~~~
Propose 5 cyber security alerts with for each of the systems. provide 1. alert description 2. alert logic (i.e. 5 detections in 1 minute, etc) 3. example log lines with detection 4. logfile location. Systems:

- Windows OS (security event logs)
- Linux OS (auditd raw logs)
- nginx access logs
- apache access logs
- IIS access logs
- bigip F5 ASM detection logs
- DAM Ipmerva logs


- Keycloak APPSSO logs
- chceckpoint firewall logs
- jboss logs
- tomcat logs


- alerts based on SELECT statements output - propose SQL SELECT statements to obtain security relevant data, i.e. from V$ views or other on Oracle DB
- alerts based on SELECT statements output - propose SQL SELECT statements to obtain security relevant data, i.e. fromsystem or other tables or views on MSSQL
- alerts based on SELECT statements output - propose SQL SELECT statements to obtain security relevant data, i.e. fromsystem or other tables or views on Mysql DB

==========

Sure, here are the proposed cyber security alerts for each of the systems:

### Windows OS (Security Event Logs)

1. **Alert Description**: Multiple failed login attempts
   - **Alert Logic**: 5 failed login attempts within 1 minute
   - **Example Log Lines**:
     ```
     EventID: 4625, LogonType: 2, AccountName: user1, FailureReason: Unknown user name or bad password
~~~

~~~~
     EventID: 4625, LogonType: 2, AccountName: user1, FailureReason: Unknown user name or bad password
     EventID: 4625, LogonType: 2, AccountName: user1, FailureReason: Unknown user name or bad password
     EventID: 4625, LogonType: 2, AccountName: user1, FailureReason: Unknown user name or bad password
     EventID: 4625, LogonType: 2, AccountName: user1, FailureReason: Unknown user name or bad password
     ```
   - **Logfile Location**: `C:\Windows\System32\winevt\Logs\Security.evtx`

2. **Alert Description**: Unauthorized access to sensitive files
   - **Alert Logic**: Access to a sensitive file by an unauthorized user
   - **Example Log Lines**:
     ```
     EventID: 4663, ObjectName: C:\SensitiveData\confidential.txt, AccessMask: 0x2, AccountName: unauthorized_user
     ```
   - **Logfile Location**: `C:\Windows\System32\winevt\Logs\Security.evtx`

3. **Alert Description**: New user account creation
   - **Alert Logic**: Creation of a new user account
   - **Example Log Lines**:
     ```
     EventID: 4720, AccountName: new_user, AccountDomain: DOMAIN
     ```
   - **Logfile Location**: `C:\Windows\System32\winevt\Logs\Security.evtx`

4. **Alert Description**: User added to administrators group
   - **Alert Logic**: Addition of a user to the administrators group
   - **Example Log Lines**:
     ```
     EventID: 4732, GroupName: Administrators, MemberName: user1
     ```
   - **Logfile Location**: `C:\Windows\System32\winevt\Logs\Security.evtx`

5. **Alert Description**: Service installation
   - **Alert Logic**: Installation of a new service
   - **Example Log Lines**:
     ```
~~~~

```
     EventID: 7045, ServiceName: NewService, ServiceFileName: C:\Program Files\NewService\service.exe
     ```
   - **Logfile Location**: `C:\Windows\System32\winevt\Logs\System.evtx`

### Linux OS (auditd Raw Logs)

1. **Alert Description**: Multiple failed login attempts
   - **Alert Logic**: 5 failed login attempts within 1 minute
   - **Example Log Lines**:
     ```
     type=USER_LOGIN msg=audit(1622542800.123:1234): user pid=1234 uid=0 auid=1000 ses=1 msg='op=login id=1000 exe="/usr/bin/login" hostname=? addr=? terminal=tty1 res=failed'
     ```
   - **Logfile Location**: `/var/log/audit/audit.log`

2. **Alert Description**: Unauthorized file access
   - **Alert Logic**: Access to a sensitive file by an unauthorized user
   - **Example Log Lines**:
     ```
     type=AVC msg=audit(1622542800.123:1234): avc:  denied  { read } for pid=1234 comm="cat" name="confidential.txt" dev="sda1" ino=12345 scontext=unconfined_u:unconfined_r:unconfined_t:s0 tcontext=system_u:object_r:admin_home_t:s0 tclass=file
     ```
   - **Logfile Location**: `/var/log/audit/audit.log`

3. **Alert Description**: New user account creation
   - **Alert Logic**: Creation of a new user account
   - **Example Log Lines**:
     ```
     type=USER_ACCT msg=audit(1622542800.123:1234): pid=1234 uid=0 auid=0 ses=1 msg='op=add_user id=1001 exe="/usr/sbin/useradd" hostname=? addr=? terminal=tty1 res=success'
     ```
   - **Logfile Location**: `/var/log/audit/audit.log`

4. **Alert Description**: User added to sudoers
   - **Alert Logic**: Addition of a user to the sudoers file
```
```

