25 Use Cases

SPLUNK SIEM

for PCI DSS Compliance

Rajneesh Gupta
@rajneeshcyber

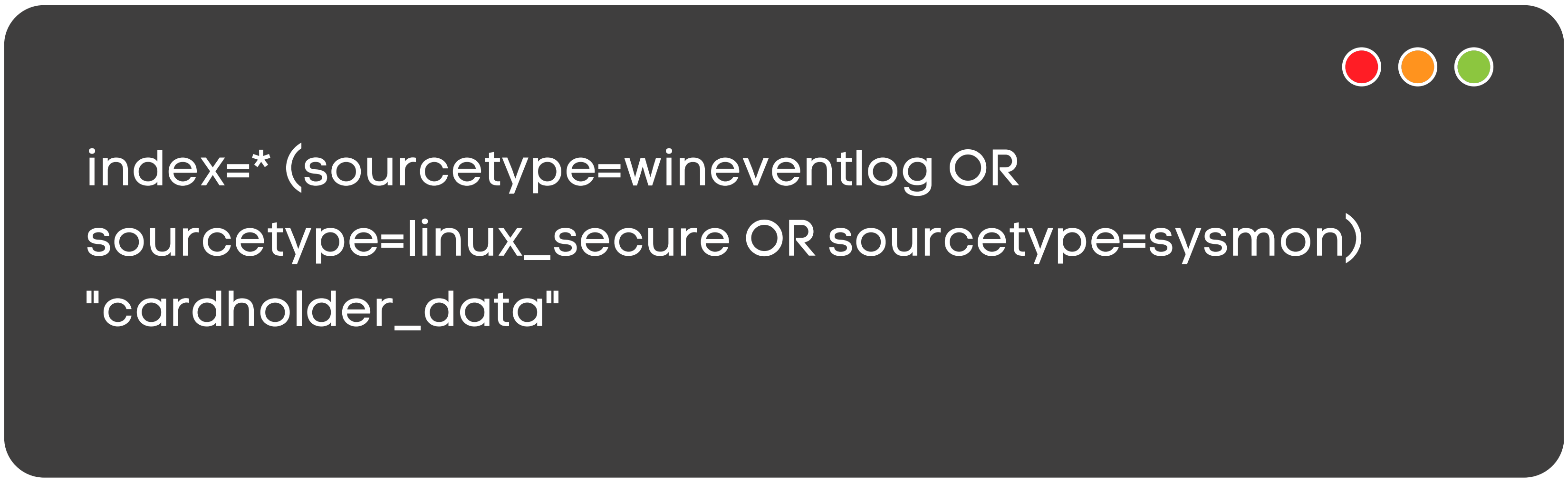
1

CARDHOLDER DATA ACCESS MONITORING

Purpose

Track access to cardholder data as required by PCI DSS v4.0 3.1

Example query



```
index=* (sourcetype=wineventlog OR
sourcetype=linux_secure OR sourcetype=sysmon)
"cardholder_data"
```

Outcome

Detect which users accessed cardholder data and when. Helps ensure that only authorized individuals have access.

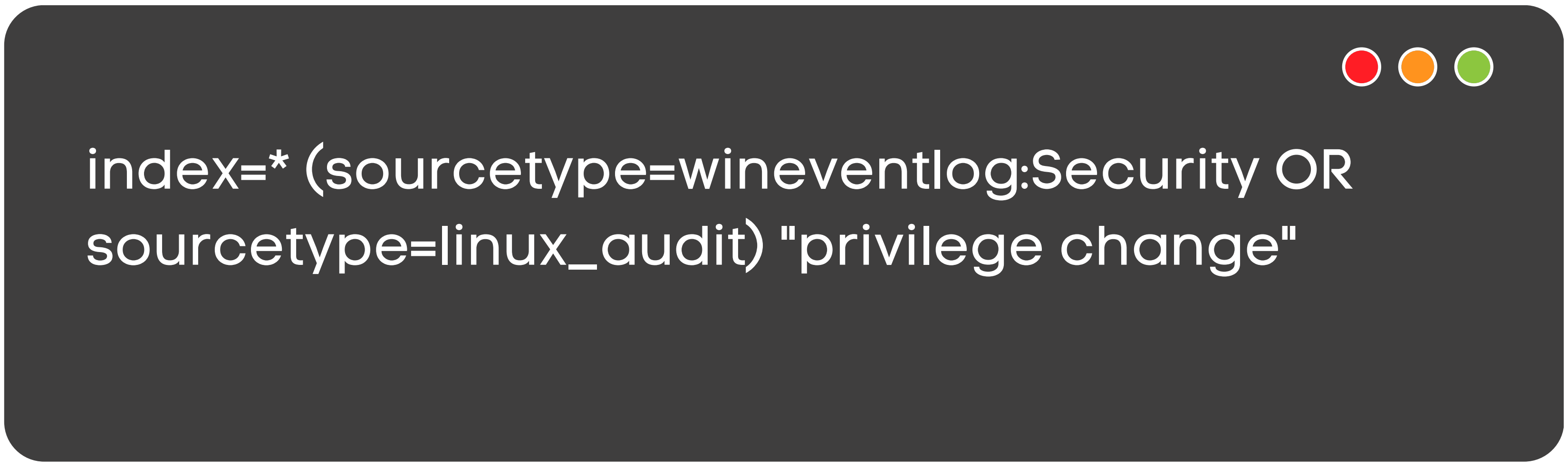
2

USER ACCESS PRIVILEGE AUDITS

Purpose

Audit changes to user access privileges as mandated by PCI DSS v4.0 7.1.2

Example query



```
index=* (sourcetype=wineventlog:Security OR
sourcetype=linux_audit) "privilege change"
```

Outcome

Identify unusual changes to user privileges. This helps ensure users don’t have unnecessary access to sensitive data.

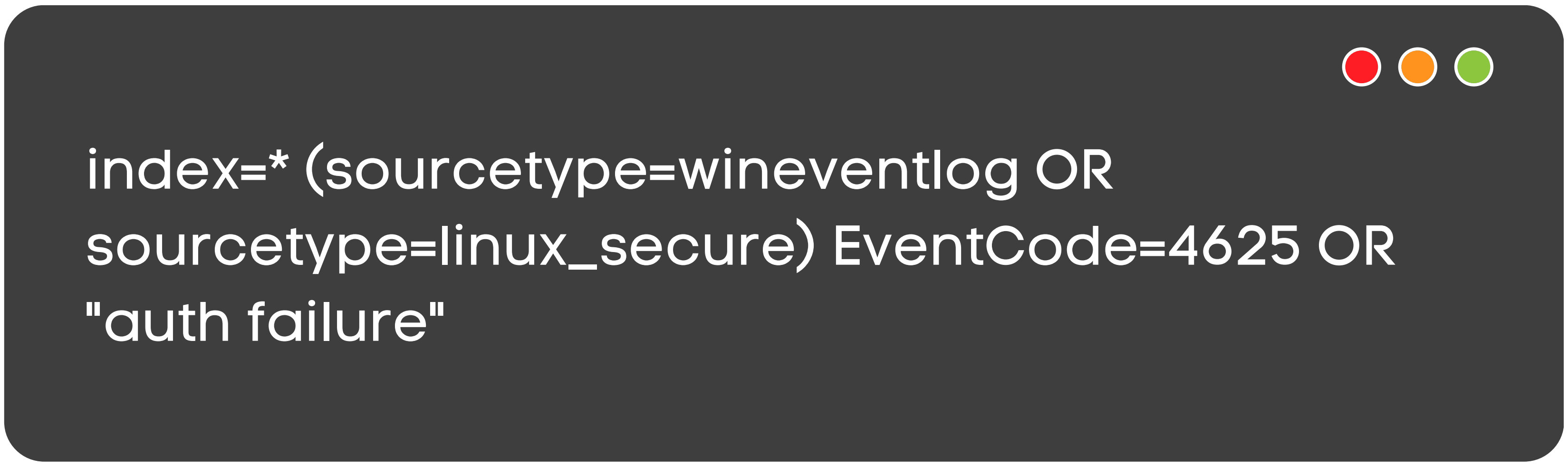
3

FAILED LOGIN ATTEMPTS

Purpose

Detect failed login attempts that may indicate brute-force attacks, as per PCI DSS v4.0 8.3.5

Example query



```
index=* (sourcetype=wineventlog OR
sourcetype=linux_secure) EventCode=4625 OR
"auth failure"
```

Outcome

Identify multiple failed login attempts to prevent unauthorized access. Helps detect potential brute-force attacks.

