Cybersecurity Study Guide

Table of Contents

Python & Security Automation Quiz Review

Key Concepts:

- Automate repetitive security tasks like log analysis, ACL management, and network monitoring.
- Python helps reduce manual effort and combine tasks efficiently.
- `type()`, `print()`, `len()` are essential tools for working with strings, lists, and variable data.

Practice Tip:

Use `for` loops, `if` statements, and methods like `.split()`, `.remove()`, and `.append()` to handle access lists and clean

SIEM & SPL Quick Guide

Key Concepts:

- SIEM tools gather and normalize data from across the network for analysis.
- SPL (Search Processing Language) in Splunk uses pipes `|`, wildcards `*`, and filters like `!=` and `=`.
- Google Chronicle uses metadata tagging and UDM (unified data model) to structure logs.

Study Notes:

- Practice reading and writing simple SPL queries.
- Know how to interpret alert logs and track suspicious IPs.

Escalation-Focused Quiz Summary

Key Concepts:

- Incident escalation involves identifying, triaging, and passing critical alerts to senior analysts.
- Entry-level analysts often handle improper usage, suspicious logins, and unauthorized software.

Key Notes:

- PII-related incidents = high urgency.
- Always escalate both policy violations and malware detection.
- Know your escalation policy and roles/responsibilities.

Lab Summary: Suricata IDS Lab

Overview:

You worked with Suricata to create custom detection rules and monitor network traffic.

Code Snippet:

```
alert http any any -> any any (msg:"Suspicious HTTP Traffic"; sid:1000001;)
```

Takeaways:

- Understand rule syntax and log file differences.
- Practice detection logic.

Lab Summary: Python Log File Filtering

Overview:

Used Python to open, read, and analyze a log file for failed login attempts.

Code Snippet:

```
with open("logs.txt", "r") as file:
for line in file:
if "failed" in line:
print("Failed login:", line)
```

Takeaways:

- Use conditionals and automation for log review.

Lab Summary: USB Attack Vector Scenario

Overview:

Analyzed risks from USB baiting attacks.

Key Takeaways:

- Never use unknown USB devices.

- Understand physical attack surfaces and response protocols.

Lab Summary: SIEM & SPL Search Activities

Overview:

Used Splunk to write SPL queries and investigate logs.

Sample SPL:

```
index=security sourcetype=linux_secure | stats count by src_ip
```

Takeaways:

- Master common query formats and visualize log anomalies.

Lab Summary: Python Debugging Practice

Overview:

Debugged Python scripts by identifying and resolving syntax and logic errors.

Example Fix:

Before:

```
for item in failed_login:
print(item)
```

After:

```
for item in failed_login:
print(item)
```

Takeaways:

- Practice clean code formatting and use of print tracing.

Cybersecurity Study Guide - Quiz Section Summary

Security Mindset & Analyst Awareness

- Recognize and evaluate potential system breaches

- Understand data types to protect: financials, credentials, trade secrets

- Stay updated on vulnerabilities

- Report suspicious activity immediately

- Escalate high-impact events like malware or unauthorized access

Incident Types & Response

- Malware infection: Malicious software disrupts systems

- Improper usage: Violation of acceptable use policies

- Unauthorized access: Gaining system access without permission

- Escalation: All high-impact or sensitive events must be reported

Escalation & Reporting

- Use the organization's escalation policy

- Always escalate: PII breaches, policy violations, repeated suspicious behavior

- Entry-level analysts must know when to involve supervisors

Python for Automation & Log Analysis

- Automate log reviews, ACL updates, and alert notifications

- Tools: `with open()`, `.split()`, `if`, `for`, `print()`

- Use Python to filter failed login attempts, parse timestamps

Regular Expressions (Regex)

- `\w` matches alphanumeric characters

- `+` matches one or more

- Use `re.findall()` to scan logs for patterns

- Useful for email validation, ID detection, IP filtering

SIEM & SPL Querying

- SPL syntax: `index=`, `sourcetype=`, `| stats count by`

- Used in Splunk to analyze and visualize log patterns

- Track failed logins, anomalous IP activity, login spikes

File Handling & Parsing

- Use `with open()` to read logs safely

- `.read()` retrieves content, `.split()` parses into lists

- Filter log entries with `if 'failed' in line:`

Debugging & Error Types

- Syntax errors: Break code before running (e.g., missing colon)

- Logic errors: Code runs but output is wrong

- Exceptions: Crashes at runtime (e.g., IndexError)

- Use print statements to trace variable flow