~~~~markdown
   - **Example Log Lines**:
     ```
     type=USER_ROLE_CHANGE msg=audit(1622542800.123:1234): pid=1234 uid=0 auid=0 ses=1 msg='op=add_to_sudoers id=1001 exe="/usr/sbin/usermod" hostname=? addr=? terminal=tty1 res=success'
     ```
   - **Logfile Location**: `/var/log/audit/audit.log`

5. **Alert Description**: Service installation
   - **Alert Logic**: Installation of a new service
   - **Example Log Lines**:
     ```
     type=SERVICE_START msg=audit(1622542800.123:1234): pid=1234 uid=0 auid=0 ses=1 msg='op=start_service id=1001 exe="/usr/sbin/service" hostname=? addr=? terminal=tty1 res=success'
     ```
   - **Logfile Location**: `/var/log/audit/audit.log`

### nginx Access Logs

1. **Alert Description**: Multiple 404 errors
   - **Alert Logic**: 10 404 errors within 1 minute from the same IP
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "GET /nonexistentpage HTTP/1.1" 404 169 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/nginx/access.log`

2. **Alert Description**: SQL Injection attempt
   - **Alert Logic**: Detection of SQL keywords in URL
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "GET /index.php?id=1' OR '1'='1 HTTP/1.1" 200 169 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/nginx/access.log`

3. **Alert Description**: Brute force login attempts
~~~~

~~~markdown
   - **Alert Logic**: 5 failed login attempts within 1 minute from the same IP
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "POST /login HTTP/1.1" 401 169 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/nginx/access.log`

4. **Alert Description**: Access to restricted URL
   - **Alert Logic**: Access to a restricted URL
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "GET /admin HTTP/1.1" 403 169 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/nginx/access.log`

5. **Alert Description**: Large file download
   - **Alert Logic**: Download of a file larger than 100MB
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "GET /largefile.zip HTTP/1.1" 200 104857600 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/nginx/access.log`

### Apache Access Logs

1. **Alert Description**: Multiple 404 errors
   - **Alert Logic**: 10 404 errors within 1 minute from the same IP
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "GET /nonexistentpage HTTP/1.1" 404 169 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/apache2/access.log`

2. **Alert Description**: SQL Injection attempt
   - **Alert Logic**: Detection of SQL keywords in URL
~~~

```
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "GET /index.php?id=1' OR '1'='1 HTTP/1.1" 200 169 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/apache2/access.log`

3. **Alert Description**: Brute force login attempts
   - **Alert Logic**: 5 failed login attempts within 1 minute from the same IP
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "POST /login HTTP/1.1" 401 169 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/apache2/access.log`

4. **Alert Description**: Access to restricted URL
   - **Alert Logic**: Access to a restricted URL
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "GET /admin HTTP/1.1" 403 169 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/apache2/access.log`

5. **Alert Description**: Large file download
   - **Alert Logic**: Download of a file larger than 100MB
   - **Example Log Lines**:
     ```
     192.168.1.1 - - [12/Oct/2023:14:55:36 +0000] "GET /largefile.zip HTTP/1.1" 200 104857600 "-" "Mozilla/5.0"
     ```
   - **Logfile Location**: `/var/log/apache2/access.log`

### IIS Access Logs

1. **Alert Description**: Multiple 404 errors
   - **Alert Logic**: 10 404 errors within 1 minute from the same IP
   - **Example Log Lines**:
```

~~~
     ```
     2023-10-12 14:55:36 192.168.1.1 GET /nonexistentpage - 80 - 192.168.1.1 Mozilla/5.0 404 0 0 169
     ```
   - **Logfile Location**: `C:\inetpub\logs\LogFiles\W3SVC1\u_exYYMMDD.log`

2. **Alert Description**: SQL Injection attempt
   - **Alert Logic**: Detection of SQL keywords in URL
   - **Example Log Lines**:
     ```
     2023-10-12 14:55:36 192.168.1.1 GET /index.php?id=1' OR '1'='1 - 80 - 192.168.1.1 Mozilla/5.0 200 0 0 169
     ```
   - **Logfile Location**: `C:\inetpub\logs\LogFiles\W3SVC1\u_exYYMMDD.log`

3. **Alert Description**: Brute force login attempts
   - **Alert Logic**: 5 failed login attempts within 1 minute from the same IP
   - **Example Log Lines**:
     ```
     2023-10-12 14:55:36 192.168.1.1 POST /login - 80 - 192.168.1.1 Mozilla/5.0 401 0 0 169
     ```
   - **Logfile Location**: `C:\inetpub\logs\LogFiles\W3SVC1\u_exYYMMDD.log`

4. **Alert Description**: Access to restricted URL
   - **Alert Logic**: Access to a restricted URL
   - **Example Log Lines**:
     ```
     2023-10-12 14:55:36 192.168.1.1 GET /admin - 80 - 192.168.1.1 Mozilla/5.0 403 0 0 169
     ```
   - **Logfile Location**: `C:\inetpub\logs\LogFiles\W3SVC1\u_exYYMMDD.log`

5. **Alert Description**: Large file download
   - **Alert Logic**: Download of a file larger than 100MB
~~~