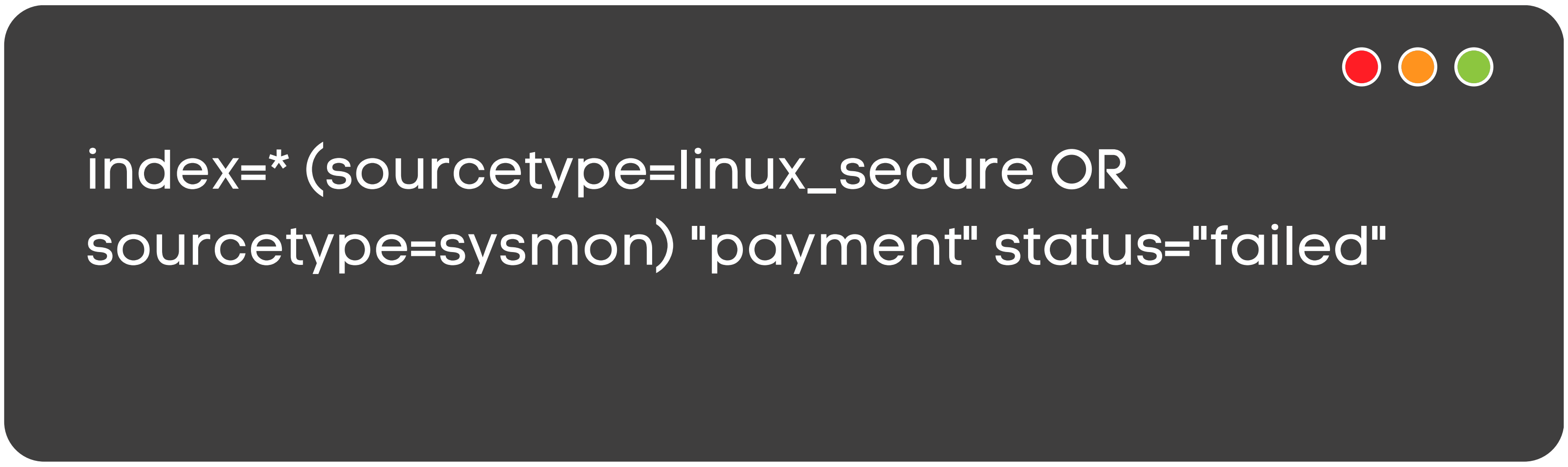
4

UNSUCCESSFUL PAYMENT TRANSACTIONS

Purpose

Track unsuccessful payment transactions as required by PCI DSS v4.0 3.2

Example query



```
index=* (sourcetype=linux_secure OR
sourcetype=sysmon) "payment" status="failed"
```

Outcome

Identify failed transactions and potential fraud indicators. Helps mitigate risks related to transaction failures.

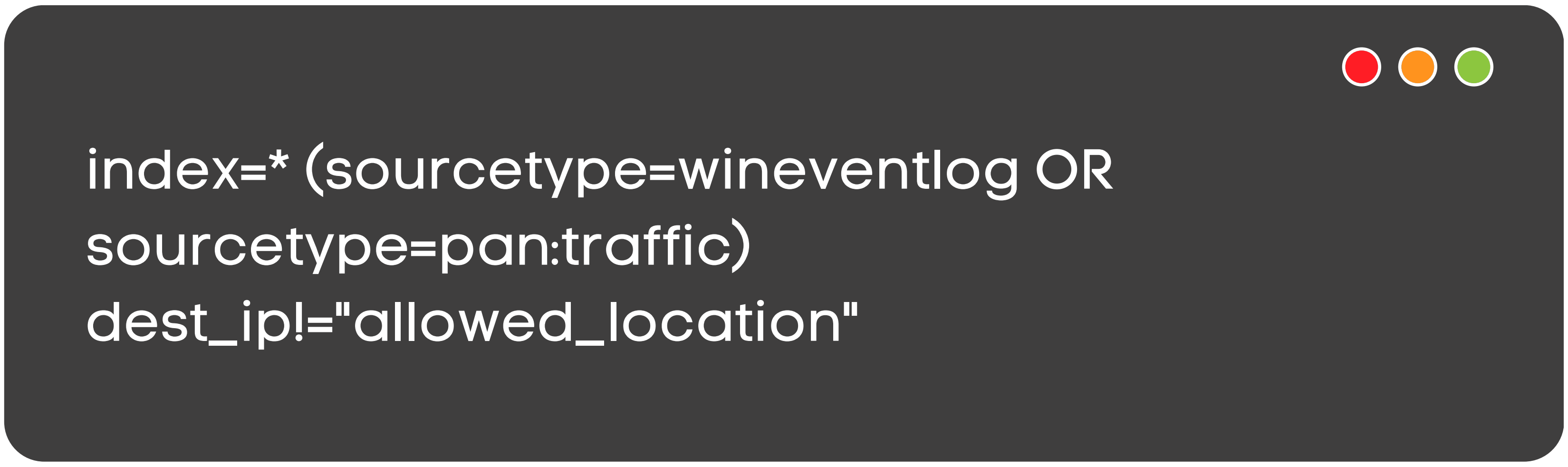
5

ACCESS FROM UNUSUAL LOCATIONS

Purpose

Monitor access to systems from unexpected geographic locations in line with PCI DSS v4.0 8.6

Example query



```
index=* (sourcetype=wineventlog OR
sourcetype=pan:traffic)
dest_ip!="allowed_location"
```

Outcome

Detect unauthorized access attempts from unfamiliar geographic locations. Helps prevent security breaches from foreign access.

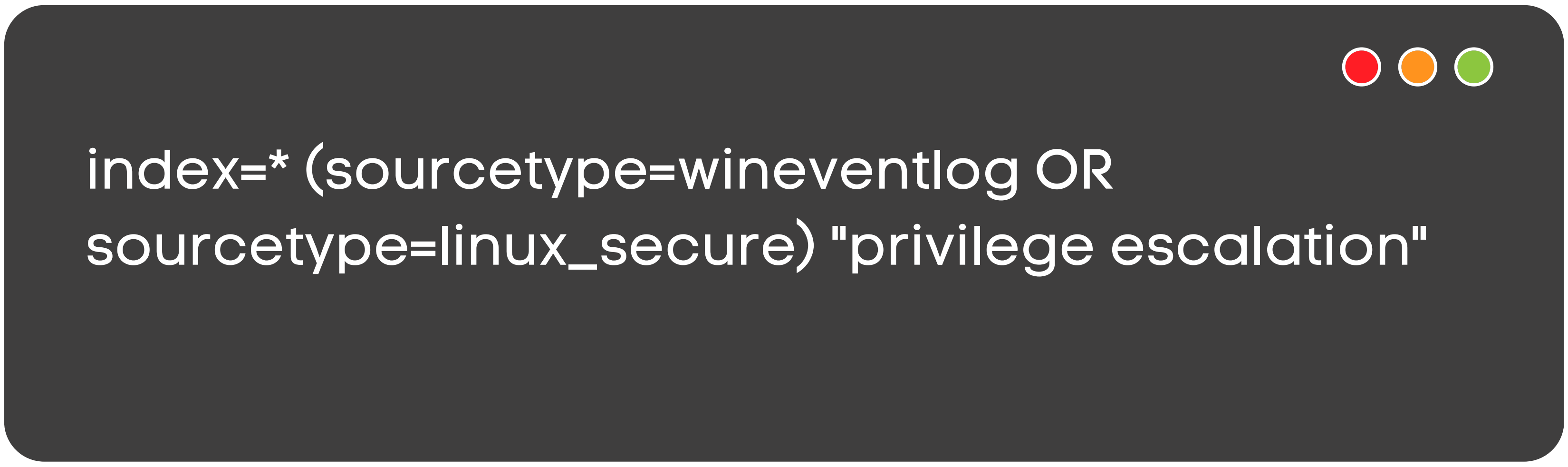
6

MONITORING USER ACTIVITY AFTER PRIVILEGE ESCALATION

Purpose

Track user activity after privilege escalation as per PCI DSS v4.0 7.2

Example query



```
index=* (sourcetype=wineventlog OR
sourcetype=linux_secure) "privilege escalation"
```

Outcome

Ensure that users with elevated privileges are not engaging in malicious activities. Helps ensure proper oversight of privileged users.

