**Datensatz:**

1. Generell:

Es herrscht großer Bedarf an verlässlichen Benchmark Datensätzen, die für die Auswertung von Intrusion Detection Systems (IDS) basierend auf einer vorhandenen Grundwahrheit verwendet werden können. Bei der Verwendung von Daten realer Infrastrukturen ist eine solche Klassifikation von Log-Einträgen in „verdächtig“ und „unbedenklich“ nicht verlässlich erstellbar. Abgesehen von dem Angriff kann häufig nicht ausnahmslos davon ausgegangen werden, dass nur benigne Aktivitäten im Netzwerk ausgeführt wurden. Reale Daten können zudem aufgrund von Datenschutz nicht ohne ausführliche Anonymisierung veröffentlicht werden. Auch die Verwendung von „Honeypots“ ist nicht trivial, da es nicht möglich ist, alle Angriffe auf diesen korrekt zu kennzeichnen. Die Aktivität dieser ähnelt zudem nicht notwendigerweise der von echten Nutzern. Auch eignen sich viele speziell für den Einsatz als Benchmark erstellte Datensätze nur für sehr eingeschränkte Verwendungszwecke.

1. Voraussetzungen (an den Datensatz)
   1. Vollständige Dokumentation: Es sollte eine klare Beschreibung der zugrundeliegenden Infrastruktur des Systems vorliegen; insbesondere, welches Szenario abgebildet wird. Informationen bzgl des Betriebssystems, der Unternehmensarchitektur usw. erlauben ein klare Definition des Umfangs der Simulation, ihrer Anwendungsmöglichkeiten und gegebener Limitationen. Eine ausführliche Dokumentation über alle Aspekte des verwendeten Datensatzes verhindert inkorrekte Interpretation der Daten.
   2. Simulierte Nutzer-Aktivitäten: Datensätze, die auf echten Daten basieren sind zwar realistischer, Simulationen erlauben aber eine perfekte Klassifikation zwischen normalem Verhalten und bösartigen Einträgen. Um synthetische Daten zu generieren, ist also ein Skript nötig, das im Voraus auf angemessenem Detail-level normale Nutzer-Aktivität auf dem System simulieren. Periodisch auftretendes Verhalten wie Cron Jobs oder auch menschliches Verhalten, welches aufgrund von Arbeitszeiten einen bestimmten täglichen oder wöchentlichen Rhythmus abbildet, verbessert die Realitätsnähe der Daten und sollte demnach abgebildet werden.
   3. Angriffe: Es ist wichtig, dass moderne Angriffe auf das System gestartet werden, die die aktuellen Möglichkeiten und Realität berücksichtigt und nicht beispielsweise veraltete Angriffsarten umfassen. Leichte Variationen der Angriffe durch veränderliche Angriffs-Parameter bei ihrer Ausführung bilden zudem eine realistischere Grundlage für die spätere Analyse.
   4. Sammeln von Log-Dateien: Die gesammelten Logs sollten nicht vor-prozessiert sein. Bei der Verwendung von synthetischen Daten ist zudem keine Anonymisierung der Daten notwendig, da „Dummy“-Dateien ohne kritischen Inhalt verwendet werden. Diverse Angriffe hinterlassen auch Spuren in Netzwerk-Logs, weshalb diese zusätzlich zu den Systems-Logs verfügbar gemacht werden sollten.
   5. Groundtruth Labels: Log-Einträge, die Spuren von Angriffen aufweisen, sollten mit eindeutigen, aussagekräftigen Labels versehen sein, um eine Auswertung der Performanz der LLMs im Bezug auf die gestellten Analyse-Aufgaben basierend auf diesem Groundtruth zu ermöglichen.
2. Mögliche Datensätze:
   1. KDD Cup 1999: vielseitige Anwendung, kritisiert als outdated & zu simpel
   2. ADFA-LD und ADFA-WD: sehr variables Nutzerverhalten (Web-Browsing usw.) aber Log-Dateien vor-prozessiert und stripped und demnach nicht repräsentativ für echte Dateien
   3. VAST Challenge 2017: keine synthetischen Daten
   4. AIT Log Data Set V2.0: Erfüllt alle gegebenen Voraussetzungen
3. Auswahl: AIT Log Data Set V2.0:
   1. Übersicht

