Analysis in Practice

The previous section defined network traffic analysis, the dependencies for performing traffic analysis, and its importance. This section will break down a workflow for performing traffic analysis, and we will become familiar with the key components.

This is not an exact science. It can be a very dynamic process and is not a direct loop. It is greatly influenced by what we are looking for (network errors vs. malicious actions) and where you have visibility into your network. Analysis can be distilled down to a few basic tenets, however.

Descriptive Analysis

Descriptive analysis is an essential step in any data analysis. It serves to describe a data set based on individual characteristics. It helps to detect possible errors in data collection and/or outliers in the data set.

1. What is the issue?
    - Suspected breach? Networking issue?
2. Define our scope and the goal. (what are we looking for? which time period?)
    - Target: multiple hosts potentially downloading a malicious file from bad.example.com
    - When: within the last 48 hours + 2 hours from now.
    - Supporting info: filenames/types 'superbad.exe' 'new-crypto-miner.exe'
3. Define our target(s) (net / host(s) / protocol)
    - Scope: 192.168.100.0/24 network, protocols used were HTTP and FTP.

Using our workflow, we will determine our issue, what we are looking for, when, and where to find it. Descriptive analysis covers these critical concepts for our analysis.

Diagnostic Analysis

Diagnostic analysis clarifies the causes, effects, and interactions of conditions. In doing so, it provides insights that are obtained through correlations and interpretation. Characteristic here is a backward-looking view, as in the closely related descriptive analytics, with the subtle difference that it tries to find reasons for events and developments.

4. Capture network traffic
    - Plug into a link with access to the 192.168.100.0/24 network to capture live traffic to try and grab one of the executables in transfer. See if an admin can pull PCAP and/or netflow data from our SIEM for the historical data.
5. Identification of required network traffic components (filtering)
    - Once we have traffic, filter out any packets not needed for this investigation to include; any traffic that matches our common baseline and keep anything relevant to the scope of the investigation. For example, HTTP and FTP from the subnet, anything transferring or containing a GET request for the suspected executable files.
6. An understanding of captured network traffic
    - Once we have filtered out the noise, it is time to dig for our targets—filter on things like `ftp-data` to find any files transferred and reconstruct them. For HTTP, we can filter on `http.request.method == "GET"` to see any GET requests that match the filenames we are searching for. This can show us who has acquired the files and potentially other transfers internal to the network on the same protocols.

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

By capturing traffic around the source of our issue, clearing out any known good data, and then taking the time to inspect and understand what is left, we can determine if it is the cause of our problem. In doing so, we just performed diagnostic analysis. We are validating the cause of our problems and examining the events surrounding them.

Predictive Analysis

By evaluating historical and current data, predictive analysis creates a predictive model for future probabilities. Based on the results of descriptive and diagnostic analyses, this method of data analysis makes it possible to identify trends, detect deviations from expected values at an early stage, and predict future occurrences as accurately as possible.

7. Note-taking and mind mapping of the found results
    - Annotating everything we do, see, or find throughout the investigation is crucial. Ensure we are taking ample notes, including:
    - Timeframes we captured traffic during.
    - Suspicious hosts within the network.
    - Conversations containing the files in question. (to include timestamps and packet numbers)
8. Summary of the analysis (what did we find?)
    - Finally, summarize what we have found explaining the relevant details so that superiors can decide to quarantine the affected hosts or perform more significant incident response.
    - Our analysis will affect decisions made, so it is essential to be as clear and concise as possible.

By performing an evaluation of the data we have found, comparing it to our baseline traffic, and known bad data such as markers of infiltration or exploitation (like signatures for viruses and other hacking tools), we are performing Predictive Analysis. In this process, we paint a clear picture so that appropriate actions can be taken in response.

Prescriptive Analysis

Prescriptive analysis aims to narrow down what actions to take to eliminate or prevent a future problem or trigger a specific activity or process. Using the results of our workflow, we can make sound decisions as to what actions are required to solve the problem and prevent it from happening again. To prescribe a solution is the culmination of this workflow. Once done and the problem is solved, it is prudent to reflect on the entire process and develop lessons learned. These lessons, when documented, will enable us to make our processes stronger—document what was done correctly, what actions failed to help, and what could improve.

This workflow is an example of how to begin the analysis process on captured traffic. Above we broke it down into its parts to explain where they fit within the analysis process and with which type of analysis it belongs. We include it here again as a whole so that it can serve as a template.

1. What is the issue?
    - Suspected breach? Networking issue?
2. Define our scope and the goal (what are we looking for? which time period?)
    - target: multiple hosts potentially downloading a malicious file from bad.example.com
    - when: within the last 48 hours + 2 hours from now.
    - supporting info: filenames/types 'superbad.exe' 'new-crypto-miner.exe'
3. Define our target(s) (net / host(s) / protocol)
    - scope: 192.168.100.0/24 network protocols used were HTTP and FTP.
4. Capture network traffic
    - plug into a link with access to the 192.168.100.0/24 network to capture live traffic to try and grab one of the executables in transfer. See if an admin can pull PCAP and/or netflow data from our SIEM for the historical data.
5. Identification of required network traffic components (filtering)
    - once we have traffic, filter out any traffic not needed for this investigation to

include; any traffic that matches our common baseline and keep anything relevant to the scope. `HTTP and FTP from the subnet, anything transferring or containing a GET request for the suspected executable files.