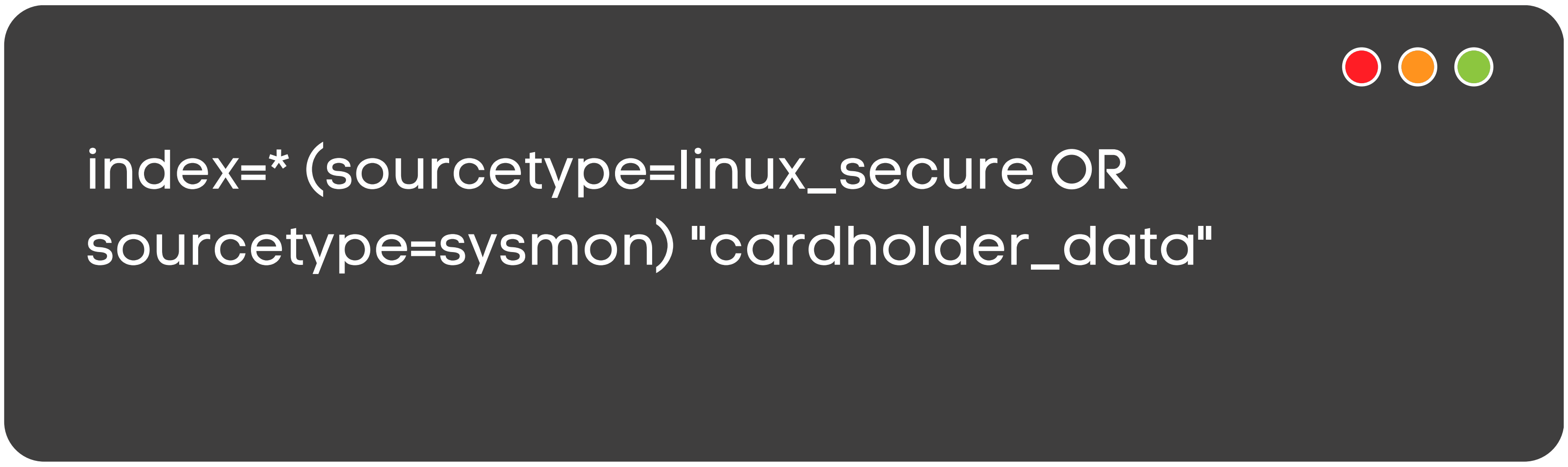
7

DATABASE ACCESS LOGGING

Purpose

Track access to PCI-relevant databases as mandated by PCI DSS v4.0 3.5

Example query



```
index=* (sourcetype=linux_secure OR
sourcetype=sysmon) "cardholder_data"
```

Outcome

Logs access to databases containing cardholder information. Helps identify potential unauthorized database access.

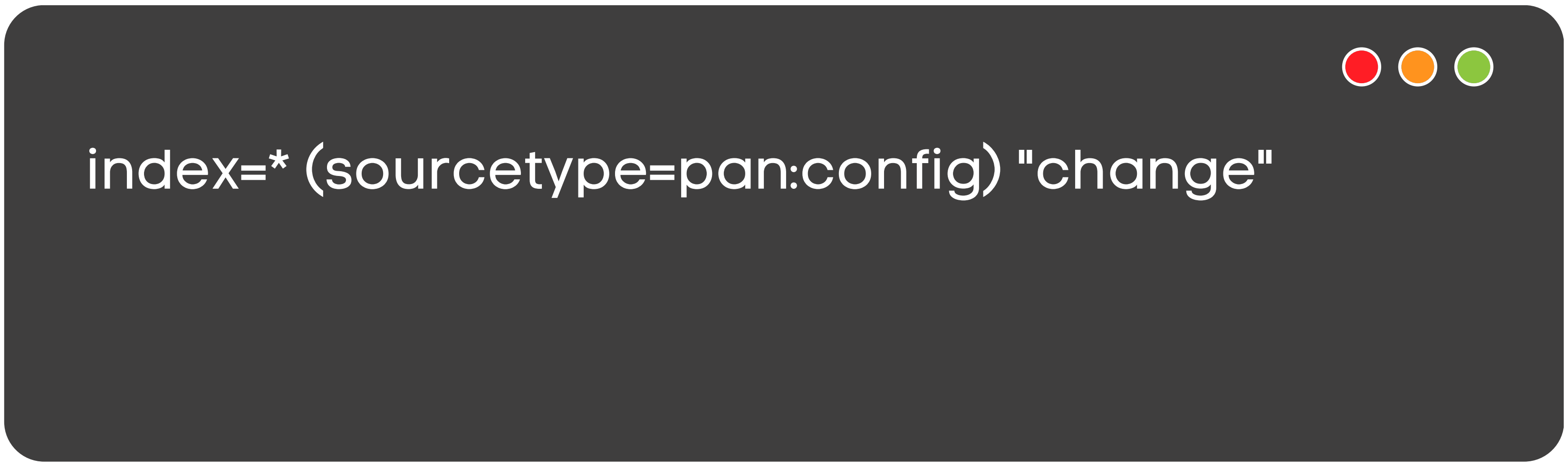
8

FIREWALL CONFIGURATION CHANGES

Purpose

Detect unauthorized changes to firewall configurations as per PCI DSS v4.0 1.1.4

Example query



```
index=* (sourcetype=pan:config) "change"
```

Outcome

Detect configuration changes that could weaken the firewall's defenses. Helps protect against unauthorized access or data leakage.

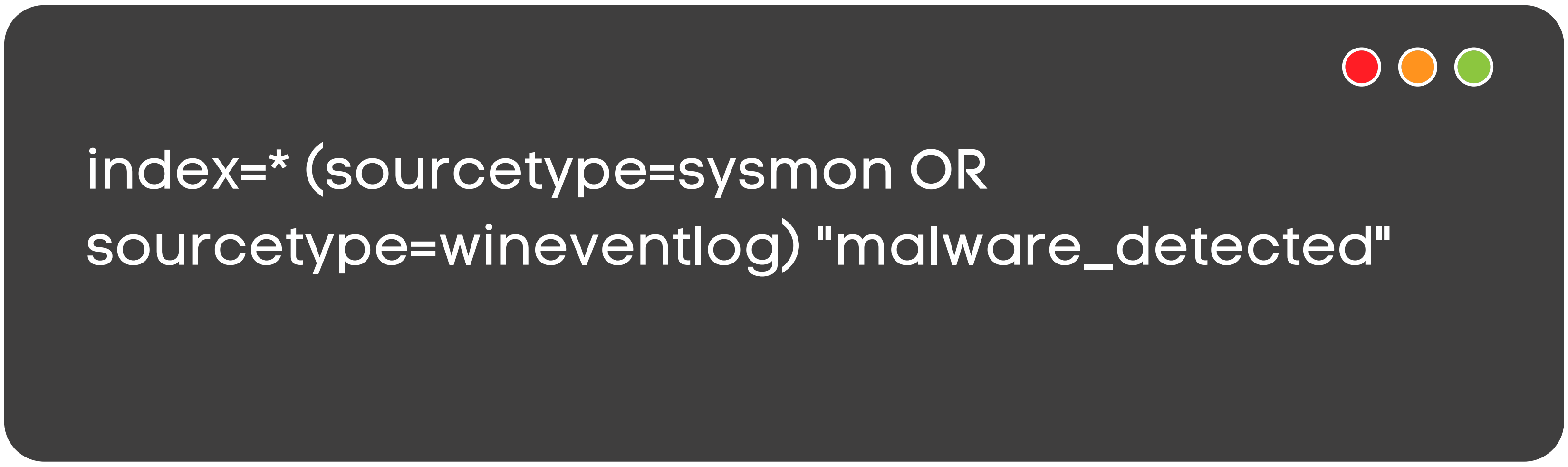
9

MALWARE DETECTION AND BLOCKING

Purpose

Monitor for malware detections in PCI environments, as required by PCI DSS v4.0 5.3.3.

Example query



```
index=* (sourcetype=sysmon OR
sourcetype=wineventlog) "malware_detected"
```

Outcome

Identify malware threats and block them immediately. Helps prevent data breaches caused by malicious software.

