# *Elastic SIEM Rule Creation Demo*

***Written by: Thomas Dominach Jr.***

## *Introduction*

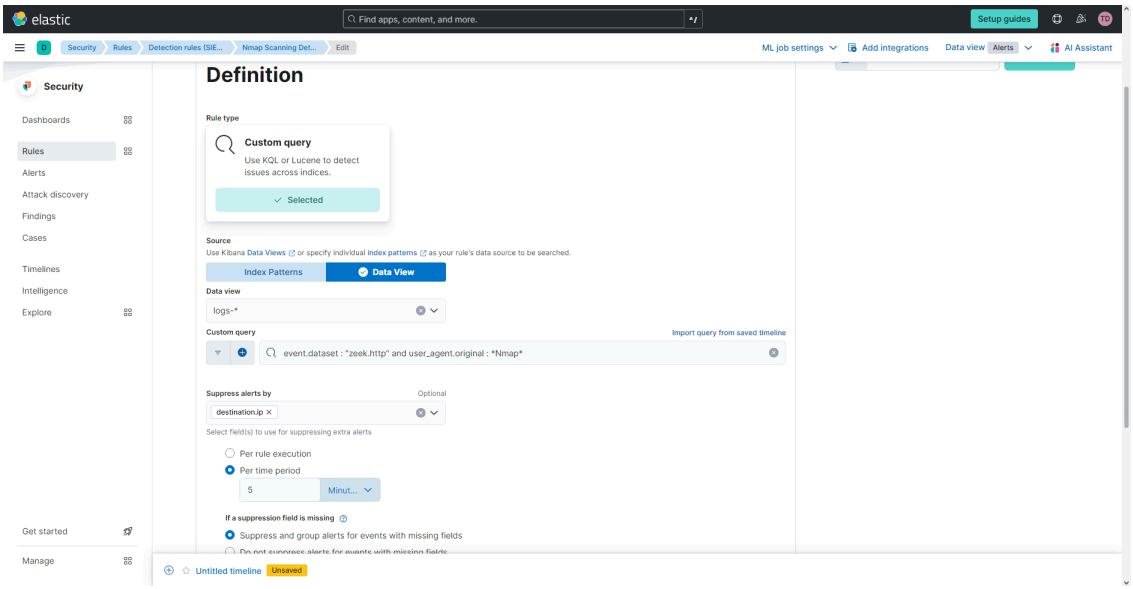
In this demonstration I created two new rules for my Elastic SIEM. One rule utilizes a custom query to detect and raise an alert for Nmap scans. The second is a threshold-based rule which detects possible DoS attacks.