~~~
   - **Example Log Lines**:
     ```
     2023-10-12 14:55:36 192.168.1.1 GET /largefile.zip - 80 - 192.168.1.1
Mozilla/5.0 200 0 0 104857600
     ```
   - **Logfile Location**:
`C:\inetpub\logs\LogFiles\W3SVC1\u_exYYMMDD.log`

### BigIP F5 ASM Detection Logs

1. **Alert Description**: SQL Injection attempt
   - **Alert Logic**: Detection of SQL keywords in request
   - **Example Log Lines**:
     ```
     <134>Oct 12 14:55:36 bigip1 ASM:attack_type="SQL Injection"
src_ip="192.168.1.1" request="GET /index.php?id=1' OR '1'='1"
     ```
   - **Logfile Location**: `/var/log/asm`

2. **Alert Description**: Cross-Site Scripting (XSS) attempt
   - **Alert Logic**: Detection of XSS keywords in request
   - **Example Log Lines**:
     ```
     <134>Oct 12 14:55:36 bigip1 ASM:attack_type="Cross-Site Scripting"
src_ip="192.168.1.1" request="GET
/index.php?name=<script>alert('XSS')</script>"
     ```
   - **Logfile Location**: `/var/log/asm`

3. **Alert Description**: Brute force login attempts
   - **Alert Logic**: 5 failed login attempts within 1 minute from the
same IP
   - **Example Log Lines**:
     ```
     <134>Oct 12 14:55:36 bigip1 ASM:attack_type="Brute Force"
src_ip="192.168.1.1" request="POST /login"
     ```
   - **Logfile Location**: `/var/log/asm`

4. **Alert Description**: Access to restricted URL
~~~

~~~
   - **Alert Logic**: Access to a restricted URL
   - **Example Log Lines**:
     ```
     <134>Oct 12 14:55:36 bigip1 ASM:attack_type="Access Violation" src_ip="192.168.1.1" request="GET /admin"
     ```
   - **Logfile Location**: `/var/log/asm`

5. **Alert Description**: Large file download
   - **Alert Logic**: Download of a file larger than 100MB
   - **Example Log Lines**:
     ```
     <134>Oct 12 14:55:36 bigip1 ASM:attack_type="Large File Download" src_ip="192.168.1.1" request="GET /largefile.zip"
     ```
   - **Logfile Location**: `/var/log/asm`

### DAM Imperva Logs

1. **Alert Description**: Unauthorized database access
   - **Alert Logic**: Access to a database by an unauthorized user
   - **Example Log Lines**:
     ```
     <134>Oct 12 14:55:36 imperva1 DB:attack_type="Unauthorized Access" src_ip="192.168.1.1" user="unauthorized_user" db="sensitive_db"
     ```
   - **Logfile Location**: `/var/log/imperva`

2. **Alert Description**: SQL Injection attempt
   - **Alert Logic**: Detection of SQL keywords in query
   - **Example Log Lines**:
     ```
     <134>Oct 12 14:55:36 imperva1 DB:attack_type="SQL Injection" src_ip="192.168.1.1" query="SELECT * FROM users WHERE id=1' OR '1'='1"
     ```
   - **Logfile Location**: `/var/log/imperva`

3. **Alert Description**: Data exfiltration attempt
   - **Alert Logic**: Large data export from the database
   - **Example Log Lines**:
~~~

~~~
     ```
     <134>Oct 12 14:55:36 imperva1 DB:attack_type="Data Exfiltration"
src_ip="192.168.1.1" query="SELECT * FROM sensitive_table"
     ```
   - **Logfile Location**: `/var/log/imperva`

4. **Alert Description**: Privilege escalation
   - **Alert Logic**: Granting of elevated privileges to a user
   - **Example Log Lines**:
     ```
     <134>Oct 12 14:55:36 imperva1 DB:attack_type="Privilege Escalation"
src_ip="192.168.1.1" user="user1" action="GRANT ALL PRIVILEGES"
     ```
   - **Logfile Location**: `/var/log/imperva`

5. **Alert Description**: Unauthorized schema change
   - **Alert Logic**: Schema change by an unauthorized user
   - **Example Log Lines**:
     ```
     <134>Oct 12 14:55:36 imperva1 DB:attack_type="Schema Change"
src_ip="192.168.1.1" user="unauthorized_user" action="ALTER TABLE
sensitive_table ADD COLUMN new_column"
     ```
   - **Logfile Location**: `/var/log/imperva`

### Keycloak APPSSO Logs

1. **Alert Description**: Multiple failed login attempts
   - **Alert Logic**: 5 failed login attempts within 1 minute
   - **Example Log Lines**:
     ```
     2023-10-12 14:55:36,123 WARN  [org.keycloak.events]
...
Sure, here are five cybersecurity alerts for each of the specified
systems:

