AD-Sicherheit: Angriffsspuren analysieren

Aufgespürt

Fabian Murer, Gregor Wegberg

Das Windows-Ereignisprotokoll spielt eine zentrale Rolle beim Detektieren und Analysieren von Angriffen. Wie das Aufspüren typischer Angriffsmuster in Logs mithilfe frei verfügbarer Regeln geht, zeigt dieser Artikel am Beispiel einer Windows-Domäne.

Für das Erkennen von Angriffen ist der sicherheitsrelevante Teil des Windows-Ereignisprotokolls, das Windows-Sicherheitsereignisprotokoll der Windows Domain Controller (DC), eine reichhaltige Informationsquelle. In ihm hält jeder Domänencontroller unter anderem erfolgreiche und fehlgeschlagene Anmeldeversuche fest. Große Unternehmen leiten solche Protokollereignisse an ein SIEM-System (Security Information and Event Management) weiter [1]. Es sammelt sicherheitsrelevante Protokolle zentral, schützt sie vor unautorisierten Änderungen und erlaubt die effiziente Suche in den Ereignissen sowie deren Auswertung. Mit gezielten Abfragen wird in den gesammelten Daten nach unerwünschtem Verhalten gesucht.

Diese Suchabfragen sind vergleichbar mit einem SQL SELECT für die Abfrage von Daten aus einer relationalen Datenbank. Eine solche Abfrage kann beispielsweise die Suche nach erfolglosen Anmeldeversuchen mit unterschiedlichen Nutzerkonten sein, die alle vom selben Gerät stammen. Das weist auf ein potenziell infiziertes Gerät hin, das dazu genutzt wird, sich unerlaubten Zugang zu Nutzerkonten zu verschaffen, zum Beispiel mit einem Password-Spraying-Angriff [2]. Unternehmen, die eine proaktive und fortlaufende Suche nach Angriffen anstreben, werden solche und andere Angriffsmuster mit Suchabfragen beschreiben und als Alarm im SIEM einrichten. Findet das SIEM während der Verarbeitung von Protokolldaten eine Übereinstimmung, löst es Alarm aus.

Wenn Unternehmen ein SIEM neu einführen, stellen sie schnell fest, dass sie das Rad immer wieder neu erfinden: Die meisten Organisationen betreiben ähnliche IT-Systeme und die Angreifer setzen immer wieder die gleichen Angriffstechniken erfolgreich ein. Jedes Unternehmen betreibt einen zentralen Authentisierungsdienst, beispielsweise ein Active Directory. Dieser wird früher oder später einem Angriff auf die Benutzerpasswörter ausgesetzt sein, da schwache Passwörter nach wie vor weit verbreitet sind. (Wie man sie vermeidet, beschreibt der Artikel „Schwierige Wahl“ ab Seite 116.)

Ressourcen sparen durch Wissensaustausch

Angriffstechniken wie Brute Force oder Password Spraying erzielen daher immer noch ihre Erfolge. Das MITRE ATT&CK-Framework listet die gängigsten auf (siehe ix.de/zybc). Organisationen könnten ihre Ressourcen besser einsetzen, wenn solche Suchabfragen nach speziellen Angriffen einfach zu tauschen und universell einsetzbar wären. Ein öffentlicher Grundstock an Abfragen, der häufige Angriffsmuster für oft eingesetzte IT-Systeme bereitstellt, wäre eine weitere Effizienzsteigerung. Die durch das Vermeiden doppelter Arbeit freigewordenen Ressourcen lassen sich für die Entwicklung neuer oder unternehmensspezifischer Suchabfragen investieren und damit lässt sich das Sicherheitsniveau weiter ausbauen.

Diesen Wissensaustausch behindern die produktspezifischen Abfragesprachen. Eine Suchabfrage in einem SIEM-Produkt

iX-TRACT

- Um Angriffe zu entdecken und nachzuvollziehen, durchsuchen Sicherheitsexperten zahlreiche Logs, darunter die Windows-Ereignisprotokolle. Hilfreich sind dabei Sigma und Chainsaw.
- Sigma ist ein generisches Format zur Beschreibung von Angriffen, die in Protokollen entdeckt werden können. Zahlreiche frei verfügbare Sigma-Regeln der Open-Source-Community können in Suchabfragen für unterschiedliche SIEM konvertiert werden.
- Mit dem freien Werkzeug Chainsaw oder einem SIEM können Sigma-Regeln zum Aufspüren typischer und bekannter Angriffsmuster direkt in Windows-Ereignisprotokollen genutzt werden.

Einbrüche erkennen mithilfe von Microsoft-Werkzeugen

Microsofts Advanced Threat Analytics (ATA; siehe ix.de/zybc) kann in Active-Directory-Umgebungen unmittelbare Angriffe auf Konten und darüber hinaus verdächtige Aktivitäten erkennen. Dazu wird in der Regel auf Domänencontrollern ein Agent eingerichtet, der Daten an das ATA Center weiterleitet, das auf einem Windows-Server installiert wird. Sofort nach Inbetriebnahme erkennt ATA eindeutige Attacken wie Golden Ticket [5] oder DCSync – und nach einer dreiwöchigen Lernphase zudem Aktivitäten, die für die überwachte Umgebung ungewöhnlich sind.