## *Purpose*

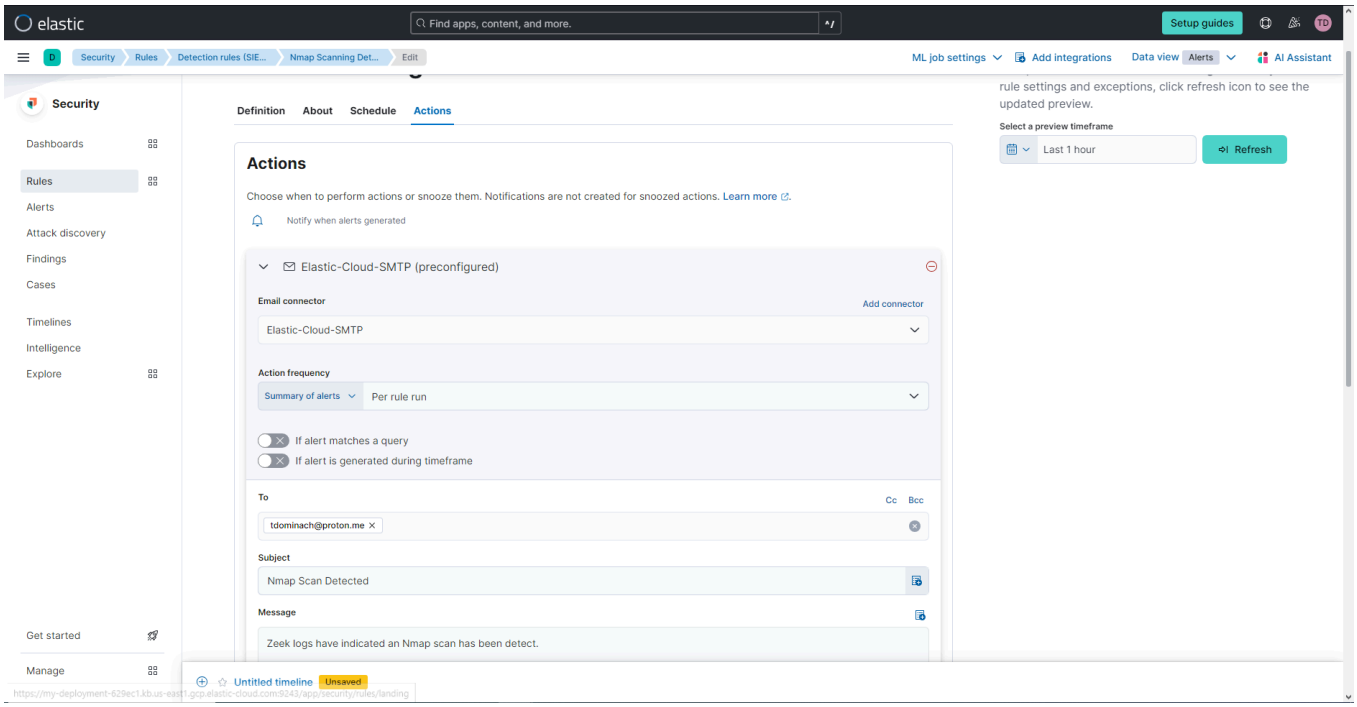
Creating custom rules for my SIEM enables me to closely analyze logs so I can learn to recognize malicious web events. My goal is to get a general idea of what it's like working with SIEM logs as an analyst. This demo works as an exercise for recognizing patterns of suspicious web traffic.

# *Execution*

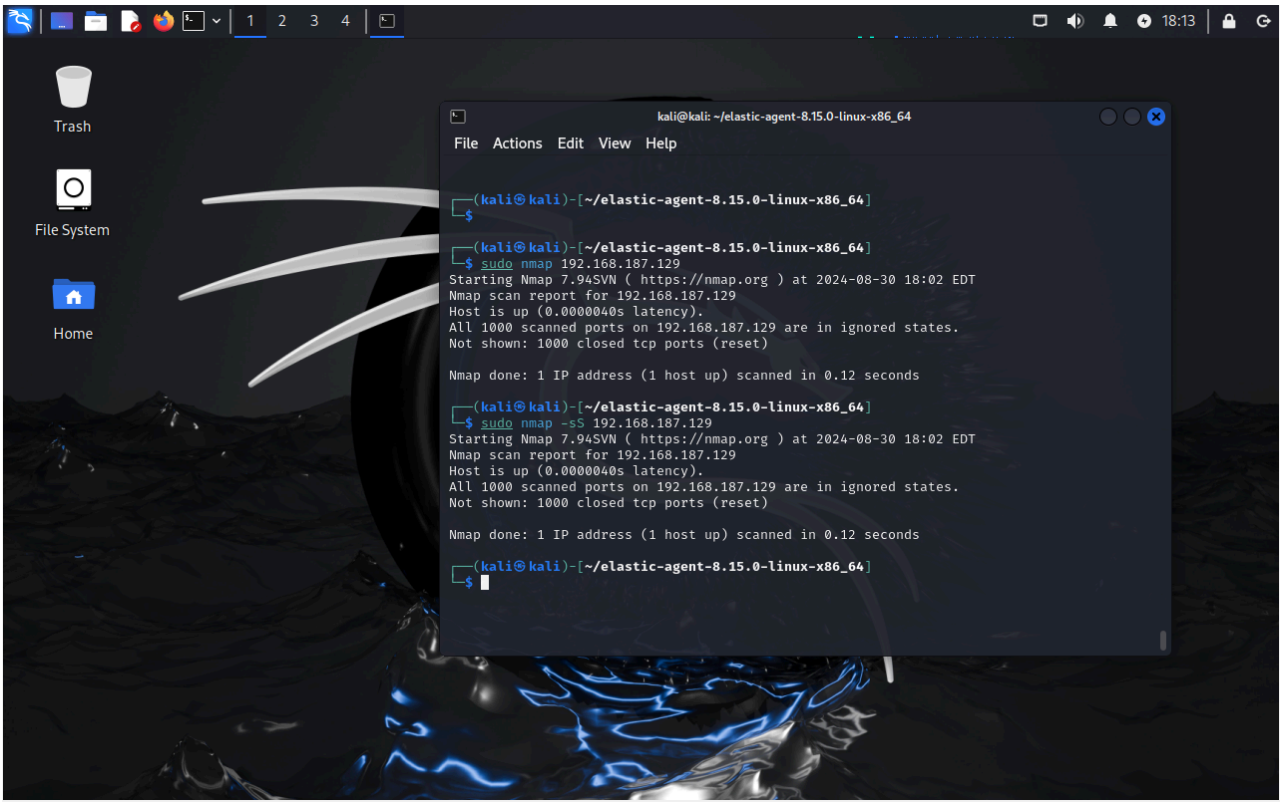
The first custom rule to be implemented into my Elastic SIEM is a query rule. The rule will detect any Nmap scans that are being directed towards the client VM. Within the Elastic web application there are multiple rule types. As seen in the figure below the rule type being used will be a custom query. Thanks to Zeek, which is an open source network traffic analyzer, I'm able to obtain detailed logs of web events. By utilizing the Elastic event field "event.dataset" in my query I'm able to narrow down my rule to only include logs produced by Zeek. In order to equate event.dataset to the Zeek dataset I enter the following condition *"event.dataset : zeek.http"*. An additional condition will need to be added to the query so Nmap scans can be detected. The Elastic user agent field user_agent.original provides the information needed to narrow down any Nmap scans. User_agent.original provides the unparsed string in a log, which can include any mentions of Nmap. The second condition will be written as *"user_agent.orginal : *Nmap"* in order to find logs that contain "Nmap" in the user agent string. Putting both conditions together with the "and" operator results in the query that will detect any Nmap scans on my client VM.



