

Pie charts are great for looking at percentages of data as they relate to others. Again, I strongly recommend avoiding, although easy to use, the multilayered pie charts.

Line charts

Line charts allow you to look at data over time. This is helpful when you're building a timeline of events. As you apply filters to data, being able to see when the event took place is helpful, especially if you're trying to build a pattern of life for the event (or surrounding events).

You can create a line chart in the same way you created a data table, bar chart, and pie chart. However, this time, in the **Buckets** menu, aggregate using a **Date Histogram** in the **@timestamp** field. Add a **Split Series** in the **network.protocol** **Terms** field, and you'll see the network protocols observed over time:

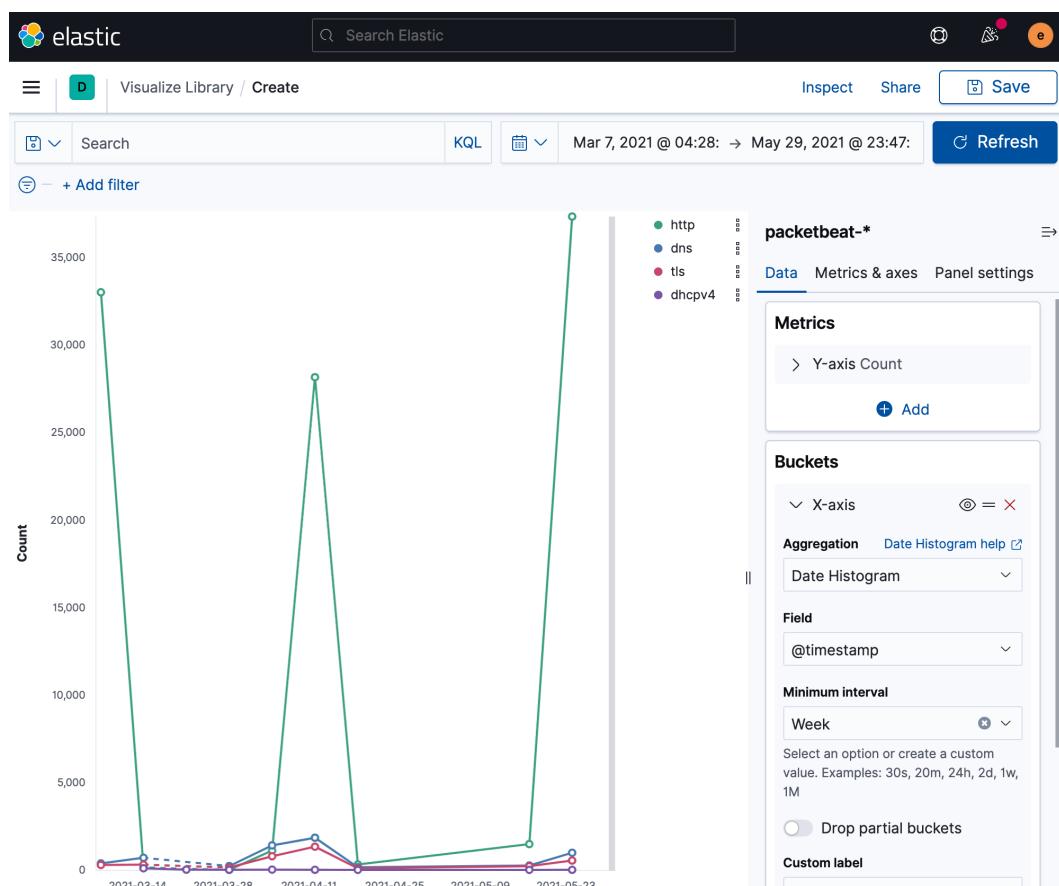


Figure 7.20 – A date histogram line chart of network protocols

Line charts are great for looking at data over time as they relate to others. Again, I strongly recommend avoiding the multitiered pie charts, even though they are easy to use.

Others

There are several other visualizations that you can experiment with using the same methods described earlier. All of the different visualizations have some uses but some more than others.

Lens

Lens is a new implementation that Elastic is rolling out to make the creation of visualizations much easier. Namely, you can drag and drop instead of having to manually select all of the different bucket options.

I like Lens; however, currently, I don't use it because of some of the limitations in terms of how filtering works and the fact that you cannot currently build Lens visualizations from saved searches, which introduces a performance impact for large dashboards.

Exercise

Using the saved searches from our **Discover App** section, let's create a data table, a pie chart, a bar chart, and a line chart.

You can create each of these visualizations in the same way you did in the preceding examples, but instead of selecting the **Packetbeat** index pattern, find your corresponding saved search (for example, Chapter 6 - HTTP Traffic instead of packetbeat-*).

Using the Chapter 6 - HTTP Traffic saved search, create the following visualizations:

- Data table: destination.domain

Save this as Chapter 6 - HTTP Destination Domain, and add a *security* tag to make it easier to find in the future.

- Data table: url.full

Save this as Chapter 6 - HTTP URL, and add a *security* tag to make it easier to find in the future.

- Bar chart: http.response.status_phrase

Save this as Chapter 6 - HTTP Response Phrase, and add a *security* tag to make it easier to find in the future.

- Pie chart: http.request.method

Save this as Chapter 6 - HTTP Request Method, and add a *security* tag to make it easier to find in the future.

- Line chart: **Date Histogram** on @timestamp

Save this as Chapter 6 - HTTP Timeline, and add a *security* tag to make it easier to find in the future.

Using the Chapter 6 - TLS Traffic saved search, create the following visualizations:

- Data table: destination.domain

Save this as Chapter 6 - TLS Destination Domain, and add a *security* tag to make it easier to find in the future.

- Data table: tls.client.ja3

Save this as Chapter 6 - TLS Client JA3, and add a *security* tag to make it easier to find in the future.

- Bar chart: tls.server.x509.subject.organization

Save this as Chapter 6 - TLS Subject Organization, and add a *security* tag to make it easier to find in the future.

- Pie chart: tls.version

Save this as Chapter 6 - TLS Version, and add a *security* tag to make it easier to find in the future.

- Line chart: **Date Histogram** on @timestamp

Save this as Chapter 6 - TLS Timeline, and add a *security* tag to make it easier to find in the future.

Using the Chapter 6 - DNS Traffic saved search, create the following visualizations:

- Data table: dns.question.name

Save this as Chapter 6 - DNS Question Name, and add a *security* tag to make it easier to find in the future.

- Data table: dns.question.subdomain

Save this as Chapter 6 - DNS Subdomain, and add a *security* tag to make it easier to find in the future.

- Bar chart: dns.question.top_level_domain

Save this as Chapter 6 - DNS Top Level Domain, and add a *security* tag to make it easier to find in the future.

- Pie chart: destination.port

Save this as Chapter 6 - DNS Destination Port, and add a *security* tag to make it easier to find in the future.

- Line chart: **Date Histogram** on @timestamp

Save this as Chapter 6 - DNS Timeline, and add a *security* tag to make it easier to find in the future.

In this section, we discussed the Visualization app, explored several different visualization options, and learned how different visualizations have different uses. Next, we'll be assembling visualizations into dashboards.

The Dashboard app

Dashboards are a great way in which to display multiple visualizations at once.

Dashboards in Kibana, like almost everything else, are interactable. This means that you can hover over them and apply filters. Additionally, we can use the visualizations we created earlier to associate the data together. For example, using dashboards and filters, we can filter out a specific domain and view the HTTP, TLS, and DNS data all at once.

To get started, click on the **hamburger** menu and select **Dashboards**. From there, you'll see several dashboards that are included with the various Beats that we're using. For example, the Packetbeat Overview ECS dashboard is a fantastic example of what you can do with visualizations and dashboards.

That said, let's create three dashboards, one each for HTTP, TLS, and DNS traffic. Click on **Create dashboard**, and then click on the **Add from Library** button. Here are all the saved searches and visualizations. Change the type to **Visualizations** and type in **Chapter 6 - HTTP**. You'll see all of the HTTP visualizations that you created. Simply click on those five visualizations, and they'll be dropped onto your dashboard pallet:

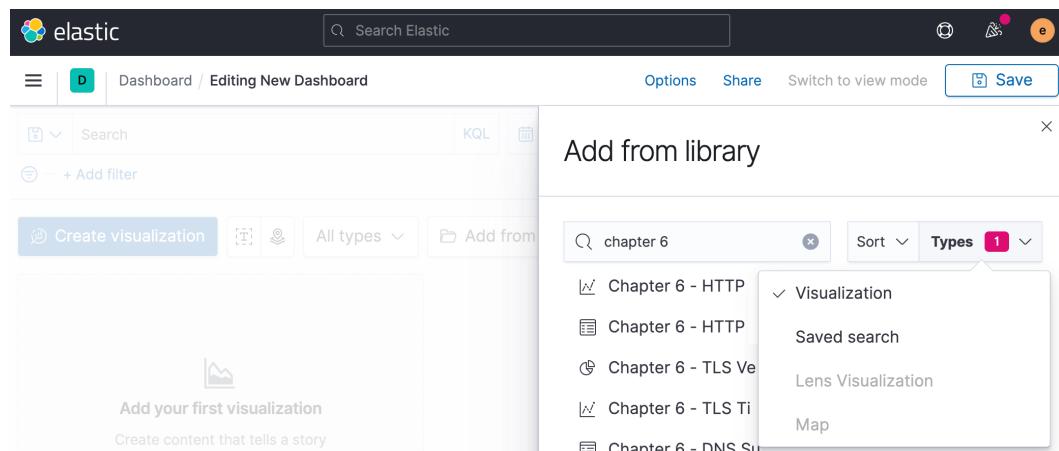


Figure 7.21 – Adding visualizations to the dashboard

The default setting in the **Date Picker** is *15 minutes*, so change that to *Last 24 hours* (or similar), and you should see your dashboard populate with all of the HTTP data. Again, this is performing only one search, and then we're applying filters from the visualizations to display the data differently.

You'll likely want to reorganize these visualizations. You can do that by simply resizing them and dragging them around in a way that makes sense to you. I like to arrange the visualizations at the top and the data tables at the bottom – that's just the way I prefer to do it. Any way that works for your personal or organizational processes is perfectly acceptable.

Once you're done, click on **Save** and name the dashboard **Chapter 6 - HTTP Dashboard**. Repeat these steps for TLS and DNS. Save them with the appropriate names. Make sure you tag them with *security*:

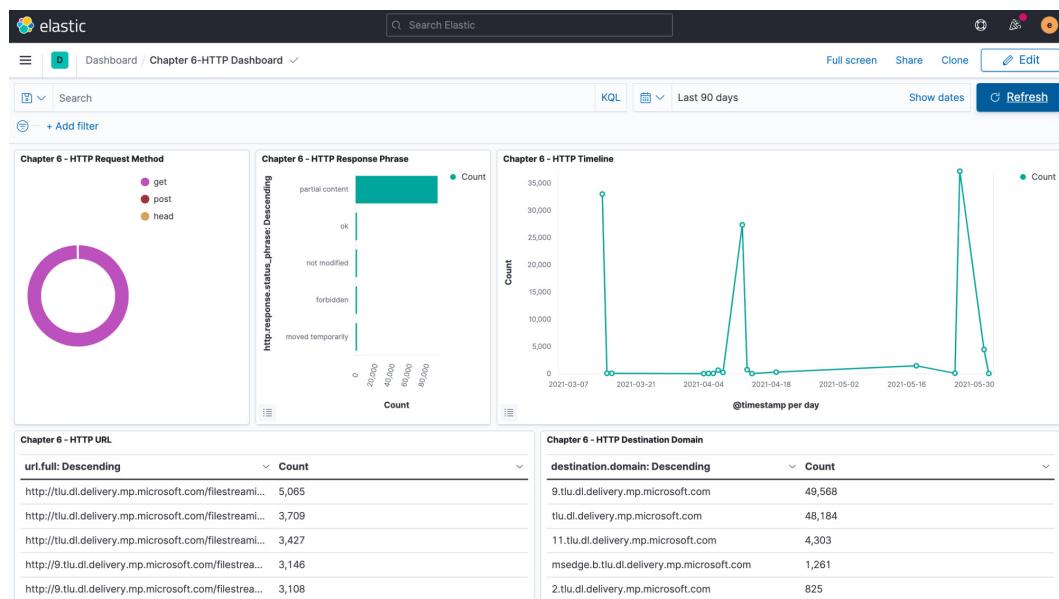


Figure 7.22 – The HTTP dashboard

Finally, as I mentioned earlier, you can click to apply filters in the dashboard, and the entire dashboard will rearrange itself with that filter applied. For example, on the TLS dashboard, I clicked on TLS version 1.3, the filter was applied, and the entire dashboard rearranged itself to only show me TLS data that used version 1.3. This can be a very powerful tool to identify the relationships between data.