Beim jüngeren Produkt Microsoft Defender for Identity (MDI, siehe ix.de/zybc), zuvor Azure Advanced Threat Protection (Azure ATP), werden ebenfalls leichtgewichtige Agenten auf jedem Domänencontroller im AD installiert, zusätzlich bindet es weitere Datenquellen wie Event Tracing for Windows ein. Defender wertet die Daten jedoch nicht lokal aus, sondern leitet sie in die Azure-Cloud weiter. Durch die Azure-Anbindung ist die Verzahnung mit anderen Cloud-Diensten wie Microsoft 365 Defender größer und das Entwicklungstempo höher: MDI erkennt mehr verdächtiges Verhalten als ATA und neu bekannt gewordene Schwachstellen, beispielsweise AS-REP Roasting [1] oder PetitPotam [6].

Seit Januar 2021 wird ATA nicht mehr weiterentwickelt, kann aber von Kunden, die nicht in die Cloud wollen, noch eingerichtet werden. Support und Sicherheitsupdates bietet Microsoft bis Januar 2026. Installationen von ATA können nach Defender for Identity migriert werden (siehe ix.de/zybc). ATA wie auch Defender erfordern in der Regel eine teure E5-Lizenz, Defender kann auch einzeln lizenziert werden.

ist nicht ohne Weiteres in einem anderen zu verwenden. So werden immer wieder die gleichen oder ähnliche Angriffsmuster in SIEM-Abfragen gegossen. Zwar steuern die SIEM-Hersteller und Drittanbieter durch eigene Marktplätze, Standardbibliotheken und ähnliche Vorhaben gegen. Die fehlende gemeinsame SIEM-übergreifende Sprache bleibt trotz allem ein Hindernis für eine effiziente Zusammenarbeit über Unternehmens- oder Produktgrenzen hinweg.

Vor dem gleichen Problem stehen auch Organisationen ohne SIEM. Spätestens wenn sie angegriffen werden, werden sie die Protokolle analysieren wollen. Andere Unternehmen wiederum möchten auch ohne konkreten Hinweis ihre zentralen Dienste auf Spuren eines Angriffs prüfen. In all diesen Situationen stehen die Unternehmen vor der Herausforderung, dass sie unterschiedliche Protokollformate und Log-Management-Systeme durchsuchen und nicht für jedes einzelne die gleichen Abfragen entwickeln möchten.

Sicherheitsforscher wiederum entdecken regelmäßig neue Angriffsmuster, Schadsoftware und Sicherheitslücken. Dieses Wissen wollen sie möglichst breit streuen und der Gesellschaft dabei helfen, sich zu schützen. Ohne eine allgemeine Sprache können sie jedoch keine einfach nutzbaren Suchabfragen bereitstellen. Es bleibt wieder an jedem Unternehmen, Ressourcen zu investieren und die in Worten beschriebenen Abläufe und Hinweise auf einen Angriff (Indicators of Compromise, IOC) in Suchabfragen zu übersetzen.

