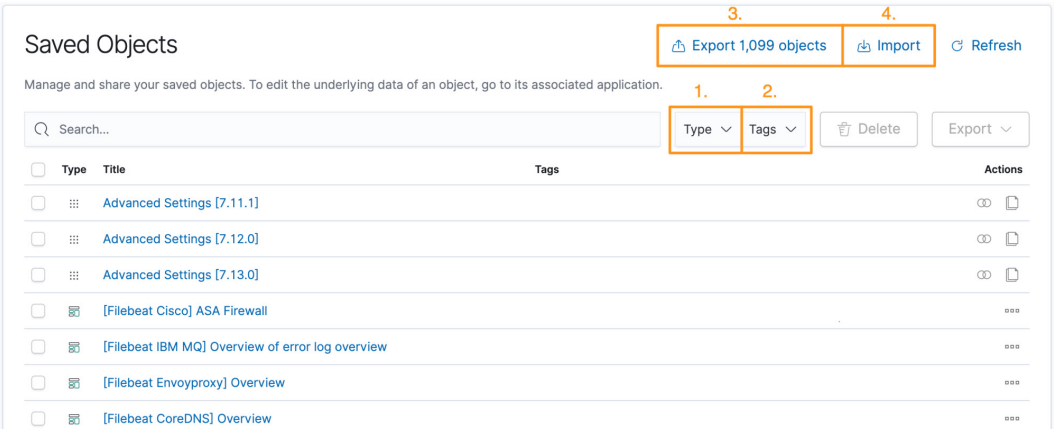


Figure 12.1 – Kibana saved object section

In this section, there are four main buttons to interact with, as follows:

1. **Type**: This allows you to filter by only certain types of saved objects.
2. **Tags**: This allows you to filter by only certain tags that have been applied to saved objects.
3. **Export**: This allows you to export selected saved objects.
4. **Import**: This allows you to import saved objects.

Let's explore these buttons and how they can be used to interact with your saved objects.

Type

The **Type** menu allows you to filter your view for only specific object types. As an example, if you filter on only visualizations, you'll only see those types of saved objects:

Figure 12.2 – Filtering on visualization saved objects

As you can see, when you select **visualizations** (or any other saved object type), it applies it as a filter to the search bar. This isn't mandatory; you can always use the drop-down menus, but this allows for faster searching across many objects if you prefer to do it that way.

There are many types of saved objects, and they all have different uses. As it relates to threat hunting, I would state that the index pattern, searches, visualizations, dashboards, and tags are the most important because they are paramount to describing, visualizing, and organizing data the same way across teams and organizations. Later in this chapter, we'll discuss how to import and export specific saved objects, as well as their related objects.

Tags

Just like **Type**, **Tags** also allows you to apply filters to your saved objects. What's helpful is that the filters stack, so by using the filter menus (or a manual search), you can chain filters together to get exactly what you're looking for. As an example, I am searching for visualization saved objects that have been tagged with either **threat intel** or **security**:

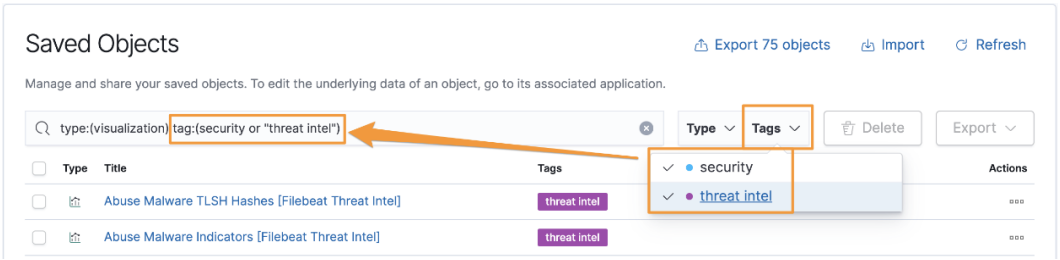


Figure 12.3 – Filtering saved objects with tags

As you can see, the filters are not only chained across type and tag but also use the `OR` operator within the tag filter. Again, both approaches will give you the same result, but you may prefer to use a manual approach that requires fewer clicks.