### Keycloak APPSSO Logs

#### Alert 1: Multiple Failed Login Attempts
~~~

```
1. **Alert Description**: Detect multiple failed login attempts within a short period, indicating a potential brute-force attack.
2. **Alert Logic**: 5 failed login attempts within 1 minute.
3. **Example Log Lines**:
   ```
   2023-10-01 12:00:01,123 WARN [org.keycloak.events] (default task-1) type=LOGIN_ERROR, realmId=example, clientId=app, userId=null, ipAddress=192.168.1.1, error=invalid_user_credentials
   2023-10-01 12:00:10,456 WARN [org.keycloak.events] (default task-1) type=LOGIN_ERROR, realmId=example, clientId=app, userId=null, ipAddress=192.168.1.1, error=invalid_user_credentials
   2023-10-01 12:00:20,789 WARN [org.keycloak.events] (default task-1) type=LOGIN_ERROR, realmId=example, clientId=app, userId=null, ipAddress=192.168.1.1, error=invalid_user_credentials
   2023-10-01 12:00:30,012 WARN [org.keycloak.events] (default task-1) type=LOGIN_ERROR, realmId=example, clientId=app, userId=null, ipAddress=192.168.1.1, error=invalid_user_credentials
   2023-10-01 12:00:40,345 WARN [org.keycloak.events] (default task-1) type=LOGIN_ERROR, realmId=example, clientId=app, userId=null, ipAddress=192.168.1.1, error=invalid_user_credentials
   ```
4. **Logfile Location**: `/opt/keycloak/logs/server.log`

#### Alert 2: Successful Login After Multiple Failures
1. **Alert Description**: Detect a successful login following multiple failed attempts, which could indicate a compromised account.
2. **Alert Logic**: 1 successful login after 3 failed attempts within 5 minutes.
3. **Example Log Lines**:
   ```
   2023-10-01 12:00:01,123 WARN [org.keycloak.events] (default task-1) type=LOGIN_ERROR, realmId=example, clientId=app, userId=null, ipAddress=192.168.1.1, error=invalid_user_credentials
   2023-10-01 12:00:10,456 WARN [org.keycloak.events] (default task-1) type=LOGIN_ERROR, realmId=example, clientId=app, userId=null, ipAddress=192.168.1.1, error=invalid_user_credentials
   2023-10-01 12:00:20,789 WARN [org.keycloak.events] (default task-1) type=LOGIN_ERROR, realmId=example, clientId=app, userId=null, ipAddress=192.168.1.1, error=invalid_user_credentials
```
```

```
   2023-10-01 12:01:00,123 INFO  [org.keycloak.events] (default task-1)
type=LOGIN, realmId=example, clientId=app, userId=12345,
ipAddress=192.168.1.1
   ```
4. **Logfile Location**: `/opt/keycloak/logs/server.log`

#### Alert 3: Admin Role Assignment
1. **Alert Description**: Detect when a user is assigned an admin role,
which could indicate privilege escalation.
2. **Alert Logic**: Any role assignment event where the role is "admin".
3. **Example Log Lines**:
   ```
   2023-10-01 12:05:00,123 INFO  [org.keycloak.events] (default task-1)
type=ROLE_UPDATE, realmId=example, clientId=app, userId=12345,
ipAddress=192.168.1.1, role=admin
   ```
4. **Logfile Location**: `/opt/keycloak/logs/server.log`

#### Alert 4: Unusual IP Address Access
1. **Alert Description**: Detect logins from an unusual IP address, which
could indicate unauthorized access.
2. **Alert Logic**: Login from an IP address not seen in the last 30 days.
3. **Example Log Lines**:
   ```
   2023-10-01 12:10:00,123 INFO  [org.keycloak.events] (default task-1)
type=LOGIN, realmId=example, clientId=app, userId=12345,
ipAddress=203.0.113.1
   ```
4. **Logfile Location**: `/opt/keycloak/logs/server.log`

#### Alert 5: Configuration Change
1. **Alert Description**: Detect any changes to the Keycloak
configuration, which could indicate tampering.
2. **Alert Logic**: Any configuration change event.
3. **Example Log Lines**:
   ```
   2023-10-01 12:15:00,123 INFO  [org.keycloak.events] (default task-1)
type=CONFIG_UPDATE, realmId=example, clientId=app, userId=12345,
ipAddress=192.168.1.1, config=realm-settings
   ```
```
```

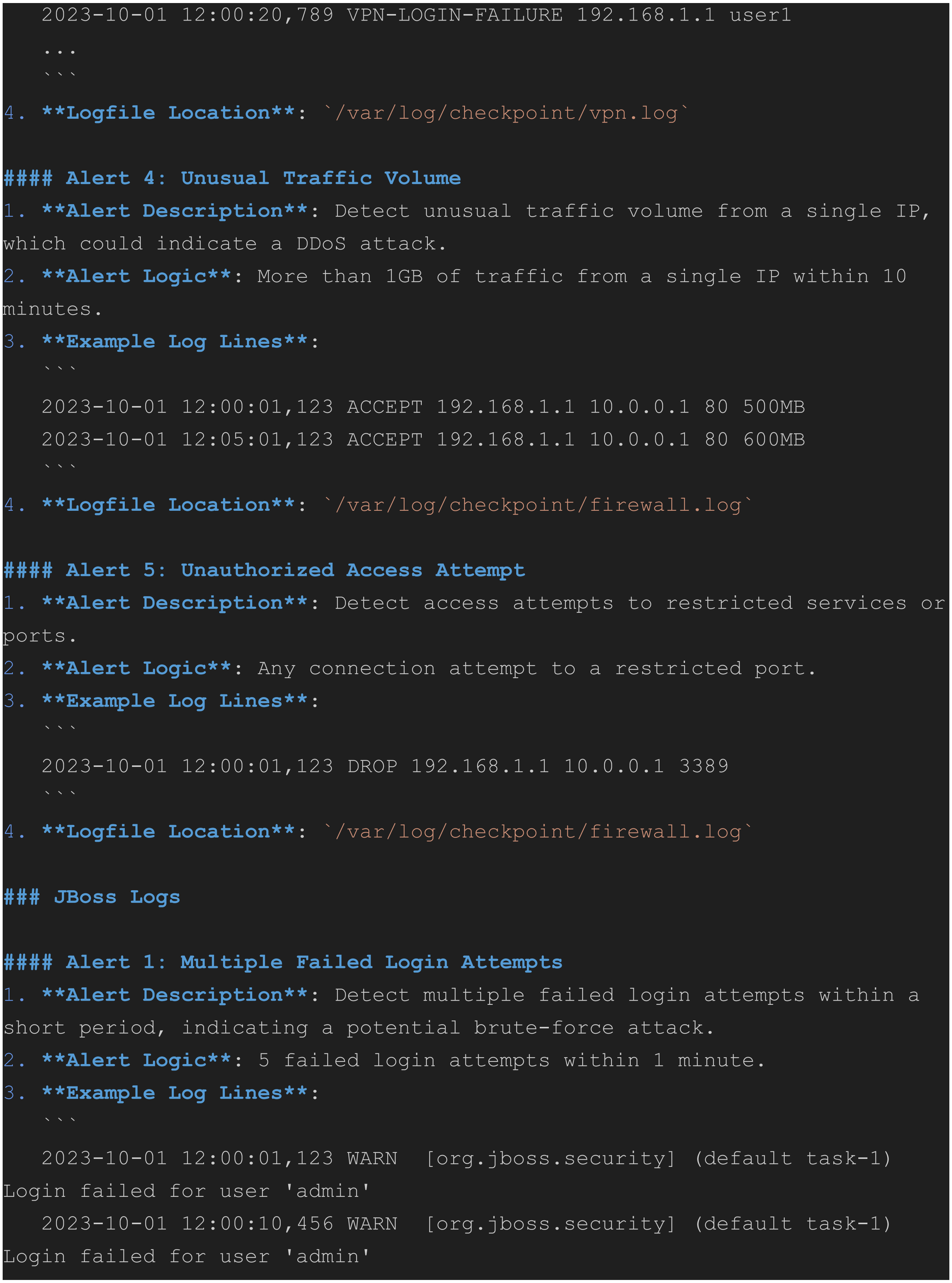
~~~markdown
4. **Logfile Location**: `/opt/keycloak/logs/server.log`