Gesucht: Lingua Franca für Angriffsmuster

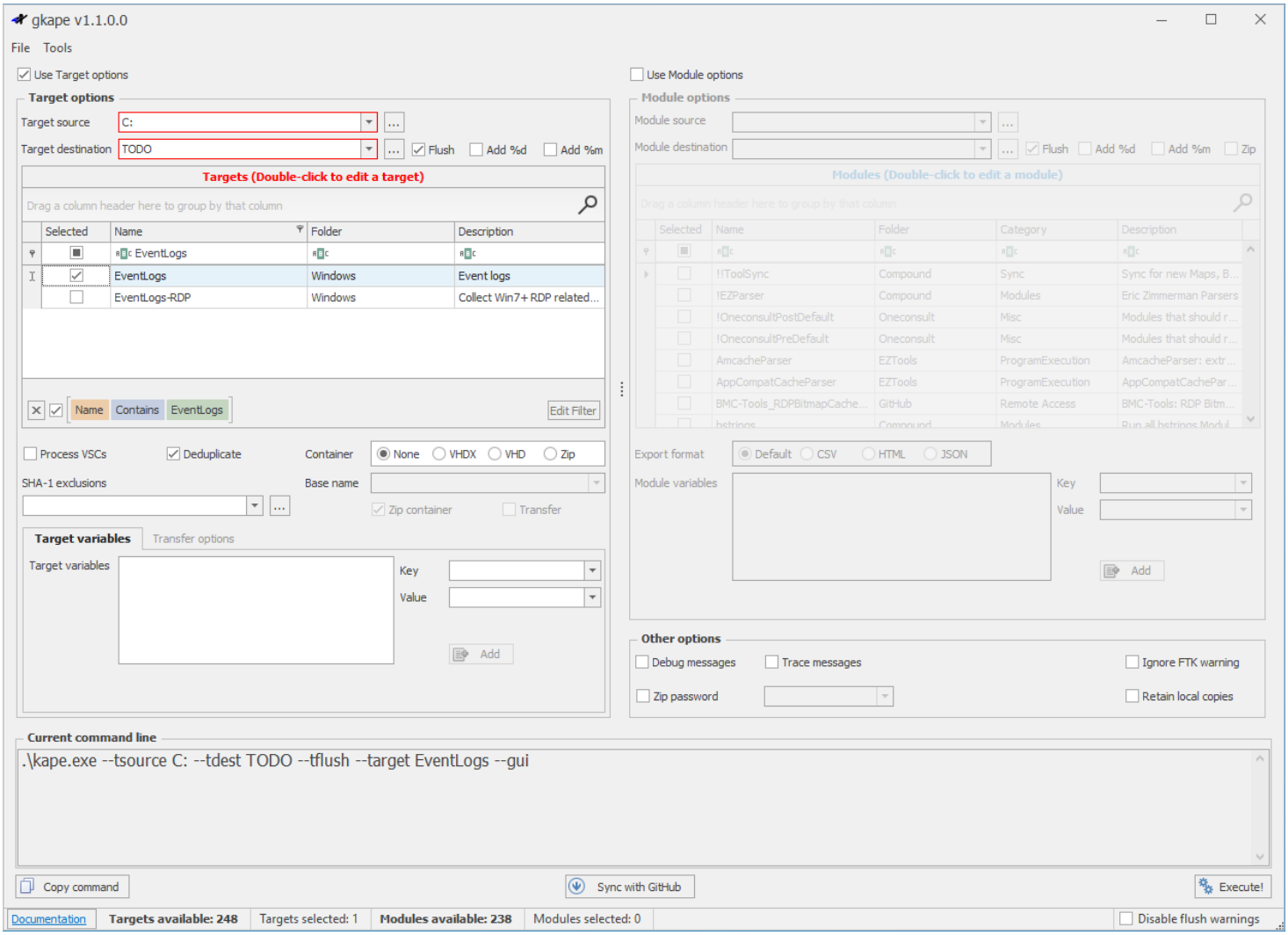
Es besteht also Bedarf für eine universelle Sprache, um die Suche nach verdächtigen Inhalten in Logs zu beschreiben. Ist sie definiert, lassen sich Muster tauschen und ein Grundstock aufbauen. Hier kommt das Sigma-Projekt ins Spiel (siehe ix.de/zybc).

Das Sigma-Projekt besteht aus drei zentralen, auf GitHub frei verfügbaren Elementen. Das erste Element ist Sigma, ein offenes und generisches Format zur Beschreibung von Angriffs- und Erkennungsmustern, die normalerweise als proprietäre Suchabfragen in einem SIEM geschrieben werden. Listing 1 zeigt eine solche Sigma-Regel zur Entdeckung von Computern, mit denen das SIEM mehr als drei fehlgeschlagene Anmeldeversuche unterschiedlicher Nutzerkonten assoziiert. Das Hauptziel des Projektes ist es, Security-Analysten eine einfache und strukturierte Form für die Definition eines Erkennungsmusters für Protokolle bereitzustellen, die geteilt und auf verschiedene Zielplattformen angewendet werden kann.

Hier kommt der zweite Baustein des Projekts ins Spiel, die Konsolenapplikation Sigmac (siehe ix.de/zybc). Sie kann Sigma-Regeln in eine Vielzahl von SIEM-Abfragesprachen konvertieren. Damit wird Sigma zu einer allgemeingültigen Sprache für den Austausch von Suchabfragen, unabhängig vom eingesetzten SIEM-Produkt. Die Applikation unterstützt nahezu jedes auf dem Markt relevante SIEM.

Listing 1: Sigma-Regel zum Aufspüren verdächtiger Geräte mit mehreren fehlgeschlagenen Anmeldeversuchen

```
title: Failed Logins with Different Accounts from Single Source System
id: e98374a6-e2d9-4076-9b5c-11bdb2569995
description: Detects suspicious failed logins with different user accounts from a single source
system
author: Florian Roth
date: 2017/01/10
modified: 2021/09/21
tags:
    - attack.persistence
    - attack.privilege_escalation
    - attack.t1078
logsource:
    product: windows
    service: security
detection:
    selection1:
        EventID:
            - 529
            - 4625
        TargetUserName: '*'
        WorkstationName: '*'
    condition: selection1 | count(TargetUserName) by WorkstationName > 3
falsepositives:
    - Terminal servers
    - Jump servers
    - Other multiuser systems like Citrix server farms
    - Workstations with frequently changing users
level: medium
```



Beispielkonfiguration für das Sammeln der Windows-Ereignisprotokolle (Abb. 1)

Der dritte Baustein des Projekts ist das Sigma GitHub Repository, eine öffentliche und zentrale Sammlung qualitativ hochwertiger Sigma-Regeln (siehe ix.de/zybc). Die in Listing 1 gezeigte Sigma-Regel stammt daraus. Interessierte Unternehmen können diese Regeln mit der Sigmac-Applikation für ihr SIEM konvertieren und frei verwenden.

Eine Kettensäge zur Verteidigung

Sigma-Regeln lassen sich nicht direkt auf das proprietäre Format der Windows-Ereignisprotokolle anwenden. Dazu müssten die Ereignisprotokolle erst in ein SIEM eingelesen und die Regeln in entsprechende Suchabfragen umgewandelt werden. Nutzt das eigene Unternehmen jedoch kein SIEM, wird es schwierig, die Protokolle mal eben schnell auf mögliche Angriffe zu untersuchen.