I recommend that you spend some time exploring this data and the dashboards. By now, you even have some experience in how to send data to the Elastic Stack. The more data you can send, the more insights you'll uncover.

In this section, we explored creating, editing, and organizing Kibana dashboards using saved searches and visualizations.

Summary

In this chapter, we explored the Discover, Visualize, and Dashboard apps and learned multiple query languages. We created saved searches and then built visualizations and dashboards off of those saved searches.

You have gained the skills that are needed to search for events using the Discover app, write multiple types of queries (such as Lucene, KQL, and EQL), and create visualizations that will facilitate threat hunting.

In the next chapter, we'll learn about the Elastic Security app, and in the chapter after that, we'll be using the lessons learned in this chapter to hunt through simulated threat data.

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 one of the most powerful features of Kibana?
 - a. Visualizations
 - b. Dashboards
 - c. Spaces
 - d. Filters
2. What type of visualization is most commonly used to show data over time?
 - a. A pie chart
 - b. A line chart
 - c. A data table
 - d. Word cloud
3. To make performant visualizations, what is it best to build them on?
 - a. Dashboards
 - b. Queries

- c. Saved searches
 - d. Lucene
4. Dashboards allow you to apply filters to your data. Is this true or false?
- a. True
 - b. False
5. Which one of the following allows you to find saved objects faster?
- a. Names
 - b. Types
 - c. Tags
 - d. Descriptions

Further reading

To learn more about this topic, please refer to the following resources:

- *Kibana Query Language* in the official Elastic documentation: <https://www.elastic.co/guide/en/kibana/current/kuery-query.html>
- *Lucene query syntax* in the official Elastic documentation: <https://www.elastic.co/guide/en/kibana/current/lucene-query.html>
- *EQL search* in the official Elastic documentation: <https://www.elastic.co/guide/en/elasticsearch/reference/current/eql.html>
- Kibana's saved searches in the official Elastic documentation: <https://www.elastic.co/guide/en/kibana/current/save-open-search.html>
- Kibana dashboards in the official Elastic documentation: <https://www.elastic.co/guide/en/kibana/7.11/dashboard.html>
- *Elastic Common Schema (ECS) Reference* in the official Elastic documentation: <https://www.elastic.co/guide/en/ecs/current/index.html>
- Machine learning in the official Elastic documentation: <https://www.elastic.co/guide/en/security/current/machine-learning.html>

8

The Elastic Security App

We have spent a great amount of time leading up to this, the Elastic Security app. The Elastic Security app is the central point for all security-related data and information. This was formerly referred to as the Elastic SIEM (Security Information and Event Management) and is how we can explore specific host and network data, analyze security events, leverage the detection engine, manage cases, and dig deep into data with timelines.

In this chapter, you will learn how to use the Elastic Security app to identify abnormal endpoint and network traffic, perform tailored detections of those events, and create detection logic based on your analysis.

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

- The Elastic Security app overview
- The detection engine
- Hosts
- Network
- Timelines
- Cases
- Administration

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/2UODWi6>

The Elastic Security app overview

The **Elastic Security app** is the central point for Elastic's security solution. It includes a security news feed, host and network data, detections, timelines, cases, and an abstracted view into the administration of the Elastic endpoint configuration.

To get to the **Elastic Security** app, click on the hamburger menu and select **Overview** under the **Security** heading. This landing page will show you the highlights of the events that are in the security app. From here we can jump into specific sections that show their relevant data:

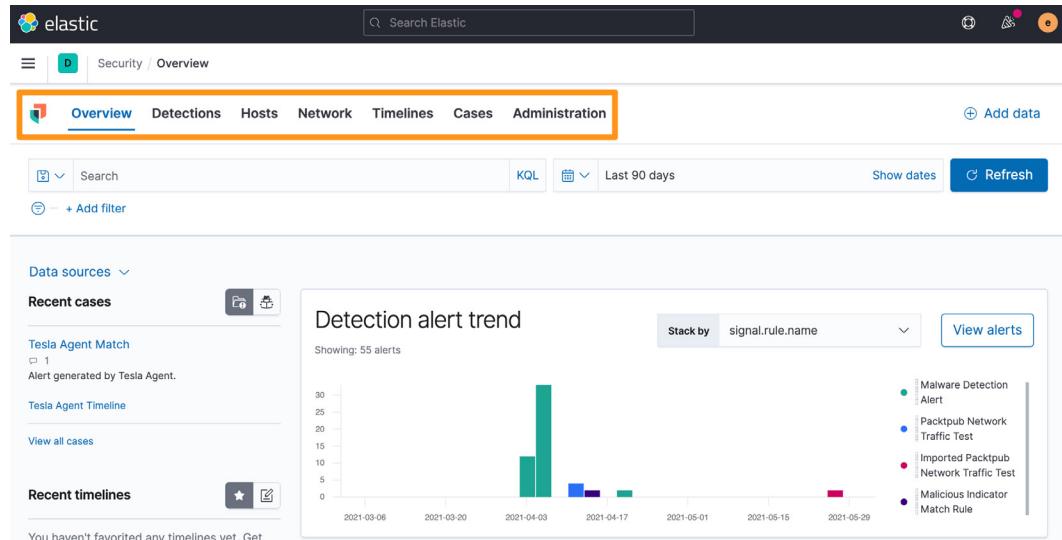


Figure 8.1 – Security app overview

You can scroll through this **Overview** section to get a high-level look at the different types of data that are reflected in the app. Most notably, at the bottom of the **Overview** page, there is a breakdown of the different datasets, separated by host and network, that we're sending into the Elastic Stack:

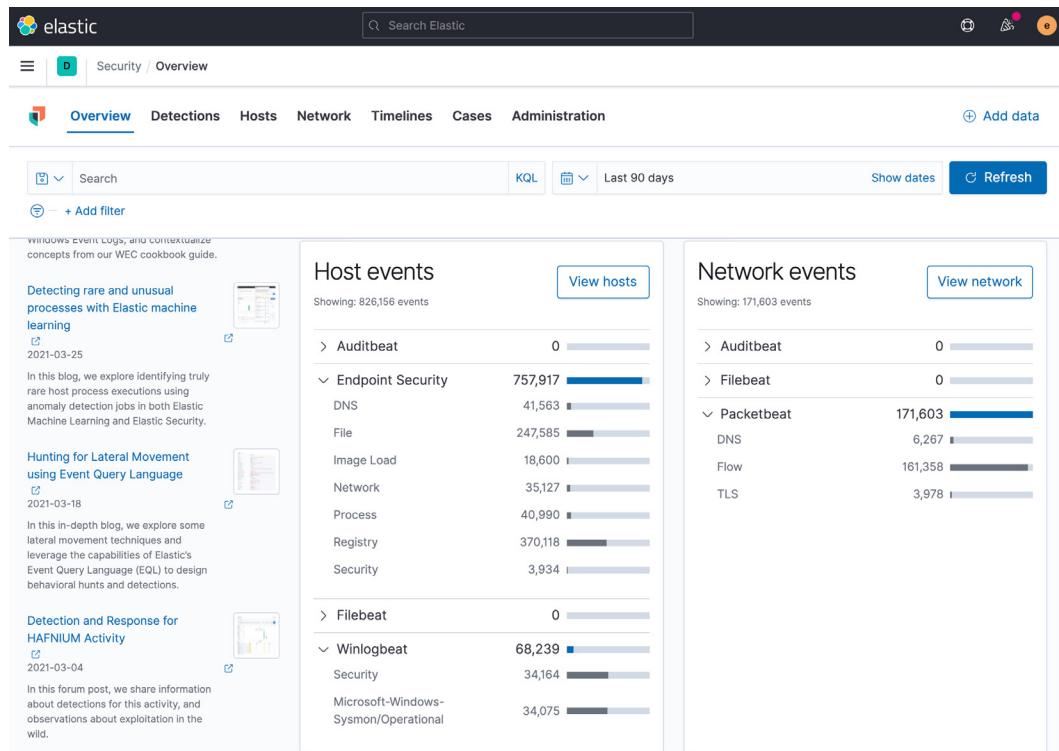


Figure 8.2 – Security app datasets

This **Overview** page allows us to see important information that is relevant across the entire Security app. To get additional information, we will use the section tabs at the top of the **Overview** page.

In this section, we learned how to get to the Elastic Security app **Overview** page. Next, we'll explore the detection engine.

The detection engine

The **Detections** section is used to investigate and create detection logic. Detection logic can be the results of a malware hit from a signature or behavior as well as potentially malicious activity. As of Elastic version 7.12, the detection engine has over 500 pre-built rules that are created by the Elastic Intelligence and Analytics Team and the Elastic community:

A screenshot of the Elastic Security app's interface. At the top, there is a navigation bar with the 'elastic' logo, a search bar labeled 'Search Elastic', and several icons. Below the navigation bar, the main menu includes 'Overview', 'Detections' (which is currently selected), 'Hosts', 'Network', 'Timelines', 'Cases', and 'Administration'. On the right side of the header, there are 'ML job settings' and 'Add data' buttons. The main content area has a 'Search' input field, a 'KQL' button, a date range selector set to 'Last 24 hours', a 'Show dates' button, and a 'Refresh' button.

Figure 8.3 – Detection section

Keeping with the theme across the rest of Kibana, you can apply specific queries directly into the **Detections** section or apply a date picker selection to narrow any searches.

In this section, we began to explore the detection engine of the Elastic Security app. In the next section, we'll learn about managing the detection rules.

Managing detection rules

One of the most powerful features of the Elastic Security app is the detection rules. **Detection rules** are pre-configured queries that compare events from various data sources to identify non-signature-based malicious activity.

As an example, perhaps you want to know whether you have systems that are receiving **Remote Desktop Protocol (RDP)** connections from outside your network, whether someone is trying to brute force through **Secure Shell (SSH)**, or someone is trying to export your Windows Registry Hive. These things could be malicious in your environment, but these are events that will not be detected by traditional anti-virus.

