Detektion von Angriffsvektoren in Active Directory Umgebungen mittels BloodHound - Forensische Nachweisbarkeit der Datensammlung

IT-Forensik Projekt II

eingereicht von: Sebastian Häuser

Studiengang IT-Forensik

Betreuer: Prof. Dr.-Ing. Antje Raab-Düsterhöft

Mülheim an der Ruhr, den 24.06.2023

Aufgabenstellung

Die Projektarbeit beschäftigt sich damit, zunächst ein grundlegendes Verständnis über die Arbeitsweise der Open-Source Software BloodHound zu entwickeln, wie damit Angriffsvektoren und Angriffspfade in Active Directory Umgebungen erkannt werden können. Der Fokus liegt auf der Datensammlung mit dem Datensammler SharpHound und der Auswertung von Ereignissen die in den Windows eigenen Ereignisprotokollen der Informationssammlung zuzuordnen sind. Es ist zu prüfen, ob sich Muster bzw. Pattern finden lassen, die zur Erkennung genutzt werden können.

Es ist eine Testumgebung aufzubauen, in der BloodHound in einem Domänen Netzwerk ausgeführt wird und die Ergebnisse und Logdateien anschließend analysiert werden können.

Es sollen folgende Fragestellungen betrachtet werden:

- Wie sammelt BloodHound die Daten, damit es zur Schwachstellenanalyse von Active Directory Umgebung genutzt werden kann und welche Voraussetzungen müssen erfüllt sein?

- Welche Spuren hinterlässt die Datensammlung in den Windows eigenen Ereignisprotokollen? Was wird im Standard-Zustand protokolliert? Lässt sich die Erkennung durch weitere Anpassungen optimieren?

- Sind Unterschiede in Protokollen zu erkennen, wenn Zugangsdaten von Benutzerkonten mit und ohne administrative Berechtigungen verwendet werden?

- Kann ein Pattern erkannt und die Nutzung forensisch nachgewiesen werden?

Inhaltsverzeichnis

1 Einleitung

Das Active Directory bzw. die Active Directory Domain Services (AD DS) ist ein von Microsoft entwickelter Verzeichnisdienst und stellt eine der zentralen und wichtigsten Komponenten in einem auf Windows basierendem Netzwerk dar. Das Active Directory verwaltet und organisiert Ressourcen wie Benutzer- und Computerkonten, Serversysteme, Gruppen und Gruppenrichtlinien und fungiert als Authentifizierungskomponente.

Das Active Directory bildet das Herzstück einer Domäne, da hier sämtliche Informationen zentral gespeichert werden. Aufgrund dessen ist es ein primäres Ziel für Angreifer, da die Kompromittierung eines Active Directory mit der Kompromittierung der gesamten Domäne eines Unternehmens einhergeht. Sollte ein Angreifer administrativen Zugriff auf das AD erlangen, so kann er jegliche Eingriffe in Ressourcen vornehmen, diese kontrollieren, verändern und löschen. Es ist daher von entscheidender Bedeutung, dem Active Directory die notwendige Sicherheitsbetrachtung zukommen zu lassen. Schwachstellen und Angriffspunkte des Verzeichnisdienstes müssen erkannt und beseitigt werden, um so die Gesamtsicherheit des Netzwerkes zu erhöhen.

BloodHound ist eine Open-Source-Software zur Schwachstellenanalyse von Active Directory Umgebungen.[1] Mithilfe der Software werden komplexe Beziehungen innerhalb des Active Directory zwischen Benutzerkonten, Computerkonten, Gruppen und weiteren Objekten aufgedeckt und grafisch dargestellt. Hierdurch sollen insbesondere verborgene und unbeabsichtigte Beziehungen erkannt werden. BloodHound wird sowohl von Angreifern als auch von „Verteidigern“ genutzt, um Angriffspfade zu lokalisieren und auszunutzen bzw. diese zu eliminieren.

Die Projektarbeit besteht darin, die Datensammlung anhand zweier Sammelmethoden durchzuführen und die auf den Maschinen in den Ereignisprotokollen erfassten Ereignisse zu sichern und auszuwerten. Es soll untersucht werden, ob und wie die Datensammlung von BloodHound im Netzwerk erkannt und forensisch nachgewiesen werden kann. Es ist zu prüfen, ob sich Muster als eine Art Pattern feststellen lassen, die eine Erkennung

ermöglichen.

Wie in der Aufgabenstellung festgelegt wird hierfür zunächst eine Testumgebung aufgebaut, in der die Voraussetzungen geschaffen werden, damit BloodHound lauffähig ist. Anschließend wird die Datensammlung mit dem Datensammler SharpHound durchgeführt und die Logdateien auf den Maschinen gesichert und ausgewertet. Die Sammlung wird anhand von zwei ausgewählten Sammelmethoden im Kontext von zwei verschiedenen Benutzerkonten durchgeführt, um so auch mögliche Unterschiede durch unterschiedliche Berechtigungen der Zugangsdaten in den Protokollen feststellen zu können.

2 Grundlagen

2.1 Active Directory

Active Directory (AD) ist ein Verzeichnisdienst von Microsoft, der in Windows-basierten Netzwerken zentral zur Verwaltung und Organisation von Ressourcen genutzt wird. Im AD wird die Domänenstruktur eines Netzwerkes verwaltet und durch Organisationseinheiten der Aufbau eines Unternehmens abgebildet. Es werden unter anderem Benutzerkonten, Sicherheitsgruppen, Computerkonten und Richtlinien eines Netzwerkes verwaltet. Zusammengefasst werden im AD alle Objekte, Ressourcen und Richtlinien zur Anmeldung und zur Zugriffssteuerung verwaltet.[2]

Der Verzeichnisdient Active Directory wurde mit der Betriebssystemversion Windows 2000 eingeführt. Seit der Version Windows Server 2008 wird er unter der Bezeichnung Active Directory Domain Services geführt und bündelt die folgenden Serverrollen:

- Active Directory Domain Services (AD DS)
- Active Directory Lightweight Directory Services (ADLDS)
- Active Directory Federation Services (ADFS)
- Active Directory Rights Management Services (ADRMS)
- Active Directory Certificate Services (ADCS)

Netzwerkumgebungen die auf den Microsoft Betriebssystemen wie Windows Server aufbauen, nutzen Active Directory als Verzeichnisdienst. Aufgrund der weiten Verbreitung und der idealen Integration von weiteren Microsoft Systemen, wie beispielsweise Microsoft Exchange Server und Netzwerkdiensten wie DNS und DHCP, gilt das Active Directory als Standard. Das Betriebssystem Windows kommt auf einen Marktanteil von mehr als 60%.[3]

Daneben gibt es noch weitere Verzeichnisdienste wie beispielsweise OpenLDAP [4], Novell eDirectory [5] oder FreeIPA [6], die zur zentralen Verwaltung von Ressourcen in Netzwerken eingesetzt werden.

In dieser Projektarbeit werden ausschließlich Informationen aus dem Active Directory beschafft, andere Verzeichnisdienste werden nicht ausgewertet.

2.2 Enumeration

Die Enumeration steht für die Aufzählung von Informationen und ist ein grundlegender Teil in der Aufklärungsphase eines Cyberangriffs.[7]

Enumerationstools sind Sicherheitswerkzeuge, die zur Informationsgewinnung verwendet werden. Angreifer und Sicherheitsverantwortliche versuchen mithilfe dieser Tools möglichst viele Informationen über das vorliegende Netzwerk zu erlangen, um so ein detailliertes Bild über den Aufbau und die Struktur der Umgebung zu erhalten. Es werden alle relevanten Einträge über Hosts, Benutzerkonten, Gruppen, Dienste und weitere Ressourcen im Netzwerk gesammelt.

Zur Enumeration von Verzeichnisdiensten existieren diverse Tools, die jeweils eigene Vor- und Nachteile mit sich bringen, verschiedene Voraussetzungen zur Ausführung benötigen und unterschiedliche Schwerpunkte in der Informationsanalyse forcieren.

BloodHound ist ein solches Enumerationstool und wird zur Informationsanalyse eines Active Directory verwendet. Im Gegensatz zu vielen anderen Enumerationstools liegt der Unterschied bei BloodHound darin, dass die gesammelten Daten grafisch dargestellt und Beziehungen sichtbar gemacht werden.

Die Projektarbeit basiert auf der Enumeration mittels der Software BloodHound und den notwendigen Komponenten. Es wird kein Vergleich zwischen weiteren Enumerationstools gezogen. Es macht aber sowohl aus der Sicht eines „Verteidigers“ als auch eines Angreifers Sinn, verschiedene Tools zu nutzen und den Schwerpunkt auf verschiedene Netzwerkbereiche legen zu können. Durch die Zusammenstellung der Ergebnisse kann ein detailliertes Gesamtbild entwickelt werden. Weitere Enumerationstools sind beispielsweise PowerView [8], ADRecon [9] oder Group3r [10].

2.3 BloodHound

BloodHound ist eine Open-Source-Software, die sowohl von IT-Sicherheitsforschern und Penetrationstestern als auch von Angreifern zur Schwachstellenanalyse und Angriffspfaderkennung in Active Directory Umgebungen eingesetzt wird. Laut dem Red Canary Threat Report des Jahres 2022 wird BloodHound auf Platz 9 der am häufigsten erkannten Bedrohungen ausgewiesen.[11] Die Software dient dabei als Grundlage für darauf aufbauende Angriffsszenarien wie beispielsweise das Auslesen von Passwort-Hashwerten mit Tools wie Mimikatz, weshalb Informationen über Sitzungsdaten von besonderem Interesse sind.

BloodHound ist eine JavaScript-Webanwendung, die sich die Graphentheorie zur Visualisierung von Datenmengen zu Nutze macht. Mit Hilfe der Software werden Daten aus dem Verzeichnisdienst Active Directory und von Systemen im Netzwerk gesammelt und grafisch dargestellt. Hierdurch können teils sehr komplexe und oftmals unbekannte Beziehungen der Objekte aufgedeckt und veranschaulicht werden. Die Interpretation der Ergebnisse wird durch die Visualisierung im Gegensatz zu textuellen Enumerationstools erleichtert. Durch Abfragen können verschiedene Graphen auf Grundlage der gesammelten Daten erstellt werden.

Die Anwendung BloodHound ist zur Darstellung und Visualisierung der Daten zuständig, die eigentliche Informationsbeschaffung erfolgt durch den von BloodHound bereitgestellten Datensammler SharpHound. Die gesammelten Daten werden in der Graphdatenbank Neo4j [12] gespeichert.

Um Abfragen auf den Daten zu formulieren, stellt BloodHound bereits einige vorgefertigte Datenabfragen in der grafischen Oberfläche zur Verfügung. Diese sind beispielsweise:

Finde

- alle Domänenadministratoren
- Domänenvertrauensstellungen
- den kürzesten Pfad zu Domänenadministratoren
- Accounts die sich für Kerberroasting-Angriffe [13] eignen

- Systeme, auf denen Benutzer lokale Adminrechte besitzen

Auf Abbildung 1 ist ein Ausschnitt der Abfragen zu sehen.

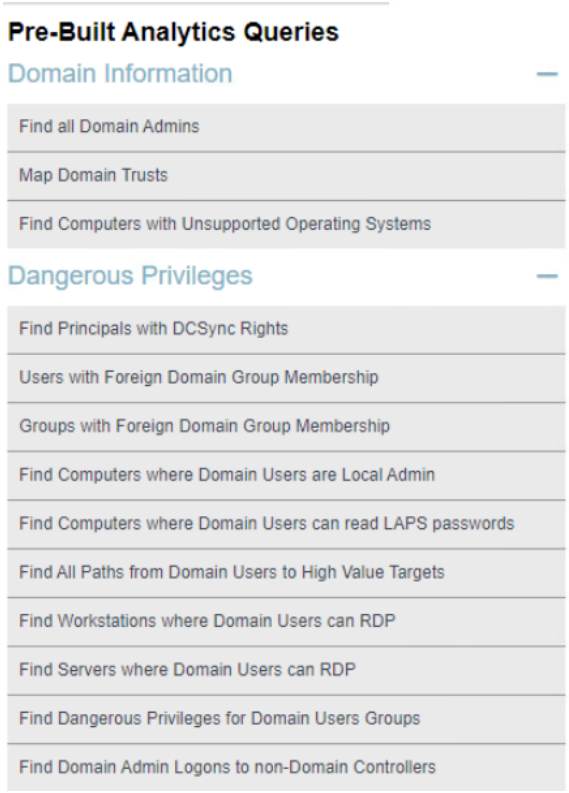

Abbildung 1 - Vorgefertigte Abfragen BloodHound

2.3.1 SharpHound

Für die Informationsbeschaffung wird der offizielle Datensammler SharpHound verwendet. SharpHound wird als ausführbare Datei und als PowerShell Skript über das GitHub-Repository von BloodHound zur Verfügung gestellt. Daneben wird auch eine Version für Azure-Umgebungen bereitgestellt (AzureHound.md), womit auch Verzeichnisdienste in der Cloud enumeriert werden können. In dieser Projektarbeit wird die Datensammlung mittels der ausführbaren Datei analysiert, eine Auswertung eines Azure-AD in der Cloud wird nicht durchgeführt.