Für diesen Zweck hat Countercept, die Incident-Response-Abteilung der Sicherheitsfirma F-Secure, eine Software entwickelt, mit der sich Sigma-Regeln direkt auf Windows-Ereignisprotokolle anwenden lassen. Mit dem ebenfalls auf GitHub veröffentlichten Chainsaw (siehe ix.de/zybc) und den Sigma-Regeln lassen sich mit wenig Aufwand bekannte Bedrohungen in den Protokolldaten entdecken. Zusätzlich verfügt Chainsaw über weitere Funktionen zur Erkennung verdächtiger Aktivitäten:

Chainsaw Brute Force mit eingebauter Logik, die (zu) viele fehlgeschlagene Anmeldeversuche erkennt (Abb. 2)

- Extraktion und Analyse von Windows-Defender-, F-Secure-, Sophos- und Kaspersky-Antimalware-Protokollen;
- Erkennung, ob wichtige Ereignisprotokolle gelöscht wurden oder der Ereignisprotokolldienst gestoppt wurde;
- Benutzer neu erstellen oder zu sensiblen Benutzergruppen (zum Beispiel administrativen Gruppen) hinzufügen;
- Brute-Force-Angriffe auf lokale Nutzerkonten und
- Hinweise auf RDP-Anmeldungen.

Chainsaw ist ein Kommandozeilentool und kann auf allen gängigen Betriebssystemen (Windows, Linux und macOS) installiert und ausgeführt werden. Um es auf einem System zu installieren, genügt es, die neuste Version herunterzuladen (siehe ix.de/zybc). Zusätzlich zur Software enthält das Paket bereits eine Auswahl an Sigma-Regeln. Bei Bedarf können weitere Regeln aus dem Sigma-Projekt nachgeladen werden, sodass gleich der aktuelle Regelgrundstock genutzt wird.

Das Werkzeug kann sowohl Livedaten, also Ereignisprotokolle aus dem laufenden Betrieb, als auch Offlinedaten aus einer Datensicherung interpretieren. Da man es bei der Vorfallsbewältigung (Incident Response) meistens mit Protokollen aus zuvor gesicherten forensischen Artefakten zu tun hat, wird Chainsaw in den folgenden Beispielen Offlinedaten überprüfen.

Windows-Ereignisprotokolle sammeln

Von einer direkten Untersuchung auf einem laufenden System ist grundsätzlich abzuraten. Während des Betriebs generieren Applikationen neue Protokolleinträge, die aufgrund von Ressourcenbegrenzungen oder Systemkonfigurationen zum Löschen alter Ereignisse führen können. Damit besteht die Gefahr, während der Untersuchung wertvolle Hinweise zu verlieren. Forensikexperten raten daher, Windows-Ereignisprotokolle erst zu sichern und dann zu analysieren. Bei der Bewältigung von Vorfällen und in der IT-Forensik eignet sich dafür der Kroll Artifact Parser and Extractor, kurz KAPE (siehe ix.de/zybc; wie man mit KAPE arbeitet, zeigt ein in *iX* erschienenes vierteiliges Tutorial [3]).

Bei der Untersuchung eines Computers übernimmt KAPE zwei Aufgaben: Das Werkzeug sammelt für die Analyse relevante Dateien („Triage“) und verarbeitet sie mit Drittsoftware, die daraus die IT-forensisch bedeutenden Informationen extrahiert und zur Analyse und Bewertung aufbereitet. Die beiden Arbeitsschritte können getrennt voneinander stattfinden und sind durch Targets und Modules abgebildet. Nachfolgend werden ausschließlich die Windows-Ereignisprotokolle mit KAPE gesichert. Die Auswertung wird anschließend mit Chainsaw durchgeführt.

Standard-Mapping-Datei

Ereignis	Ereignisprotokoll	Event-ID
Erstellung eines neuen Prozesses	Microsoft-Windows-Sysmon/Operational	1
neue Netzwerkverbindung	Microsoft-Windows-Sysmon/Operational	3
Laden eines Moduls in einen Prozess	Microsoft-Windows-Sysmon/Operational	7
Erstellung einer Datei	Microsoft-Windows-Sysmon/Operational	11
Ereignisse in der Windows-Registry	Microsoft-Windows-Sysmon/Operational	13
PowerShell-Skriptblöcke	Microsoft-Windows-PowerShell/Operational	4104
neuer Prozess wurde erstellt	Security	4688
geplante Aufgabe wurde erstellt	Security	4698
Erstellung eines neuen Dienstes	System	7045