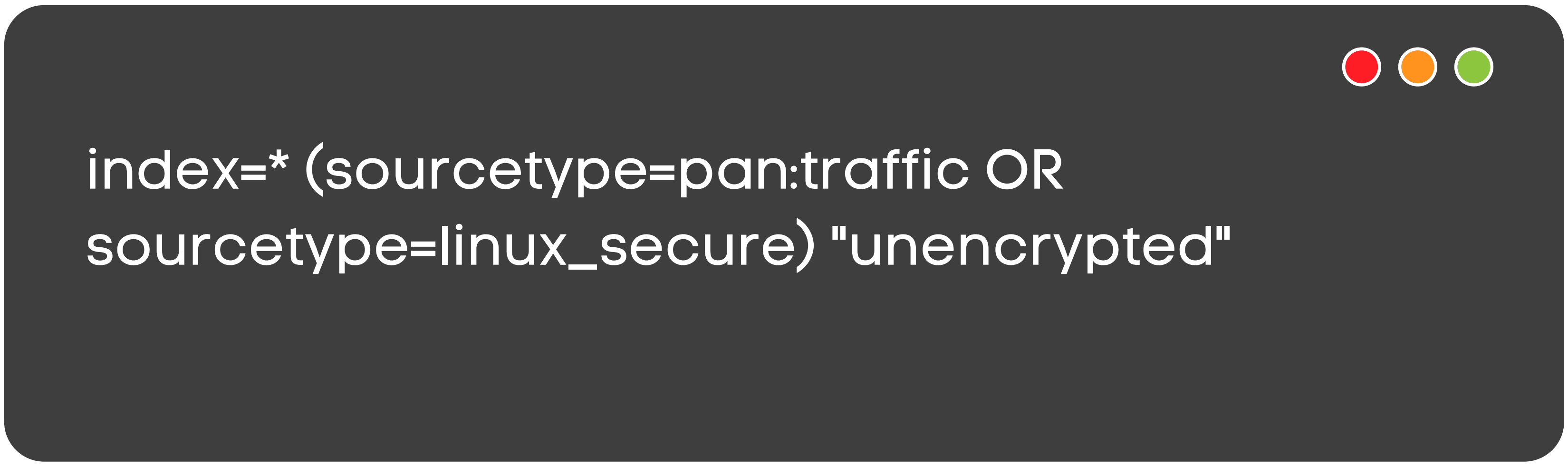
10

UNENCRYPTED CARDHOLDER DATA TRANSMISSION

Purpose

Monitor for unencrypted cardholder data being transmitted as required by PCI DSS v4.0 4.1

Example query



```
index=* (sourcetype=pan:traffic OR
sourcetype=linux_secure) "unencrypted"
```

Outcome

Detect unencrypted transmission of cardholder data. Ensures compliance by identifying risky transmissions.

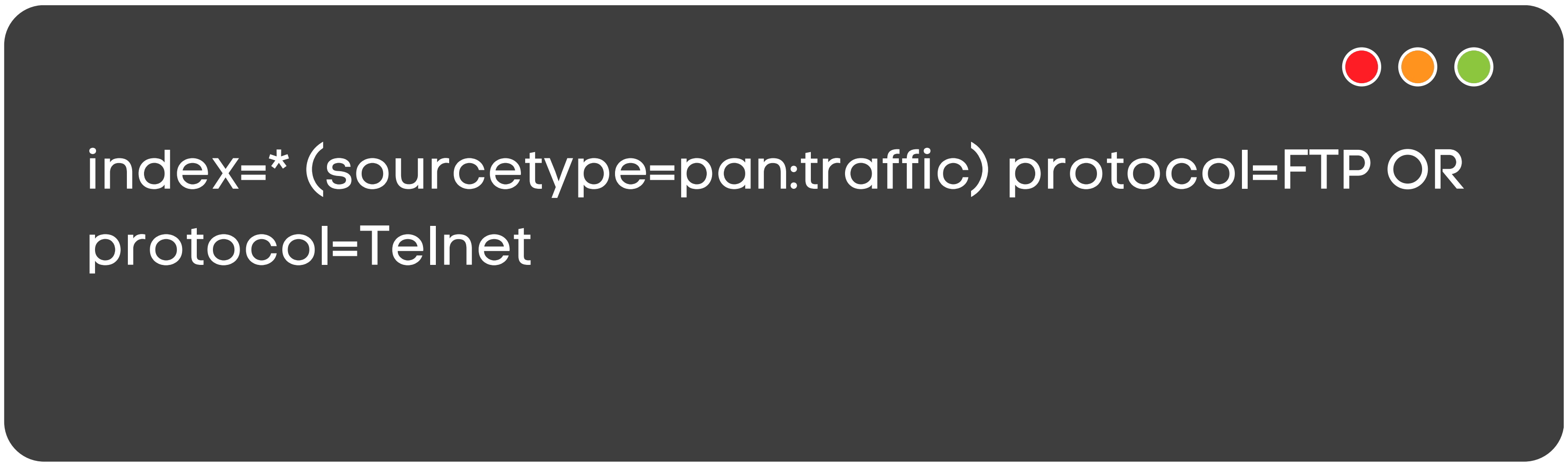
11

INSECURE PROTOCOL USE (FTP, TELNET)

Purpose

Detect the use of insecure protocols as required by PCI DSS v4.0 4.2

Example query



```
index=* (sourcetype=pan:traffic) protocol=FTP OR
protocol=Telnet
```

Outcome

Detects use of insecure communication protocols. Helps ensure encrypted channels are used for sensitive data transmission.

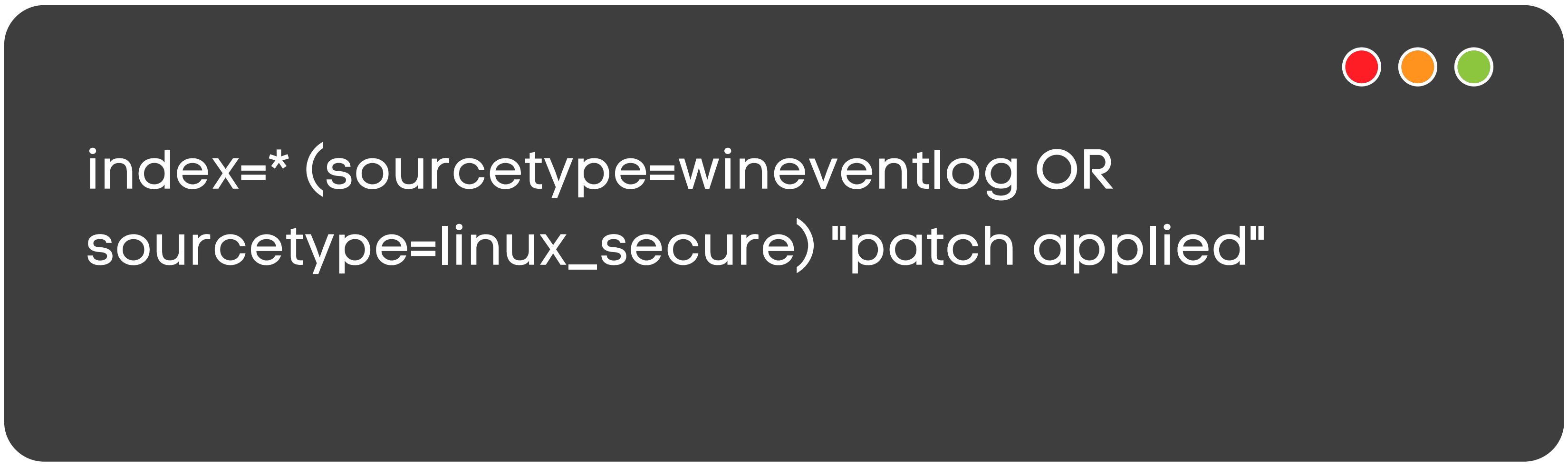
12

SECURITY PATCH APPLICATION MONITORING

Purpose

Monitor the application of security patches as per PCI DSS v4.0 6.3

Example query



```
index=* (sourcetype=wineventlog OR
sourcetype=linux_secure) "patch applied"
```

Outcome

Verifies that security patches are applied promptly. Ensures the systems remain secure and updated.

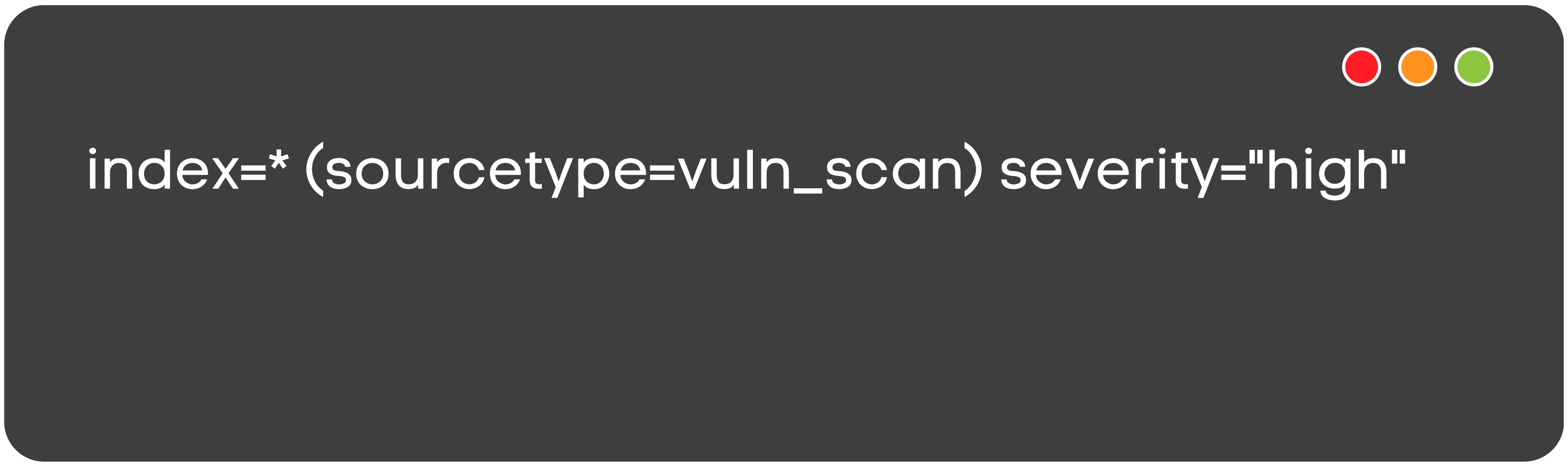
13

VULNERABILITY SCANNING ALERTS

Purpose

Monitor and report vulnerability scanning results as required by PCI DSS v4.0 11.3.4