Export

Now that we have used the **Tags** and **Type** dropdowns to filter on the different types of saved objects we want to share, let's use the **Export** button to save some targeted objects to our local system.

Still in the **Saved Objects** section, apply a filter for dashboards and the security tag. You should have the three dashboards that we created in *Chapter 7*, *Using Kibana to Explore and Visualize Data*. These dashboards were for the HTTP, TLS, and DNS protocols.

Select the checkboxes next to all three of the dashboards and then click the **Export 3 objects** button; alternatively, you can click the **Export** button. In either case, you'll be asked if you want to include related objects. We want those related objects, so we'll ensure the switch is toggled and then click **Export**.

Related objects are other saved objects that are used for the object we're exporting. Examples of this would be our index patterns, saved searches, visualizations, and tags. This does not include data:

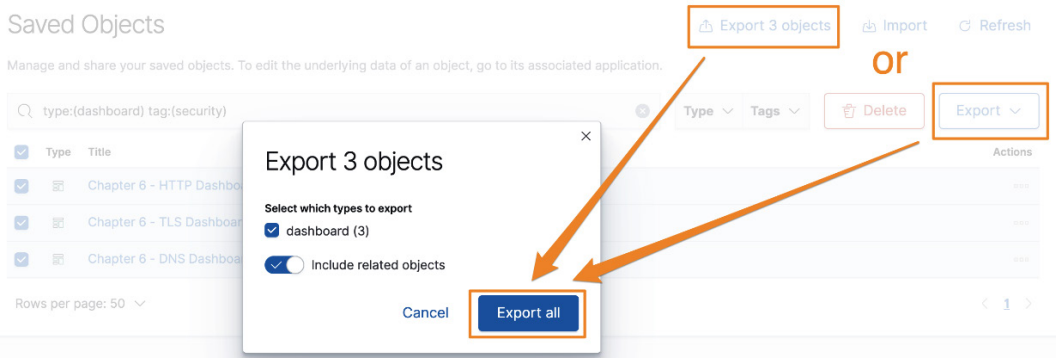


Figure 12.4 – Exporting saved objects

This will save a newline delimited JSON document to your local system. This is simply a flattened JSON document that puts one JSON object per line. If you look in the file, you'll see attributes that look similar to the fields we selected when we made the saved search, visualizations, tags, and dashboard. At the bottom of the document, you will notice that there are 23 objects that have been exported, not three. This is because we included the related objects:

```
{"exportedCount":23,"missingRefCount":0,"missingReferences":[]}
```

Now that we've exported this saved object, we can share it with others in our organization or with our peers so that they can simply import it into their Kibana instance. This will allow them to have the exact same view into the data that we did (provided they're using Packetbeat or another network data source that is using ECS).

Next, we'll show you how to import a saved object.

Import

Previously, we exported some saved objects so that we could share them with others. However, if we're the recipient of some great dashboards, we need to be able to import them as well.

> **Important Note**
>
> There are two approaches you can try here. You can import and simply override your existing saved objects, or you can delete your dashboards and visualizations and "perform without a net." I'll let you make your own call on how you want to import.

Still in Kibana, in the **Saved Objects** section, click on the **Import** button. You'll be given a few options on how to deal with conflicts. The safest way is to create new objects to avoid any loss of preexisting saved objects. That said, I normally use the **request action** option and make decisions along the way based on what I'm importing, where I'm getting it from, and what kind of data I have in my cluster (whether it's production data, test data, customer data, and so on).