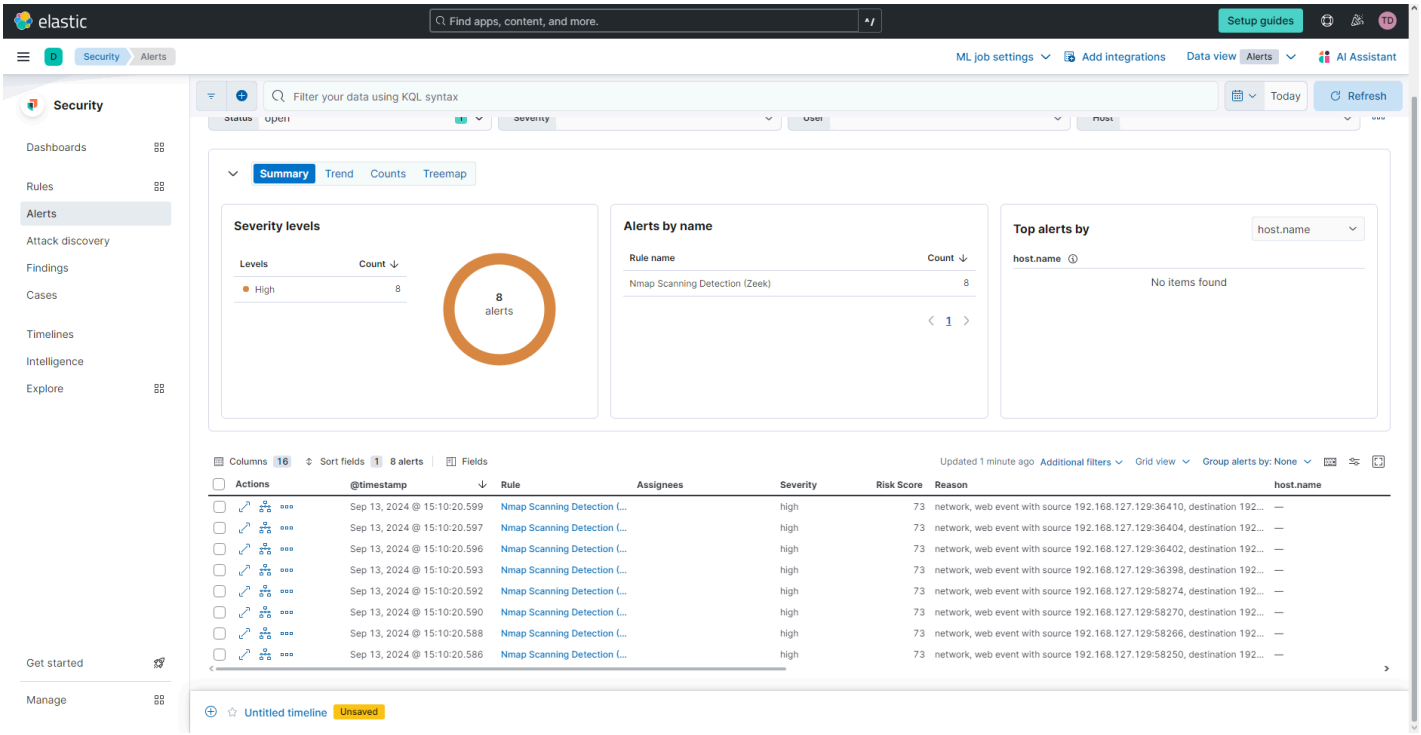
Elastic provides a numerous amount of actions that can occur in the event a rule is detected on the SIEM. I chose to have an email notification be sent to my inbox in order to confirm my rule works.