Elastic has released hundreds of open source rules for the detection engine (<https://www.elastic.co/blog/elastic-security-opens-public-detection-rules-repo>) and has made them all available on GitHub (<https://github.com/elastic/detection-rules>). As I mentioned before, there are 546 rules available for free. Not only are the rules available on GitHub, but they are also automatically loaded in Kibana. You may remember in *Chapter 5, Building Your Hunting Lab – Part 2*, we loaded all of the prebuilt rules:

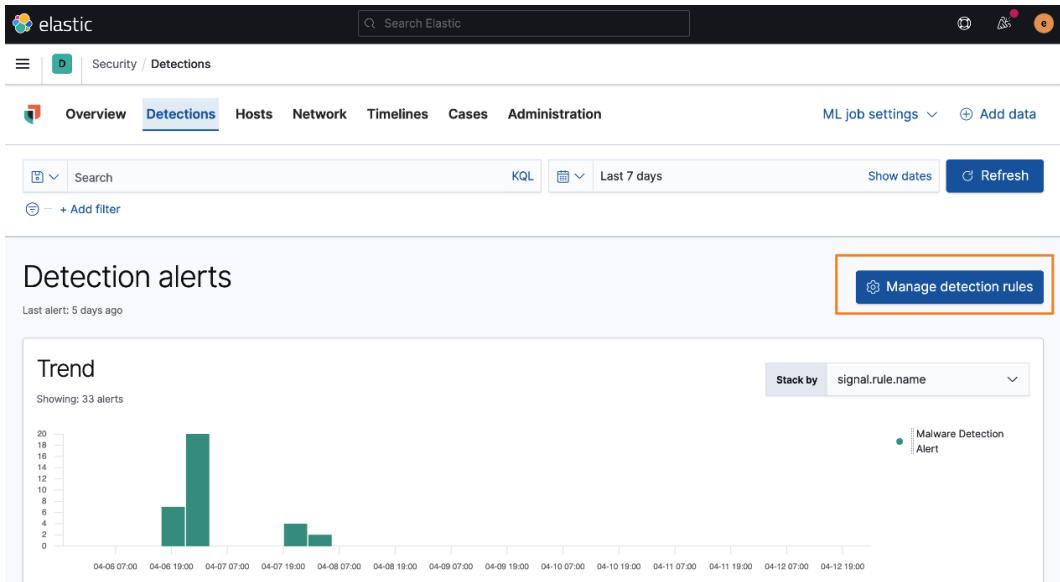


Figure 8.4 – Manage detection rules

If you click on the **Manage detection rules** button, it will open the **Detection Rules** management section.

From here, we can see the **Rules**, **Rules Monitoring**, and **Exception Lists** tabs.

Rules

From this tab, we can enable different rules, search for rules by their names or tags, or dig into the rules to learn more about them:

The screenshot shows the 'Detection rules' section of the Elastic Security App. At the top, there are navigation links for Overview, Detections (which is selected), Hosts, Network, Timelines, Cases, and Administration. Below the navigation is a search bar and buttons for 'Upload value lists', 'Import rule', and 'Create new rule'. The main area displays a table of 'All rules' with columns for Rule, Risk score, Severity, Last run, Last response, and Last updated. One row is expanded to show details: 'Imported Packtpub Network Traffic Test' with a risk score of 21, low severity, and last run 7 minutes ago. To the right of the table, a modal window titled 'Search tags' is open, containing a search input field and a list of tags: APM, Application, Asset Visibility, AWS, Azure, Cloud, Collection, and Command and Control. Each tag has a 'Activated' toggle switch and three dots for more options.

Figure 8.5 – Rules overview

Clicking on a rule will open the rule so that you can inspect the metadata about the rule, where the data must come from, what the query is, and so on:

Important note

You cannot modify the Elastic-provided rules, but you may make a duplicate and modify the duplicate if necessary.

The screenshot shows the Elastic Stack Detections interface. At the top, there's a navigation bar with the elastic logo, a search bar, and various icons. Below it, the breadcrumb navigation shows Security / Detections / Detection rules / Public IP Reconnaissance Activity. The main menu includes Overview, Detections (which is selected), Hosts, Network, Timelines, Cases, and Administration. On the right, there are ML job settings and Add data buttons.

Below the menu, there are search and filter options: a dropdown for 'Search' (set to 'KQL'), a date range selector for 'Last 7 days', a 'Show dates' button, and a 'Refresh' button.

The title of the page is 'Public IP Reconnaissance Activity'. It includes a creation timestamp ('Created by: elastic on Feb 21, 2021 @ 20:28:22.677'), an update timestamp ('Updated by: elastic on Apr 12, 2021 @ 21:29:07.695'), and a last response timestamp ('Last response: succeeded at Apr 12, 2021 @ 21:29:12.777'). There are also 'Activate' and 'Edit rule settings' buttons.

The page is divided into several sections:

- About:** Includes details like Author (Elastic), Severity (Low), Risk score (21), Reference URLs (two links), False positive examples (a note about authorized workflows), License (Elastic License v2), MITRE ATT&CK™ (Discovery (TA0007) and System Network Configuration Discovery (T1016)), and Timestamp override (event.ingested).
- Definition:** Contains the index pattern ('packetbeat-*'), a custom query ('event.category:network AND event.type:connection AND server.domain:(ipecho.net OR ipinfo.io OR ifconfig.co OR ifconfig.me OR icanhazip.com OR myexternalip.com OR api.ipify.org OR bot.whatismyipaddress.com OR ip.ansysrc.net OR wtfismyip.com) AND NOT http.response.status_code:302 AND status:OK AND NOT _exists:_http.request.referrer'), Rule type (Query), and Timeline template (None).
- Schedule:** Shows the rule runs every 5m with an additional look-back time of 1m.

Figure 8.6 – Public IP Reconnaissance Activity network rule

This overview of the rules will help you determine what rules you want to enable and what rules don't make sense in your environment. As an example, if you aren't collecting cloud security rules, or Linux or macOS events, it doesn't make sense to enable those rules.

Rule monitoring

Clicking on the **Rule Monitoring** tab will give you a view of the amount of time the rules take to run:

Rule	Indexing Time (ms)	Query Time (ms)	Last Gap (if any)	Last run	Last response	Activated
Setgid Bit Set via chmod	—	67.48	—	19 minutes ago	● succeeded	active
SSH Authorized Keys File Modification	—	83.01	—	19 minutes ago	● succeeded	active
Sensitive Files Compression	—	45.06	—	19 minutes ago	● succeeded	active
WebProxy Settings Modification	—	58.86	—	19 minutes ago	● succeeded	active
Public IP Reconnaissance Activity	—	6.23	—	16 minutes ago	● succeeded	active
Endpoint Security	—	1.76	3 hours	15 minutes ago	● succeeded	active

Figure 8.7 – Rule Monitoring

Rule Monitoring can be helpful if you're noticing a performance impact; you can look at what rules are taking the most time and decide if you need to increase the resources for your stack or if those rules are needed.

Exception Lists

The **Exception Lists** tab is where you can view any exceptions you've created for rules or the endpoint:

The screenshot shows the Elastic Detection rules interface. At the top, there's a navigation bar with the Elastic logo, a search bar, and various icons. Below it, a secondary navigation bar includes 'Overview', 'Detections' (which is underlined), 'Hosts', 'Network', 'Timelines', 'Cases', and 'Administration'. To the right are 'ML job settings' and 'Add data' buttons. The main content area is titled 'Detection rules' with a 'Back to detections' link. It features several buttons: '+ Install 1 Elastic prebuilt rule', '+ Upload value lists', '+ Import rule', and '+ Create new rule'. Below these are tabs for 'Rules', 'Rule Monitoring', and 'Exception Lists' (which is currently selected). A search bar with placeholder text 'e.g. Example List Name' is also present. The main table displays two exception lists:

List ID	Name	Number of rules as...	Rules assigned to	Date created	Last edited
1d8ecb43-f88c-4d94-8ce0-2de8cf96f356	Endpoint Security	1	Endpoint Security	11 Apr 2021 21:25	2021-04-12T02:25:19.5... Edit Delete
endpoint_list	Endpoint Security Exception List	1	Endpoint Security	21 Feb 2021 20:28	2021-02-22T02:28:20.3... Edit Delete

At the bottom left, there's a 'Rows per page' dropdown set to 20, and at the bottom right, navigation arrows.

Figure 8.8 – Exception Lists

We'll talk more about the exception framework in the *Event actions* section, a bit further on in the chapter.

Creating a detection rule

I mentioned that Elastic provides 546 rules for you, but we can also create rules that fit a specific threat profile for our environment.

Rules can either be created using a Python module that Elastic provides and has made available, or be created and made available through Kibana.

To get started, click the blue **Create new rule** button:

The screenshot shows the 'Detection rules' section of the Elastic Security App. At the top, there's a navigation bar with tabs for Overview, Detections (which is selected), Hosts, Network, Timelines, Cases, and Administration. Below the navigation is a search bar labeled 'Search Elastic'. On the right side of the header, there are icons for ML job settings, Add data, and user profile. The main content area is titled 'Detection rules' and has a sub-header 'All rules'. It shows a table with one row: 'Public IP Reconnaissance Activity' (Risk score: 21, Severity: Low, Last run: 17 minutes ago, Last response: succeeded, Last updated: Apr 12, 2021 @ 21:29:07.695, Version: 3). There are filters for Tags, Elastic rules (546), and Custom rules (0). At the bottom of the table, there are buttons for Discovery, Elastic, Network, and See all. A blue box highlights the 'Create new rule' button at the top right of the page.

Figure 8.9 – Create new rule

There are five types of rules that you can create:

- A **Custom query** (KQL or Lucene rule)
- A **Machine Learning** rule
- A **Threshold** rule
- An **Event Correlation** rule
- An **Indicator Match** rule

Next, we'll walk through the creation of these rule types.

Important note

For the detection rules, the first section (**Define rule**) will change depending on the rule type that you're going to use, but the three follow-on sections will all be the same (**About rule**, **Schedule rule**, and **Rule actions**). We'll go all the way through the four sections for the **Custom query** type and just the first section for the other four rule types.

Custom query rule

By default, the rule type will be **Custom query**. This is how you'd create a KQL or Lucene query.

Below the rule type, you can select what index patterns your data will be in. By default, all of the possible index patterns are added. As we only have Endpoint, Packetbeat, and Winlogbeat data, we can safely remove the unused datasets.

Next, we can enter our query. As an example, I am looking for network connections to the domain `packtpub.com`:

The screenshot shows the Elasticsearch Detections interface for creating a new rule. The top navigation bar includes the elastic logo, a search bar, and various navigation links like Overview, Detections, Hosts, Network, Timelines, Cases, Administration, ML job settings, and Add data. The main title is 'Create new rule'. Step 1, 'Define rule', is selected. Under 'Rule type', the 'Custom query' option is highlighted with an orange border and labeled 'Selected'. Other options like 'Machine Learning' and 'Threshold' are also shown. Below this, 'Event Correlation' and 'Indicator Match' are listed with 'Select' buttons. The 'Index patterns' section contains three selected indices: 'logs-*', 'packetbeat-*', and 'winlogbeat-*'. A note says 'Enter the pattern of Elasticsearch indices where you would like this rule to run. By default, these will include index patterns defined in Security Solution advanced settings.' At the bottom, the 'Custom query' field contains the KQL query: `event.category: "network" and event.type: "connection" and destination.domain: "packtpub.com"`. There are buttons for 'Import query from saved timeline' and 'KQL'.

Figure 8.10 – Custom query

Here we can see what the first part of the rule will look like.

