Activity File: Domain-Specific Interview Questions

Question I: Setting Alerts in a New Monitoring System

"How do you determine which alerts to set in a new monitoring system?"

Note: In Project 3, you configured a series of alerts based on your knowledge of the tactics used to infiltrate the target machine. This question provides an opportunity to explain how you reasoned through configuring their thresholds and metrics.

1. Restate the Problem

A new monitoring system must be carefully matched to business needs to protect against potential exploits.

2. Provide a Concrete Example Scenario

- Describe the network provided for Project 3. Identify the VMs on the network and what they do.

The virtual network in Project 3 consisted of a Virtual RDP host, two target machines, a Kibana server for monitoring and a Kali attack machine.

- Which VMs had public Internet access? The RDP/Hyper-V host and the target machines, Kali.

- Which VMs did not have public Internet access? Kibana.

- Which VMs do you expect to receive traffic from the Internet, if any? Target1 as it had HTTP running on open port 80.

- Which VMs do you expect to receive traffic from the local network, if any?

The target machines since Kali can attack them, and Kibana which can monitor the target hosts.

- Which protocols did you observe on the Project 3 network?

Target1 – HTTP:80 and SSH:22.

3. Explain the Solution Requirements

- Based on the likely origins, sources, and protocols identified above, which kinds of malicious traffic are most likely to appear on the network?

Unsecured internet is at risk for Brute Forcing attempts, app scanning, and the observation of data. SSH is at risk for malicious actors gaining a shell into the system with potential root access.

- How would you baseline your network to validate your expectations?

- For each type of malicious traffic:

- Which metric would you set?

- What threshold would you set?

- Why?

Brute Forcing – establish a baseline over ‘normal’ level of failed login attempts. HTTP 400 series responses, typically Unauthorized messages. Any level 50% higher than the top incidence of failures during established stable operations.

App Scanning – Excessive HTTP requests to alert when an agent has an inordinate requisition for data outside ‘normal’ bounds. Vital system file sizes should be considered in setting the threshold, as some applications can be storage intensive, and some not.

SSH Compromise – Similar to attempts on HTTP, you’ll want to look for excess failed logins as Brute Forcing may be underway. Email alerts are recommended when someone attempts to login in this manner.

4. Explain the Solution Details

- Which tools in Project 3 did you use to set these alerts?

In Kibana, different ‘beats’ are used to monitor the system: Filebeat for log collection, Metricbeat for monitoring system hardware resources, and Packetbeat to assess network traffic. Metricbeat is useful to see when the CPU is receiving excess computation requests (when an attack may be underway); Packetbeat is crucial to monitor incoming traffic attempting to potentially break-in the system or exfiltrate data.

- Which steps did you take to configure them?

Counts, sums and levels are set for each beat complete with thresholds which an alert is triggered afterward. Kibana also has a Watcher feature that provides the ability to see alerts triggered in real time (when an attack is potentially underway) as well as a history of said alerts.

5. Identify Advantages and Disadvantages of the Solution

In the case of Excessive HTTP Requests, that alert was firing in real-time while an attacker WPscan was underway (an attempt to compromise a vulnerable WordPress blog app).

- Are there any malicious circumstances that the alert(s) discussed above do not address?

A less complex (or less storage intensive app) may not tax the CPU as much during an invasion.

Also monitoring excessive HTTP errors may fail if an attacker was able to ‘guess’ a password in just a few attempts (or early on during Brute Forcing).