Example query



```
index=* (sourcetype=vuln_scan) severity="high"
```

Outcome

Identifies high-risk vulnerabilities. Helps prioritize patching and remediation efforts to prevent potential exploits.

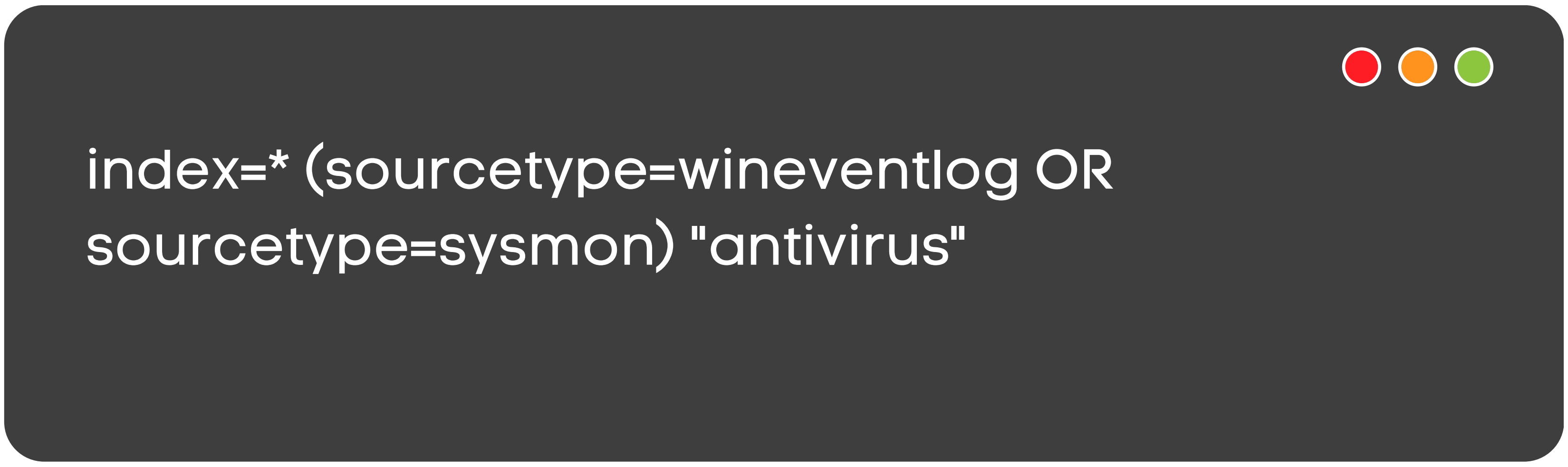
14

ANTIVIRUS MONITORING

Purpose

Track antivirus alerts and actions in line with PCI DSS v4.0 5.4.

Example query



```
index=* (sourcetype=wineventlog OR
sourcetype=sysmon) "antivirus"
```

Outcome

Detects and logs antivirus activity such as quarantining. Helps monitor real-time detection and response to threats.

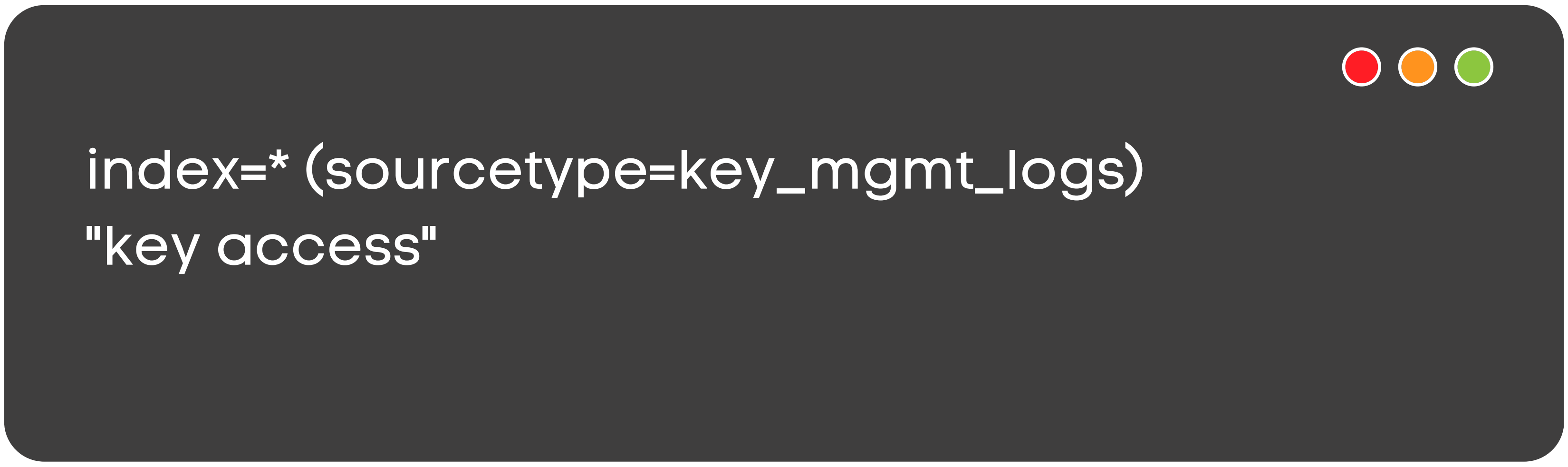
15

ENCRYPTION KEY ACCESS MONITORING

Purpose

Monitor access to encryption key management systems as per PCI DSS v4.0 3.5.1

Example query



```
index=* (sourcetype=key_mgmt_logs)
"key access"
```

Outcome

Logs and monitors access to encryption keys. Ensures only authorized users can handle sensitive encryption materials.

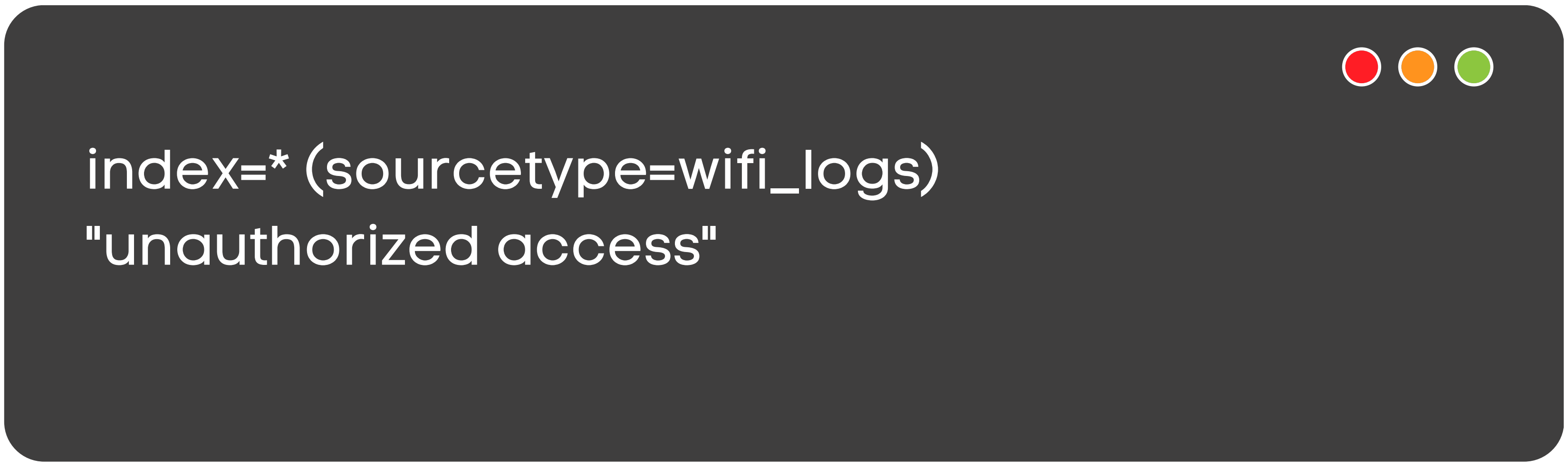
16

WIRELESS NETWORK SECURITY

Purpose

Track security on wireless networks in PCI environments, aligned with PCI DSS v4.0 4.2.