### Checkpoint Firewall Logs

#### Alert 1: Port Scanning
1. **Alert Description**: Detect port scanning activity, which could indicate reconnaissance.
2. **Alert Logic**: More than 100 connection attempts to different ports from the same IP within 1 minute.
3. **Example Log Lines**:
   ```
   2023-10-01 12:00:01,123 ACCEPT 192.168.1.1 10.0.0.1 80
   2023-10-01 12:00:02,456 ACCEPT 192.168.1.1 10.0.0.1 443
   2023-10-01 12:00:03,789 ACCEPT 192.168.1.1 10.0.0.1 22
   ...
   ```
4. **Logfile Location**: `/var/log/checkpoint/firewall.log`

#### Alert 2: Multiple Denied Connections
1. **Alert Description**: Detect multiple denied connection attempts, which could indicate an attack.
2. **Alert Logic**: 10 denied connections from the same IP within 1 minute.
3. **Example Log Lines**:
   ```
   2023-10-01 12:00:01,123 DROP 192.168.1.1 10.0.0.1 80
   2023-10-01 12:00:10,456 DROP 192.168.1.1 10.0.0.1 443
   2023-10-01 12:00:20,789 DROP 192.168.1.1 10.0.0.1 22
   ...
   ```
4. **Logfile Location**: `/var/log/checkpoint/firewall.log`

#### Alert 3: VPN Login Failure
1. **Alert Description**: Detect multiple failed VPN login attempts, which could indicate a brute-force attack.
2. **Alert Logic**: 5 failed VPN login attempts within 1 minute.
3. **Example Log Lines**:
   ```
   2023-10-01 12:00:01,123 VPN-LOGIN-FAILURE 192.168.1.1 user1
   2023-10-01 12:00:10,456 VPN-LOGIN-FAILURE 192.168.1.1 user1
~~~



~~~
   2023-10-01 12:00:20,789 VPN-LOGIN-FAILURE 192.168.1.1 user1
   ...
   ```
4. **Logfile Location**: `/var/log/checkpoint/vpn.log`