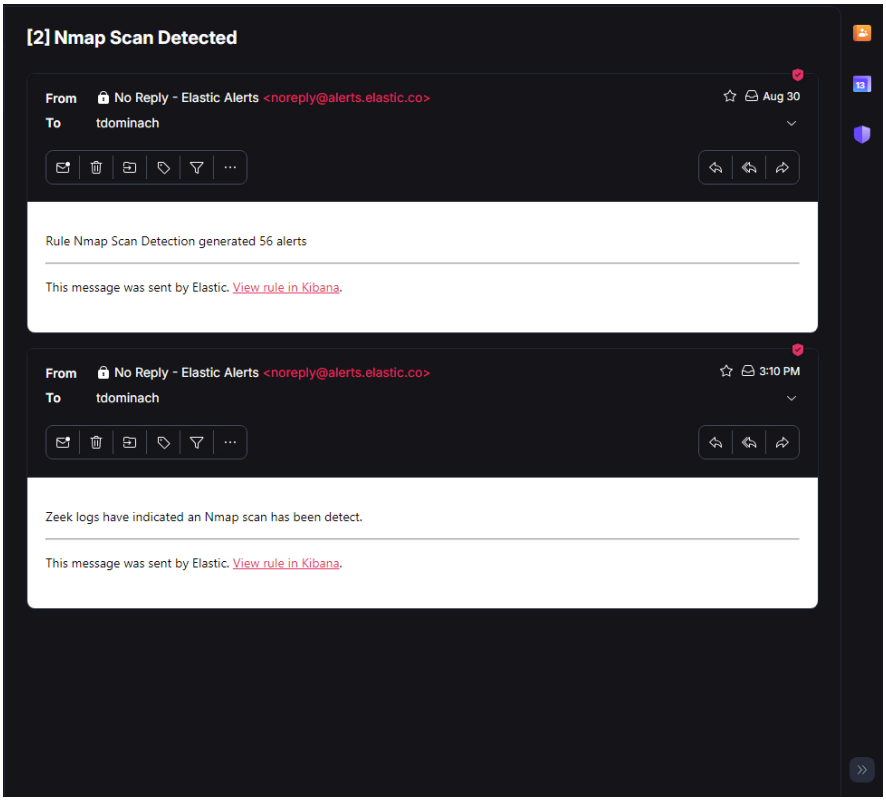

After setting up the Nmap rule, I logged into the Kali attacker VM and performed an Nmap scan of the client VM.