6. An understanding of captured network traffic
    - Once we have filtered out the noise, it's time to dig for our targets—filter on things like ftp-data to find any files transferred and reconstruct them. For HTTP, we can filter on http.request.method == "GET" to see any GET requests that match the filenames we are searching for. This can show us who has acquired the files and potential other transfers internal to the network on the same protocols.
7. Note-taking and mind mapping of the found results.
    - Annotating everything we do, see, or find throughout the investigation is crucial. Ensure we are taking ample notes, including:
    - Timeframes we captured traffic during.
    - Suspicious hosts within the network.
    - Conversations containing the files in question. (to include timestamps and packet numbers)
8. Summary of the analysis (what did we find?)
    - Finally, summarize what has been found, explaining the relevant details so that superiors can make an informed decision to quarantine the affected hosts or perform more significant incident response.
    - Our analysis will affect decisions made, so it is essential to be as clear and concise as possible.

Often this process is not a once-and-done kind of thing. It is usually cyclic, and we will need to rerun steps based on our analysis of the original capture to build a bigger picture. This could have been a much larger attack than what is in the examples. Suppose a full-scale incident response is deemed necessary. In that case, we may have to reanalyze the PCAP previously captured to look at any conversations that involve the affected hosts within several minutes of the executable transfer to ensure it did not spread over another route, as an example.

Key Components of an Effective Analysis

1. Know your environment

There are several key components to perform traffic analysis effectively. First, know the environment. If we are unsure if a host belongs in the network, how can we determine if it is rogue or not? Keeping asset inventories and network maps is vital. These will aid in the analysis process.

2. Placement is Key

Next, the placement of our host for capturing traffic is a critical thing. Closest to the source of the issue is the ideal placement of our capturing tool. If the traffic in question is coming from the internet, listening to the inbound links is a great way to see the complete picture. It is as close to the source as we, the administrators, can get. If the problem seems to be isolated to one host on our internal network, try placing the capture tools in the same segment as the problem host and see what traffic is happening within the segment.

3. Persistence

Persistence is the next critical component for us. The issue will not always be easy to spot. It may not even be a frequent event on the network. For example, an attacker's Command and Control server reaching out to the victim's computers may only happen on a time interval of once every several hours, or even once a day or less. This means that if we did not catch it the first time around, it might be a while before it appears in our logs. Don't lose the drive to find the problem. It could mean the difference between stopping the attacker and a full-scale breach like a ransomware attack.

Analysis Approach

We have spent some time discussing the analysis process and how to start a basic workflow when performing our

tasks. Let's take a second to discuss some easy wins when looking at traffic and finding problems.

Start with `standard protocols first` and work our way into the `austere and specific` only to the organization. Most attacks will come from the internet, so it has to access the internal net somehow. This means there will be traffic generated and logs written about it. HTTP/S, FTP, E-mail, and basic TCP and UDP traffic will be the most common things seen coming from the world. Start at these and clear out anything that is not necessary to the investigation. After these, check standard protocols that allow for communications between networks, such as SSH, RDP, or Telnet. When looking for these types of anomalies, be mindful of the security policy of the network. Does our organization's security plan and implementations allow for RDP sessions that are initiated outside the enterprise? What about the use of Telnet?

Look for `patterns`. Is a specific host or set of hosts checking in with something on the internet at the same time daily? This is a typical Command and Control profile setup that can easily be spotted by looking for patterns in our traffic data.

Check anything `host to host` within our network. In a standard setup, the user's hosts will rarely talk to each other. So be suspicious of any traffic that appears like this. Typically hosts will talk to infrastructure for IP address leases, DNS requests, enterprise services and to find its route out. We will also see hosts talking with local webservers, file shares, and other critical infrastructure for the environment to function like Domain controllers and authentication apps.

Look for `unique` events. Things like a host who usually visits a specific site ten times a day changing its pattern and only doing so once is curious. Seeing a different User-Agent string not matching our applications or hosts talking to a server out on the internet is also something to be concerned with. A random port only being bound once or twice on a host is also of note. This could be an opening for things like C2 callbacks, someone opening a port to do something non-standard, or an application showing abnormal behavior. In large environments, patterns are expected, so anything sticking out warrants a look.

`Don't be afraid to ask for help.` This may seem overstated and obvious, but after a bit of time staring at packet captures, things can blend together, and we may not see the whole picture. Having a second set of eyes on the data can be a huge help in spotting stuff that may get glossed over.

In summary, the analysis process is a very dynamic task, and our days will never be the same. Keep learning, understand what is going on around us, and as your skills grow, so will the ability to detect threats. This process does not solely rely on the use of tools such as tcpdump and Wireshark. There are many helpful tools like Snort, Security Onion, Firewalls, and SIEMs that can help enrich our understanding of the environment and provide better protection. Do not be afraid to utilize these in investigations.