Example query



```
index=* (sourcetype=wifi_logs)
"unauthorized access"
```

Outcome

Detects unauthorized access to wireless networks. Helps secure the transmission of sensitive cardholder data over wireless connections.

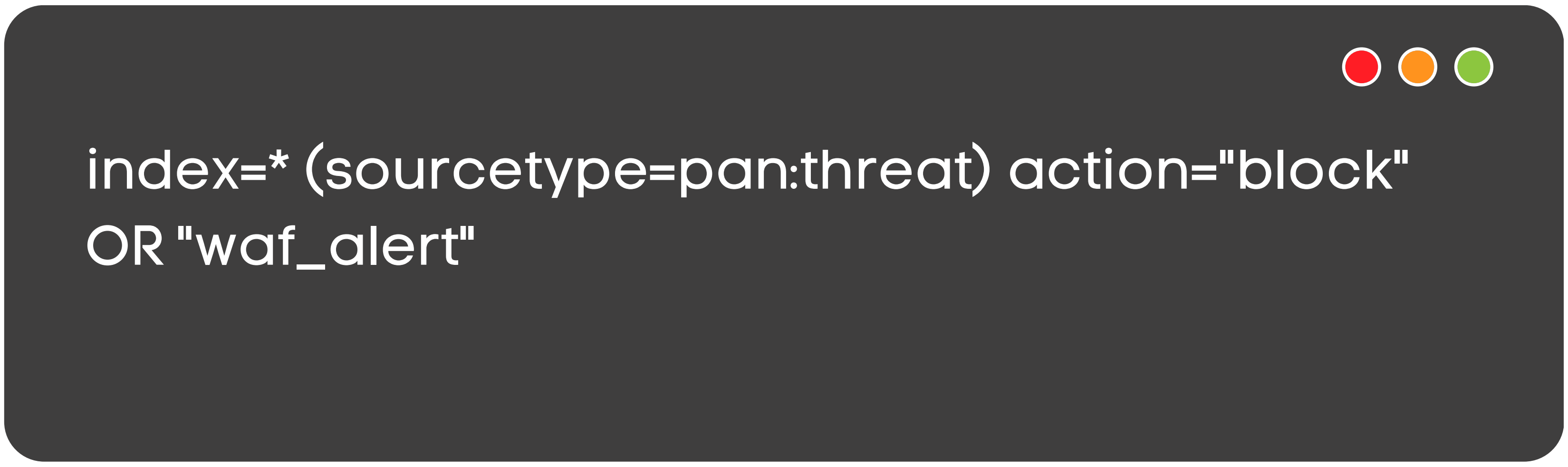
17

WEB APPLICATION FIREWALL (WAF) ALERTS

Purpose

Monitor WAF alerts for PCI-relevant websites, as required by PCI DSS v4.0 6.6

Example query



```
index=* (sourcetype=pan:threat) action="block"
OR "waf_alert"
```

Outcome

Tracks WAF block/allow events. Ensures web-based applications are protected from threats like SQL injection or cross-site scripting.

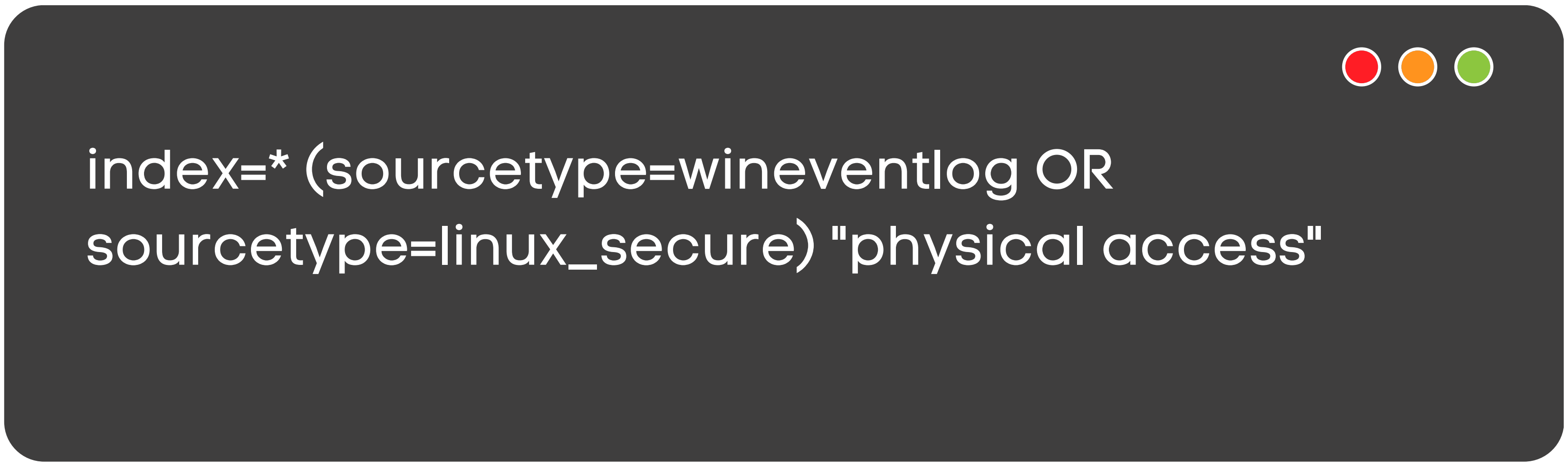
18

PHYSICAL SECURITY ACCESS LOGS

Purpose

Monitor physical access to PCI DSS environments as per PCI DSS v4.0 9.1

Example query



```
index=* (sourcetype=wineventlog OR
sourcetype=linux_secure) "physical access"
```

Outcome

Tracks physical access to secure areas housing PCI systems. Helps ensure only authorized personnel have access.

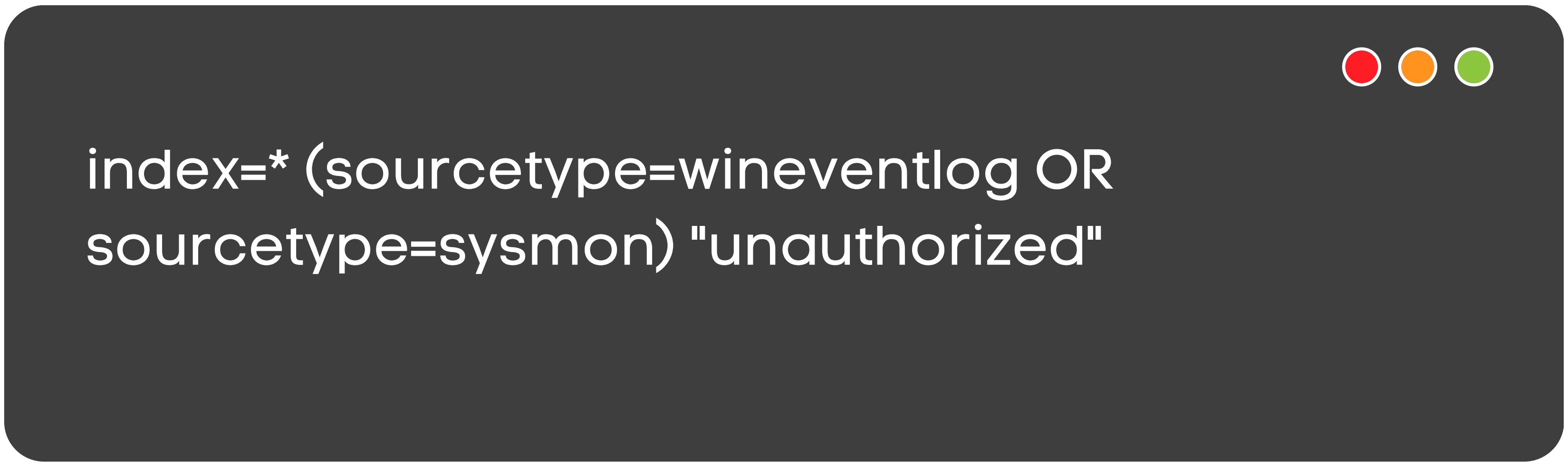
19

UNAUTHORIZED ACCESS ATTEMPTS DETECTION

Purpose

Detect attempts to access restricted PCI systems, aligned with PCI DSS v4.0 8.3

Example query



```
index=* (sourcetype=wineventlog OR
sourcetype=sysmon) "unauthorized"
```

Outcome

Detects unauthorized attempts to access PCI systems. Helps prevent potential breaches by flagging these access attempts.