Das Skript selbst stellt zahlreiche Abfragen an Domänencontroller und erreichbare Clients im Netzwerk und speichert die gesammelten Informationen in JSON-Dateien (s. Abschnitt 2.6) ab. Die Ergebnisdateien lassen sich dann wiederum zur Bildung von Graphen in BloodHound importieren.

Durch optionale Filter, genannt Flags, kann die Informationssammlung beeinflusst werden. Es können sowohl einzelne Flags gesetzt werden, um so nur bestimmte Informationen abzufragen als auch Sammelmethoden ausgewählt werden, die die Ausführung von mehreren Flags kombinieren. Es kann außerdem auf das Sammelverhalten Einfluss genommen werden. (s. Abschnitt 4.2)

2.3.2 Neo4j

Neo4j ist eine quelloffene Graphdatenbank zur Speicherung von vernetzten Informationen.[14] Die Daten werden im Gegensatz zu relationalen Datenbanken anstatt in Tabellen in einer Graphstruktur gespeichert. Das Design empfiehlt sich aus diesem Grund besonders zur Speicherung und Darstellung von Daten, die in Beziehung zueinanderstehen, wie es in Verzeichnisdiensten der Fall ist. Die Daten werden in Form von Kanten und Knoten organisiert und verknüpft.

Neo4j bietet mit der Abfragesprache Cypher [15] die Möglichkeit eigene Abfragen über einen Graphen zu formulieren, um so Muster und Zusammenhänge der Daten zu analysieren.

2.4 LDAP - Lightweight Directory Access Protocol

Das Lightweight Directory Access Protocol (LDAP) ist ein Protokoll, welches zur Kommunikation mit Verzeichnisdiensten verwendet wird. Durch Nutzung des Protokolls können Abfragen und Änderungen in einem verteilten Verzeichnisdienst ausgeführt werden. Es ist eine „vereinfachte“ Version des Directory Access Protocol (DAP) und wurde 1993 an der Universität von Michigan entwickelt [16] und durch die IETF in RFC 4510, 4511 und 4532 dokumentiert.[17]

LDAP zählt zu den Hauptkomponenten eines Active Directory Verzeichnisdienstes. Daneben kommt das Protokoll auch in anderen (verteilten) Verzeichnissen wie beispielsweise OpenLDAP zum Einsatz. Es ermöglicht Anwendungen den standardisierten Zugriff um Abfragen, Suchen und Aktualisierungen von Informationen durchzuführen.

BloodHound bzw. SharpHound nutzt dieses Protokoll zur Kommunikation mit dem Domänencontroller, um Informationen aus dem AD abzufragen.

Es besteht sowohl die Möglichkeit einer ungesicherten oder einer gesicherten Übertragung. Standardmäßig erfolgt die Kommunikation ungesichert über Port 389. Eine gesicherte Übertragung erfolgt über TLS über Port 636 (LDAPS).

2.5 RPC über SMB

Zur Informationsgewinnung werden neben dem Verzeichnisdienst auch im Netzwerk erreichbare Clients von BloodHound abgefragt. Die Abfrage erfolgt dann über Remoteprozeduraufrufe (RPC) über das Server Message Block (SMB) Protokoll.

RPC über SMB wird zur Kommunikation von Computern und Servern in Windows Netzwerken verwendet und ist ein wichtiger Bestandteil für Client-Server Anwendungen. Es basiert auf dem Mechanismus des Remoteprozeduraufruf und stellt eine Methode zum Funktionsaufruf von entfernten Systemen bereit. Auf einem Client werden Prozeduren aufgerufen, die aus Sicht des Clients wie eine lokale Ausführung dargestellt werden.[18]

Die Kommunikationsverwaltung und der Transport der Daten erfolgen über das SMB-Protokoll über Port 445.[19]

BloodHound nutzt RPC über SMB, um Abfragen an Clients im Netzwerk zu stellen, damit beispielsweise Daten über gespeicherte Benutzersitzungen oder lokale Gruppenmitgliedschaften ermittelt werden können.

2.6 JavaScript Object Notation – JSON

Das JavaScript Object Notation Format (JSON) ist ein Format, das zur strukturierten Darstellung und zum Austausch von Daten verwendet wird. Es basiert auf einer Untermenge der Programmiersprache JavaScript, stellt aber keine eigene Programmiersprache dar. Durch den strukturierten Aufbau ist die Notation für den Menschen gut lesbar und für Maschinen einfach zu parsen.[20]

JSON ermöglicht die Darstellung von Daten in Form von Schlüssel-Wert-Paaren, Arrays und verschachtelten Objekten. Das Format ist sprachunabhängig und wird von den meisten modernen Programmiersprachen unterstützt. Durch die gute Interpretierbarkeit und Interoperabilität ist es zu einem gängigen Standard für den Austausch von Daten zwischen Systemen geworden.

Die Ergebnisse von SharpHound werden im JSON-Format strukturiert gespeichert und in BloodHound importiert.

2.7 Windows Ereignisanzeige

Windows Betriebssysteme protokollieren eine Reihe von System Ereignissen in der eigenen Ereignisanzeige. Diese spielen u.a. eine wichtige Rolle bei der Überwachung und Fehlerbehebung von Systemen. Die Ereignisse in der Ereignisanzeige unterteilen sich in die folgenden Kategorien:

- **Anwendung**
  In der Kategorie Anwendung werden Informationen, Fehlermeldungen und Anwendungsereignisse protokolliert.
- **Sicherheit**
  In der Kategorie Sicherheit werden sicherheitsrelevante Ereignisse wie Anmeldungen, Änderungen von Berechtigungen und Zugriffsversuche protokolliert.
- **Installation**
  Installationsereignisse von Anwendungen und Aktualisierungen werden in der Kategorie Installation protokolliert.
- **System**
  Sämtliche Ereignisse, die das Systemverhalten betreffen werden in System protokolliert, dazu zählen bspw. Ereignisse über den Systemstart sowie das Herunterfahren.
- **Weitergeleitete Ereignisse**
  In dieser Kategorie werden weitergeleitete Ereignisse von anderen Systemen oder Ereignisprotokollen gesammelt.

Für die Suche nach Ereignissen, die bei der Nutzung von SharpHound entstehen, werden die protokollierten Ereignisse des Domänencontrollers sowie der Clients ausgewertet. Die Kategorie Sicherheit ist hierbei von besonderem Interesse, da dort sicherheitsrelevante Systemereignisse protokolliert werden.

3 Testumgebung und Software

3.1 Versuchsaufbau

Damit BloodHound ausreichend Daten zur Visualisierung zur Verfügung stehen, werden in einer Testumgebung zwei Server und zwei Clients in einem Domänennetzwerk erstellt, sodass die Datensammlung mit SharpHound und die anschließende Auswertung durchgeführt werden kann.

Es wird die Domäne **testlab-bloodhound.de** bereitgestellt. Der Aufbau der Testumgebung ist in Abbildung 2 dargestellt.

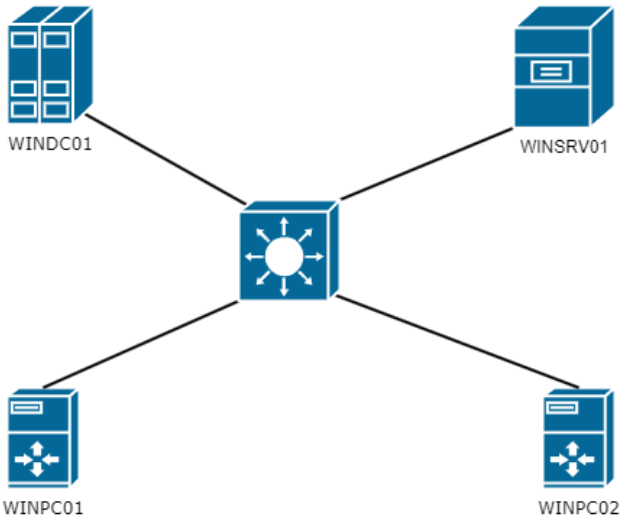

Abbildung 2 – Aufbau Testumgebung

3.2 Hardware

Für die Bereitstellung der benötigten Maschinen kann auf einen Server aus dem Arbeitsumfeld zugegriffen werden. Der Server befindet sich in einem abgeschotteten Netzwerk und kann zur Bereitstellung einer Testumgebung verwendet werden. Auf der Hardware ist der Hypervisor VMWare ESXi [21] in der Version 6.5 installiert, der zur Bereitstellung der virtuellen Infrastruktur verwendet wird.

Der Server hat folgende Ausstattung:

- 12 CPUs Intel Xeon CPU E5-2620
- 48 GB RAM
- 800 GB Plattenspeicher

3.3 Benutzerkonten

Es werden drei Domänenkonten erstellt. Diese sind in Tabelle 1 aufgelistet.

Benutzerkonto	Benutzername	Berechtigungen
Anton Admin	aadmin	Domänenadministrator
Alice	alice	Domänenbenutzer ohne lokale Adminrechte
Bob	bob	Domänenbenutzer mit lokalen Adminrechten auf WINPC01 und WINPC02

Tabelle 1 - Benutzerkonten

3.4 Virtuelle Maschinen

Im Folgenden werden die virtuellen Server und Clients beschrieben.

3.4.1 WINDC01

Der Server WINDC01 in der Version Windows Server 2022 ist der Domänencontroller und stellt die Domäne *testlab-bloodhound.de* bereit. Nach der Installation des Betriebssystems wird hierfür die Rolle AD DS installiert und der Server nach Abschluss zum Domänencontroller heraufgestuft. Zwingende Voraussetzung für die Aktivierung der Domänendienste ist ein DNS-Server. Die DNS-Rolle wird zur Vereinfachung auch auf dem Domänencontroller aktiviert und

konfiguriert.

Der Domänencontroller ist für die Organisation der gesamten Domänenstruktur zuständig. Er verwaltet die Ressourcen im Netzwerk und wird die LDAP-Abfragen während der Datensammlung beantworten.

3.4.2 WINSRV01

Der zweite Server WINSRV01 wird ebenfalls in der Version Windows Server 2022 bereitgestellt und stellt als Anwendungs-/Fileserver Dokumente bereit. Nach der Installation des Betriebssystems wird eine Ordnerfreigabe eingerichtet, die jedem Domänenrechner als Netzlaufwerk zugewiesen wird. Hierdurch sollen BloodHound weitere Daten zur Visualisierung zur Verfügung gestellt werden.

3.4.3 WINPC01

Auf dem Client WINPC01 wurde ein Windows 10 Enterprise LTSC 2021 installiert. An dem System meldet sich das Benutzerkonto Alice an. Das Konto besitzt keine lokalen Adminrechte, sodass auf diesem Client die Datensammlung als Standard-Domänenbenutzer durchgeführt wird.

3.4.4 WINPC02

Auf dem Client WINPC02 wurde ein Windows 10 Enterprise LTSC 2021 installiert. An dem System meldet sich der Domänenbenutzer Bob an. Das Benutzerkonto besitzt auf beiden Clients WINPC02 und WINPC01 lokale Adminrechte, sodass auf diesem Client die Datensammlung mit höheren Privilegien durchgeführt wird.