Elastic has added the **Preview results** feature so you can test to see whether your query is working as intended:

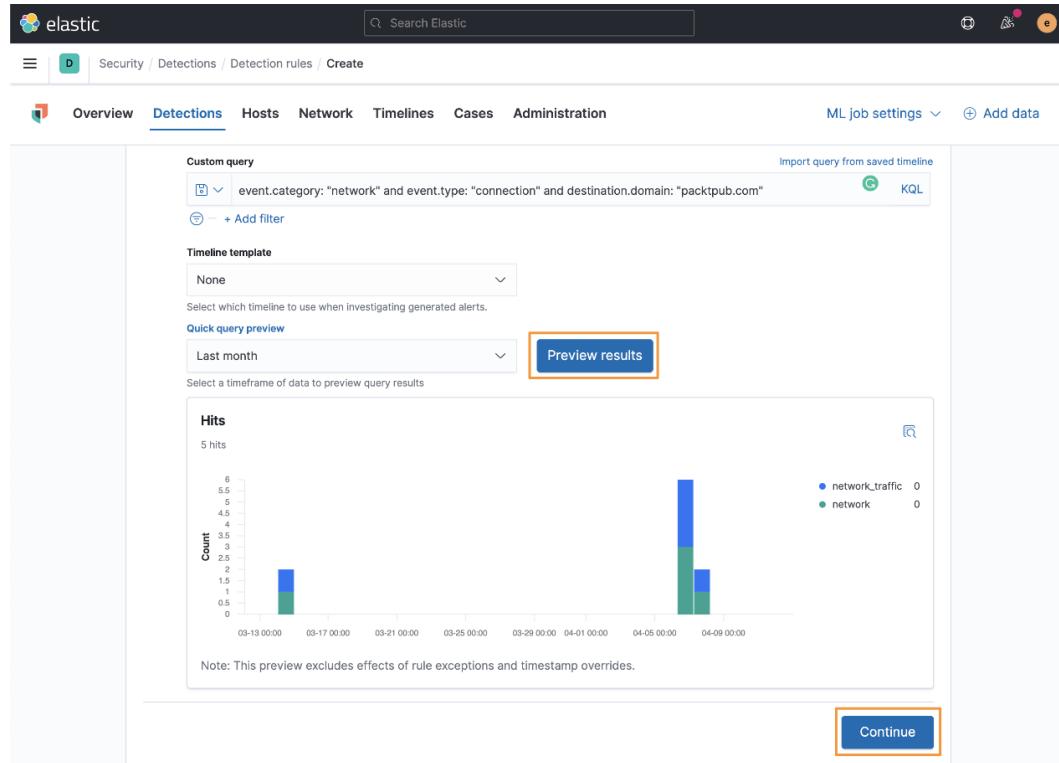


Figure 8.11 – Preview results

After we've set up the rule type and the data sources, written our query, and tested the results, we can click on **Continue** to move onto the next step.

In the second section, we can define the name, description, severity, and risk score, and add any organization tags:

The screenshot shows the Elastic Detections interface with the following details:

- Name:** Packtpub Network Traffic Test
- Description:** This rule will identify when network traffic is observed going to the Packt domain.
- Default severity:** Low (selected)
- Severity override:** Checked. A table shows source fields and their corresponding source values mapped to severities:

Source field	Source value	Severity
		Low
		Medium
		High
network.protocol	http	Critical

 A note below states: "For multiple matches the highest severity match will apply. If no match is found, the default severity will be used."
- Default risk score:** 21 (selected from a range of 0 to 100)
- Risk score override:** Unchecked
- Tags:** Test, Network (Optional)

Figure 8.12 – Rule description

The default severity is **Low** and the risk score is **21**. As you change the severity, the risk score will automatically adjust. This can also be manually tuned if you have organizational policies that dictate a severity and risk matrix.

You can also override the defaults for severity and risk. In the preceding figure, I have changed the severity from **Low** to **Critical** if the network traffic is unencrypted over the HTTP protocol instead of TLS.

Clicking on the **Advanced** settings dropdown, we can add some additional information about the rule. These settings are optional:

Tooltip in Markdown

Advanced settings

Reference URLs
https://www.packtpub.com

False positive examples
If a test is performed to validate the rule, this could be a false positive and should be marked as such.

MITRE ATT&CK™ threats

- MITRE ATT&CK™ tactic: Command and Control (TA0011)
- MITRE ATT&CK™ technique: Application Layer Protocol (T1071)
- MITRE ATT&CK™ subtechnique: Web Protocols (T1071.001)

Investigation guide
https://policies.soc.internal

Figure 8.13 – Advanced settings

Here we can define any reference URLs that can help provide some context when performing an investigation. I like to put the source of the threat reports that may have led to the rule creation here.

We can provide some false positive examples.

We can add MITRE ATT&CK tactics, techniques, and subtechniques. The ATT&CK model was covered in *Chapter 1, Introduction to Cyber Threat Intelligence, Analytical Models, and Frameworks*.

We can also add investigative notes. This can be used to link to organization-specific documentation, points of contact, and so on. This guide renders GitHub-flavored Markdown (<https://github.github.com/gfm/>), which is helpful to use for inserting hyperlinks and tooltips.

GitHub-flavored Markdown can create hyperlinks using [URL Text] (url) and tooltips with !{tooltip [anchor text] (helpful description)}.

Moving on, we can populate the **Author** field and add the appropriate **License** details:

The screenshot shows the 'Create' page for a detection rule in the Elastic interface. The 'Detections' tab is selected. The 'Author' field contains 'PacktPub X'. The 'License' field contains 'Apache 2.0'. Both fields have an orange border around them. A 'Continue' button at the bottom right is also highlighted with an orange border.

Figure 8.14 – Detection alerts trend sorting

We can also choose to apply any exceptions to this rule; we don't have any yet, but if we wanted to apply exceptions, we could check that box.

Building block rules are low-risk rules that we can create that will still write to the rules indices but not be displayed in the **Alerts** table in the main **Detections** view. This is helpful if you want to create a rule for context for other rules but not clutter up your view.

Rule name override will allow you to override the rule name we set before with the value of a field from the event. As an example, we could use the `destination.domain` field to name this rule `packtpub.com` when it is displayed in the **Alerts** table.

If we want to use a different timestamp than the default `@timestamp`, we can define that here.

Next, we can click **Continue** to move on.

We can define the schedule for the rule. This is how often the rule will run and how far back it should look. The lookback is to ensure there aren't specific events that happen to fall between the rule executions that could be missed.

The default is to run every 5 minutes with a 1-minute lookback. I prefer to change this to run every 9 minutes with a 1-minute lookback:

The screenshot shows the third step of a rule creation wizard, titled "Schedule rule". It has two main sections: "Runs every" and "Additional look-back time".

- Runs every:** A dropdown menu showing "9 Minutes". Below it is a note: "Rules run periodically and detect alerts within the specified time frame."
- Additional look-back time:** A dropdown menu showing "1 Minutes" with the note "Optional". Below it is a note: "Adds time to the look-back period to prevent missed alerts."

At the bottom right is a blue "Continue" button, which is highlighted with an orange border.

Figure 8.15 – Schedule the rule

After we have set the schedule and the lookback, we can click **Continue** to move on to **Rule actions**.

We can define how often the rule actions are performed, but they allow you to send notifications to third-party services. Actions to these external services are a paid feature.

Finally, we can create the rule. I always prefer to create the rule without activating it (meaning it will generate an event in the **Detections** section) so I can take one final look at the completed rule before I run it:

Figure 8.16 – Create the new rule

Once we've created the new rule, we'll land back on the **Detection rules** management page. We can click on **Custom rules** to view our new rule:

Rule	Risk score	Severity	Last run	Last response	Last updated	Version	Tags	Activated
Packtpub Network Traffic Test	21	Low	6 minutes ago	succeeded	Jun 2, 2021 @ 00:12:32,189	5	Network, Test	Activated

Figure 8.17 – View custom rules

Clicking on the rule name will open the rule and we can make a final check, make any necessary changes, and activate (or deactivate) it when you are ready to generate events with it:

The screenshot shows the Elastic Security App interface with the following details:

- Header:** elastic, Search Elastic, ML job settings, Add data.
- Breadcrumbs:** Security / Detections / Detection rules / Packtpub Network Traffic Test.
- Navigation:** Overview, Detections (selected), Hosts, Network, Timelines, Cases, Administration.
- Search Bar:** Search, KQL, Today, Show dates, Refresh.
- Rule Title:** Packtpub Network Traffic Test.
- Rule Details:**
 - Created by: elastic on Apr 12, 2021 @ 22:48:18.259 Updated by: elastic on Apr 12, 2021 @ 22:56:39.797
 - Last response: succeeded at Apr 12, 2021 @ 22:56:41.891
 - Activate button (disabled)
 - Edit rule settings button
 - Save button
- About Section:**
 - This rule will identify when network traffic is observed going to the Packt domain.
 - Author:** PacktPub
 - Severity:** Low
 - Severity override:** network.protocol: http → Critical
 - Risk score:** 21
 - Reference URLs:** <https://www.packtpub.com>
 - False positive examples:** If a test is performed to validate the rule, this could be a false positive and should be marked as such.
 - License:** Apache 2.0
 - MITRE ATT&CK™:** Command and Control (TA0011)
 - Application Layer Protocol (T1071)
 - Web Protocols (T1071.001)
 - Tags:** Test, Network
- Definition Section:**
 - Index patterns:** logs-* | packetbeat-* | winlogbeat-*
 - Custom query:** event.category: "network" and event.type:"connection" and destination.domain:"packtpub"
 - Rule type:** Query
 - Timeline template:** None
- Schedule Section:**
 - Runs every:** 9m
 - Additional look-back time:** 1m

Figure 8.18 – Reviewing the new rule

As a real-world example, you notice in the preceding figure that the destination domain isn't right. I can click on the **Edit rule settings** button and adjust the domain to include the **.com Top-Level Domain (TLD)**:

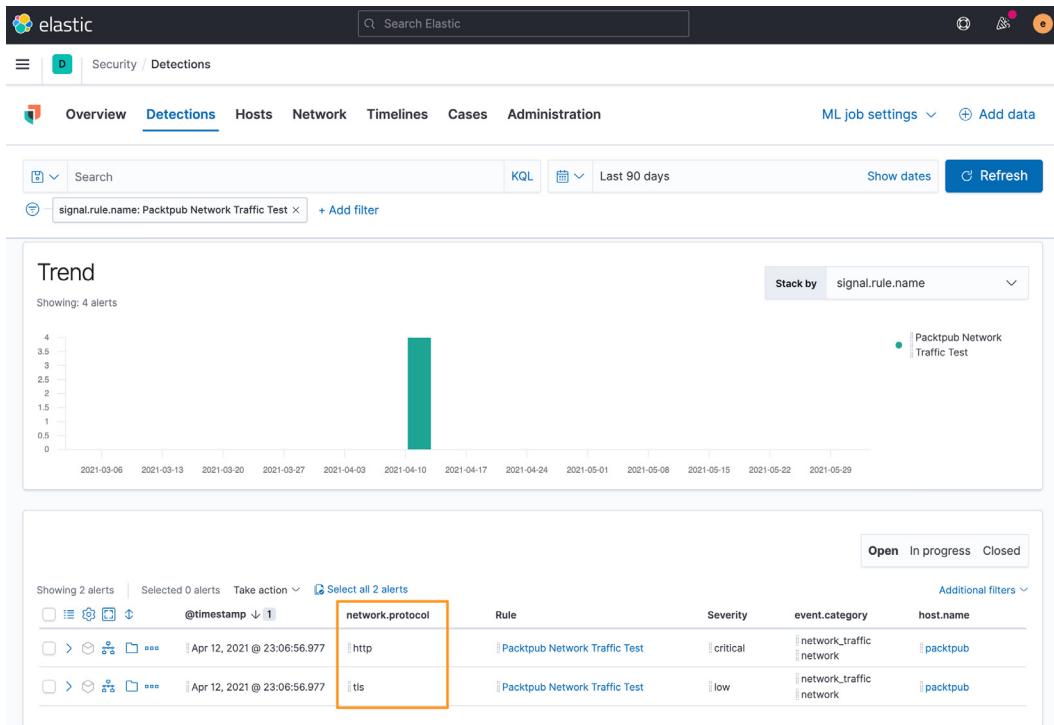


Figure 8.19 – Packtpub network rule execution

After adjusting my mistake, I can see an execution of the rule and we can even see the severity override we did for the HTTP connection.