#### Alert 4: Unusual Traffic Volume
1. **Alert Description**: Detect unusual traffic volume from a single IP, which could indicate a DDoS attack.
2. **Alert Logic**: More than 1GB of traffic from a single IP within 10 minutes.
3. **Example Log Lines**:
   ```
   2023-10-01 12:00:01,123 ACCEPT 192.168.1.1 10.0.0.1 80 500MB
   2023-10-01 12:05:01,123 ACCEPT 192.168.1.1 10.0.0.1 80 600MB
   ```
4. **Logfile Location**: `/var/log/checkpoint/firewall.log`

#### Alert 5: Unauthorized Access Attempt
1. **Alert Description**: Detect access attempts to restricted services or ports.
2. **Alert Logic**: Any connection attempt to a restricted port.
3. **Example Log Lines**:
   ```
   2023-10-01 12:00:01,123 DROP 192.168.1.1 10.0.0.1 3389
   ```
4. **Logfile Location**: `/var/log/checkpoint/firewall.log`

### JBoss Logs

#### Alert 1: Multiple Failed Login Attempts
1. **Alert Description**: Detect multiple failed login attempts within a short period, indicating a potential brute-force attack.
2. **Alert Logic**: 5 failed login attempts within 1 minute.
3. **Example Log Lines**:
   ```
   2023-10-01 12:00:01,123 WARN  [org.jboss.security] (default task-1) Login failed for user 'admin'
   2023-10-01 12:00:10,456 WARN  [org.jboss.security] (default task-1) Login failed for user 'admin'
~~~

~~~
   2023-10-01 12:00:20,789 WARN  [org.jboss.security] (default task-1)
Login failed for user 'admin'
   ...
   ```
4. **Logfile Location**: `/opt/jboss/standalone/log/server.log`

#### Alert 2: Deployment of New Application
1. **Alert Description**: Detect the deployment of a new application,
which could indicate unauthorized changes.
2. **Alert Logic**: Any deployment event.
3. **Example Log Lines**:
   ```
   2023-10-01 12:05:00,123 INFO  [org.jboss.as.server]
(DeploymentScanner-threads - 1) JBAS015876: Starting deployment of
"newapp.war"
   ```
4. **Logfile Location**: `/opt/jboss/standalone/log/server.log`

#### Alert 3: Configuration Change
1. **Alert Description**: Detect any changes to the JBoss configuration,
which could indicate tampering.
2. **Alert Logic**: Any configuration change event.
3. **Example Log Lines**:
   ```
   2023-10-01 12:10:00,123 INFO  [org.jboss.as.controller]
(management-handler-thread - 1) JBAS014774: Configuration changed:
subsystem=logging
   ```
4. **Logfile Location**: `/opt/jboss/standalone/log/server.log`

#### Alert 4: High CPU Usage
1. **Alert Description**: Detect high CPU usage by the JBoss server, which
could indicate a performance issue or attack.
2. **Alert Logic**: CPU usage above 90% for more than 5 minutes.
3. **Example Log Lines**:
   ```
   2023-10-01 12:15:00,123 WARN  [org.jboss.as.server] (ServerService
Thread Pool -- 1) JBAS015960: High CPU usage detected: 95%
   ```
4. **Logfile Location**: `/opt/jboss/standalone/log/server.log`
~~~

~~~markdown
#### Alert 5: Unauthorized Access Attempt
1. **Alert Description**: Detect access attempts to restricted services or resources.
2. **Alert Logic**: Any unauthorized access attempt.
3. **Example Log Lines**:
   ```
   2023-10-01 12:20:00,123 WARN  [org.jboss.security] (default task-1) Unauthorized access attempt detected for user 'guest'
   ```
4. **Logfile Location**: `/opt/jboss/standalone/log/server.log`

### Tomcat Logs

#### Alert 1: Multiple Failed Login Attempts
1. **Alert Description**: Detect multiple failed login attempts within a short period, indicating a potential brute-force attack.
2. **Alert Logic**: 5 failed login attempts within 1 minute.
3. **Example Log Lines**:
   ```
   01-Oct-2023 12:00:01.123 WARNING [http-nio-8080-exec-1] org.apache.catalina.realm.LockOutRealm.authenticate An attempt was made to authenticate the locked user "admin"
   01-Oct-2023 12:00:10.456 WARNING [http-nio-8080-exec-1] org.apache.catalina.realm.LockOutRealm.authenticate An attempt was made to authenticate the locked user "admin"
   01-Oct-2023 12:00:20.789 WARNING [http-nio-8080-exec-1] org.apache.catalina.realm.LockOutRealm.authenticate An attempt was made to authenticate the locked user "admin"
   ...
   ```
4. **Logfile Location**: `/var/log/tomcat/catalina.out`

#### Alert 2: Deployment of New Application
1. **Alert Description**: Detect the deployment of a new application, which could indicate unauthorized changes.
2. **Alert Logic**: Any deployment event.
3. **Example Log Lines**:
   ```
~~~

~~~~
   01-Oct-2023 12:05:00.123 INFO [localhost-startStop-1]
org.apache.catalina.startup.HostConfig.deployWAR Deploying web application
archive [/var/lib/tomcat/webapps/newapp.war]
   ```
4. **Logfile Location**: `/var/log/tomcat/catalina.out`