3.5 Software

3.5.1 BloodHound und Softwarekomponenten

Auf beiden Clients wurde das Github Repository von BloodHound in der Version 4.3.1 heruntergeladen. BloodHound und die benötigten Komponenten werden

anhand der Installationsanleitung [22] konfiguriert. Dabei werden jeweils folgende Schritte abgearbeitet:

- Installation der Java OpenJDK 11
- Installation und Konfiguration der Datenbank Neo4j
- Datenbankanbindung BloodHound

3.5.2 Windows Ereignisanzeige

Die Ereignisanzeige ist Bestandteil eines jeden Windows Betriebssystems und muss nicht separat installiert werden. Durch Gruppenrichtlinien kann aber das Protokollierungsverhalten beeinflusst werden, sodass weitere Ereignisse erfasst oder detailliertere Informationen protokolliert werden. Bei der Recherche wurde dieser Blogeintrag [23] gefunden, in dem die Anpassung eines Gruppenrichtlinienobjekts beschrieben wird, wonach ein weiteres Ereignis erfasst werden kann, welches durch die Datensammlung ausgelöst wird. Dieses Ereignis protokolliert die Überprüfung eines Netzwerkobjekts, um festzustellen, ob dem Client der gewünschte Zugriff gewährt werden kann.[24]

Das Gruppenrichtlinienobjekt „*Objektzugriffsversuche überwachen*“ wird bei „Erfolg“ aktiviert. Das Objekt befindet sich im Pfad:

Computerkonfiguration → Richtlinien → Windows-Einstellungen → Sicherheitseinstellungen → Lokale Richtlinien → Überwachungsrichtlinie

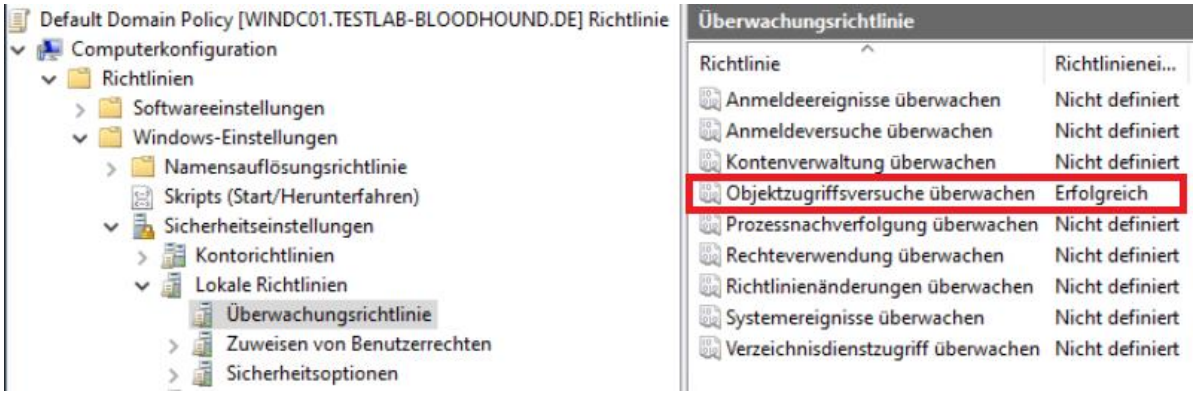

Abbildung 3 - Anpassung der Gruppenrichtlinie

Die Änderungen werden in der Default Domain Policy durchgeführt. Damit diese wirksam werden, muss die Gruppenrichtlinie entweder erzwungen oder es muss ein Systemneustart durchgeführt werden.

4 Informationsgewinnung mit SharpHound

4.1 Voraussetzungen

Für die Anwendung von SharpHound müssen die nachfolgend beschriebenen Voraussetzungen bestehen, die in der Testumgebung gegeben sind.

4.1.1 Zugangsdaten

Damit der Datensammler SharpHound im Netzwerk ausgeführt werden kann, werden gültige Zugangsdaten eines Domänenbenutzers benötigt. Ein Angreifer müsste in den Besitz solcher gelangen oder sich durch Infiltration eines Zielsystems Zugriff verschaffen. Hier sind verschiedene Angriffsszenarien denkbar, wie beispielsweise der Versand von schadhaften E-Mails zur Installation einer Remote Shell oder die Beschaffung von Zugangsdaten durch Social Engineering.

In der Projektarbeit wird die Datensammlung mit gültigen Zugangsdaten der Domänenbenutzer Alice und Bob ausgeführt.

4.1.2 Berechtigungen

Für die Anwendung einzelner Flags werden administrative Berechtigungen benötigt. Damit alle Flags gesetzt und so mögliche Unterschiede in den Ergebnissen festzustellen sind, besitzen die Benutzerkonten Alice und Bob wie beschrieben unterschiedliche Zugriffsrechte.

4.1.3 Virenscanner

Aktuelle Virenscanner sollten die Ausführung von SharpHound erkennen, ebenso zeigen gängige aktuelle Browser beim Herunterladen der Software eine Warnung an. Dies kann jedoch nicht als vollständiger Schutz angesehen werden, da BloodHound eine Open-Source Software ist und der Code verändert und das Programm neu kompiliert werden kann. Auch Ausführungen außerhalb des von Virenscannern überwachten Speicherbereichs sind denkbar. Zu Testzwecken kann eine Ausnahme für die Software definiert werden.[25]

4.2 Flags

Flags sind optionale Filter, die zur Steuerung und Konfiguration der Informationsbeschaffung gesetzt werden können. Allgemein werden Flags direkt nach dem Aufruf der ausführbaren Datei in einer Kommandozeile mit einem doppelten Minus (--) angegeben. Die Syntax lautet:

SharpHound.exe --FLAG

Eine Sammelmethode vereint mehrere Flags und führt diese in einem Durchlauf aus. Durch Setzen der Flag „-c“ kann die jeweilige Sammelmethode angegeben werden:

SharpHound.exe -c SAMMELMETHODE

Es lassen sich auch mehrere einzelne Flags aus den nachfolgend beschriebenen Kategorien kombinieren. Die Flags werden dann hintereinander gesetzt:

SharpHound.exe --FLAG --FLAG

4.2.1 Kategorien

Flags können die Informationsgewinnung durch SharpHound für folgende Kategorien beeinflussen:

- **Aufzählungsoptionen**:
  Mit Flags dieser Kategorie wird festgelegt, welche Informationen von SharpHound gesammelt werden sollen. Es können entweder einzelne Flags und so nur bestimmte Informationen abgefragt werden, oder es lassen sich Sammelmethoden auswählen, die mehrere Flags in einem Sammellauf kombinieren.

- **Ausgabeoptionen**:
  Die Ergebnisse, die in JSON-Dateien gespeichert werden, lassen sich anpassen. Es kann der Ausgabepfad und Ausgabename,

Verschlüsselungsoptionen und das Format der JSON-Dateien angepasst werden.

- **Schleifenoptionen**:
  In größeren Netzwerkumgebungen ist es denkbar, dass zum Ausführungszeitpunkt der Datensammlung nicht alle Systeme eingeschaltet und erreichbar sind. Hierfür besteht die Möglichkeit die Informationssammlung in Schleifendurchläufen (Loop) zu konfigurieren, sodass die Sammlung über einen längeren, definierten Zeitraum durchgeführt wird. Hierdurch lassen sich unter Umständen mehr Rechner im Netzwerk erreichen und der Informationsgehalt wächst.

- **Verbindungsoptionen**
  Mithilfe von Flags dieser Kategorie können Verbindungsoptionen zum Domänencontroller angepasst werden. Es lassen sich Ports und Benutzerdaten zur Authentifizierung verändern.

- **Performanceoptionen**
  Performance-Flags sind dafür da, um Portchecks und Unterbrechungen von SharpHound zu konfigurieren. Aus Sicht eines Angreifers kann das Laufverhalten angepasst werden, sodass die Datensammlung ggf. unauffälliger durchgeführt wird, wenn bspw. auf massenhafte Portchecks verzichtet wird oder Pausenzeiten gesetzt werden, um weniger Aufmerksamkeit zu erlangen.

- **Cacheoptionen**
  Mit Flags der letzten Kategorie kann Cache-Verhalten verändert werden.

Die folgende Abbildung 4 gibt einen Überblick über die Sammelmethoden, welche Flags durch die jeweilige Methode genutzt oder einzeln gesetzt werden können, welches Protokoll zur Abfrage verwendet wird und welche weiteren Möglichkeiten der Beeinflussung der Performance, des Cache-Verhaltens und der Ausgabeoptionen bestehen.

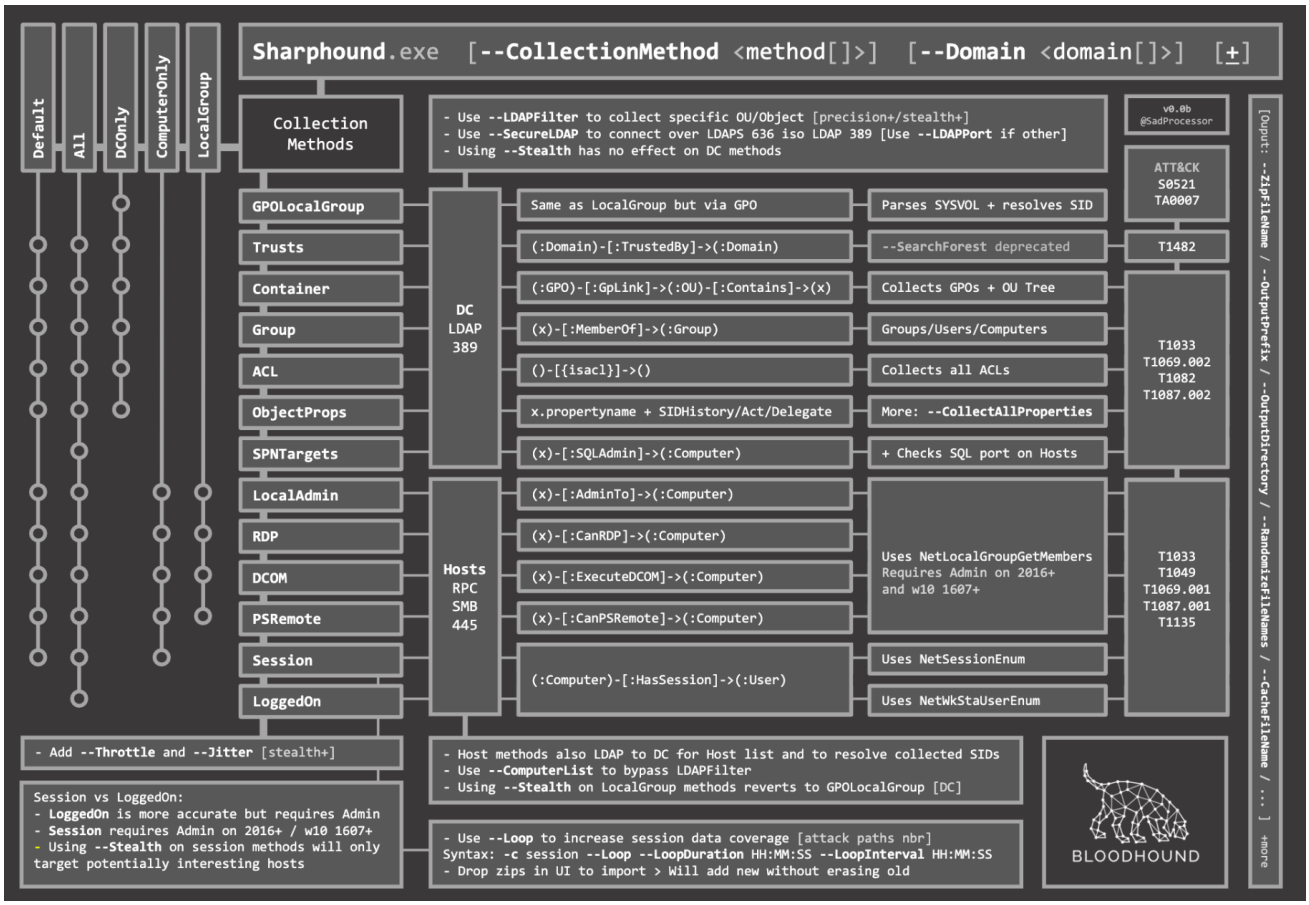

Abbildung 4 - Übersichtsmatrix SharpHound – Flags [26]

Im linken Bereich der Abbildung unterhalb von „Collection Methods“ sind alle vorhandenen Flags aufgelistet. Links davon sind die vorhandenen Sammelmethoden zu sehen. Anhand des Kreissymbols ist zu erkennen welche Flag bei der jeweiligen Sammelmethode gesetzt wird. Alle Sammelmethoden mit den jeweiligen Optionen sind dem Anhang der Tabelle A.1 zu entnehmen.