We created a custom query rule with KQL; next we'll explore the machine learning rule.

Machine learning rule

Machine learning rules are available in the detection engine; however, as we discussed in *Chapter 7, Using Kibana to Explore and Visualize Data*, it is a licensed feature, so we'll not be able to explore them with our lab environment.

If you do add a license for your lab, you will be able to enable machine learning detection rules or create your own.

Next, we'll discuss threshold rules.

Threshold rules

Threshold rules are run against indices and then create an alert when the defined number of events occurs.

Using the **Group By** field in the **Define rule** section will create Boolean logic using AND for multiple fields that you define. You can also leave this blank and it will use the threshold value irrespective of any defined fields.

Additionally, you can set **Count field**; this will identify the number of unique values for a defined field.

In this example, I again used the same query as I did for the custom query previously to identify traffic to packtpub.com (`event.category: "network" and event.type: "connection" and destination.domain: "packtpub.com"`). As we only have one system, I opted to group by the `network.protocol` field with over two results and then count by the number of unique values for the `source.port` field. So, this rule will trigger when there are more than two network protocols and more than two source ports.

This rule would create alert rules only when the preceding criteria are met:

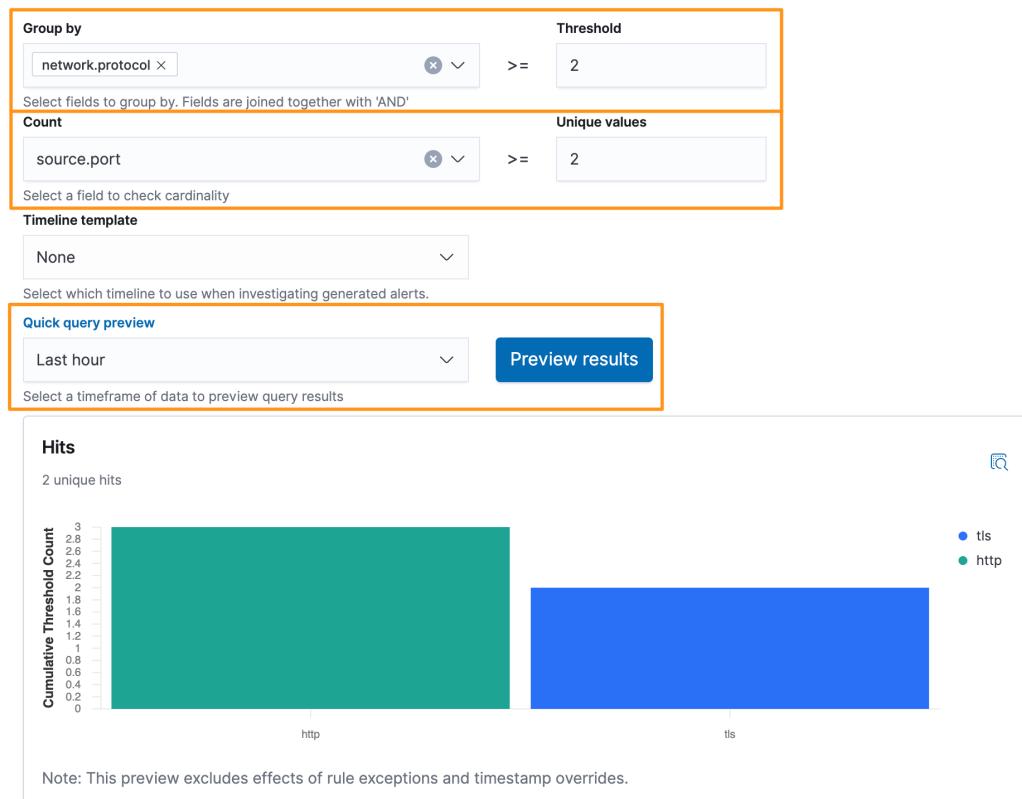


Figure 8.20 – Detection rule – threshold rule

Threshold rules are valuable when you are looking for things such as multiple processes calling the same domain or maybe user logins from more than one IP address.

In this section, we discussed creating a threshold detection rule; in the next section, we'll discuss creating a correlation rule with EQL.

Event correlation rule

As we discussed in *Chapter 7, Using Kibana to Explore and Visualize Data*, using the **Event Query Language (EQL)**, we can create a rule to correlate multiple events together. This is extremely helpful when correlating process and network events together.

In this example, I am using `process.entity_id`, which is a unique identifier for a process, to connect the process and network event together. I am correlating events where the cURL process starts and makes a connection:

```
sequence by process.entity_id
[process
  where event.type in ("start", "process_started")
    and process.name == "curl.exe"]
[network
  where event.type == "connection"]
```

Here we can see how this event correlation rule looks in Kibana:

The screenshot shows the Kibana Detection Rule configuration interface. At the top, there is an orange box highlighting the EQL query section. The query is as follows:

```

EQL query
sequence by process.entity_id
[process
  where event.type in ("start", "process_started")
  and process.name == "curl.exe"]
[network
  where event.type == "connection"]

```

Below the EQL query is a link to the "Event Query Language (EQL) Overview".

Underneath the EQL query is the "Timeline template" section, which is currently set to "None". A note below it says: "Select which timeline to use when investigating generated alerts."

Next is the "Quick query preview" section, which is set to "Last hour". To its right is a blue "Preview results" button.

At the bottom of the preview section is a note: "Select a timeframe of data to preview query results".

The main area is titled "Hits" and shows a histogram of the count of hits over time. The y-axis is labeled "Count" and ranges from 0 to 2. The x-axis shows dates from 06-01 22:40 to 06-01 23:35. There are two bars: one at 06-01 23:00 with a count of 2, and another at 06-01 23:30 with a count of 1.

A small note at the bottom of the histogram area says: "Note: This preview excludes effects of rule exceptions and timestamp overrides, and is limited to 100 results."

Figure 8.21 – Detection rule – correlation rule

This rule type can be used for any EQL rules or when you need to correlate multiple data types together.

Next, we'll explore the last rule type, the indicator match rule.

Indicator match rule

The indicator match rule is used to match local observations with indicators that are provided either by a previously ingested list (as we discussed using the Data Visualizer in *Chapter 7, Using Kibana to Explore and Visualize Data*) or from a threat feed.

In this section, we spent time creating and managing detection rules. Back on the main **Detection** page of the Security app, we'll continue down the page with the trend timeline.

The easiest way to do this will be to use the Threat Intel Filebeat module we set up in *Chapter 5, Building Your Hunting Lab – Part 2*.

The easiest way to generate some samples will be to identify a malicious domain and browse to it; after all, we're in a sandbox, right?

In the **Discover** app, go to your threat feed index pattern (`filebeat-*`). Let's apply a few filters so we can zero in on a good indicator to test:

```
event.dataset:threatintel.anomali and threatintel.indicator.type:domain-name
```

This will narrow our view to data provided by Anomali and only domains. This search conforms with the threat ECS fieldset.

Let's add the `threatintel.indicator.domain` field as a column and pick any domain:

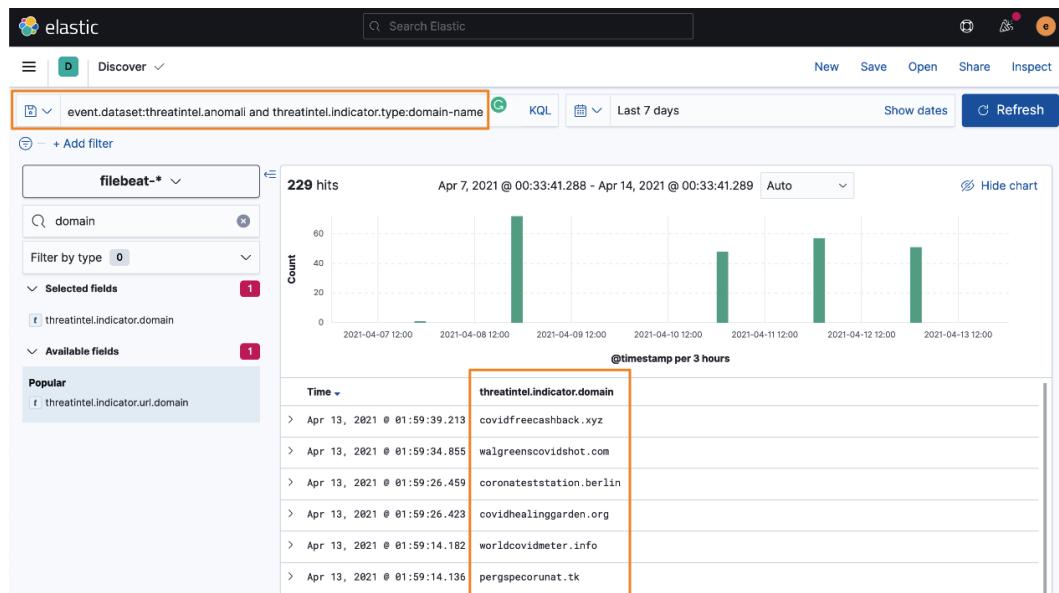


Figure 8.22 – Sample domain indicators

Now that we have the domain, let's go to our Windows box and use cURL to reach out and touch the domain:

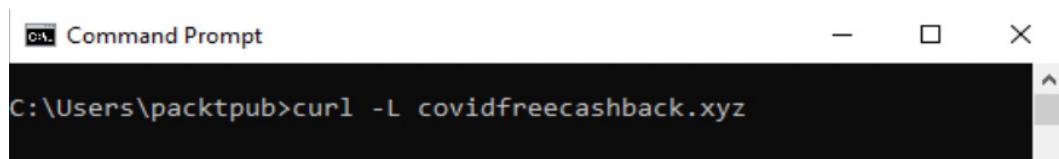


Figure 8.23 – Using cURL to generate indicator match traffic

Now that we've generated some data, let's go back to the **Detections** section of the Security app and create our indicator match rule.

Here, we'll use the same index patterns that we've been using.

We'll write a custom query for our local data. The default is `* : *`, meaning "match everything," but we're looking for domain traffic, so we can write a more specific rule to make it more performant:

```
event.category:"network" and event.type:"connection" and
destination.domain:*
```

Next up is the indicator index pattern; remember, we're using `filebeat-*`.

The indicator query can be tuned to require that the domain indicator exists, again, to make it more performant:

```
threatintel.indicator.type:"domain-name" and threatintel.
indicator.domain:*
```

Finally, we'll define what observation fields match what indicator fields. Here we're saying that the value of `destination.domain` must match `threatintel.indicator.domain`. You can extend this to include multiple fields to match, such as file hashes, IP addresses, and any fields that are present in both your local index patterns and your indicator data:

The screenshot shows the 'Detections' configuration page in the Elastic Security App. It is divided into four main sections: 'Custom query', 'Indicator index patterns', 'Indicator index query', and 'Indicator mapping'.