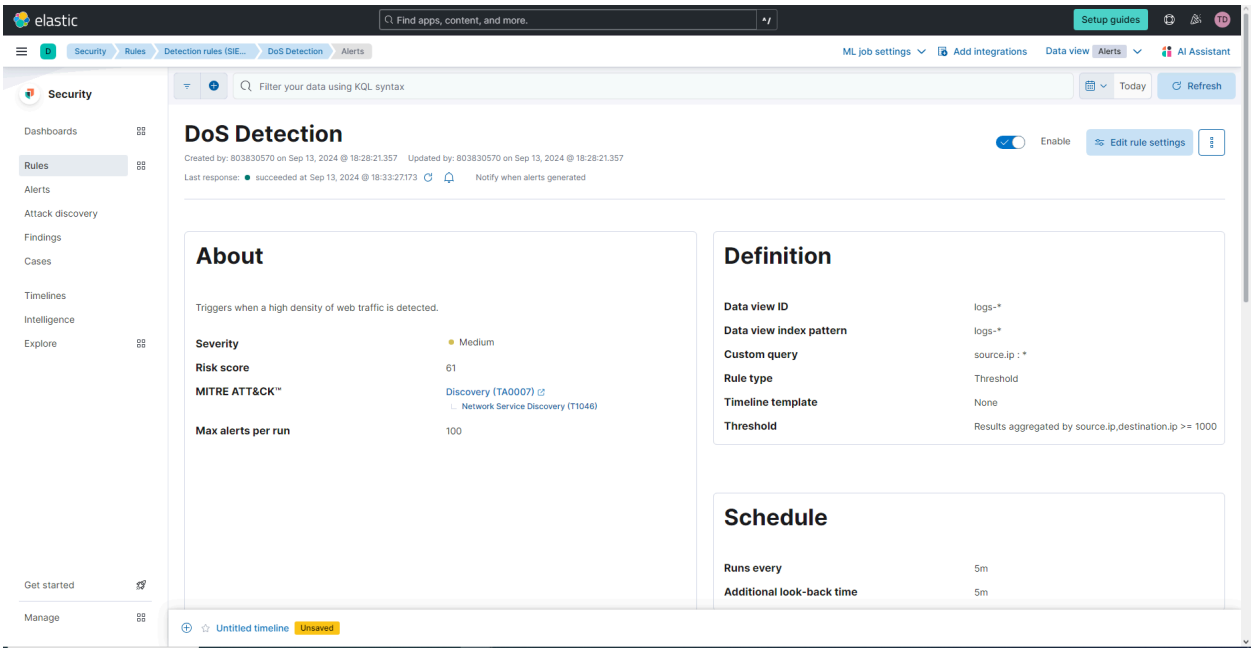
After the Nmap scan, the SIEM is able to detect the rule and produce alerts succesfully. As seen in the screenshot below 8 alerts were produced after performing the Nmap scan on the attacker VM.




I was able to confirm an email notification reached my inbox as well.