Rechts neben den Flags ist das jeweilige Protokoll zu sehen, welches für die Abfrage der Informationen verwendet wird. Für Abfragen an Domänencontroller wird LDAP standardmäßig unter Port 389 (unverschlüsselt) genutzt. Für alle Abfragen an Clients wird das Protokoll RPC und SMB über Port 445 verwendet.

Optional kann durch Verbindungsoptionen durch Setzen der Flag „--LDAPFilter“ der Suchbereich auf bestimmte Organisationseinheiten eingeschränkt wird. Der der Port zur Kommunikation genutzt wird kann durch Setzen der Flag

„--SecureLDAP" angepasst werden.

Im unteren Bereich der Abbildung 4 sind Schleifenoptionen zu erkennen.

Eine Beschreibung aller Flags, abgeleitet von der offiziellen Dokumentation von BloodHound, findet sich im Anhang unter dem Punkt A.2.

In dieser Projektarbeit werden für die Informationsgewinnung mit SharpHound keine einzelnen Flags verwendet, sondern es werden die beiden Sammelmethoden „Default“ und „All“ verwendet, sodass Flags gebündelt ausgeführt werden und die größtmögliche Informationsgewinnung gegeben ist.

5 Durchführung

5.1 Datensammlung

Die Datensammlung wird im Kontext des Domänenbenutzers Alice ohne lokale Adminrechte auf dem Client WINPC01 und im Kontext des Domänenbenutzers Bob mit lokalen Adminrechten auf beiden Clients auf dem Domänenrechner Client ausgeführt.

Für beide Benutzerkonten wird der Sammellauf mit den Flags „Default“ und „All“ durchgeführt, um so mögliche Unterschiede in den Ergebnissen sowie in den Logdateien erkennen zu können und ausreichend Daten zur Auswertung zu generieren. Zur besseren Übersichtlichkeit der Ergebnisse wird mit der Flag „--outputprefix" der Name des Ausgabeordners angepasst.

5.1.1 Ausführung ohne lokale Adminrechte

Die ersten beiden Sammelläufe werden im Kontext des Domänenbenutzers Alice auf dem Client WINPC01 ausgeführt. Alice besitzt keine administrativen Rechte auf Domänenrechnern.

Sammelmethode „Default“:

```
SharpHound.exe --outputprefix alice_default
```

Abbildung 5 - Ausführung SharpHound - Alice - Default

Sammelmethode „all“:

```
SharpHound.exe -c all --outputprefix alice_all
```

Abbildung 6 - Ausführung SharpHound - Alice - All

5.1.2 Ausführung mit lokalen Adminrechten auf Clients

Der dritte und vierte Sammellauf wird im Kontext des Domänenbenutzers Bob auf dem Client WINPC02 ausgeführt. Der Benutzer Bob besitzt administrative

Berechtigungen auf den beiden Clients WINPC01 und WINPC02.

Sammelmethode „Default“:

```
SharpHound.exe --outputprefix bob_default
```

Abbildung 7 - Ausführung SharpHound - Bob - Default

Sammelmethode „all“:

```
SharpHound.exe -c all --outputprefix bob_all
```

Abbildung 8 - Ausführung SharpHound - Bob - All

5.2 Windows Eventlogs

Vor jedem Sammellauf werden die Einträge in den Windows Ereignisprotokollen auf jedem System geleert. Alle Ereignisse, die während der Ausführung der Datensammlung protokolliert werden, werden nach Beendigung des Durchlaufs gespeichert. Dies betrifft die Logdateien auf dem Domänencontroller WINDC01 und auf den beiden Clients WINPC01 und WINPC02.

5.3 Datenimport

Nach Fertigstellung der Sammlung werden die generierten JSON-Dateien in BloodHound importiert. Nach erfolgreichem Abschluss werden Statistiken zu den gesammelten Daten angezeigt und es können erste Abfragen gestellt werden.

In der folgenden Abbildung ist zu sehen, dass der Sammellauf Daten erfasst hat. BloodHound kann beispielsweise alle Domänenadministratoren anzeigen wie in der Abbildung 9 zu erkennen:

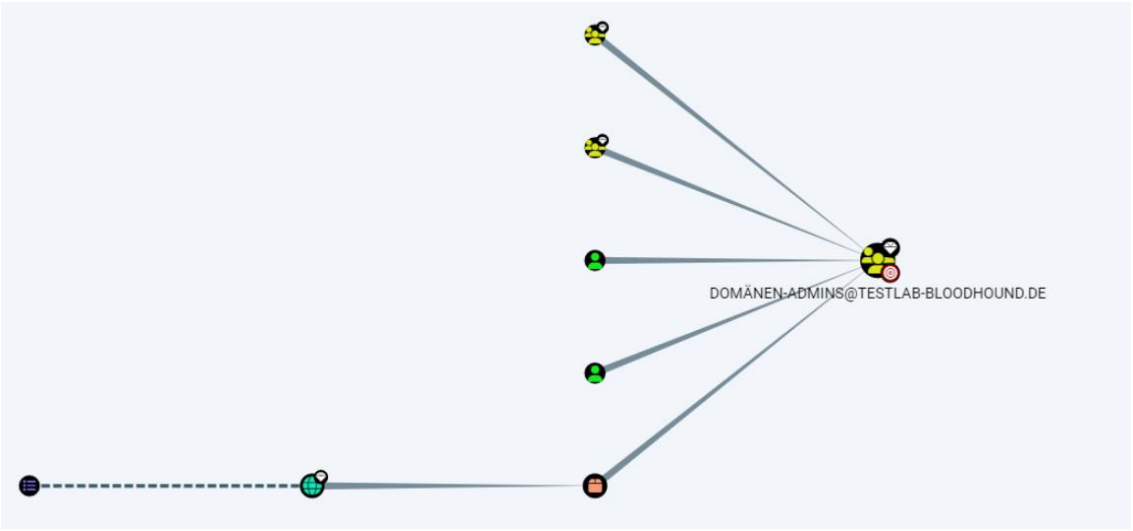

Abbildung 9 - BloodHound - Domänenadministratoren

Auch mit einer überschaubaren Anzahl an Systemen im AD werden mit der Abfrage nach „High Value Targets“ komplexere Graphen erzeugt, wie in Abbildung 10 zu sehen:

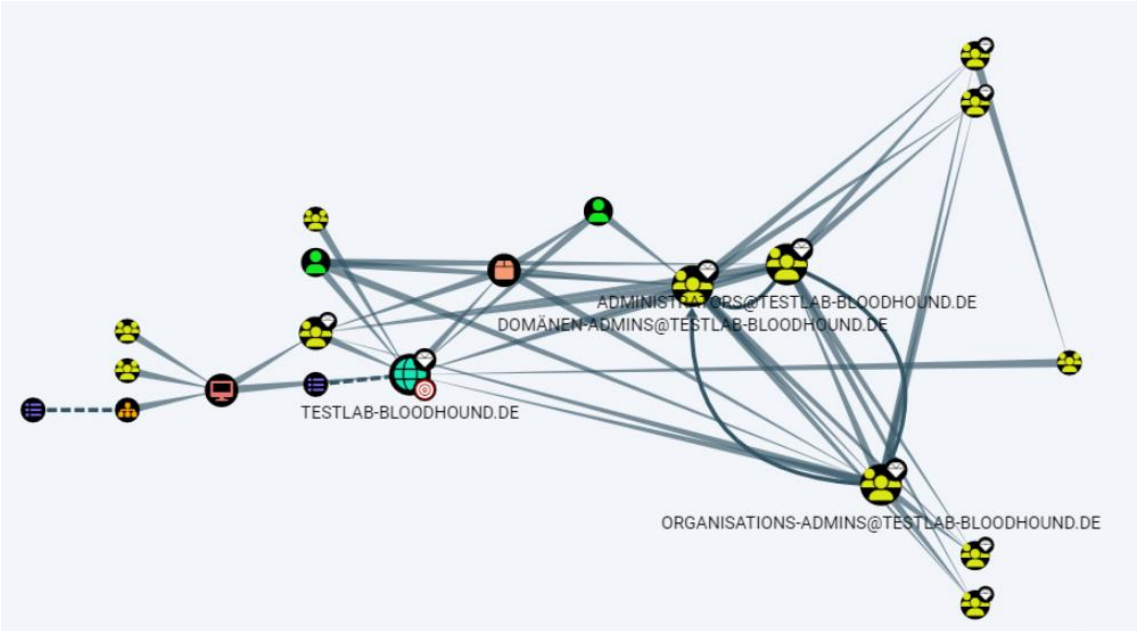

Abbildung 10 - BloodHound – High Value Targets

6 Auswertung

Während der Datensammlung konnten Ereignisse protokolliert werden, die dahingehend ausgewertet werden, welche Ereignisse der Ausführung von SharpHound zuzuordnen sind und ob sich Muster erkennen lassen.

6.1 Domänencontroller

Auf dem Domänencontroller wurden Ereignisse protokolliert, die sich der Datensammlung zuordnen lassen. Die aufgetretenen Ereignisse sind mit einer kurzen Beschreibung des jeweiligen Ereignisses der Tabelle 2 zu entnehmen. Im Anschluss werden die Ereignisse detailliert ausgewertet, die eine Erkennung von SharpHound ermöglichen. Die Auswertung gilt für beide Sammelmethoden.

Ereignis-ID	Aufgabenkategorie	Beschreibung
4624	Logon	Ein Konto wurde erfolgreich angemeldet
4634	Logoff	Ein Konto wurde abgemeldet.
4658	Other Object Access Events	Ein Handle zu einem Objekt wurde geschlossen.
4661	SAM	Ein Handle zu einem Objekt wurde angefordert.
4769	Kerberos Service Ticket Operations	Ein Kerberos-Dienstticket wurde angefordert.
4799	Security Group Management	Eine sicherheitsaktivierte lokale Gruppenmitgliedschaft wurde aufgezählt.
5140	File Share	Es wurde auf ein Netzwerkfreigabeobjekt zugegriffen.
5145	Detailed File Share	Ein Netzwerkfreigabeobjekt wurde überprüft, um zu ermitteln, ob dem Client der gewünschte Zugriff gewährt werden kann.

5156	Filtering Platform Connection	Die Windows-Filterplattform hat eine Verbindung zugelassen.
5158	Filtering Platform Connection	Die Windows-Filterplattform hat die Bindung an einen lokalen Anschluss zugelassen.
5379	User Account Management	Die Anmeldeinformationen in der Anmeldeinformationsverwaltung wurden gelesen.

Tabelle 2 - Protokollierte Ereignisse Domänencontroller

Ereignis-ID 4624 und 4634

Von dem von SharpHound ausführendem Benutzer werden kurz nach Start und während des Sammellaufs Anmeldevorgänge unter der Ereignis-ID 4624 protokolliert.

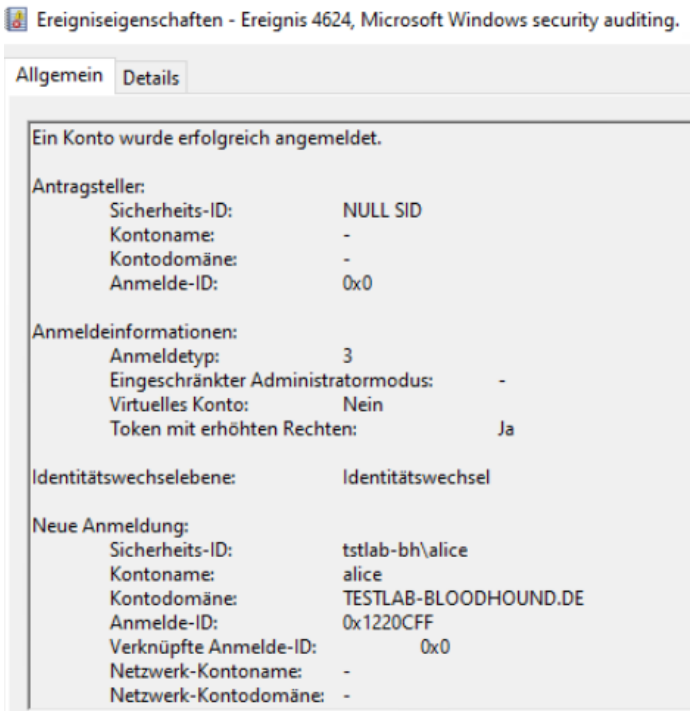

Abbildung 11 – Ereignis-ID 4624 - Anmeldung Alice

Nach Fertigstellung des Durchlaufs werden entsprechende Abmeldevorgänge unter der Ereignis-ID 4634 erfasst.