Der AIT Log Datensatz V2.0 umfasst 8 simulierte Maschinen, die jeweils 8 Log-Dateien mit entsprechender Grundwahrheit besitzen. Diese Logs umfassen System-Logs als auch aufgezeichneten Netzwerkverkehr. Auf den Simulationen wurden zudem 10 verschiedene Angriffsschritte ausgeführt.  
Der Datensatz ähnelt dem Aufbau eines mittelgroßen Unternehmens, welches empfohlene Sicherheitsrichtlinien befolgt und gängige Open-Source Komponenten verwendet. Die Simulation normalen Nutzerverhaltens erfolgt durch „State-Machines“ und der Zuweisung zufälliger Nutzer-Rollen, um Angestellte mit ausreichender Komplexität und Detail-tiefe innerhalb des Netzwerkes darzustellen. Dieses Verhalten umfasst Periodische Muster bezüglich Arbeitszeiten über mehrere Tage.   
Die Angriffe, die im Datensatz auftauchen sind repräsentativ, umfassen moderne Exploits und sind in diverse Schritte aufzuteilen. Die Datensätze variieren, während die Angriffe mit leicht abgewandelten Variablen die gleiche Struktur aufweisen.  
Die Grundwahrheit zu 8 verschiedenen Log-Typen umfasst Ereignis-spezifische Labels, korrespondiert zu verschiedenen Angriffs-Schritten, erstellt durch ein festes Labeling Framework.  
Die Dokumentation des AIT Log Datensatzes V2.0 ist aufschlussreich und detailliert.

* 1. Layout

Das Layout des Datensatzes soll einem kleinen bis mittelgroßen Unternehmen ähneln, welches allgemein bekannte und verbreitete Sicherheitsrichtlinien befolgt, als auch gängige Open-Source Komponenten. Kleine bis mittelgroße Firmen sind zudem auch häufig Opfer von Cyber Angriffen, während oder vielleicht eben weil sie häufig kein ausreichendes Budget für umfassenden Schutz aufweisen.   
Das Testbed unterteilt sein Netzwerk in Zonen, entsprechend der gängigen Sicherheits-Richtlinien für Unternehmen. Die erste Zone umfasst das Intranet; hier existieren Linus Hosts für alle Angestellten und ein Intranet Server, auf dem eine WordPress Instanz als auch ein Samba File-Share eingerichtet ist. Die zweite Zone besteht aus dem VPN Server, dem Proxy Server als auch dem Cloud Share Server. Die dritte Zone ist das Internet mit globalem DNS, Hosts für externe Mitarbeiter, welche sich mithilfe eines VPN verbinden sowie der Angreifer Host.  
Alle Technologien, wie sie in diesem Datensatz verwendet werden, sind auch in der Realität in echten Netzwerken so vorzufinden. Die Netzwerkgröße der acht generierten Simulationen variiert ähnlich wie die Zuweisung der IP-Adressen und Host-Namen. So werden jeweils 3 bis 9 interne, Remote-, und externe Angestellte generiert, also eine Belegschaft von insgesamt 9 bis 27 Personen.

* 1. Simulierte Nutzer-Aktivitäten

Um den Daten eine zugrundeliegende Authentizität zu verleihen, wird während der kompletten Laufzeit normales Nutzerverhalten simuliert. Wird eine realistische Nutzung der Systeme durch die Angestellten nicht ausreichend abgebildet, führt dies gegebenenfalls zu nicht-repräsentativen Daten. Sollten diese Daten wiederum bei der Auswertung von IDS verwendet werden, resultiert daraus eine zu niedrige „False Positive“ Quote, denn menschliches Verhalten ist teils erratisch und kann so Maschinenzustände auslösen, die fälschlicherweise als malicious eingestuft werden könnten.   
Die synthetischen Nutzer-Aktivitäten werden durch eine Web Automation Software generiert, die in der Lage ist, auf Webseiten unter Anderem Links und Knöpfe anzuklicken, sich erfolgreich/-los anzumelden, sich für variable Zeit in einen „Ruhezustand“ zu begeben und andere Webseiten zu besuchen. Weitere simulierte Aktivitäten umfassen das Sichten und Beantworten von Mails, das Lesen oder Erstellen neuer WordPress Beiträge, das Ausführen von Befehlen aus einer vordefinierten Liste und das Aufbauen von SSH-Verbindungen. Auch Pausen zwischen der Ausführung von Verhalten wird generiert, von wenigen Sekunden bis zu mehreren Minuten an Inaktivität.  
Bezüglich der möglichen auszuführenden Befehle wird zwischen verschiedenen Rollen unterschieden, die den Angestellten ebenso zufällig zugeordnet werden können: ssh Administrator, WordPress Editor, Manager etc. Alle Passwörter der Hosts werden zufällig generiert, die Nutzernamen aus einer vordefinierten Liste ausgewählt. Jeder Nutzer erhält individuell verschiedene Arbeitszeiten, einen präferierten Internet-Browser, zufällige auf dem System vorhandene Dateien und Kontakte. Die Dateien werden ausgewählt aus zuvor generierten „Dummy-Dateien“ ohne real sensible Inhalte.