- Custom query:** Contains the query `event.category:"network" and event.type:"connection" and destination.domain:*`. The entire input field is highlighted with an orange box.
- Indicator index patterns:** Shows the index pattern `filebeat-*`. This section is also highlighted with an orange box.
- Indicator index query:** Contains the query `threatintel.indicator.type:"domain-name" and threatintel.indicator.domain:*`. The entire input field is highlighted with an orange box.
- Indicator mapping:** Shows a mapping between the **Field** `destination.domain` and the **Indicator index field** `threatintel.indicator.domain`. This section is highlighted with an orange box.

At the bottom left, there are buttons for `+ AND` and `+ OR`. At the top right, there are buttons for `G` (Green) and `KQL`.

Figure 8.24 – Detection rule – indicator match

Running this rule, we can see that our generated malicious traffic generated a rule alert:

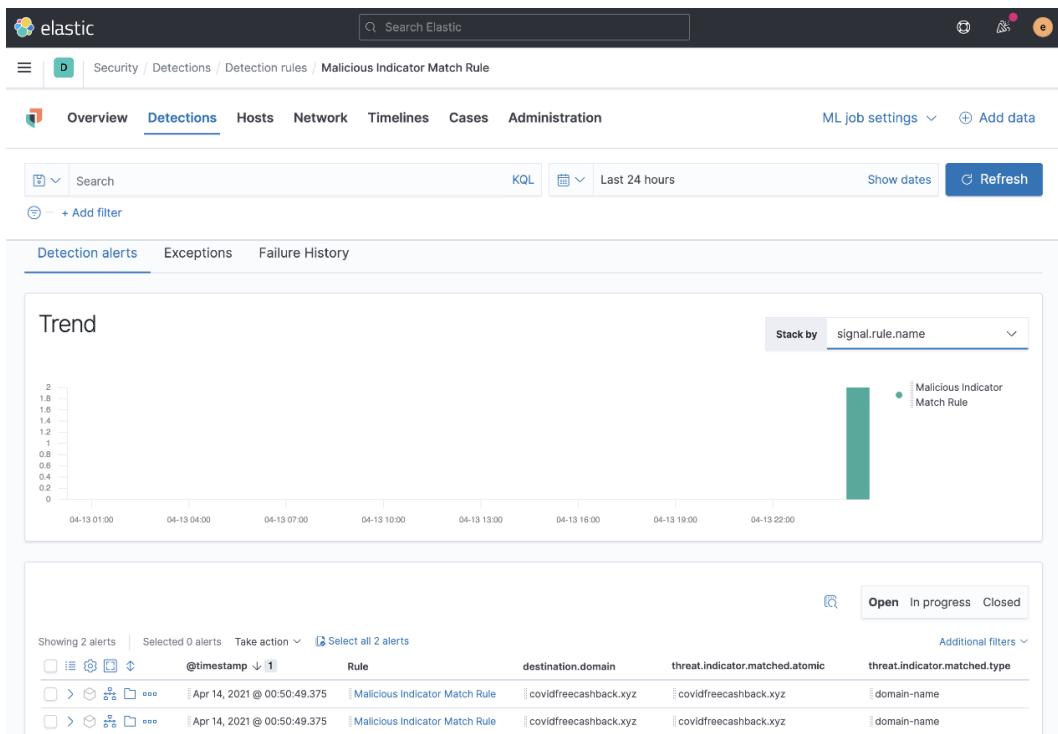


Figure 8.25 – Detection rule – indicator match alert

You may notice that there are fields that end in `.matched`. We're going to explore those in *Chapter 11, Enriching Data to Make Intelligence*, when we talk about indicator enrichment.

In this section, we created five detection rule types in the Security app. Next, we'll learn more about the **Detection alerts** page.

Trend timeline

Using the trend timeline, you can sort all events by their specific metadata. This is a view into all of the events and alerts that are generated by the detection engine. This is very helpful in identifying priority, risk, and criticality based on the detection rule settings. We'll discuss that more as we create our own rules.

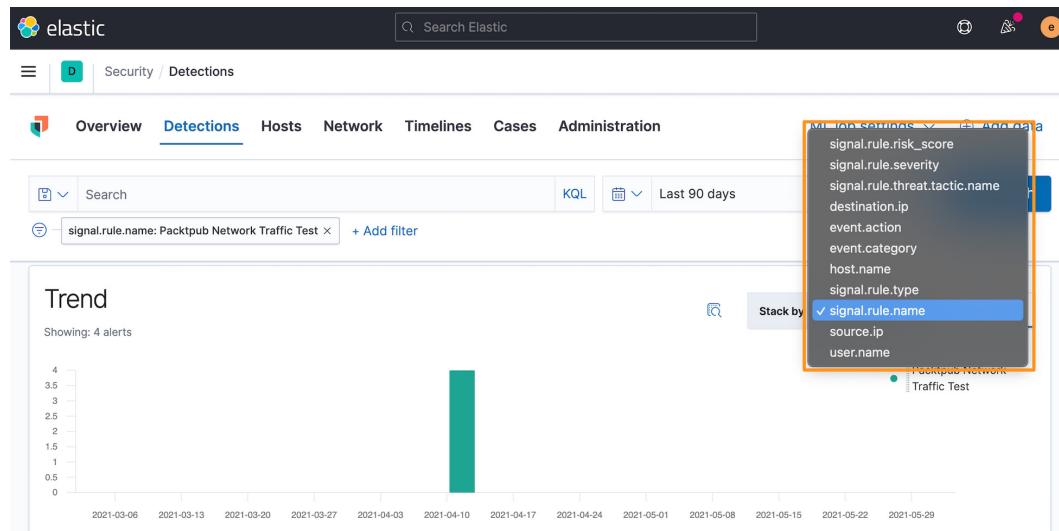


Figure 8.26 – Detection alerts trend sorting

If we continue down the page, we'll see more detailed information for each event that has occurred. While there is a lot in this single visualization, it is laid out in a way that makes it fairly intuitive.

From this visualization, we can customize the columns that are in our view, adjust the renderers, move to full screen, and sort the events:

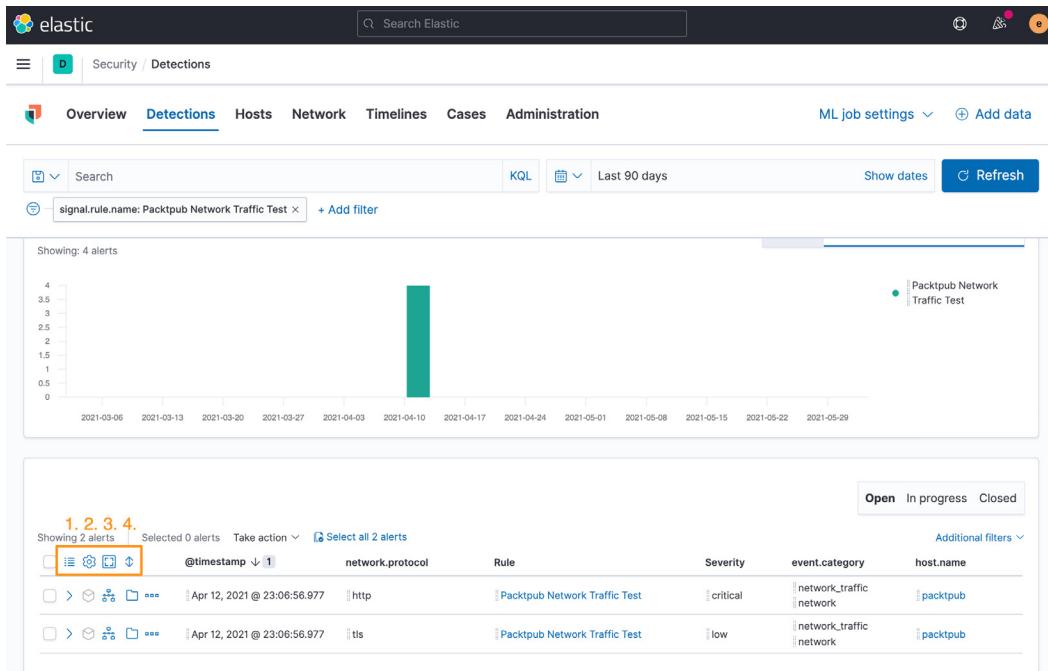


Figure 8.27 – Detection event organization

In the preceding screenshot, you can see four buttons (the icons labeled 1 through 4):

1. **Customize Columns**
2. **Customize Event Renderers**
3. **Full Screen**
4. **Sort**

Clicking on **Customize Columns** will allow you to select the different information that you'd like to show in the event details. I usually change this to include just the timestamp, rule name, module, category, host name, user name, filename, and destination domain. This allows me to get a quick look at the important information. There is plenty more to look at, but this provides you with the basics.

To change the columns, click on **Customize Columns** and then you can type the field names in the query and simply put a checkmark next to the ones you want to add:

The screenshot shows the 'Detection alerts' interface. At the top right is a button labeled 'Manage detection rules'. Below it, a message says 'Last alert: 1 minute ago'. On the left, a modal window titled 'Customize Columns' is open. It has a search bar containing 'file.name' and a table with two categories: 'file' and 'log'. Under 'file', there is a row for 'file.name' with a checked checkbox. An orange arrow points from the search bar to this checkbox. Another orange arrow points from the checked checkbox to the 'file.name' column header in the main event list below. The main event list shows a single alert for 'eicar.exe' with details about its detection as a malicious file.

Figure 8.28 – Adding columns to the detection event view

You can also search for any columns you want to remove, but I find it's faster to just click the **x** next to the column name:

This screenshot shows the 'Detection alerts' interface with the 'Customize Columns' modal open. The 'Method' column header has an orange box around its 'x' icon, indicating it is being removed. The main event list below shows the same alert for 'eicar.exe' as in Figure 8.28, but the 'Method' column is now empty.

Figure 8.29 – Removing columns from the detection event view

You can also click and drag columns around to reorganize them. This is helpful for your specific analytical process – organizing data in a logical way that makes sense to you.

Next to the **Customize Columns** button, there is the **Customize Event Renders** button. Adjusting the renderers allows you to change your view to a specific context. By default, we're just looking at the basic information about the events. If we click on **Customize Event Renderers**, we can select **Alerts**:

The screenshot shows the 'Customize Event Renderers' interface. At the top, there is a search bar labeled 'Search...' and two buttons: 'Disable all' and 'Enable all'. Below the search bar, there is a table with three columns: 'Name ↑', 'Description', and 'Example'. The 'Name ↑' column is sorted by name. The 'Description' column contains two entries: 'Alerts' (selected) and 'Auditd'. The 'Example' column shows event logs for both contexts. For 'Alerts', it shows a log entry from 'win2019-endpoint-1' preventing a malicious process. For 'Auditd', it shows a log entry for an audit event from the Linux Audit Framework.

Name ↑	Description	Example
<input checked="" type="checkbox"/> Alerts	Alerts are displayed when malware or ransomware is prevented and detected	<pre>win2019-endpoint-1 was prevented from executing a malicious process C:\Users\sean\Downloads\3be13acde2f4dcdded4fd8d518a513bfc9882407a6e384ffb17d12710db7d76fb.exe (6920) C:\Users\sean\Downloads\3be13acde2f4dcdded4fd8d518a513bfc9882407a6e384ffb17d12710db7d76fb.exe with result success # 3be13acde2f4dcdded4fd8d518a513bfc9882407a6e384ffb17d12710db7d76fb</pre>
<input type="checkbox"/> Auditd	Auditd audit events convey security-relevant logs from the Linux Audit Framework.	<pre>Session # 246 alices@zeek-london connected using > wget (1490) wget www.example.com with result success Destination 192.168.216.34 : 80</pre>

Figure 8.30 – Customize event renderers

Selecting **Alerts** organizes our events into a context of security events instead of a generic view.