Abbildung 12 - Ereignis-ID 4634 - Abmeldung Alice

Ereignis-ID 4658 und 4661

Unter der Ereignis-ID 4661 wird protokolliert, wenn ein Handle zu einem Objekt angefordert wird. Der ausführende Benutzer von SharpHound stellt Anforderungen an die Objekttypen „SAM_DOMAIN“ und „SAM_ALIAS“ wie in Abbildung 13 und 14 zu erkennen.

Abbildung 13 - Ereignis-ID 4661 - Handle - SAM_DOMAIN

Abbildung 14 - Ereignis-ID 4661 - Handle - SAM_ALIAS

SAM steht für Security Account Manager und ist eine Datenbank, die auf jedem Windows Client vorhanden ist und für die Speicherung von Benutzerkonten verwendet wird. Der Objekttyp „SAM_DOMAIN“ gibt an, dass auf die Datenbank eines Active Directory zugegriffen wird. Der Objekttyp „SAM_ALIAS“ gibt an, dass auf eine lokale Gruppe zugegriffen wurde.[27]

Unter der Ereignis-ID 4658 wird ein Handle zu einem angeforderten Objekt wieder geschlossen.

Ereignis-ID 4769

Dieses Ereignis überwacht die Anforderung eines Kerberos-Diensttickets. Es werden mehrfach Kerberos-Diensttickets des ausführenden Benutzers angefordert.

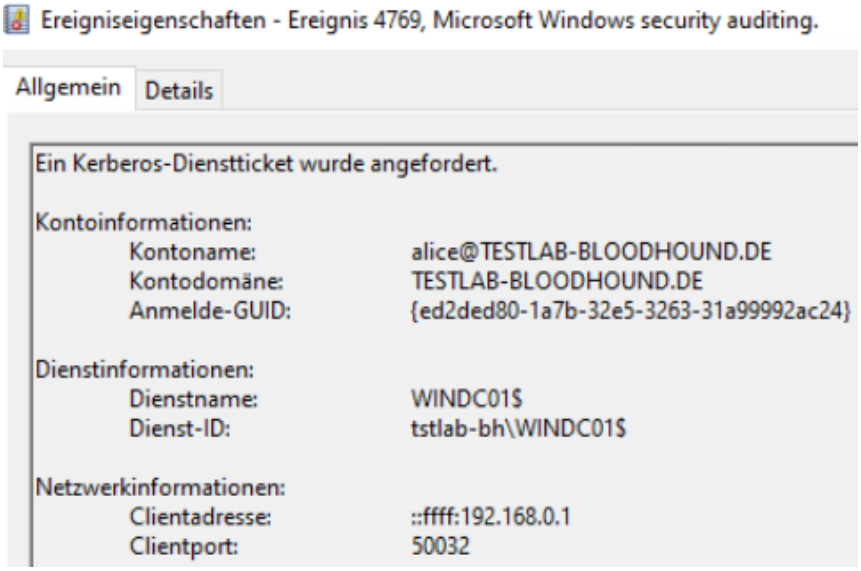

Abbildung 15 - Ereignis-ID 4769 - Anforderung Kerberos-Dienstticket

Ereignis-ID 4799

Unter der Ereignis-ID 4799 wird immer dann ein Ereignis protokolliert, wenn Mitgliedschaften einer lokalen Gruppe aufgezählt werden.[28] Für das Auslesen jeder lokalen Gruppe wird ein entsprechendes Ereignis generiert. Dies ist insbesondere auf Domänencontrollern auffällig, da lokale Gruppen von dem System selbst nicht eingesehen werden können und solche Abfragen im regulären Betrieb als unüblich einzuordnen sind.

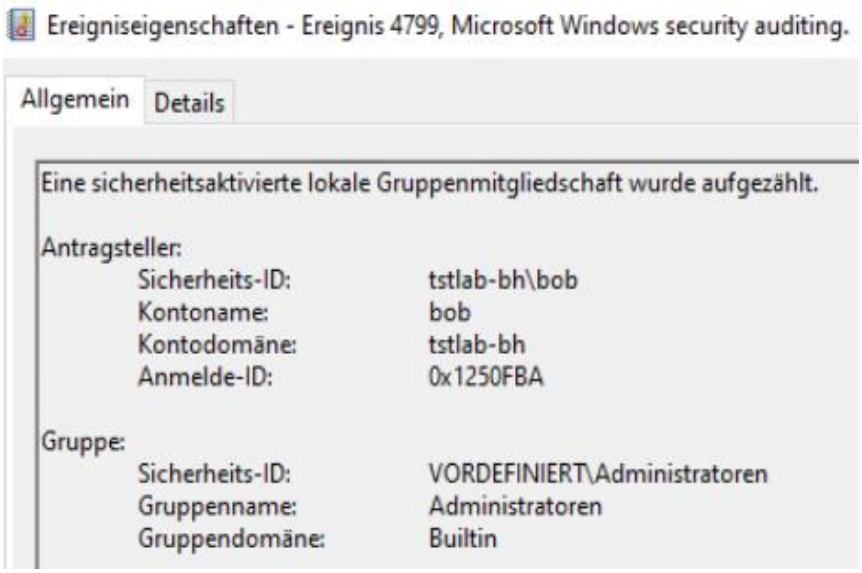

Abbildung 16 - Ereignis-ID 4799 – Aufz. lokaler Gruppenmitgliedschaften

Ereignis-ID 5140

Das Ereignis mit der ID 5140 taucht ebenfalls bei jedem Sammellauf auf und protokolliert den Zugriff auf ein Netzwerkfreigabeobjekt. Es wird auf die IPC$-Freigabe zugegriffen. Diese Freigabe wird auch als NULL-Sitzungsverbindung bezeichnet und ermöglicht das Aufzählen der Namen von Domänenkonten und Netzwerkfreigaben.[29]

Außerdem wird der Zugriff auf die SYSVOL-Freigabe des Domänencontrollers erfasst.

Die Zugriffe auf beide Freigaben sind auf Abbildung 17 und 18 zu sehen.

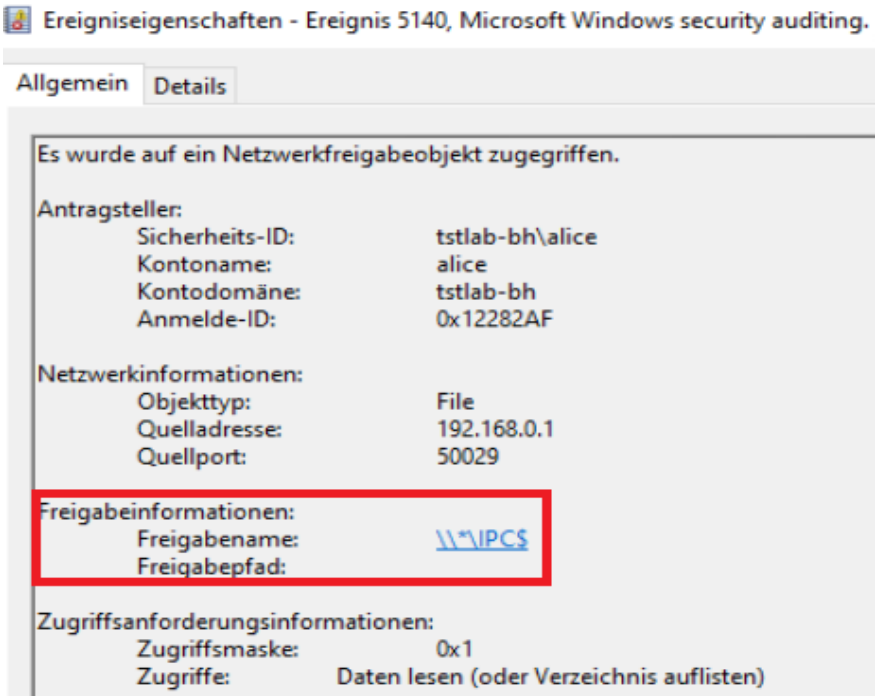

Abbildung 17 - Ereignis-ID 5140 - IPC$

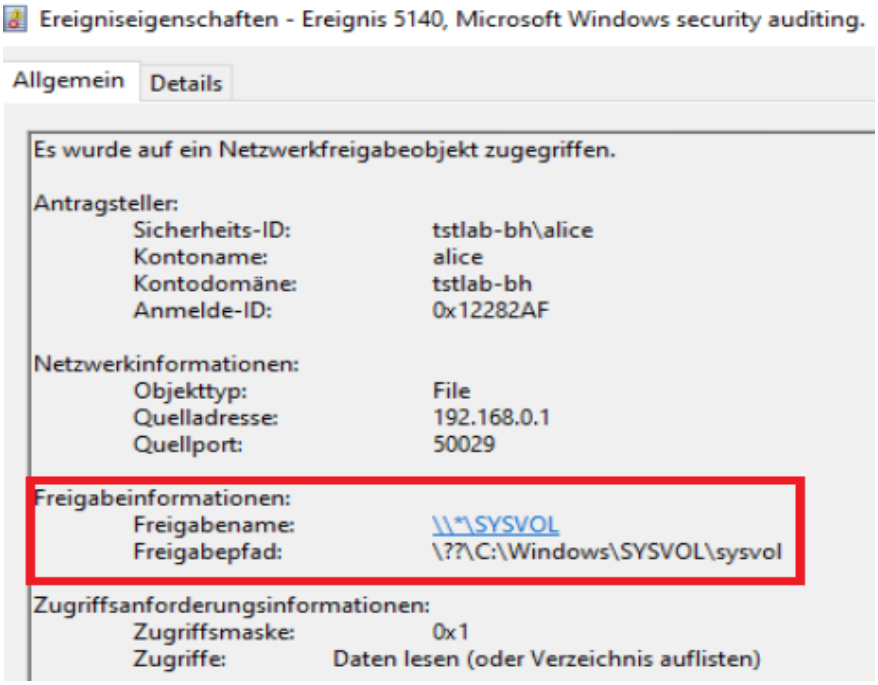

Abbildung 18 - Ereignis-ID 5140 - SYSVOL

Ereignis-ID 5145

Dieses Ereignis wird aufgrund der angepassten Konfiguration der Gruppenrichtlinie (s. Abschnitt 3.5.2) protokolliert und liefert Zusatzinformationen zum Ereignis 5140.

Es wird erneut der Zugriff auf die IPC$-Freigabe protokolliert mit Zusatzinformationen auf welchen relativen Zielnamen zugegriffen wird. Die protokollierten Zielnamen mit der jeweiligen Funktion sind in Tabelle 3 gelistet:

Relativer Zielname	Beschreibung / Funktion
srvsvc	Aufzählung von Systeminformationen
samr	Aufzählung von Domänen- und Benutzerinformationen
winreg	Aufzählung der Windows Remote Registry
wkssvc	Aufzählung von angemeldeten Benutzerkonten

Tabelle 3 - Zielnamen - IPC$-Freigabe

Auf den folgenden Abbildungen 19-22 ist der Zugriffsversuch auf die jeweiligen Zielnamen zu erkennen.