That said, in this situation, to *practice what I preach*, I have deleted every saved object with the `security` tag. This was every saved object we created. So, when we do the import, you'll know if this really worked or if I'm just going through the motions:

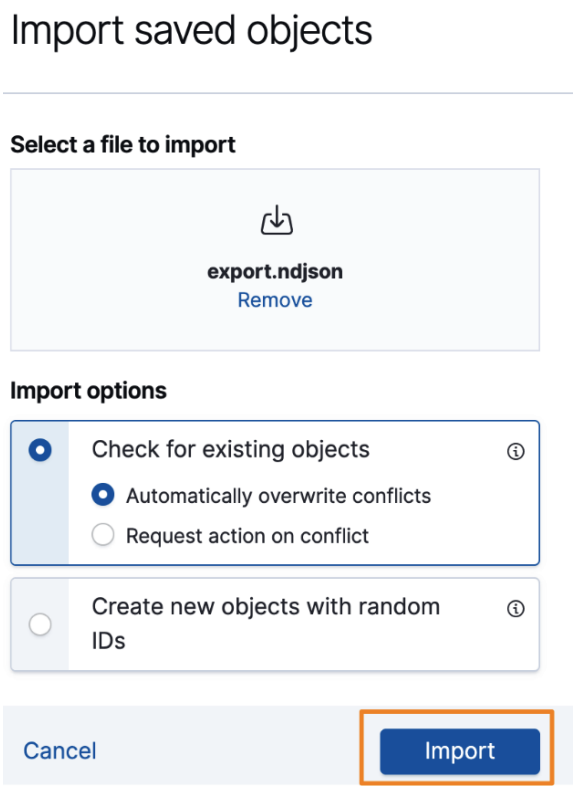


Figure 12.5 – Importing saved objects

Once the import is complete, you can just click **Done**. Let's go check to see if this worked.

In Kibana, go to **Dashboards** and select one of the dashboards that we just imported. You will see that all the data is displayed on the dashboard, as expected:

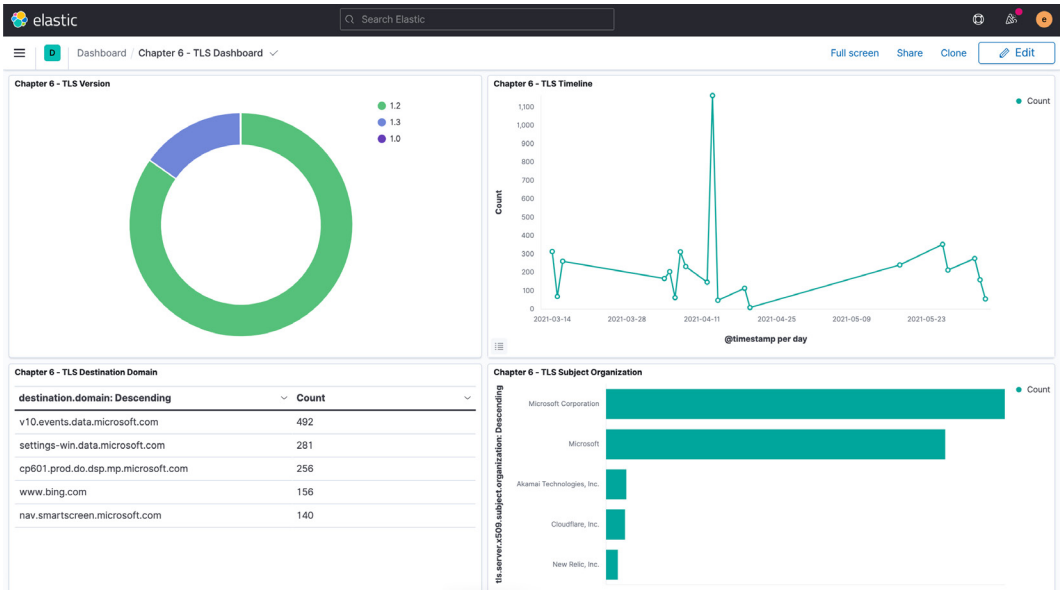


Figure 12.6 – Testing the imported dashboard

This is an extremely powerful capability. In this example, I became the partner organization. I became the peer. I had Packetbeat data, but because I deleted all the saved objects, I had no way to visualize the data other than by manually creating dashboards myself. This saved me hours of work simply by importing saved objects provided by my peers through information sharing. This wouldn't be possible without the Elastic Common Schema.