#### Alert 3: Configuration Change
1. **Alert Description**: Detect any changes to the Tomcat configuration,
which could indicate tampering.
2. **Alert Logic**: Any configuration change event.
3. **Example Log Lines**:
   ```
   01-Oct-2023 12:10:00.123 INFO [localhost-startStop-1]
org.apache.catalina.startup.HostConfig.deployDescriptor Deploying
configuration descriptor
[/var/lib/tomcat/conf/Catalina/localhost/manager.xml]
   ```
4. **Logfile Location**: `/var/log/tomcat/catalina.out`

#### Alert 4: High Memory Usage
1. **Alert Description**: Detect high memory usage by the Tomcat server,
which could indicate a performance issue or attack.
2. **Alert Logic**: Memory usage above 90% for more than 5 minutes.
3. **Example Log Lines**:
   ```
   01-Oct-2023 12:15:00.123 WARNING [http-nio-8080-exec-1]
org.apache.catalina.util.SessionIdGeneratorBase.createSessionId High
memory usage detected: 95%
   ```
4. **Logfile Location**: `/var/log/tomcat/catalina.out`

#### Alert 5: Unauthorized Access Attempt
1. **Alert Description**: Detect acc\attempts to restricted services or
resources.
2. **Alert Logic**: Any unauthorized access attempt.
3. **Example Log Lines**:
   ```
   01-Oct-2023 12:20:00.123 WARNING [http-nio-8080-exec-1]
org.apache.catalina.realm.LockOutRealm.authenticate Unauthorized access
attempt detected for user 'guest'
~~~~

~~~markdown
   ```
4. **Logfile Location**: `/var/log/tomcat/catalina.out`

These alerts can help in identifying potential security incidents and
taking appropriate actions to mitigate risks.

### Oracle DB Alerts

#### Alert 1: Unusual Number of Failed Login Attempts
1. **Alert Description**: Detects an unusual number of failed login
attempts within a short period, indicating a potential brute-force attack.
2. **Alert Logic**: More than 5 failed login attempts within 1 minute.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT USERNAME, COUNT(*)
   FROM DBA_AUDIT_SESSION
   WHERE RETURN_CODE != 0 AND TIMESTAMP > SYSDATE - INTERVAL '1' MINUTE
   GROUP BY USERNAME
   HAVING COUNT(*) > 5;
   ```
4. **Logfile Location**: Oracle Audit Trail (DBA_AUDIT_SESSION)

#### Alert 2: Privilege Escalation Attempt
1. **Alert Description**: Detects attempts to grant DBA privileges to a
user.
2. **Alert Logic**: Any GRANT statement that includes DBA privileges.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT USERNAME, SQL_TEXT
   FROM DBA_AUDIT_TRAIL
   WHERE ACTION_NAME = 'GRANT' AND SQL_TEXT LIKE '%DBA%';
   ```
4. **Logfile Location**: Oracle Audit Trail (DBA_AUDIT_TRAIL)

#### Alert 3: Unauthorized Data Access
1. **Alert Description**: Detects unauthorized access to sensitive tables.
2. **Alert Logic**: SELECT statements on sensitive tables by
non-privileged users.
3. **Example Log Lines with Detection**:
   ```sql
~~~

```
   SELECT USERNAME, SQL_TEXT
   FROM DBA_AUDIT_TRAIL
   WHERE ACTION_NAME = 'SELECT' AND OBJECT_NAME IN ('SENSITIVE_TABLE1', 'SENSITIVE_TABLE2') AND USERNAME NOT IN ('PRIVILEGED_USER1', 'PRIVILEGED_USER2');
   ```
4. **Logfile Location**: Oracle Audit Trail (DBA_AUDIT_TRAIL)

#### Alert 4: Data Exfiltration Attempt
1. **Alert Description**: Detects large data exports.
2. **Alert Logic**: SELECT statements that return a large number of rows.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT USERNAME, SQL_TEXT, ROWS_PROCESSED
   FROM DBA_AUDIT_TRAIL
   WHERE ACTION_NAME = 'SELECT' AND ROWS_PROCESSED > 1000;
   ```
4. **Logfile Location**: Oracle Audit Trail (DBA_AUDIT_TRAIL)

#### Alert 5: Changes to Database Structure
1. **Alert Description**: Detects unauthorized changes to the database structure.
2. **Alert Logic**: DDL statements executed by non-privileged users.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT USERNAME, SQL_TEXT
   FROM DBA_AUDIT_TRAIL
   WHERE ACTION_NAME IN ('CREATE', 'ALTER', 'DROP') AND USERNAME NOT IN ('PRIVILEGED_USER1', 'PRIVILEGED_USER2');
   ```
4. **Logfile Location**: Oracle Audit Trail (DBA_AUDIT_TRAIL)

### MSSQL Alerts

#### Alert 1: Unusual Number of Failed Login Attempts
1. **Alert Description**: Detects an unusual number of failed login attempts within a short period, indicating a potential brute-force attack.
2. **Alert Logic**: More than 5 failed login attempts within 1 minute.
3. **Example Log Lines with Detection**:
   ```sql
```

~~~
   SELECT LOGIN_NAME, COUNT(*)
   FROM sys.fn_get_audit_file('path_to_audit_file', NULL, NULL)
   WHERE event_time > DATEADD(MINUTE, -1, GETDATE()) AND action_id = 'LGIF' AND succeeded = 0
   GROUP BY LOGIN_NAME
   HAVING COUNT(*) > 5;
   ```
4. **Logfile Location**: SQL Server Audit Logs