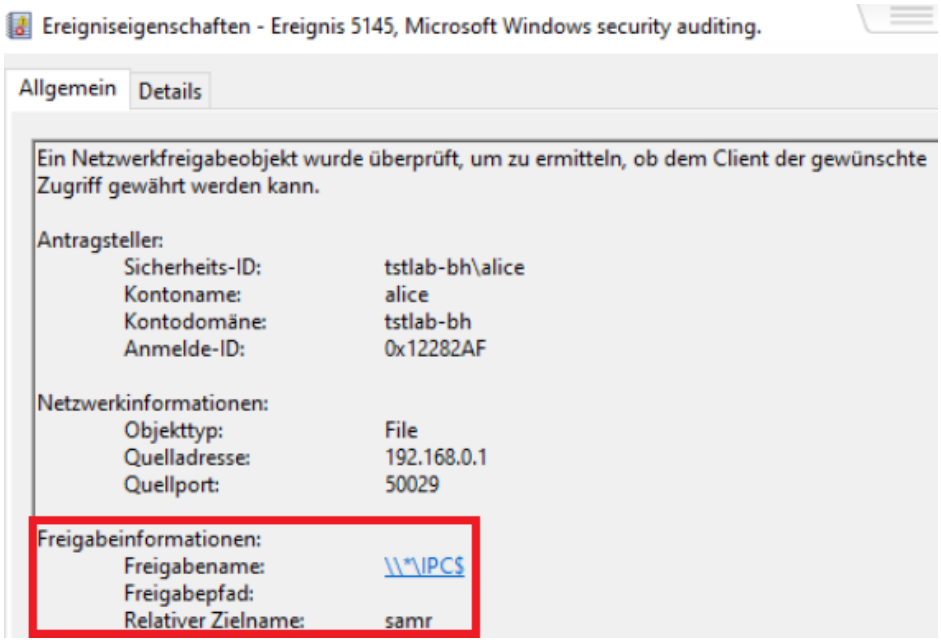

Abbildung 19 - Ereignis-ID 5145 - samr

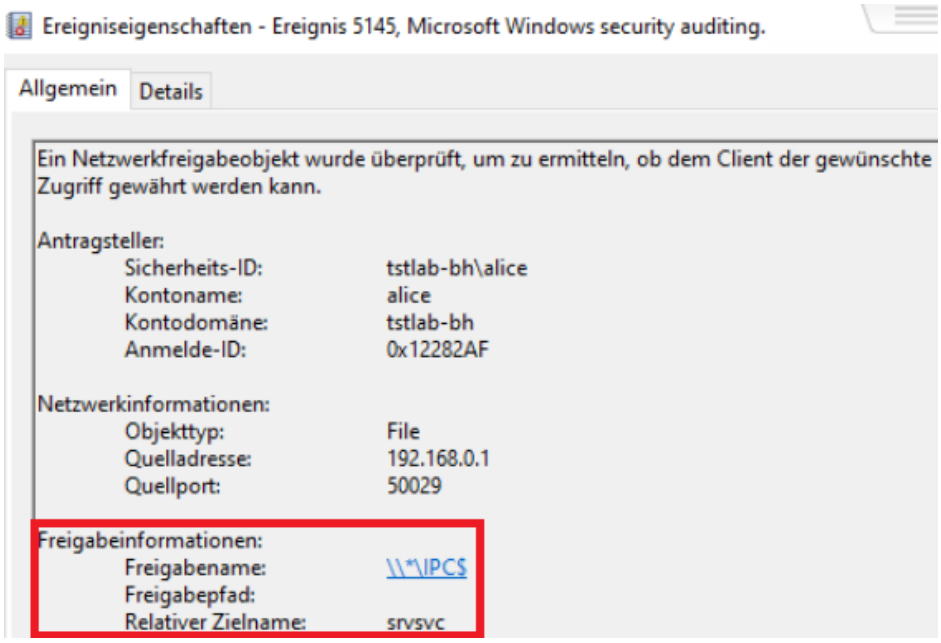

Abbildung 20 - Ereignis-ID 5145 - srvsvc

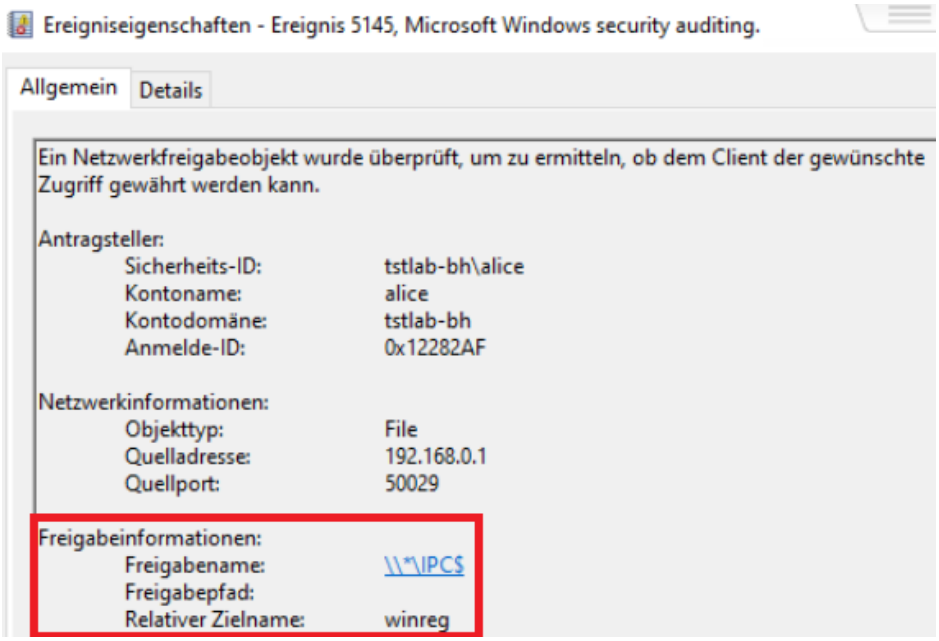

Abbildung 21 - Ereignis-ID 5145 – winreg

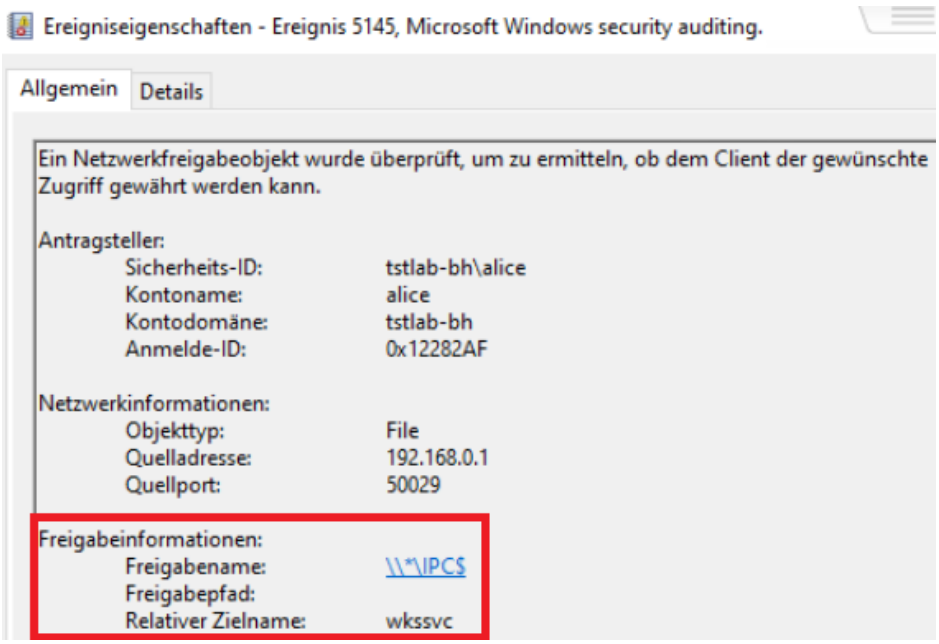

Abbildung 22 - Ereignis-ID 5145 - wkssvc

Ereignis-ID 5156 und 5158

Dieses Ereignis wird immer dann erfasst, wenn ein Verbindungsaufbau der Windows-Filterplattform zugelassen wurde und wenn die Bindung eines lokalen Anschlusses zugelassen wurde.

Die ID 5156 wird zu Beginn der Datensammlung und währenddessen mehrfach protokolliert. Es wird ein Verbindungsaufbau zu den Anwendungen *Lsass.exe* und *Svchost.exe* aufgebaut. Das Quellsystem ist der Client, auf dem SharpHound ausgeführt wird, der Zielport ist 389 (LDAP). Abbildung 23 zeigt das Ereignis.

Abbildung 23 - Ereignis-ID 5156 - lsass.exe - Port 389

Lsass steht für Local Security Authority Subsystem Service. Dieser Prozess prüft u.a. die Gültigkeit von Benutzeranmeldungen und verarbeitet Passwortänderungen. Der Lsass-Prozess authentifiziert Benutzerkonten für den Winlogon-Dienst und erstellt Zugriffstoken, die wiederum von anderen Prozessen, die gestartet werden, übernommen werden.[30]

Svchost ist ein gemeinsamer Dienstprozess, der zum Laden und Starten von DLL-Dateien genutzt wird, die eine effiziente Ausführung von Windows-Prozessen unterstützen.[31]

Im Weiteren wird unter diesem Ereignis auch der Verbindungsaufbau zum Zielport 445 (RPC over SMB) protokolliert, wenn Abfragen an Clients und nicht an das AD gestellt werden. (s. Abb. 24)

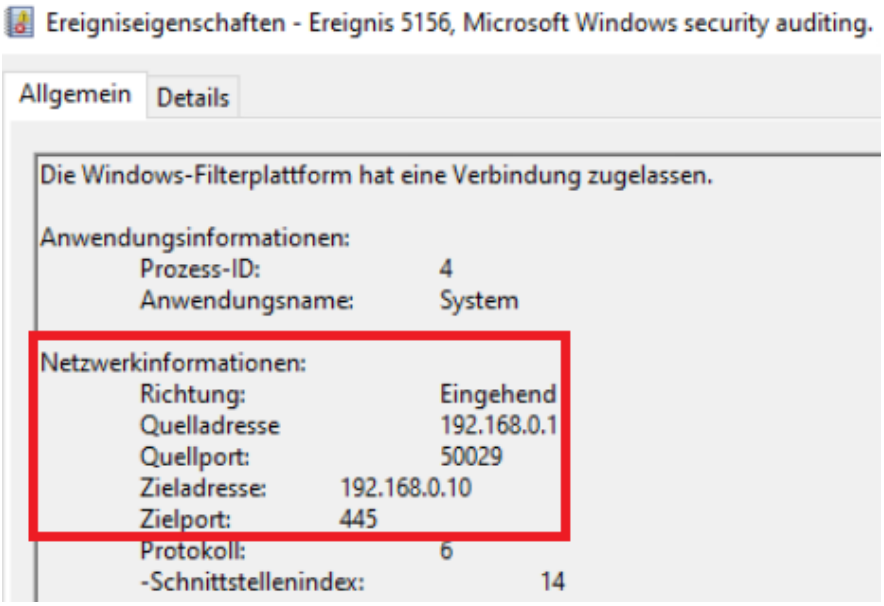

Abbildung 24 - Ereignis-ID 5156 - Port 445

6.2 WINPC01 / WINPC02

Auf den beiden Clients konnten ebenfalls Ereignisse während der Datensammlung protokolliert werden. Die Ereignisse gleichen sich mit denen auf dem Domänencontroller, mit Ausnahme dieser die sich explizit auf das AD unter Port 389 beziehen.

Die aufgetretenen Ereignisse sind mit ihrer Ereignis-ID, der zugeordneten Aufgabenkategorie, einer kurzen Beschreibung und der Angabe des Systems, auf dem das jeweilige Ereignis protokolliert wurde, der Tabelle 4 zu entnehmen.

Ereignis-ID	Aufgabenkategorie	Beschreibung	WINPC01	WINPC02
4624	Logon	Ein Konto wurde erfolgreich angemeldet	X	X
4634	Logoff	Ein Konto wurde erfolgreich abgemeldet	X	X
4799	Security Group Management	Eine sicherheitsaktivierte lokale Gruppenmitgliedschaft wurde aufgezählt	X	

5140	File Share	Es wurde auf ein Netzwerkfreigabeobjekt zugegriffen	x	x
5145	Detailed File Share	Ein Netzwerkfreigabeobjekt wurde überprüft, um zu ermitteln, ob dem Client der gewünschte Zugriff gewährt werden kann	x	x
5156	Filtering Platform Connection	Von der Windows-Filterplattform wurde eine Verbindung zugelassen	x	x

Tabelle 4 - protokollierte Ereignisse auf Clients

Es werden erneut Anmeldevorgänge (4624) und entsprechende Abmeldevorgänge (4634) des ausführenden Benutzers erfasst.

Im Weiteren werden auch auf den Clients Zugriffe auf die jeweiligen Netzwerkfreigabeobjekte unter der Ereignis-ID 5140 protokolliert.