Additionally, the saved objects that I exported are available in the code link at the beginning of this chapter. Even if you didn't do the exercises in *Chapter 7*, *Using Kibana to Explore and Visualize Data*, you can simply import the saved objects as we showed previously and ECS will do the rest.

In this section, we discussed how to interact with ECS-compliant saved objects through filtering, export them so that they can be shared with peer and partner groups, and import shared objects that have been provided to our organization.

In the next section, we'll discuss how to develop our own detection logic for the Detection Engine. This can be shared with partners and peers or even contributed directly to the Elastic security community.

Developing and contributing detection logic

While sharing Kibana objects is extremely valuable for peers and partners that have similar analysis approaches and processes for analyzing data, there are also groups that have their own Kibana objects organized in a way that works best for them. We can share information and analysis with them by using logic that's used to detect adversary activity.

The benefit of detection logic over a network **Intrusion Detection System** (**IDS**) or **Endpoint Detection and Response** (**EDR**) platform is that you can create rules based on event data that, by itself, may be benign, but when combined using ECS, can indicate malicious activity.

In *Chapter 8, The Elastic Security App*, we created a detection rule in the Detection Engine. Let's export that for sharing.

In Kibana, go to **Detection Engine**, click on **Manage detection rules**, and then click on **Custom rules**:

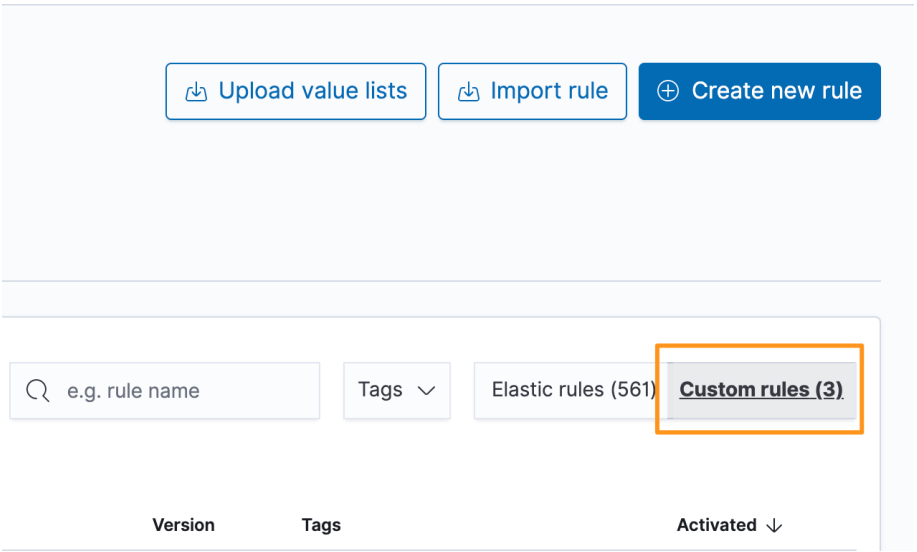


Figure 12.7 – Viewing custom detection rules

Next, click on the three dots on the right-hand side of the network traffic test we previously created and select **Export rule**. This will save the rule locally to your machine. The file will be called `rules_export.ndjson`. If we look at that file, we will see that it is a newline delimited JSON document, just like we observed with the objects we exported from Kibana. This file can be shared with others so that it can be imported into their Detection Engine.