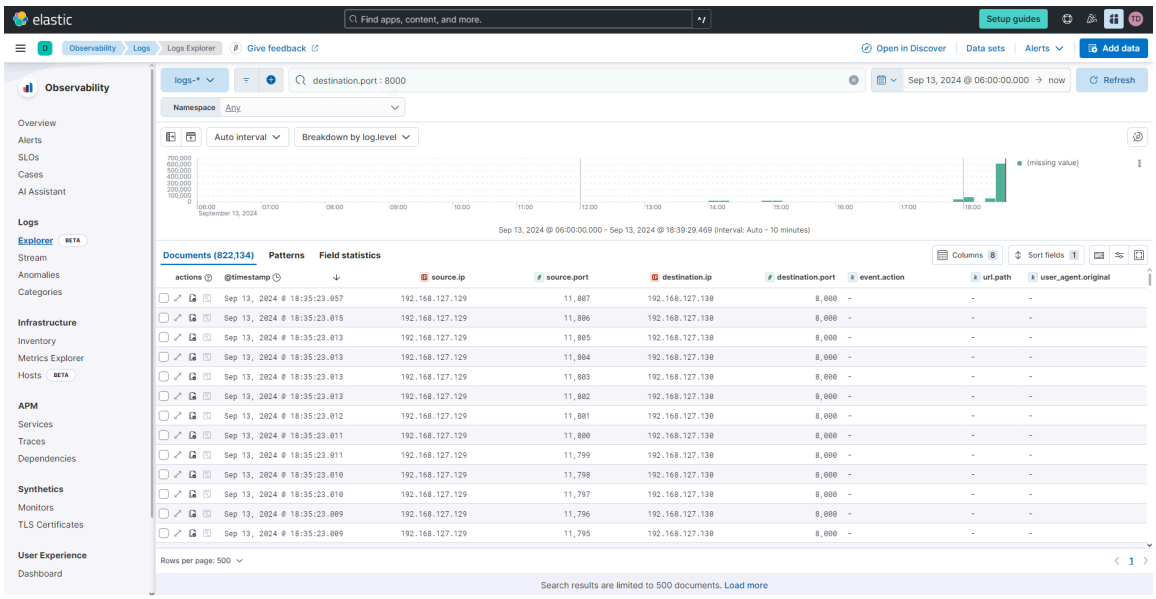


After successfully creating a custom query rule, I then moved onto making a threshold rule on the SIEM. Since I want to make a rule that will detect possible DoS attacks the threshold rule will need to take into account the *source.ip* and *destination.ip* field within the logs. After setting the threshold to both values the SIEM should be able to raise an alert if there is any repetitive traffic between the same source and destination IP in a short amount of time.

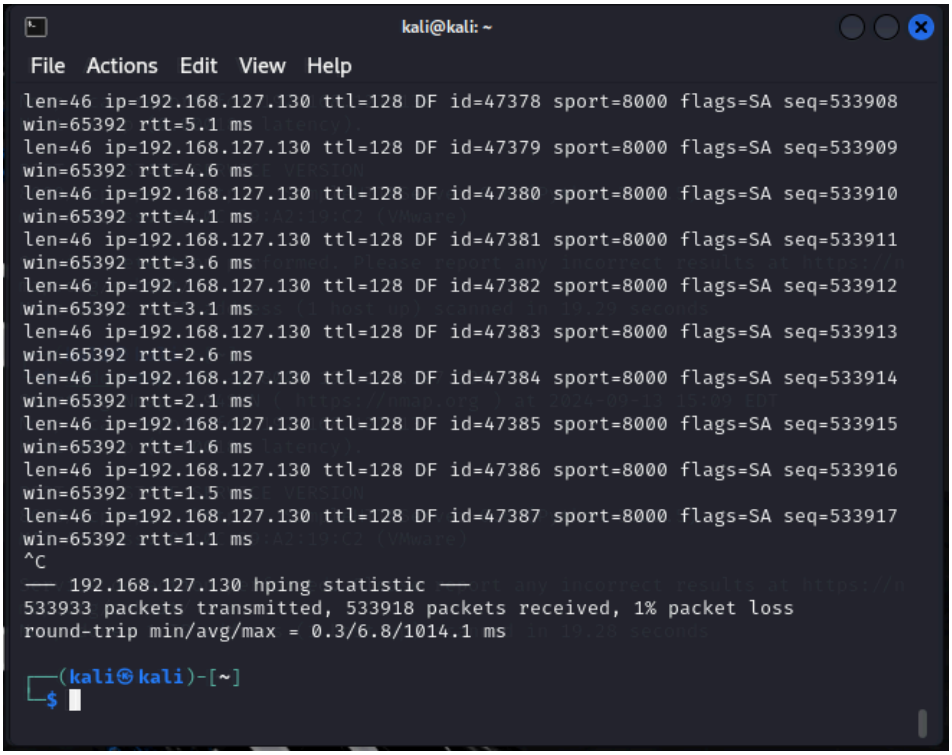



In order to test this rule a tool called hping3 on the Kail VM will continuously send TCP/IP packets to the Windows Client VM at a high rate. The following command will be used to send the packets on port 8000, which is being used to host a Python server on the Windows Client.

The hping3 command resulted in the following entry within the SIEM logs.




As you can see from the figure above hping3 was successful in sending a staggering amount of TCP/IP packets to the Windows Client. The rate at which the packets were sent triggered the alert for my DoS Detection rule almost immediately. After stopping the command on the Kali VM, it can be seen on the console just over half a million packets were sent in only a couple minutes.



Similar to the Nmap detection rule, I set the DoS detection rule to send an alert notification via email to confirm it was triggered in the SIEM. After checking my inbox I was able to confirm my second rule was successful.