* 1. Angriffe:
     1. Simulation von Angriffen:

Die Angriffe, die auf dem Datensatz während der Laufzeit der Simulation ausgeführt werden, sind Mehrstufen-Angriffe, die dem Aufbau einer typischen Cyber Kill Chain folgen. Sie geschehen unter Verwendung herkömmlicher, und in diesem Kontext auch häufig genutzter, Penetration Testing Werkzeuge und geschehen unter der Annahme, dass die Angreifer außerhalb der aufgezeichneten Simulation unrechtmäßig an VPN Login-Daten des Netzwerks gelangt sind. Bei realen Angriffen könnte dies beispielweise durch Phishing oder kompromittierte persönliche Rechner geschehen, wobei beides keine Spuren in den aufgezeichneten Log-Dateien hinterlassen würde und hier deshalb vernachlässigt wird.

Im Folgenden soll der Aufbau eines simulierten Angriffes beschrieben werden:  
Der Angreifer verwendet seine unrechtmäßig erlangte VPN-Verbindung, um eine Remote Netzwerkverbindung mit dem VPN Server aufzubauen.   
Der erste Angriffsschritt in der Attack Chain umfasst diverse Scans des Netzwerkes. Hierfür wird das Werkzeug Nmap verwendet, um DNS und Port Scans im DMZ Netzwerk des VPN Servers durchzuführen. Hierdurch erfährt der Angreifer die Classless Inter-Domain Routing (CIDR) und kann seine Scans so auf die Hosts in der Intranet Zone ausweiten. Einer der Scans liefert eine WordPress Instanz auf dem Intranet Server, weshalb der Angreifer diesen als potenziellen Ziel des Eindringens wählt und mithilfe eines Brute Force Directory Scans mit dem Werkzeug dirb nach relevanten Dateien zu suchen. Da dies keine relevanten Dateien liefert, nutzt der Angreifer im Folgenden den WordPress Security Scan, um potenziell angreifbare installierte Versionen oder Fehlkonfigurationen von Plugins in der WordPress Instanz des Servers ausfindig zu machen. So wird er auf die Existenz des angreifbaren Plugins wpDiscuz aufmerksam, dessen Verwundbarkeit im nächsten Schritt ausgenutzt wird. Die Reconnaissance Phase ist hiermit beendet.  
Die zweite Phase umfasst das Ausnutzen des verletzlichen Plugins; insbesondere verwendet der Angreifer hier den uneingeschränkten Datei-Upload (CVE-2020-24186), wodurch er eine php webshell als backdoor hochlädt und so jegliche Kommandos mit den Berechtigungen den www\_data Nutzers ausführen kann. Der Angreifer führt so also mehrere Kommandos aus, um Informationen über den Host zu sammeln, unter Anderem Informationen über Prozesse, das Betriebssystem, Dateinamen und Verbindungen des Systems als auch den Befehls-Verlauf. Irgendwann stößt der Angreifer so auf das Password der Nutzer-Datenbank in der WordPress Konfigurations-Datei und sieht so alle Nutzernamen und deren gehashte Passwörter ein. Der Angreifer versucht im Folgenden, die gehashten Psaswörter durch eine Liste von häufig genutzten Passwörtern zu knacken. Dies geschieht entweder offline, was hier also durch simuliertes „Nichts-Tun“ dargestellt wird, während der Angreifer diesen Prozess auf der eigenen Maschine ausführt, oder Online, insbesondere durch die Verwendung des Werkzeuges John the Ripper, welches installiert und dann in Verbindung mit einer Liste an häufig verwendeten Passwörtern ausgeführt wird. Wir gehen hier davon aus, dass mindestens ein Nutzer-Passwort in der verwendeten Liste vorkommt und demnach nach bestimmter Zeit offengelegt wird. Sobald der Angreifer dieses identifiziert hat, verwendet er das kompromittierte Nutzer-Konto, um über eine interaktive Reverse-Shell Root-Berechtigungen zu erlangen. Danach führt der Angreifer diverse Befehle aus, für welche diese Privilegien erforderlich sind, wie zum Beispiel das Einsehen der Shadow Datei.  
Also dritten und finalen Schritt führt der Angreifer das DNSteal Werkzeug aus, welches sensible Daten vom File-Share in der Intranet-Zone extrahiert. Das Werkzeug startet einen Prozess, welcher Dateien mancher Verzeichnisse in base64 codiert, in Abschnitte aufteilt und sie dann als DNS Anfragen durch die Firewall zu einer vom Angreifer kontrollierten Domain im globalen DNS schickt. Die Daten werden von der malicious Domain an den Host des Angreifers weitergeleitet, dort decodiert und aufbewahrt. Um das Erkennen dieses Prozesses komplizierter zu gestalten, wurde das Werkzeug, was für die Extraktion der Dateiinhalte verwendet wird, bereits am Anfang der Simulation gestartet, was es zeitlich weit vom dem Angriff trennt.   
Die Angriffe starten in jeder Simulation erst nach circa 3 Tagen, um IDS Werkzeugen zu ermöglichen, normale System-Aktivitäten zu lernen und diese von späteren Angriffen unterscheiden zu können.