To upload shared detection rules, simply click the **Import rule** button in the Detection Engine, provide the file, and click **Import**:

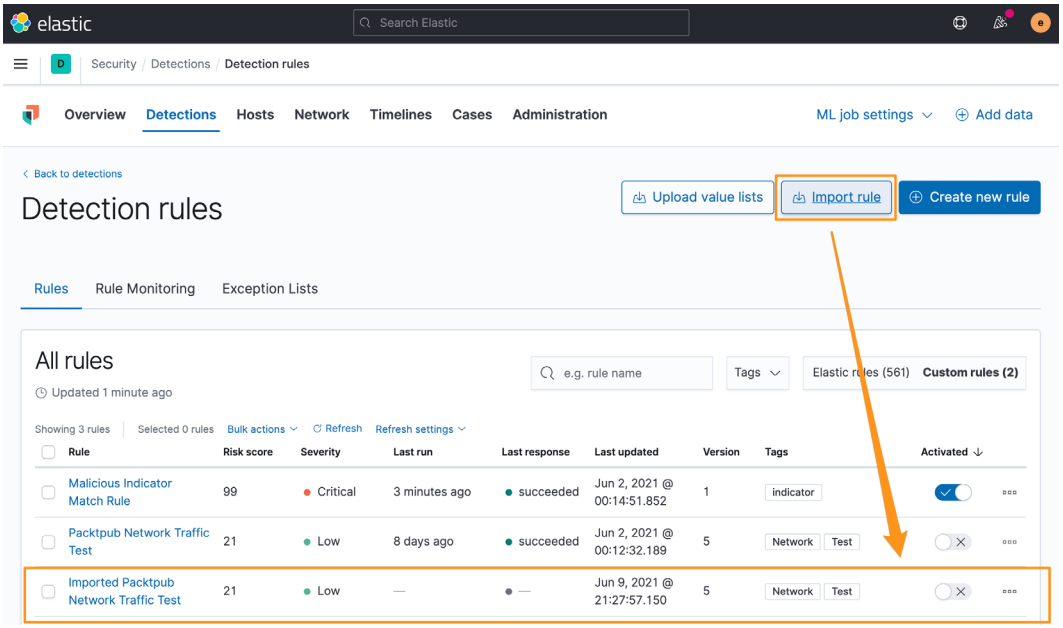


Figure 12.8 – Importing shared detection logic

Now that you've imported the shared rule, you can enable it or run it, and it will behave exactly like a rule you've created yourself:

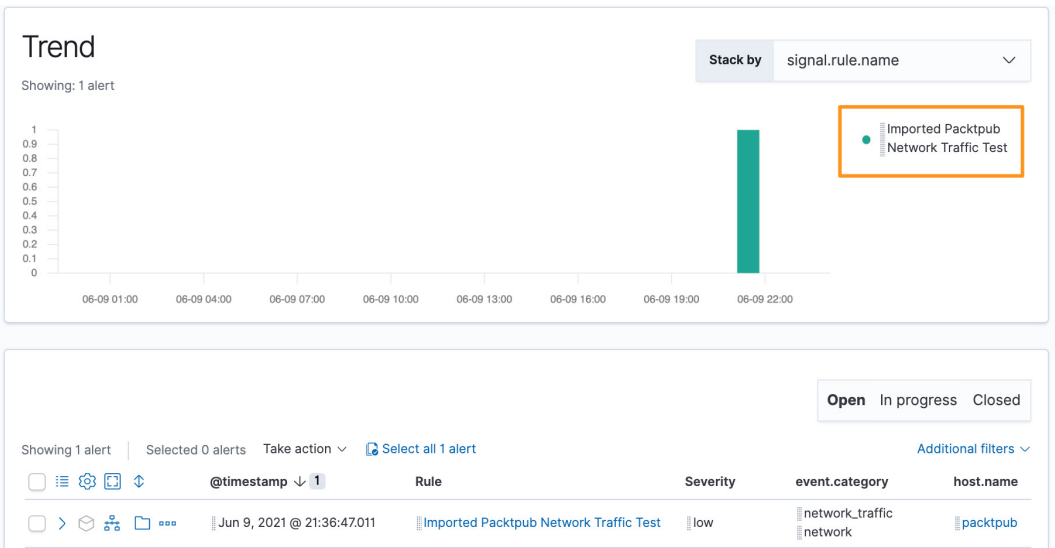


Figure 12.9 – Successfully triggered an imported rule

This is a huge force multiplier for defenders; they can find adversaries and then enable others to use your techniques to leverage the six Ds, which we discussed in *Chapter 2*, *Hunting Concepts, Methodologies, and Techniques*. The more defenders that are using shared detection logic, the harder it is for adversaries to reuse their tactics, techniques, and tools. This puts the adversary on their heels and constrains their resources if they are continually having to come up with new ways to gain access to an environment.