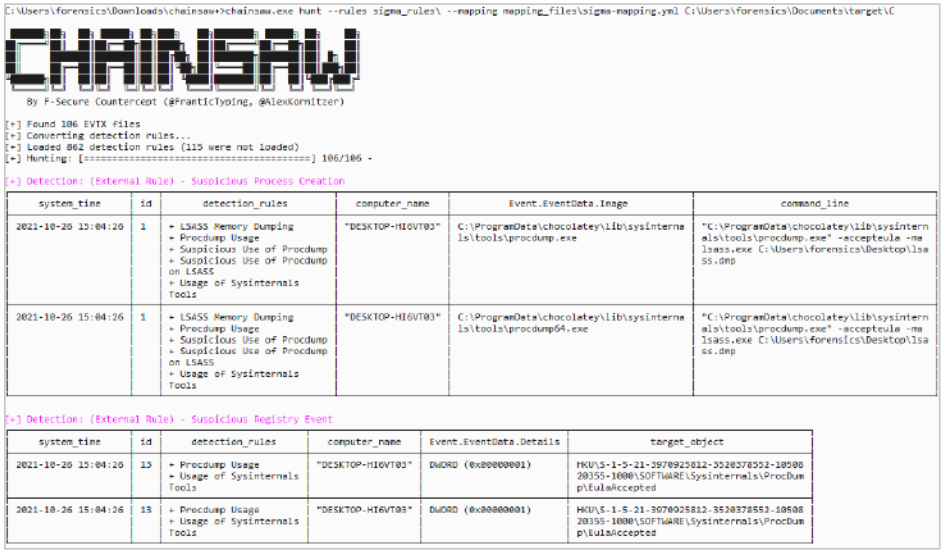
Vor der Ausführung von KAPE ist die Software und der zur Datensammlung benötigte Kommandozeilenbefehl auf einem Drittsystem vorzubereiten. Dieses System dient später als Analysesystem für die Auswertung der gesammelten Logdaten und ist nicht in den zu untersuchenden Vorfall involviert.

Zuerst lädt man KAPE von der offiziellen Webseite herunter und entpackt es auf einem externen Datenträger. Danach wird es auf den neuesten Stand gebracht: Mit dem Skript `Get-KAPEUpdate.ps1` wird die neueste Version automatisch heruntergeladen und entpackt. Im Anschluss werden die Targets und Module mit `kape.exe --sync` aktualisiert. Damit ist das Werkzeug auf dem neuesten Stand und zum Einsatz bereit.

Als Nächstes ist der KAPE-Kommandozeilenaufruf vorzubereiten. Hierzu startet man die grafische Oberfläche (gkape.exe), klickt „Use Target options“ an, setzt „Target source“ auf die Systempartition des zu untersuchenden Systems – also sehr wahrscheinlich C: –, trägt bei „Target destination“ einen Platzhalter ein und wählt das Target EventLogs aus (siehe Abbildung 1). Den Kommandozeilenbefehl unter „Current command line“ speichert man für die Ausführung auf dem zu untersuchenden System und schließt KAPE.

Nun schließt man das externe Laufwerk an das zu untersuchende System an, startet eine administrative Konsole wie PowerShell und wechselt damit in den KAPE-Ordner auf dem externen Datenträger. Danach fügt man den vorbereiteten Befehl in die Kommandozeile ein und ersetzt den Platzhalter für die „Target destination“. Im vorliegenden Beispiel wurde der externen Festplatte der Laufwerksbuchstabe E zugeordnet. Entsprechend wird der Platzhalter durch einen gültigen Ordnerpfad für die zu sammelnden Dateien ersetzt:

```
.\kape.exe --tsource C: --tdest E:\TargetDestination\ --tflush --target EventLogs --gui
```



Die Chainsaw-Ausgabe weist auf einen versuchten Angriff auf die Passworthashes der angemeldeten Benutzer hin (Abb. 3).

```
C:\Users\forensics\Downloads\chainsaw+>chainsaw.exe search -i -s mimikatz C:\Users\forensics\Documents\target\C

   By F-Secure Countercept (@FranticTyping, @AlexKornitzer)

[+] Found 106 EVTX files
[+] Searching: [---------------------------------------] 0/106 \
---
Event:
  "#attributes":
    xmlns: "http://schemas.microsoft.com/win/2004/08/events/event"
  EventData:
    MessageNumber: 1
    MessageTotal: 1
    Path: ""
    ScriptBlockId: a88d6a0f-25d4-4bad-92fa-3f86d8be6e0d
    ScriptBlockText: "\"C:\\Users\\forensics\\Downloads\\mimikatz_trunk\\x64\\mimikatz.exe\""
  System:
    Channel: Microsoft-Windows-PowerShell/Operational
    Computer: DESKTOP-HI6VT03
    Correlation:
      "#attributes":
        ActivityID: C1644660-CA70-0003-FB70-64C170CAD701
    EventID: 4104
    EventRecordID: 10570
    Execution:
      "#attributes":
        ProcessID: 664
        ThreadID: 5180
    Keywords: "0x0"
    Level: 5
    Opcode: 15
    Provider:
      "#attributes":
        Guid: A0C1853B-5C40-4B15-8766-3CF1C58F985A
        Name: Microsoft-Windows-PowerShell
    Security:
      "#attributes":
        UserID: S-1-5-21-3970925812-3520378552-1050820355-1000
    Task: 2
    TimeCreated:
      "#attributes":
        SystemTime: "2021-10-26T14:59:55.648801Z"
    Version: 1
```

Das search-Modul von Chainsaw ermöglicht weitere Suchoptionen in den Windows-Ereignisprotokollen (Abb. 4).