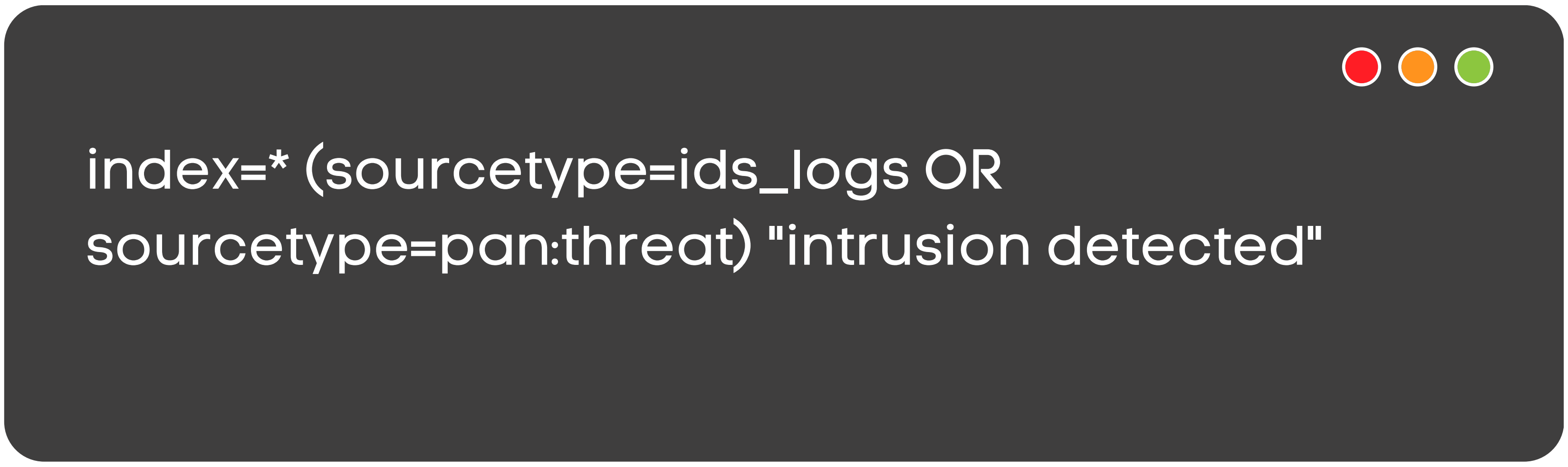
20

INTRUSION DETECTION SYSTEM (IDS) ALERTS

Purpose

Monitor IDS alerts for PCI systems as per PCI DSS v4.0 11.4

Example query



```
index=* (sourcetype=ids_logs OR
sourcetype=pan:threat) "intrusion detected"
```

Outcome

Identifies and responds to suspicious network traffic. Helps detect potential intrusions or attacks in real-time.

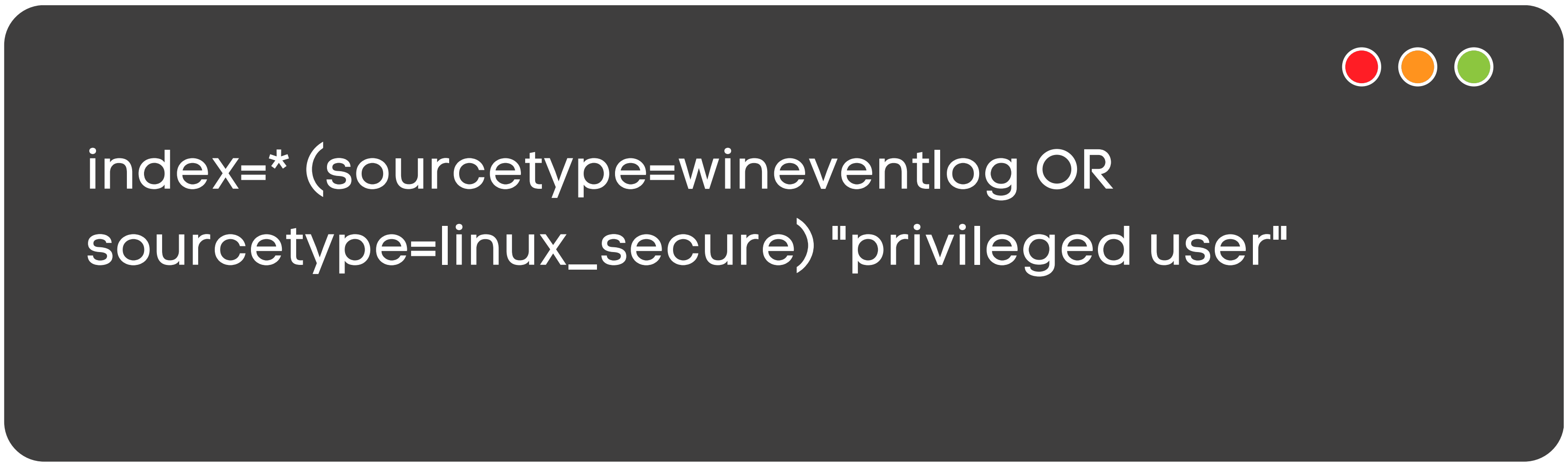
21

PRIVILEGED USER MONITORING

Purpose

Track activities of privileged users as required by PCI DSS v4.0 7.2.1

Example query



```
index=* (sourcetype=wineventlog OR
sourcetype=linux_secure) "privileged user"
```

Outcome

Monitors actions taken by privileged users.
Ensures elevated access is not being misused.

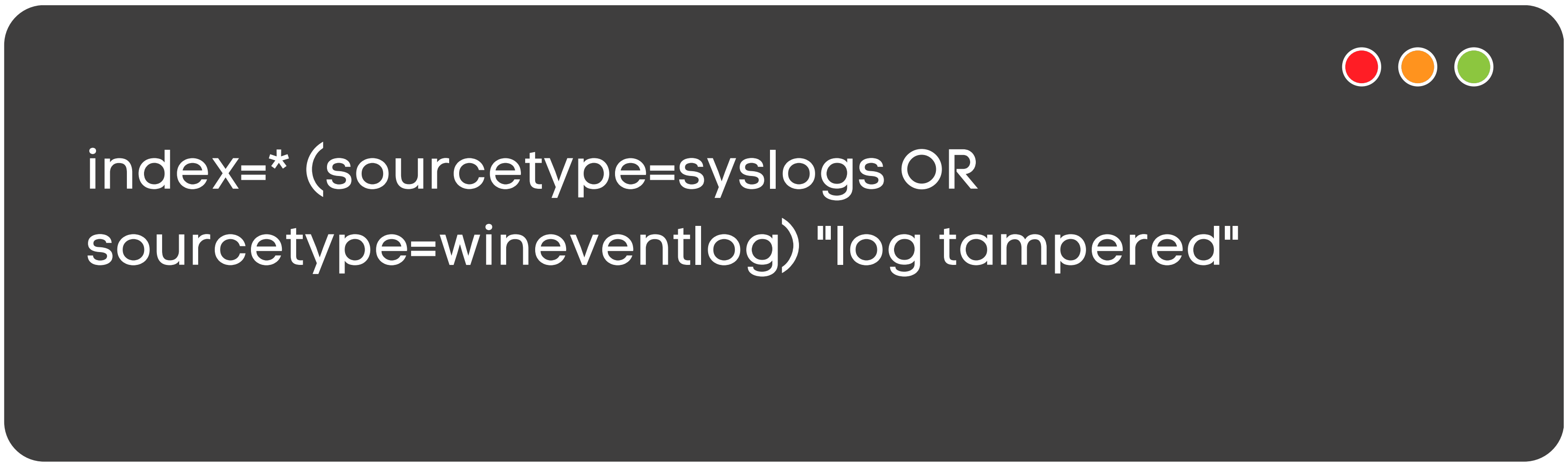
22

LOG INTEGRITY MONITORING

Purpose

Ensure the integrity of PCI-relevant logs as per PCI DSS v4.0 10.5.

Example query



```
index=* (sourcetype=syslogs OR
sourcetype=wineventlog) "log tampered"
```

Outcome

Detects tampering or alteration of logs. Ensures data integrity for forensic investigations and compliance.