Abbildung 25 - Ereignis-ID 5140 - IPC$-Freigabe

Unter der Ereignis-ID 5145 finden sich hierzu wieder detailliertere Informationen. Es wird, wie auf dem Domänencontroller, auf die relativen Zielnamen, die in Tabelle 3 aufgelistet sind, zugegriffen. Dies wiederholt sich auf jedem Client.

Der Unterschied bei der Ausführung von SharpHound als lokaler Administrator auf den jeweiligen Clients liegt in der Protokollierung des Ereignisses mit der ID 4799. Wenn der ausführende Benutzer administrative Berechtigungen auf dem jeweiligen Client besitzt (Bob), dann kann er die lokalen Gruppenmitgliedschaften auslesen, wie auf Abbildung 26 zu erkennen.

Abbildung 26 - Ereignis-ID 4799 - Aufzählen lokaler Gruppenmitgliedschaften

Wenn SharpHound als Domänenbenutzer ohne administrative Berechtigungen ausgeführt wird (Alice), tritt dieses Ereignis nicht auf.

6.3 Pattern

Für die Erkennung der Datensammlung mit SharpHound können folgende Muster festgehalten werden:

- Ereignis-ID 4624: Der ausführende Benutzer meldet sich auf allen erreichbaren Systemen an. Die Anmeldezahlen dieses Benutzerkontos steigen stark an. Je mehr Systeme im Netzwerk vorhanden sind, desto mehr Anmeldungen werden erfasst.
- Ereignis-ID 5156: Es werden mehrfach Verbindungen zur Anwendung Lsass.exe über Port 389 (AD) aufgebaut.
- Ereignis-ID 4799: Die lokalen Gruppenmitgliedschaften werden

ausgelesen (Wenn Zugangsdaten von lokalen Administratoren genutzt werden).

- Ereignis-ID 5140 und insbesondere 5145 (nach Aktivierung der GPO): Es wird mehrfach auf die IPC$- und die SYSVOL-Freigabe (DC) zugegriffen. Der Zugriff wird auf die relativen Zielnamen srvsvc, samr, winreg und wkssvc protokolliert, da so versucht wird gespeicherte Informationen abzufragen.
- Allgemein häufen sich Anfragen über Port 389 (LDAP) und 445 (RPC/SMB) während der Ausführung.

6.4 Unterschiede

In den Ereignisprotokollen konnten keine Unterschiede zwischen den Sammelmethoden „Default“ und „All“ und den protokollierten Ereignissen festgestellt werden. Es besteht aber ein Unterschied in den erlangten Informationen. Das Auslesen von Sitzungsdaten führte im Sammelmodus „Default“ zu keinen Ergebnissen, im Sammelmodus „All“ wurden jedoch Sitzungsdaten gefunden. Dies liegt daran, dass im Default-Modus die Flag „Session“ genutzt wird, worüber die Funktion „NetSessionEnum“ aufgerufen wird. Seit der Betriebssystem Version Windows 10 Build 1607 und Windows Server 2019 kann dieses Attribut jedoch nicht mehr von jedem Domänenbenutzer ausgelesen werden, so wie es in früheren Windows Versionen möglich war. Es können folglich in BloodHound keine Informationen über Sessions angezeigt werden. Abbildung 27 zeigt die Ansicht in BloodHound:

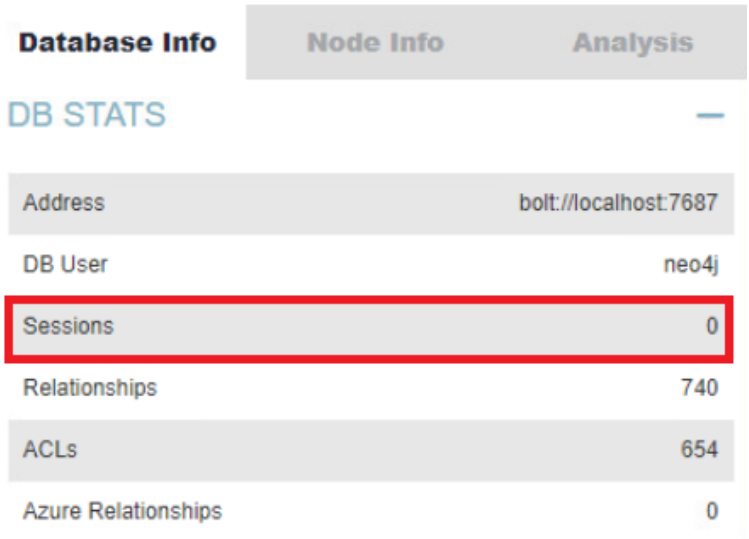

Abbildung 27 - BloodHound - Alice - keine Sessions

Wird hingegen die Sammelmethode All ausgewählt, so wird die Flag „LoggedOn“ gesetzt und es wird die Funktion „NetWkstaUserEnum“ sowie die Remote Registry verwendet. Die Remote Registry kann von jedem Domänenbenutzer ausgelesen werden, es werden keine expliziten Berechtigungen benötigt. BloodHound kann in dieser Sammelmethode auch Sitzungsinformationen anzeigen:

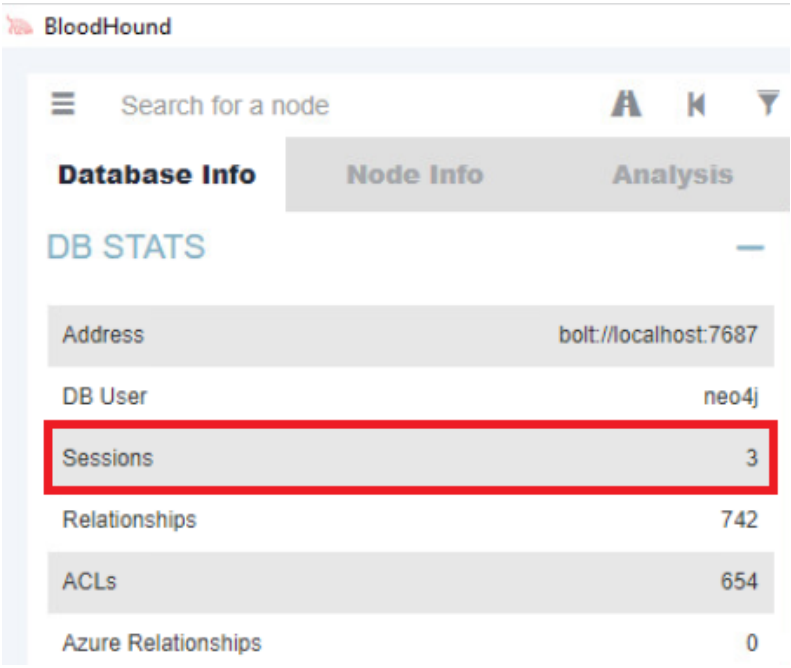

Abbildung 28 - BloodHound - Bob - Sessions

Ein weiterer Unterschied konnte zwischen der Ausführung im Kontext von Alice und Bob festgestellt werden. Wenn zur Datensammlung Zugangsdaten mit administrativen Rechten verwendet werden, wie beispielsweise von einem Client-Administrator im Netzwerk, so werden auf jedem System die lokalen Gruppenmitgliedschaften ausgelesen, auf denen der Benutzer lokaler Admin ist. Auf diesen Systemen wird das Ereignis mit der ID 4799 protokolliert.

Für BloodHound bedeutet dies, dass ohne administrative Berechtigungen die lokalen Berechtigungen auf dem Zielsystem nicht ausgelesen und keine Beziehungen von Benutzerkonten auf diesen Clients hergestellt werden können. Auf Abbildung 29 ist zu erkennen, dass BloodHound nach dem Sammellauf im Kontext von Alice keine Informationen über Berechtigungen anzeigt:

Abbildung 29 - BloodHound - Alice - keine lokale Berechtigungen

Nach der Ausführung im Kontext von Bob hingegen werden Berechtigungen angezeigt, wie in Abbildung 30 zu erkennen:

Abbildung 30 - BloodHound - Bob - lokale Berechtigungen

7 Zusammenfassung und Ausblick

Die Ausführung der Datensammlung von BloodHound mit dem Datensammler SharpHound kann forensisch nachgewiesen werden. Es treten sowohl auf dem Domänencontroller als auch auf den Clients wiederkehrende Ereignisse auf, die in den Ereignisprotokollen erfasst werden. Es konnte ein Muster erkannt werden, welches für den Nachweis der Nutzung von SharpHound als Pattern unterstützend genutzt werden kann.

Es konnten außerdem Unterschiede in den Ereignissen und Ergebnissen festgestellt werden, je nach vorhandenen Berechtigungen des ausführenden Benutzers.

Um eine Erkennung zu ermöglichen und eine forensische Auswertung auch nach der Datensammlung noch erkennen zu können, sollten die entsprechenden Protokolle der Systeme gespeichert und vorgehalten werden. Eine manuelle Auswertung ist zu Testzwecken in einer Testumgebung möglich, in Produktivumgebungen empfiehlt sich aber die zentrale Auslagerung, Speicherung und Überwachung von Protokollen, da im normalen Systembetrieb deutlich mehr Systemereignisse erfasst werden, die eine Erkennung erschweren. Eine mögliche Auslagerung der Logs könnte mit dem ELK-Stack durchgeführt werden.

Es könnte außerdem untersucht werden, ob im Active Directory weitere Objekte als eine Art Honeypot erstellt werden können, bei denen eine gesonderte Überwachung stattfindet. Wenn solche Objekte dann abgefragt werden, kann eine darauffolgende Aktion konfiguriert werden, die eine Erkennung ermöglicht.

Im Weiteren kann auch die Konfiguration von weiteren Gruppenrichtlinienobjekten geprüft, die eine Erkennung von SharpHound unterstützen.

Neben der Untersuchung von BloodHound ist auch ein Vergleich der Ereignisprotokolle bei Nutzung von weiteren Enumerationstools denkbar. Im Ergebnis könnten Pattern verschiedener Enumerationstools verglichen werden.

Die Projektarbeit hat einen Eindruck über die Möglichkeiten vermittelt, die sich bei der Visualisierung der Daten durch BloodHound zur Detektion von Angriffspfaden in Active Directory Umgebungen geben. Es werden teils komplexe Beziehungen aufgedeckt, dessen man sich oftmals nicht bewusst ist und deren Ausmaß man sonst nicht erkennt. Es ist jedem Administrator eines Active Directory zu empfehlen die Software in der eigenen Umgebung zur Analyse von Angriffspfaden zu nutzen, sodass ein Bewusstsein dafür geschaffen wird, was ein Angreifer für Informationen einsehen kann und die Vektoren eliminiert werden können.