You can view the event details in full screen mode by clicking the **Full Screen** button.

Finally, you can sort your events by the default of **@timestamp** or you can select a different field and go either ascending or descending:

The screenshot shows the 'Customize Event Renderers' interface with the 'Alerts' context selected. A dropdown menu is open under the 'Sort by' button, showing the current sort configuration: '@timestamp' (ascending). The dropdown also includes a 'Pick fields to sort by' button and a 'Clear sorting' button.

Figure 8.31 – Customize event renderers

Now that we've organized our data into an alert-centric context, let's explore the individual event detail options.

Immediately below the **View customization** section are five specific event detail options:

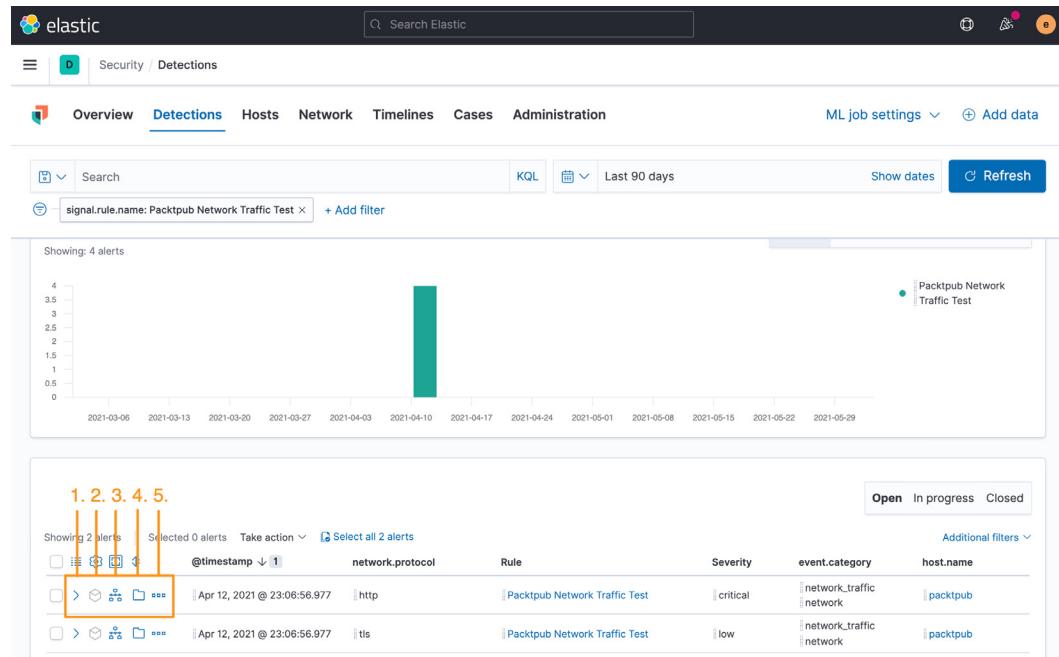


Figure 8.32 – Detection event details

In the preceding screenshot, there are five buttons (the icons labeled **1** through **5**):

1. **Event Details**
2. **Resolver**
3. **Add to Timeline**
4. **Add to Case**
5. **Event Actions**

Next, we'll walk through each of these options.

Event Details

Clicking on the **Event Details** icon will expand a slide-out pane on the right side of the screen with three different views: **Summary**, **Table**, and **JSON View**.

The **Summary** view will show you the high-level basics of the event, such as when it occurred, what rule was triggered, the severity, the risk score, the host name, and the user name. This provides basic information:

The screenshot shows the Elastic Detections interface. At the top, there's a navigation bar with the Elastic logo, a search bar, and various icons. Below the navigation bar, the main header says "Security / Detections". The left sidebar has tabs for "Overview", "Detections" (which is selected), "Hosts", "Network", "Timelines", and "Cases". Under "Detections", there's a search bar, a "KQL" button, and a "+ Add filter" link. The main content area is titled "Detection alerts" and shows a "Trend" chart for the last 11 minutes. An orange arrow points from the "Selected 0 alerts" link in the trend chart towards the detailed alert summary on the right. The summary details a "Malware Detection Alert" with the following fields:

Field	Value
signal.status	Open
@timestamp	Apr 6, 2021 @ 21:10:54.734
Rule	Malware Detection Alert
Severity	critical
Risk Score	99
host.name	packtpub
user.name	packtpub

Below the summary, there's a table showing two alerts. One row is highlighted with an orange box around the first column, which contains a file icon and a right-pointing arrow.

Figure 8.33 – Detection event details – Summary

The **Table** view shows you very granular data organized into a table similar to how an event would look in **Discover**. Like the **Customize Columns** menu we discussed earlier, you can use this view to add and remove columns:

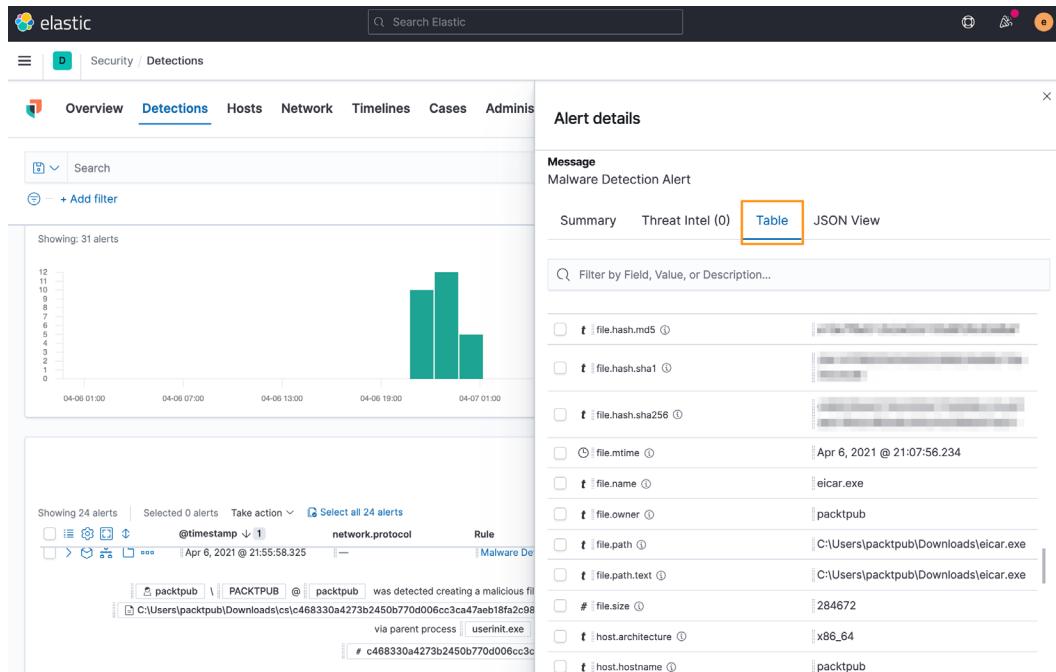


Figure 8.34 – Detection event details – Table

Finally, you can look at the raw data in JSON format. This is helpful if you have specific questions about how the data is structured. While looking at data this way can be busy, it can be helpful to look at similar events. As an example, it can be helpful to look at all the **file** or **process** information in one view:

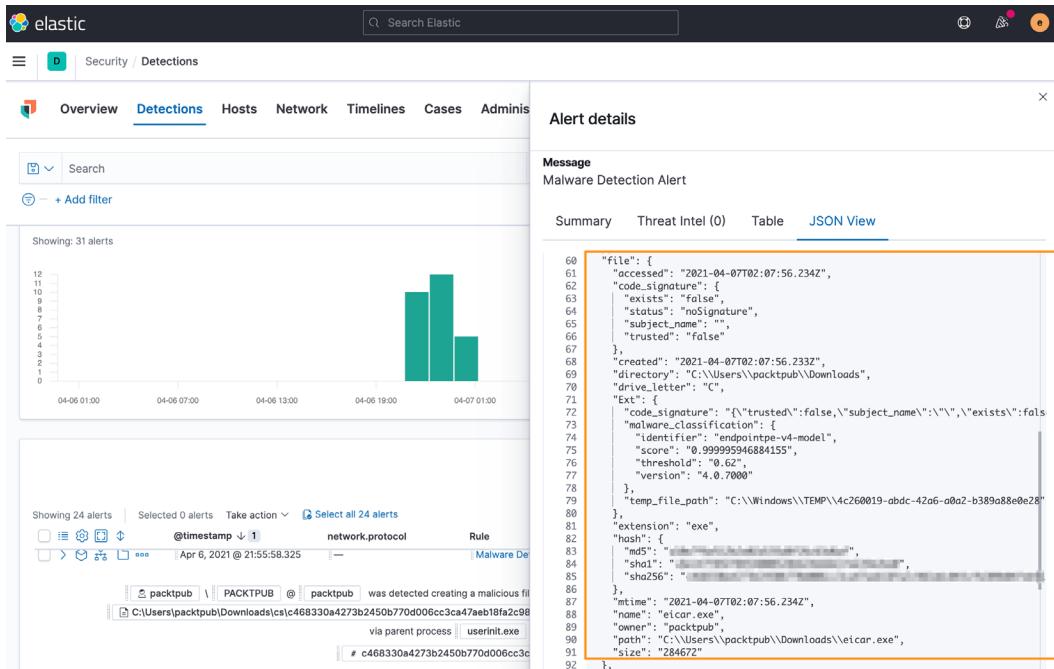


Figure 8.35 – Detection event details – JSON

Viewing the event details is often the first step in responding to events. Being able to move quickly between a summary to details in just a few clicks is very powerful, especially when you don't have to switch screens, portals, contexts, or views.

Next, we'll discuss the **Resolver** view, which is a valuable visualization to track events and their relationships.

Resolver

The **Resolver** view provides a tree-type view of file and network events. What makes **Resolver** particularly powerful is that it connects file and network events. From any security event that was generated by the Elastic Agent, there will be a **Resolver** icon that can be clicked to open the event in the **Resolver** view.

To highlight the utility of **Resolver**, I created a snapshot of my Windows VM, downloaded a malware sample of the Tesla Agent (a popular information stealer and remote access tool), and detonated it on the Windows system. As a reminder, we've configured the Elastic Agent Security Integration to only detect, not prevent.

I am going to obscure the identifying marks of the malware (hashes, network connections, and so on) because this is live malware that could absolutely cause damage. Additionally, adversary-controlled infrastructure may not be owned by them and I don't want to expose innocent victims if they are being used without their knowledge.

Executing the malware, we should take note that the filename is `tesla.exe`. This will help focus our search:

The screenshot shows a table of alerts. One alert is selected, highlighted with a yellow border. The alert details are as follows:

- Selected 0 alerts**
- Timestamp**: Apr 7, 2021 @ 22:48:09.922
- event.created**: Apr 7, 2021 @ 22:47:30.394
- Rule**: Malware Detection Alert
- file.name**: zYIPIYOP.exe
- Severity**: critical
- event.module**: endpoint
- Event Details:** packtpub \\ PACKTPUB @ packtpub was detected modifying a malicious file zYIPIYOP.exe in C:\Users\packtpub\AppData\Roaming\zYIPIYOP.exe via file tesla.exe (6172) C:\Users\packtpub\Downloads\tesla.exe via parent process explorer.exe (3704) with result success

Figure 8.36 – Malware event

Remembering the file we're tracking is called `tesla.exe`, we can look for `tesla.exe`. If we click on the **Resolver** icon and then the **4 file** button, we are shown some additional details:

The screenshot shows the Resolver view for the malware event. On the left, a list of events is displayed:

- file change** @ Apr 7, 2021 @ 22:43:14.020
C:\Users\packtpub\Downloads\tesla.exe
- file change** @ Apr 7, 2021 @ 22:43:06.281
C:\Users\packtpub\Downloads\ebd059221f...
- file creation** @ Apr 7, 2021 @ 22:42:48.022
C:\Users\packtpub\Downloads\tesla\ebd05...
- file creation** @ Apr 7, 2021 @ 22:39:26.825
C:\Users\packtpub\AppData\Roaming\Micro...