#### Alert 2: Privilege Escalation Attempt
1. **Alert Description**: Detects attempts to grant sysadmin privileges to a user.
2. **Alert Logic**: Any GRANT statement that includes sysadmin privileges.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT login_name, statement
   FROM sys.fn_get_audit_file('path_to_audit_file', NULL, NULL)
   WHERE action_id = 'GRANT' AND statement LIKE '%sysadmin%';
   ```
4. **Logfile Location**: SQL Server Audit Logs

#### Alert 3: Unauthorized Data Access
1. **Alert Description**: Detects unauthorized access to sensitive tables.
2. **Alert Logic**: SELECT statements on sensitive tables by non-privileged users.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT login_name, statement
   FROM sys.fn_get_audit_file('path_to_audit_file', NULL, NULL)
   WHERE action_id = 'SELECT' AND object_name IN ('SENSITIVE_TABLE1', 'SENSITIVE_TABLE2') AND login_name NOT IN ('PRIVILEGED_USER1', 'PRIVILEGED_USER2');
   ```
4. **Logfile Location**: SQL Server Audit Logs

#### Alert 4: Data Exfiltration Attempt
1. **Alert Description**: Detects large data exports.
2. **Alert Logic**: SELECT statements that return a large number of rows.
3. **Example Log Lines with Detection**:
   ```sql
~~~

~~~~
   SELECT login_name, statement, row_count
   FROM sys.fn_get_audit_file('path_to_audit_file', NULL, NULL)
   WHERE action_id = 'SELECT' AND row_count > 1000;
   ```
4. **Logfile Location**: SQL Server Audit Logs

#### Alert 5: Changes to Database Structure
1. **Alert Description**: Detects unauthorized changes to the database structure.
2. **Alert Logic**: DDL statements executed by non-privileged users.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT login_name, statement
   FROM sys.fn_get_audit_file('path_to_audit_file', NULL, NULL)
   WHERE action_id IN ('CREATE', 'ALTER', 'DROP') AND login_name NOT IN ('PRIVILEGED_USER1', 'PRIVILEGED_USER2');
   ```
4. **Logfile Location**: SQL Server Audit Logs

### MySQL DB Alerts

#### Alert 1: Unusual Number of Failed Login Attempts
1. **Alert Description**: Detects an unusual number of failed login attempts within a short period, indicating a potential brute-force attack.
2. **Alert Logic**: More than 5 failed login attempts within 1 minute.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT user_host, COUNT(*)
   FROM mysql.general_log
   WHERE event_time > NOW() - INTERVAL 1 MINUTE AND command_type = 'Connect' AND argument LIKE '%Access denied%'
   GROUP BY user_host
   HAVING COUNT(*) > 5;
   ```
4. **Logfile Location**: MySQL General Log

#### Alert 2: Privilege Escalation Attempt
1. **Alert Description**: Detects attempts to grant SUPER privileges to a user.
2. **Alert Logic**: Any GRANT statement that includes SUPER privileges.
~~~~

~~~
3. **Example Log Lines with Detection**:
   ```sql
   SELECT user_host, argument
   FROM mysql.general_log
   WHERE command_type = 'Query' AND argument LIKE 'GRANT%SUPER%';
   ```
4. **Logfile Location**: MySQL General Log

#### Alert 3: Unauthorized Data Access
1. **Alert Description**: Detects unauthorized access to sensitive tables.
2. **Alert Logic**: SELECT statements on sensitive tables by
non-privileged users.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT user_host, argument
   FROM mysql.general_log
   WHERE command_type = 'Query' AND argument LIKE 'SELECT%FROM
sensitive_table%' AND user_host NOT IN ('privileged_user1',
'privileged_user2');
   ```
4. **Logfile Location**: MySQL General Log

#### Alert 4: Data Exfiltration Attempt
1. **Alert Description**: Detects large data exports.
2. **Alert Logic**: SELECT statements that return a large number of rows.
3. **Example Log Lines with Detection**:
   ```sql
   SELECT user_host, argument
   FROM mysql.general_log
   WHERE command_type = 'Query' AND argument LIKE 'SELECT%' AND argument
LIKE '%LIMIT 1000%';
   ```
4. **Logfile Location**: MySQL General Log

#### Alert 5: Changes to Database Structure
1. **Alert Description**: Detects unauthorized changes to the database
structure.
2. **Alert Logic**: DDL statements executed by non-privileged users.
3. **Example Log Lines with Detection**:
   ```sql
~~~

~~~
   SELECT user_host, argument
   FROM mysql.general_log
   WHERE command_type = 'Query' AND argument LIKE 'CREATE%' OR argument
LIKE 'ALTER%' OR argument LIKE 'DROP%' AND user_host NOT IN
('privileged_user1', 'privileged_user2');
   ```
4. **Logfile Location**: MySQL General Log
~~~