Additionally, it is important to remember that while you can share detection logic with your internal organization, information security is community-driven – and threat hunting in the Elastic Stack is no different. Elastic has always been deeply connected with the community, and that goes for their Detection Rules repository too (Elastic, `https://github.com/elastic/detection-rules`). This repository allows the community to develop and contribute rules that are valuable for security monitoring, incident response, and threat hunting. This repository is a great way to share your experience with the community through tangible detection logic.

In this section, we discussed how to share ECS-compliant detection logic with peers and partners. We also showcased how to receive detection logic and import it into your environment.

Summary

In this chapter, we discussed the Elastic Common Schema and the value of using it to describe data in a standard manner. We explored how to interact with ECS-compliant saved objects in Kibana so that we can export, import, and share them. Finally, we discussed how to use the Detection Engine to import and export saved detection logic so that it can be shared with peers and partners.

In this chapter, you have learned skills that will allow you to share the objects that you've created in Kibana, such as visualizations, dashboards, and saved searches, with others. This ability helps defenders observe, and defend against, adversary activity.

As this is the final chapter of this book, you should look back at the skills that you've gained thus far. You have learned how to leverage models to track adversary actions, built an Elastic lab that you can use to detonate and analyze malware, and learned how to share your knowledge with others. It is important to remember that threat hunting is a journey and that this is just the beginning. Use these skills to build your own arsenal of defensive capabilities to frustrate and stop adversary activities in your networks. Happy hunting!

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. The Elastic Common Schema is used to do what?

   a. Provide a uniform way to describe data.

   b. Automatically convert data into a different format.

   c. Automatically share data with peers.

   d. Create generic data.

2. What is not an example of a saved object in Kibana?

   a. Saved searches

   b. Dashboards

   c. Indexed data

   d. Visualizations

3. Exporting ECS-compliant objects allows you to do what?

   a. Back up indexed data.

   b. Share the objects with peers or partners.

   c. Convert visualizations into dashboards.

   d. Repair corrupted data.

4. What is not a filter that can be applied when viewing saved objects in Kibana?

   a. Tags

   b. Visualizations

   c. Dashboards

   d. Rules

5. True or false? You can import and export detection logic from the Detection Engine.

   a. True

   b. False

Further reading

To learn more about the Elastic Common Schema, please go to `https://www.elastic.co/guide/en/ecs/current/index.html`.

Assessments

In the following pages, we will review all of the practice questions from each of the chapters in this book and provide the correct answers.

Chapter 1 – Introduction to Cyber Threat Intelligence, Analytical Models, and Frameworks

1. c. Processes and methodologies tightly coupled with, and in support of, traditional SecOps.
2. a. Information
3. b. Delivery
4. a. Lateral Movement
5. d. Infrastructure

Chapter 2 – Hunting Concepts, Methodologies, and Techniques

1. d. Disrupt
2. a. Authorized binaries abused for nefarious purposes.
3. b. A process to age indicators through response tiers

Chapter 3 – Introduction to the Elastic Stack

1. d. Elasticsearch
2. b. Input
3. a. Modules
4. c. Discover
5. d. Security

Chapter 4 – Building Your Hunting Lab – Part 1

1. b. Host
2. a. Hypervisor
3. c. Installing software on Linux
4. a. Remove the Linux ISO from VirtualBox
5. d. `sudo dnf update`