Nach Absetzen des Befehls beginnt das Sammeln der Daten. Anschließend finden sich im Zielordner (E:\TargetDestination\) die zusammenkopierten Windows-Ereignisprotokolldateien sowie drei Protokolldateien.

Für die nun anstehende Verarbeitung und Untersuchung der gesammelten Dateien wird der externe Datenträger wieder an das Analysesystem angeschlossen.

Verdächtiges in den Protokollen finden

Um die von KAPE gesicherten Ereignisprotokolle mithilfe von Chainsaw auf verdächtige Aktivitäten zu analysieren, kann dessen hunt-Modul verwendet werden. Die Standardauswertung der mit KAPE gesammelten Windows-Ereignisprotokolldateien geschieht wie folgt:

```
.\chainsaw.exe hunt E:\TargetDestination\C\Windows\System32\winevt\logs
```

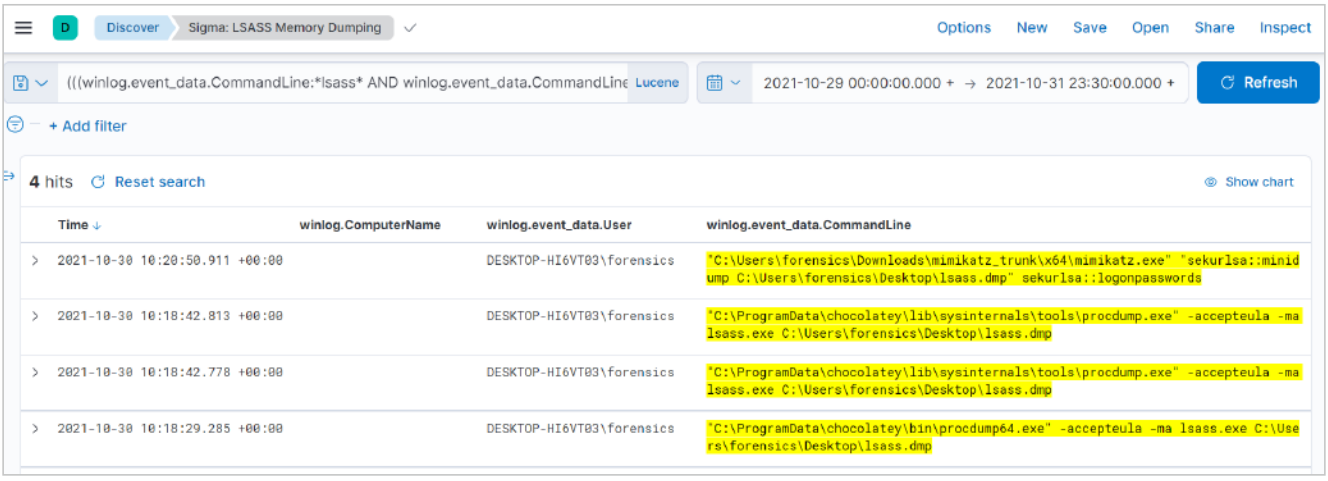
Auf diese Weise werden die eingebauten Regeln angewendet, worauf Chainsaw in seiner Ausgabe noch einmal warnend verweist. Mit deren Hilfe lassen sich jedoch bereits verdächtige Aktivitäten wie Brute-Force-Angriffe entdecken. Abbildung 2 zeigt die Ausgabe von Chainsaw in verschiedenen Tabellen auf der Kommandozeile. Zur besseren Analyse lassen sich die Resultate in eine CSV-Datei exportieren. Hierzu ist lediglich der Parameter `--csv` anzugeben.

Will man nun die Analyse mit Sigma-Regeln anreichern und weitere, ausgefeiltere Angriffstechniken identifizieren, muss man Chainsaw nur ein Verzeichnis mit Sigma-Regeln als Parameter übergeben. Der Befehl einer Suche mit Sigma-Regeln hat dann folgendes Format:

```
.\chainsaw.exe hunt --rules SIGMA_REGELN --mapping .\mapping_files\sigma_mapping.yml E:\TargetDestination\C\Windows\System32\winevt\logs
```

Aber aufgepasst, Chainsaw nutzt standardmäßig nur eine limitierte Auswahl an Ereignisprotokollen, auf die die Sigma-Regeln angewendet werden. Diese decken jedoch den größten Teil aller Angriffstechniken ab. Die für die Auswertung betrachteten Ereignisprotokolle werden dabei in einer Mapping-Datei beschrieben. Standardmäßig sind das die in der Tabelle „Standard-Mapping-Datei“ aufgeführten. Die Mapping-Datei lässt sich jedoch nach Belieben konfigurieren und ergänzen.

Wie im Beispiel in Abbildung 3 zu sehen, werden die angegebenen Sigma-Regeln so konvertiert, dass sie durch Chainsaw interpretiert und im Anschluss auf die geladenen Ereignisprotokolle angewendet werden können. Analog zur Standardausführung werden auch hier die Resultate basierend auf der Regel in Tabellen ausgegeben. Im gezeigten Beispiel ist zu sehen, dass gleich mehrere Sigma-Regeln angeschlagen haben. Die Spalte detection_rules führt sie auf.