8 Literaturverzeichnis

1. **BloodHound: Six Degrees of Domain Admin**
   https://github.com/BloodHoundAD/BloodHound
2. **Active Directory: Der Verzeichnisdienst von Windows Server kurz vorgestellt**
   https://www.wintotal.de/active-directory/
3. **Marktanteile Betriebssysteme weltweit**
   https://de.statista.com/statistik/daten/studie/157902/umfrage/marktanteil-der-genutzten-betriebssysteme-weltweit-seit-2009/
4. **OpenLDAP**
   https://www.openldap.org/
5. **Novell eDirectory**
   https://www.elektronik-kompendium.de/sites/net/0905031.htm
6. **FreeIPA**
   https://www.freeipa.org/page/Main_Page
7. **Enumeration Cyber Security**
   https://crashtest-security.com/enumeration-cyber-security/
8. **PowerView**
   https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1
9. **ADRecon**
   https://github.com/sense-of-security/ADRecon
10. **Group3r**
    https://github.com/Group3r/Group3r
11. **Red Canary Threat Report 2022 – BloodHound auf Platz 9 der erfassten Bedrohungen**
    https://redcanary.com/threat-detection-report/threats/
12. **Neo4j**
    https://neo4j.com/
13. **Kerberroasting-Angriffe**
    https://www.scip.ch/?labs.20181011

14. **Lexikon Neo4j**
https://www.datenbanken-verstehen.de/lexikon/neo4j/
15. **Cypher – Neo4j**
https://neo4j.com/developer/cypher/
16. **Lightweight Directory Access Protocol**
https://www.ip-insider.de/was-ist-ldap-lightweight-directory-access-protocol-a-581204/
17. **RFC Lightweight Directory Access Protocol**
https://datatracker.ietf.org/doc/html/rfc4511
18. **Microsoft Remote Procedure Call**
https://www.akamai.com/de/blog/security-research/msrpc-security-mechanisms
19. **Elektronik Kompendium – Server Message Block**
https://www.elektronik-kompendium.de/sites/net/2101131.htm
20. **JavaScript Object Notation**
https://www.json.org/json-de.html
21. **VMWare ESXi**
https://www.vmware.com/products/esxi-and-esx.html
22. **Installationsdokumentation BloodHound und Komponenten**
https://bloodhound.readthedocs.io/en/latest/installation/windows.html
23. **Blogeintrag Linkedin – Detecting BloodHound**
https://www.linkedin.com/pulse/detecting-bloodhound-sharphound-tool-threat-hunting-samanta-santos?trk=portfolio_article-card_title
24. **Windows Ereignis ID 5145**
https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-5145
25. **SharpHound Antivirus**
https://bloodhound.readthedocs.io/en/latest/data-collection/sharphound.html#sharphound-vs-antivirus
26. **Übersichtsgrafik SharpHound**
https://bloodhound.readthedocs.io/en/latest/data-collection/sharphound-all-flags.html
https://twitter.com/SadProcessor

27. **Ereignis-ID 4661**
https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4661
28. **Ereignis-ID 4799**
https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4799
29. **IPC$-Freigabe und NULL-Sitzungsverhalten**
https://learn.microsoft.com/de-de/troubleshoot/windows-server/networking/inter-process-communication-share-null-session
30. **Lsass.exe**
https://www.neuber.com/taskmanager/deutsch/prozess/lsass.exe.html
31. **Svchost.exe**
https://www.avast.com/de-de/c-what-is-svchost-file

9 Abbildungsverzeichnis

10 Tabellenverzeichnis

11 Abkürzungsverzeichnis

AD	Active Directory
ADDS	Active Directory Domain Services
OUs	Organizational Unit – Organisationseinheit
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
LDAP(S)	Lightweight Directory Access Protocol (Secure)
IETF	Internet Engineering Task Force
RFC	Request for comments
TLS	Transport Layer Security
RPC	Remote Procedure Call
SMB	Server Message Block
DC	Domänencontroller
LTSC	Long Term Servicing Channel
SAM	Security Account Manager
ELK	Elasticsearch, Logstash, Kibana
GPO	Group Policy Object – Gruppenrichtlinienobjekt

12 Anhang

A.1 Auflistung der Sammelmethoden mit den jeweiligen Flags

Sammelmethode	Gesetzte Flags
Default	Trusts Container Group ACL ObjectProps LocalAdmin RDP DCOM PSRemote Session
All	Trusts Container Group ACL ObjectProps SPNTargets LocalAdmin RDP DCOM PSRemote Session LoggedOn
DCOnly	GPOLocalGroup Trusts Container Group ACL ObjectProps

ComputerOnly	LocalAdmin RDP DCOM PSRemote Session
LocalGroup	LocalAdmin RDP DCOM PSRemote

Anhang A.2 Alle Flags von SharpHound

Aufzählungsoptionen

Sammelmethoden:

Default:

Durch das einfache Ausführen der Datei SharpHound.exe wird SharpHound im Standardmodus mit den Default Einstellungen gestartet. Es werden dann folgende Informationen eines Domänencontrollers abgefragt:

- Mitgliedschaften in Sicherheitsgruppen
- Domänenvertrauensstellungen
- Missbräuchliche Rechte an Active Directory-Objekten
- Gruppenrichtlinienlinks / Links in GPOs
- OU-Baumstruktur
- Mehrere Eigenschaften von Computer-, Gruppen- und Benutzerobjekten

Darüber hinaus werden von im Netzwerk erreichbaren Clients folgende Informationen abgefragt:

- Mitglieder der lokalen Administratoren-, Remotedesktop-, verteilten COM- und Remoteverwaltungsgruppen

- Aktive Sitzungen mit interaktiver Anmeldung

All:

Mit der Flag „all“ werden alle Flags außer „GPOLocalGroup“ ausgeführt.

DCOnly:

Es werden nur Daten vom Domänencontroller gesammelt. Alle weiteren der Domäne zugehörigen Windows Systeme werden nicht abgefragt. Es werden die gleichen Informationen wie im Standardmodus abgefragt, abgesehen von den Informationen die einen Client betreffen.

ComputerOnly:

Es werden nur Benutzersitzungen und lokale Gruppen von in der Domäne eingebundenen Windows Systemen gesammelt, keine weiteren Informationen, die mit der „DCOnly“ Methode abgefragt werden.

LocalGroup:

Listet alle Mitglieder von relevanten lokalen Gruppen von jedem erreichbaren Domänenrechner auf und vereint die nachfolgenden Flags LocalAdmin, RDP, DCOM und PSRemote. Für die genannten Flags werden jeweils Adminrechte benötigt.

Flags:

GPOLocalGroup:

Es werden Informationen vom Domänencontroller abgerufen, um Verbindungen von Gruppenrichtlinien zu Computerkonten herzustellen und so relevante lokale Gruppen jedes Windows Domänenrechners zu ermitteln. Es werden jedoch keine Clients direkt abgefragt.

Trusts:

Sammeln von Informationen über Domänenvertrauensstellungen.

Container:

Die Sammelmethode „Container“ liest die OU-Baumstruktur und Gruppenrichtlinienlinks aus.

Group:

Mit dem Flag Group werden Sicherheitsgruppenmitgliedschaften aus dem AD gesammelt.

ACL:

Es werden missbräuchliche / fehlerhafte Berechtigungen (ACLs) von AD-Objekten gesammelt.

ObjectProps:

Sammelt Objekteigenschaften, wann die letzte Anmeldung stattgefunden (LastLogon) und wann das Passwort zuletzt geändert wurde (PwdLastSet).

LocalAdmin:

Sammelt alle Mitglieder der lokalen Gruppe der Administratoren von jedem erreichbaren Windows Domänenrechner.

RDP:

Sammelt alle Mitglieder der lokalen Gruppe „Remotedesktopbenutzer“ von jedem erreichbaren Windows Domänenrechner.

DCOM:

Liest die Mitglieder der lokalen Gruppe „Distributed COM-Benutzer“ von jedem erreichbaren Windows Domänenrechner aus.

PSRemote:

Liest die Mitglieder der lokalen Gruppe „Remoteverwaltungsbenutzer“ von jedem erreichbaren Windows Domänenrechner aus.

Session:

Erfassung von Benutzersitzungen. Die Flag „Session“ benötigt ab der Betriebssystemversion Windows Server 2016 und Windows 10 und neuer Adminberechtigungen.

LoggedOn:

Sammeln von aktiven Benutzersitzungen auf Systemen. Es werden lokale Administratorrechte benötigt, vergleichbar mit „Session“, listet jedoch detailliertere Informationen auf.

Domain:

Mit dem Flag „-d“ kann die Domäne angegeben werden, über die Informationen gesammelt werden sollen. Voraussetzung ist eine funktionierende DNS-Auflösung in der Domäne.

Stealth:

Mit dem Filter „--Stealth" wird die Datensammlung so verändert, dass nur Systeme abgefragt werden, bei denen relevante Daten zu erwarten sind.

ComputerFile:

Mit diesem Parameter können Rechnernamen oder IP-Adressen mitgegeben werden, über die Informationen gesammelt werden sollen.

SearchBase:

SharpHound muss nicht das gesamte AD auslesen. Mit „--SearchBase“ kann eine bestimmte Organisationseinheit (OU) angegeben werden, um so die Ergebnisse zu limitieren.

LDAPFilter:

Es werden nur Informationen von Objekten gesammelt, die dem übergebenen LDAP-Filter entsprechen.

ExcludeDomainControllers:

Domänencontroller werden mit dem Filter bei der Informationssuche ausgeschlossen. Das Risiko der Entdeckung sinkt hierdurch.

RealDNSName:

Der DNS-Suffix kann fest angegeben werden und kann dann genutzt werden, wenn das DNS nicht im AD integriert ist und es mehr als eine mögliche Auflösung von Hostnamen gibt.

OverrideUserName:

Wenn SharpHound mit dem Parameter „runas“ als anderer Benutzer ausgeführt wird, kann der Benutzername zur Authentifizierung mitgegeben werden.

CollectAllProperties:

Erfassung von allen LDAP-Eigenschaften.

WindowsOnly:

Es werden nur Informationen über Windows Betriebssysteme gesammelt.

Ausgabeoptionen

OutputDirectory:

Standardmäßig werden die Ergebnisse von SharpHound in dem Verzeichnis abgelegt, in dem SharpHound gestartet wurde. Mit --OutputDirectory kann auch ein anderes Zielverzeichnis angegeben werden.

OutputPrefix:

Hierdurch ist es möglich den Ergebnissen (JSON und ZIP) ein Präfix im Namen zu geben.

NoZip:

Die Ergebnisse in Form der JSON-Dateien werden nicht im ZIP-Format abgelegt.

EncryptZip:

Das ZIP-Archiv wird mit einem zufälligen Passwort verschlüsselt.

ZipFileName:

Mit ZipFileName kann der Name des ZIP-Archivs angepasst werden.

RandomizeFilenames:

Die Namen der Ausgabedateien werden zufällig generiert.

PrettyJson:

Die JSON-Dateien werden eingerückt, um die Lesbarkeit auf Kosten der Dateigröße zu verbessern.

DumpComputerStatus:

Falls bei dem Verbindungsaufbau zu Computern ein Fehler auftritt, werden die Fehlercodes ausgegeben.

Schleifenoptionen

Loop:

Computerabfragen können mit „--Loop" in einer Schleife ausgeführt werden.

LoopDuration:

Standardmäßig dauert ein Schleifendurchlauf 2 Stunden. Mit „--LoopDuration" kann die Zeit angepasst werden, um so bspw. ein detaillierteres Analysebild zu erfassen.

LoopInterval:.

Mit diesem Filter kann eine Pause zwischen den Schleifendurchläufen konfiguriert werden.

Verbindungsoptionen

DomainController:

Hiermit kann konkret ein Domänencontroller anhand seines Rechnernamens oder der IP angesprochen werden

LdapPort:

Falls LDAP nicht auf den Standardports lauscht (389, XXX) kann manuell ein Port angegeben werden.

SecureLdap:

Es wird LDAPS (Secure LDAP) über Port 636 genutzt anstatt Klartext LDAP-Abfragen über Port 389 .

LdapUsername:

Falls ein alternativer Benutzer für die Verbindung zum Domänencontroller mitgegeben werden muss, kann dieser mit diesem Filter angepasst werden.

LdapPassword:

Das entsprechende Passwort des LDAP-Benutzers kann so übermittelt werden.

Performanceoptionen

PortScanTimeout:

Um Benutzerinformationen und lokale Gruppen auf entfernten Computern zu sammeln, wird Port 445 (RPC over SMB) genutzt. Damit unnötige Aufrufe für nicht erreichbare Rechner vermieden werden, wird zuerst geprüft, ob der Port auf dem Zielsystem erreichbar ist. Standardmäßig wartet SharpHound 2 Sekunden auf eine Antwort. Die Zeitspanne kann durch diesen Filter angepasst werden, z.B. wenn ein sehr schnelles oder sehr langsames Netzwerk gegeben ist.

SkipPortScan:

Der Portscan (445) kann übersprungen werden. Dies kann zur Beeinträchtigung der Performance führen, da Abfragen „ins Leere“ laufen, falls das Zielsystem nicht erreichbar ist.

Throttle:

Für jede Anfrage an einen Rechner kann eine Verzögerung konfiguriert werden. Die Angabe erfolgt in Millisekunden. (Standard = 0)

Jitter:

Fügt einen prozentualen Jitter zur Drosselung hinzu.

Cacheoptionen

CacheFileName:

Um die Geschwindigkeit der Datenerfassung zu erhöhen, erstellt SharpHound eine lokale Cache-Datei. Der Name dieser Datei kann durch diese Flag angepasst werden.

NoSaveCache:

Mit „--NoSaveCache" wird keine Cache-Datei erzeugt. Die Datenerfassung wird durch Setzen der Flag folglich langsamer durchgeführt. Dies kann zur Verringerung des Risikos der Erkennung durch Antivirensoftware oder anderer Sicherheitssoftware wie EDR-Systeme beitragen.

InvalidateCache:

SharpHound wird mit Setzen dieser Flag angewiesen eine neue Cache-Datei zu erstellen.