Chapter 5 – Building Your Hunting Lab – Part 2

1. b. `5601`
2. d. `9200`
3. a. KeyStore
4. c. Fleet
5. a. Records the content of PowerShell script blocks

Chapter 6 – Data Collection with Beats and the Elastic Agent

1. a. Windows event data
2. c. Application-type network events
3. a. Fleet
4. d. Integrations
5. b. `Winlogbeat`

Chapter 7 – Using Kibana to Explore and Visualize Data

1. d. Filters
2. b. A line chart
3. c. Saved searches
4. a. True
5. c. Tags

Chapter 8 – The Elastic Security App

1. a. Osquery
2. b. Zeek
3. d. Filebeat Threat Intel Module
4. c. EQL
5. a. Resolver

Chapter 9 – Using Kibana to Pivot through Data to Find Adversaries

1. b. Scheduled task
2. c. Setting attributes
3. a. A harmless anti-virus test file
4. d. What the scheduled task does
5. a. `attrib -h`

Chapter 10 – Leveraging Hunting to Inform Operations

1. a. True
2. b. Detection
3. a. Driving the adversary back through the Kill Chain.
4. d. Recovery
5. b. False

Chapter 11 – Enriching Data to Make Intelligence

1. c. Tactics, techniques, and sub-techniques
2. a. cURL
3. d. Token
4. b. IP addresses
5. a. The adversary could know they've been detected.

Chapter 12 – Sharing Information and Analysis

1. a. Provide a uniform way to describe data.
2. c. Indexed data
3. b. Share the objects with peers or partners.
4. d. Rules
5. a. True

Packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

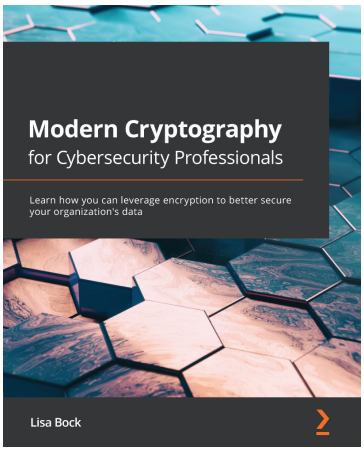
- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:



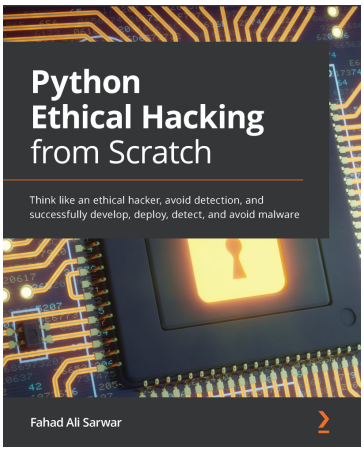
Modern Cryptography for Cybersecurity Professionals

Lisa Bock

ISBN: 978-1-83864-435-2

- Understand how network attacks can compromise data
- Review practical uses of cryptography over time
- Compare how symmetric and asymmetric encryption work
- Explore how a hash can ensure data integrity and authentication
- Understand the laws that govern the need to secure data
- Discover the practical applications of cryptographic techniques

- Find out how the PKI enables trust
- Get to grips with how data can be secured using a VPN



Python Ethical Hacking from Scratch

Fahad Ali Sarwar

ISBN: 978-1-83882-950-6

- Understand the core concepts of ethical hacking
- Develop custom hacking tools from scratch to be used for ethical hacking purposes
- Discover ways to test the cybersecurity of an organization by bypassing protection Schemes
- Develop attack vectors used in real cybersecurity tests
- Test the system security of an organization or subject by identifying and exploiting its weaknesses
- Gain and maintain remote access to target systems
- Find ways to stay undetected on target systems and local networks

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Share Your Thoughts

Now you've finished Threat Hunting with Elastic Stack, we'd love to hear your thoughts! If you purchased the book from Amazon, please click here to go straight to the Amazon review page for this book and share your feedback or leave a review on the site that you purchased it from.

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

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y