Die konvertierte Sigma-Regel funktioniert nach der Umwandlung ebenso zuverlässig bei der Entdeckung sicherheitsrelevanter Ereignisse (Abb. 5).

Daraus ist ersichtlich, dass auf dem zu analysierenden System mit großer Wahrscheinlichkeit mittels der Software ProcDump von Sysinternals (siehe ix.de/zybc) der Arbeitsspeicher des lsass-Prozesses gesichert wurde. Das ist ein bekanntes Verfahren von Angreifern, um an die Passworthashes aller angemeldeten Benutzer zu gelangen (siehe ix.de/zybc). In diesem Fall zeigt die Spalte command_line sogar den tatsächlich ausgeführten Befehl:

```
"C:\ProgramData\chocolatey\lib\sysinternals\
    tools\procdump.exe" -accepteula -ma lsass.
       exe C:\Users\forensics\Desktop\lsass.dmp
```

Nach dem erfolgreichen Sichern des lsass-Prozesses versuchen Eindringlinge häufig die Passworthashes mittels Mimikatz auszulesen. Mimikatz ist bei Angreifern ein sehr beliebtes Werkzeug, das es unter anderem erlaubt, Passwörter im Klartext aus dem Arbeitsspeicher zu lesen oder eben die Passworthashes aus dem Abbild des Speicherbereiches des lsass-Prozesses. In der Folge kann Mimikatz ebenfalls verwendet werden, um einen Pass-the-Hash-Angriff zu starten [4] und sich so im Active Directory lateral zu verbreiten.

In diesem Beispiel scheint Chainsaw mittels Sigma-Regeln jedoch keine Indizien für das Ausführen von Mimikatz entdeckt zu haben. Für eine erweiterte Suche bietet sich ein weiteres Modul von Chainsaw an, das search-Modul. Es erlaubt nicht nur die einfache Suche nach Strings, sondern auch nach expliziten Event-IDs oder Regular Expressions. Die Grundanwendung lässt sich dabei wie folgt beschreiben:

```
.\chainsaw.exe search -i -s mimikatz E:\
        TargetDestination\C\Windows\System32\
                                    winevt\logs
```

Das Flag `-i` bewirkt, dass die Groß- und Kleinschreibung nicht beachtet wird. Wie in Abbildung 4 zu sehen ist, wurde auch hier Mimikatz auf dem System ausgeführt.

Sigma-Regeln im SIEM anwenden

Chainsaw ist ideal für eine spontane Überprüfung lokaler oder gesammelter und gesicherter Ereignisprotokolle. Je nach Aufbau der Infrastruktur kann es jedoch sein, dass die Protokolle nicht mehr lokal gesammelt, sondern zwecks optimierter Auswertung und Überwachung an ein SIEM geschickt werden. In diesem Fall ist Chainsaw nicht mehr effizient, da die Protokolle je nach Aufbau und Typ des Logservers nicht im Originalformat verfügbar sind. Doch genau hier zeigt sich die Stärke von Sigma-Regeln. Mithilfe des erwähnten Konvertierungsprogramms Sigmac lassen sie sich in für verschiedene SIEMs passende Suchabfragen umwandeln.

Für das Konvertieren benötigt man zuerst ein aktuelles Set an Regeln. Diese können einfach von der GitHub-Seite heruntergeladen werden. Damit die Konvertierungssoftware weiß, welches Format benötigt wird, werden die entsprechenden Informationen als Parameter mitgegeben.

- Ziel: Dieser Parameter definiert das SIEM, sodass die Konvertierungssoftware weiß, welche Syntax für die Suchabfrage verwendet werden muss. Die Software unterstützt bereits zahlreiche SIEMs, beispielsweise den Elastic Stack, Splunk, Qradar und viele mehr. Alle verfügbaren Ziele können mit `sigmac.exe --list` angezeigt werden.
- Konfiguration: Die Konfiguration wird benötigt, damit die Konvertierungssoft-

Listing 2: Beispiel für eine selbst festgelegte Alarmierungsregel

```
es_host: localhost
es_port: 9200
name: LSASS Dumping
type: any
index: winlogbeat-*
description: Detect creation of dump files containing the memory space of lsass.exe, which contains sensitive credentials. Identifies usage of
                                        Sysinternals procdump.exe to export the memory space of lsass.exe which contains sensitive credentials.
filter:
- query:
    query_string:
        query: (((winlog.event_data.CommandLine:*lsass* AND winlog.event_data.CommandLine:*.dmp*) AND (NOT (winlog.event_data.Image:*\\
              werfault.exe))) OR (winlog.event_data.Image:*\\procdump* AND winlog.event_data.Image:*.exe AND winlog.event_data.CommandLine:*lsass*))
priority: 2
realert:
    minutes: 0
alert:
- "email"
email:
- "alert@example.com"
```

ware weiß, wie und in welchem Format die Logs an das SIEM gesandt wurden. Das ist wichtig, weil je nach SIEM die indexierten Felder andere Bezeichnungen haben.