On the right, a diagram illustrates the relationships between processes. A central node labeled "ANALYZED EVENT · RUNNING PROCESS" (explorer.exe) has four arrows pointing to other nodes, each labeled with a number from 1 to 4. The nodes are:

- 1. A blue hexagon labeled "4 file 1 library 302 registry".
- 2. A blue hexagon labeled "302 registry".
- 3. A blue hexagon labeled "4 file 1 library".
- 4. A light blue cube labeled "TERMINATED PROCESS userinit.exe".

 A timer icon indicates a "1 second" delay between the numbered steps. The entire diagram is enclosed in a light gray box with zoom controls on the right side.

Figure 8.37 – Malware event – Resolver view

Resolver shows us some analysis information and even a few artifacts of me staging the malware! There are labels **1** through **4** in the screenshot:

1. This is a screenshot of the desktop wallpaper that was stored.
2. This was me staging the malware by unzipping it from an encrypted archive. This isn't part of our analysis, but it's great that the agent tracked and associated this with the activity. This would be useful telemetry in tracking an actual event.
3. This was me renaming the malware `tesla.exe` and moving it from the default archive folder. This isn't part of our analysis, but it's great that the agent tracked and associated this with the activity. This would be useful telemetry in tracking an actual event.
4. Here is the actual file that was detonated, `tesla.exe`. We knew that already, but we can see how it's recorded by the Agent.

We'll spend more time deep-diving into `tesla.exe` in *Chapter 9, Using Kibana to Explore and Visualize Data*, but for continuity, we'll continue to use this event for our examples throughout this chapter.

Next, we'll explore how to add events to a timeline.

Adding to a timeline

We had a brief introduction to timelines in the EQL section of *Chapter 7, Using Kibana to Explore and Visualize Data*. We'll discuss using timelines to natively build EQL queries later in this chapter in the *Timelines* section, but from within the detection engine, we can create a timeline using the document identification value or even drag a specific event field into a timeline.

The easiest way to create a timeline from an event is to click on the **Investigate in timeline** button next to the event, and this will create a new timeline using the event document ID. The document ID field is named `_id` and it is a unique value that is assigned when an event is indexed:



Figure 8.38 – Malware event – add an event to timeline

From here we can see this single event has been added to a timeline:

A screenshot of the Elastic Security App showing the 'Untitled timeline' page. The interface includes a search bar, a query builder with filters and search fields, and a table of log entries. The table columns are: @timestamp, message, event.category, event.action, and host. One entry in the table is highlighted with an orange border.

@timestamp	message	event.category	event.action	host
Apr 7, 2021 @ 22:48:09.922	Malware Detection Alert	malware intrusion_detection file	creation	pac
	packtpub PACKTPUB @ packtpub was detected creating a malicious file		in	
	C:\Users\packtpub\Downloads\tesla\		via	
	> explorer.exe (3704) C:\Windows\Explorer.EXE via parent process userinit.exe (4084) with result success			

Figure 8.39 – Malware event added to timeline

Additionally, we can also click and drag fields onto the timeline slide-out to add it to a timeline:

The screenshot shows the Elastic Security interface with a search bar at the top. Below it is a table with columns: Rule, Severity, and event.category. A single alert is listed: 'Malware Detection Alert' (Severity: critical) with category 'malware', 'intrusion_detection', and 'file'. The event message details a file modification by 'zYIPYOP.exe' via 'tesla.exe' (process ID 6172). At the bottom, there's a timeline builder with a placeholder 'Drop anything highlighted here to build an OR query' and a search bar containing 'file.hash.sha256: "██████████"'.

Figure 8.40 – Malware event – click and drag event to create timeline

Naming and providing a description for the timeline is helpful when you're tracking multiple events:

The screenshot shows the Elastic Security interface with a search bar at the top. Below it is a timeline titled 'Tesla Agent Timeline' with a count of 1 event. The timeline details a file modification by 'zYIPYOP.exe'. A modal window titled 'Name Timeline' is open, showing the title 'Tesla Agent Timeline' and a description 'Possible Tesla Agent events observed.' with a 'Save' button. The background shows the main interface with various filters and a search bar.

Figure 8.41 – Malware event – naming a timeline

We can also add this directly to an existing or new case (which we'll talk about in the next section):

The screenshot shows the Elastic Security App interface. At the top, there's a navigation bar with the elastic logo, a search bar, and user icons. Below it, the 'Security / Detections' tab is selected. A timeline entry for 'Agent Tesla' is shown, with metrics: Processes (16), Users (1), Hosts (1), and Source IPs (0). The timeline is autosaved every 40 seconds ago. Below the timeline, it says 'Possible Agent Tesla events observed.' A context menu is open over the timeline entry, with options like 'Add to favorites' and 'Attach to case'. The 'Attach to case' option has a submenu with 'Attach to new case' and 'Attach to existing case...'. The main timeline view shows a query for 'file.name: "zYIPIYOP.exe"', search results, and a detailed log entry.

Agent Tesla Timeline

Autosaved 40 seconds ago

Processes	Users	Hosts	Source IPs
16	1	1	0

Possible Agent Tesla events observed. 0

Query 75 Correlation Analyzer Notes Pinned

Last 90 days Show dates Refresh All data sources

(file.name: "zYIPIYOP.exe")
 OR () + Add field

Filter Search G KQL

+ Add filter

@timestamp ↓ 1	message	event.category	event.action
Apr 19, 2021 @ 22:33:21.280	Malware Detection Alert	malware intrusion_detection file	modification
packtpub \ PACKTPUB @ packtpub was detected modifying a malicious file zYIPIYOP.exe in C:\Users\packtpub\Desktop\zYIPIYOP.exe via > cmd.exe (6688) C:\Windows\system32\cmd.exe via parent process explorer.exe (3780) with result success			

Figure 8.42 – Malware event – adding a timeline to a case

Here we learned the two ways that we can create timelines from the **Detection alerts** page. Next, we'll discuss how to create a case.

Adding to a case

Cases is a developing feature in Kibana. Currently, it is a case management capability to save links to data and make notes.

We'll work more with cases later in this chapter, but from the **Events** page, you can click the **Cases** button and add an event to a new or existing case:

The screenshot shows the Kibana Events page with the following details:

- Header: Open, In progress, Closed
- Search bar: Showing 7 alerts, Selected 0 alerts, Take action, Select all 7 alerts
- Filter: Additional filters, Severity, event.cat
- Event list:
 - Alert timestamp: Apr 19, 2021 @ 22:33:21.280
 - Rule: Malware Detection Alert
 - Severity: critical
 - Category: malware, intrusion, file
 - Description: packtpub was detected modifying a malicious file zYiPIYOP.exe in C:\Windows\system32\cmd.exe via cmd.exe (6688) process explorer.exe (3780) with result success
 - Action buttons: Add to new case, Add to existing case (highlighted with an orange box)

Figure 8.43 – Malware event – adding an event to a case

Here we showed how to add an event to a case. Next, we'll explore how to modify the status of, or even make exceptions to, an event.

Event actions

Now that we've looked at the different things we can do from an analysis perspective with an event, there are some administrative actions that we can take on an event.

We can mark an event as **In progress** or **Closed**. This simply helps with the event response organization so that multiple analysts can work on the alerts without stepping on each other's toes. When you mark an event as **In progress** or **Closed**, it is filtered from the default **Open** view. To find out who is working on an event, it must be in a case:

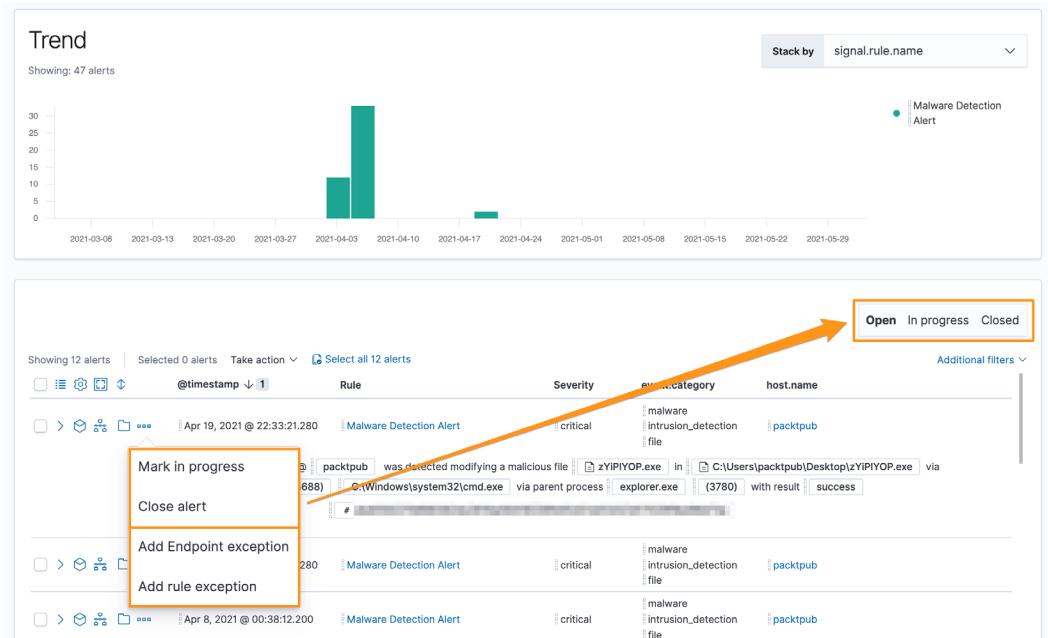


Figure 8.44 – Malware event – changing event Status

Additionally, we can also create either a rule or an endpoint exception. Both of these exception types prevent alerts from being generated when their conditions are met.

The difference between a rule exception and an endpoint exception is that the endpoint exception is evaluated on the endpoint and the rule exception is evaluated in the detection engine. This difference is extremely important if you are taking preventative measures on the endpoint, where you'd want the exception evaluated on the endpoint or the prevention would still occur.

When creating either exception type, you can choose to close the current alert as well as any other alerts that match the criteria. Endpoint exceptions can remove an endpoint from quarantine if the exception criteria are met.

Clicking on the *three dots* next to the alert allows you to create an exception of either type:

Add Endpoint Exception

Malware Detection Alert

Alerts are generated when the rule's conditions are met, except when:

Field	Operator	Value	Actions
file.Ext.code_signature	is	—	
subject_name	is	Search field value...	
trusted	is	false	
file.path.caseless	is	C:\Users\packtpub\Downloads\zYiPIYOP.exe	
file.hash.sha256	is	[REDACTED]	

AND

Close this alert

Close all alerts that match this exception and were generated by this rule (Lists and non-ECS fields are not supported)

On all Endpoint hosts, quarantined files that match the exception are automatically restored to their original locations. This exception applies to all rules using Endpoint exceptions.

Cancel

Add Endpoint Exception

Figure 8.45 – Malware event – Add Endpoint Exception