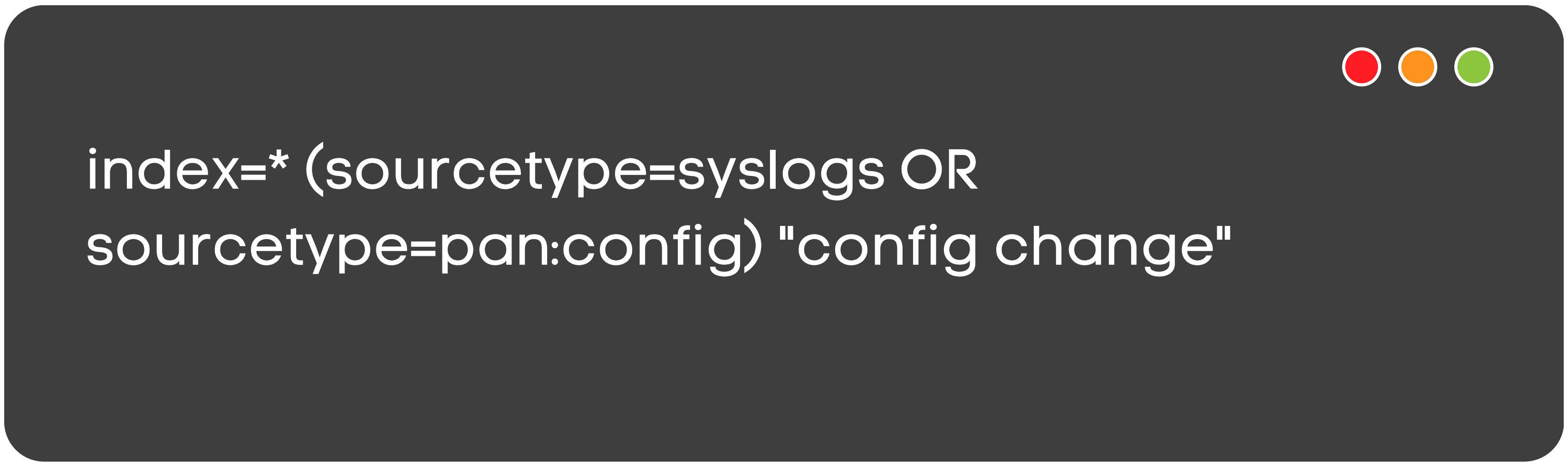
23

PCI SYSTEM CONFIGURATION CHANGES

Purpose

Track changes to system configurations in PCI environments as per PCI DSS v4.0 2.2.4.

Example query



```
index=* (sourcetype=syslogs OR
sourcetype=pan:config) "config change"
```

Outcome

Logs all system configuration changes. Helps ensure that unauthorized changes are identified and reverted.

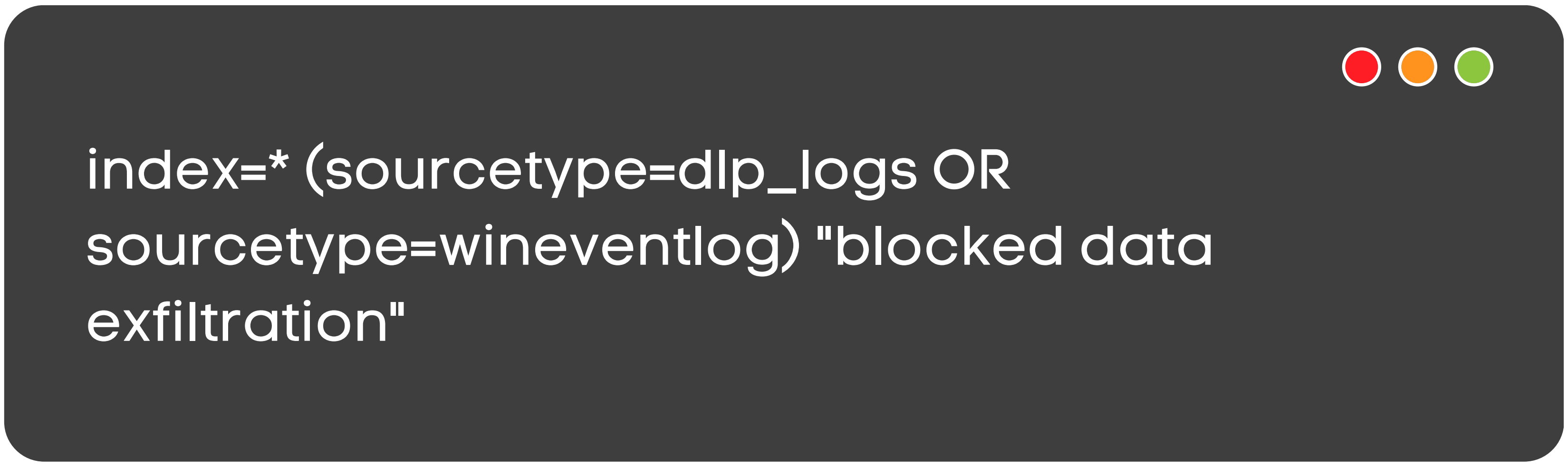
24

DATA LOSS PREVENTION (DLP) ALERTS

Purpose

Monitor DLP alerts related to cardholder data, as required by PCI DSS v4.0 9.9.3

Example query



```
index=* (sourcetype=dlp_logs OR
sourcetype=wineventlog) "blocked data
exfiltration"
```

Outcome

Detects and blocks attempts to exfiltrate cardholder data. Ensures sensitive data is not leaked outside the secure environment.

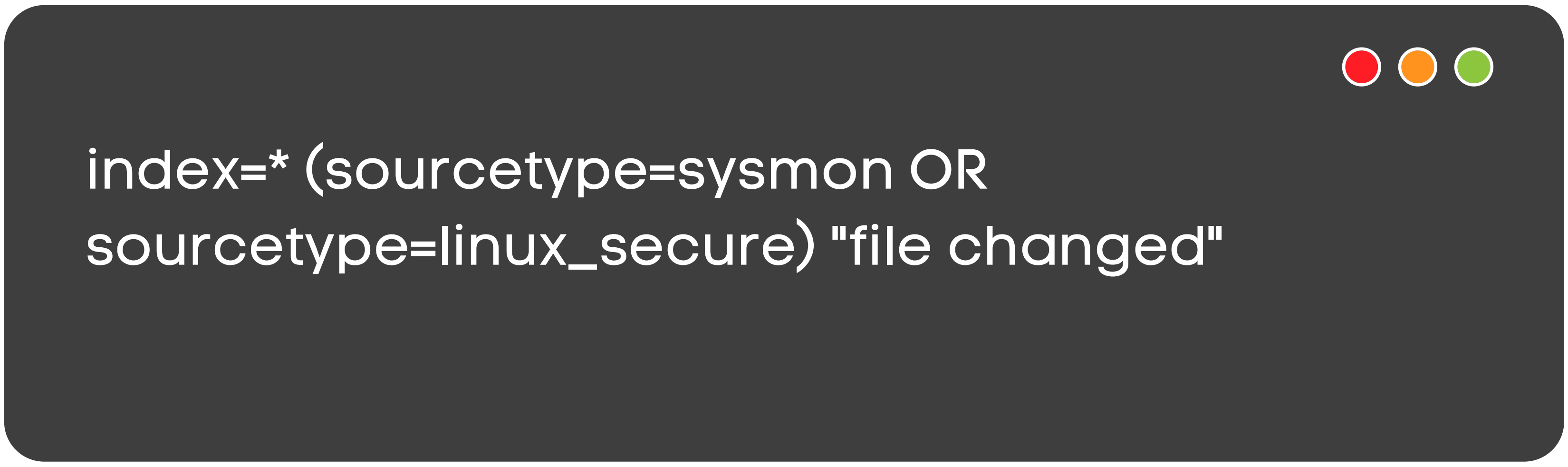
25

FILE INTEGRITY MONITORING

Purpose

Monitor file integrity in PCI environments as per PCI DSS v4.0 11.5

Example query



```
index=* (sourcetype=sysmon OR
sourcetype=linux_secure) "file changed"
```

Outcome

Detects unauthorized modifications to critical files. Helps ensure that PCI systems are not compromised by attackers.

CONCLUSION

- Comprehensive PCI DSS monitoring across key areas.
- Supports diverse log sources like Windows, Linux, and Palo Alto.
- Enables proactive detection of threats and incidents.
- Ensures data integrity with real-time alerts.
- Helps maintain security and compliance at scale.
- Adaptable for organizations of any size.

Reach us at hi@haxsecurity.com