Im folgenden Beispiel wurden die Windows-Ereignisprotokolle mithilfe des Elastic-Agenten Winlogbeat (siehe ix.de/zybc) an Elastic Stack geschickt [1]. Entsprechend wird in diesem Bespiel für den Ziel-Parameter kibana-ndjson ausgewählt. Kibana ist die grafische Oberfläche des Elastic Stack. Als Konfigurationsparameter wird die Datei winlogbeat.yml verwendet, da die Ereignisprotokolle mittels Winlogbeat an das SIEM gesendet wurden.

Mit diesen beiden Parametern können nun beliebige Sigma-Regeln in Kibana-Suchabfragen umgewandelt werden. Der folgende Befehl zeigt, wie die Sigma-Regel LSASS Memory Dumping aus Abbildung 3 in eine Suchabfrage konvertiert wird:

```
.\sigmac.exe --target kibana-ndjson --config .\
    sigma\tools\config\winlogbeat.yml .\sigma\
     rules\windows\process_creation\win_lsass_
              dump.yml > win_lsass_dump.ndjson
```

Dieser Befehl speichert die Suchabfrage im NDJSON-Format in der Datei win_lsass_dump.ndjson. Die Datei kann nun über die Benutzeroberfläche von Kibana importiert werden. Das lässt sich über das Menü „Stack Management/Saved objects/Import“ bewerkstelligen. Die konvertierte Suchabfrage wird nun in Kibana als „Sigma: LSASS Memory Dumping“ gespeichert und ist jederzeit nutzbar. Wie Abbildung 5 zeigt, findet diese Suchabfrage ebenfalls das zuvor von Chainsaw identifizierte Ereignis.

Ein großer Vorteil von Sigma-Regeln in SIEMs ist, dass sie bereits als ein Anwendungsfall für Alarmierungen verwendet werden können. So lassen sich beispielsweise basierend auf Sigma-Regeln eigene Dashboards oder sogar automatisierte Alarme erstellen, die bei einem Treffer ausgelöst werden. Die Konvertierungssoftware erlaubt bei ELK-Stack als SIEM das automatische Übersetzen der Sigma-Regel in eine Alarmierungsregel. Dazu muss lediglich der Zielparameter ausgewechselt werden:

```
.\sigmac.exe --target elastalert --config .\
    sigma\tools\config\winlogbeat.yml .\sigma\
          rules\windows\process_creation\
                              win_lsass_dump.yml
```

Dieser Befehl erzeugt eine neue Regel, die ElastAlert (siehe ix.de/zybc) interpretieren kann. Im Beispiel in Listing 2 wurden die Felder es_host, es_port und email automatisch generiert. Lediglich die Felder zur Identifikation der Elasticsearch-Instanz und die Art und Weise, wie der Alarm stattfinden soll, wurden für die ElastAlert-Regel ergänzt. Die definierte Regel würde für jede erfolgreiche Abfrage des Suchbefehls – also mindestens ein Resultat – eine E-Mail an alert@example.com schicken.

Fazit

Als freies und universelles Format erleichtert Sigma den Austausch von Erkennungsmustern für Protokolldaten. Damit werden die Kompatibilitätsgrenzen zwischen den unterschiedlichen SIEM-Produkten überwunden und Sicherheitsfachleute aller Unternehmen und Organisationen können gemeinsam einen Pool an Suchabfragen aufbauen.

Chainsaw macht sich die Stärke und Verfügbarkeit von Sigma-Regeln zu eigen. Mit ihm lassen sich Windows-Ereignisprotokolle nach besonders häufig auftretenden Angriffsmustern mit geringem Aufwand durchsuchen. Beide Werkzeuge können die IT-Sicherheit ein Stück voranbringen. (ur@ix.de)

Quellen

[1] Fabian Murer; Protokollschätze; Incident Response und Forensik – Angreifer durch Logs enttarnen; *iX* 8/2021, S. 94

[2] Frank Ully; Nach oben gehangelt; Informationsbeschaffung – was jeder Domänenbenutzer alles sieht; *iX* 10/2020, S. 58

[3] Gregor Wegberg; Tutorial IT-Forensik mit dem Werkzeug KAPE; 4 Teile ab *iX* 7/2021, S. 128

[4] Frank Ully; Fette Beute; Passwörter und Hashes – wie Angreifer die Domäne kompromittieren; *iX* 11/2020, S. 94

[5] Yves Kraft, Frank Ully; Zwischen den Wäldern; Inter-Forest und Persistenz: Wie Angreifer sich über einen AD-Forest hinaus ausbreiten und festsetzen; *iX* 4/2021, S. 102

[6] Hans-Joachim Knobloch; Und ewig grüßt das Nilpferd; PetitPotam und weitere Wege, die Kontrolle über das AD zu übernehmen; *iX* 9/2021, S. 91

[7] Alle erwähnten Projekte und Werkzeuge sind über ix.de/zybc zu finden.

Fabian Murer

ist Senior Digital-Forensics- und Incident-Response-Spezialist bei der Oneconsult AG. Er unterstützt Firmen bei der Bewältigung von Cyberattacken und untersucht als IT-Forensiker die Methoden der Angreifer bis auf den letzten Befehl.

Gregor Wegberg

unterstützt mit seinem Team bei der Oneconsult AG Organisationen bei der Bewältigung von Cyberangriffen.