* + 1. Spuren der Angriffe in den Log-Dateien

Der Directory Scan in der Reconnaissance Phase besteht aus mehreren tausend Anfragen, die innerhalb kürzester Zeit an den Web Server geschickt und im Apache Access Log verzeichnet werden. Da dieser bei normalem Verhalten nur Anfragen von Nutzern bezüglich angefragter Webseiten vermerkt, ist die Erkennung hier bereits aufgrund der dramatisch höheren Belastung im Vergleich zu normalen Systemoperationen besonders leicht (von circa 500 Anfragen zu mehr als 5000). Auch in den Monitoring Logs ist die CPU Verwendung während des Angriffs-Schrittes des Password-Crackings mit John The Ripper deutlich erhöht.  
Die veränderten oder abgewandelten Schritte jedes Angriffs haben einen Einfluss auf die jeweilige Erkennbarkeit, was das Ziel verfolgt, eine möglichst robuste und solide Simulation verschiedener Angriffe zu bieten.

* 1. Labeling

Die korrekte Zuweisung von Labels zwischen Log-Ereignissen, die normalem Nutzerverhalten entspringen im Vergleich zu diesen, die Hinweise auf einen Angriff liefern, ist in Benchmark Datensätzen von essenzieller Bedeutung. Dieses Labeling gestaltet sich in Netzwerk-basierten Datensätzen unter Umständen einfacher, indem dort eine Auswahl von Events mit IP-Adressen getätigt werden kann, die bekanntermaßen zu Angreifern gehören und demnach im Rahmen eines Angriffs generiert wurden. Einträge in System Log-Dateien besitzen solche Eigenschaften nicht unbedingt und sind gegebenenfalls nur aufgrund ihres gemeinsamen Auftretens überhaupt als verdächtig einzustufen. Bei Testbeds wie hier im AIT Log Datensatz V2.0 gestaltet sich dies noch komplizierter, da IP-Adressen des Angreifers zudem variabel sein können, oder im Falle eines kompromittierten Hosts die IP-Adresse an sich kein ausreichendes Kriterium für malicious Verhalten darstellt. Von einem solchen Host könnten gleichzeitig also benigne als auch maliziöse Verhaltensweisen ausgehen.  
Um 15 Labels bezüglich verschiedenen Angriffs-Schritten korrekt innerhalb von 8 verschiedene Log-Typen zuweisen zu können, wurden 27 Regeln entwickelt.  
Die Start- und End-Zeit eines Angriffs kann zwar Indizien auf relevante Log-Abschnitte geben, ist aber in vielen Fällen nicht ausreichend, um einzelne Log-Einträge verschiedenen Angriffs-Phasen zuzuordnen. Besonders bei langen Angriffen kann so auch benignes Verhalten False Positive als Teil eines Angriffes vermerkt werden. Labels sollten so also eher einzelnen Log-Instanzen zugeordnet werden, als ganzen Log-Ausschnitten, die einem bestimmten Zeitabschnitt entsprechen. Der für diesen Datensatz gewählte Prozess ähnelt dem Vorgang von Signature-basierten IDS; Log Ereignisse werden nach bestimmten Key-Words durchsucht und demnach als maliziös eingestuft. Manche Einträge, die nur durch ihren Kontext als verdächtig eingestuft werden können, entziehen sich dieser Analyse jedoch. Aus diesem Grund werden für jede Simulation sogenannte „Fakten“ gesammelt, die verwendet werden können, um Labeling Rule Templates auszuführen, die dann während des Labeling-Prozesses stetig mit Artefakten wie IP-Adressen, Domain Namen und Nutzernamen erweitert werden können.

* + 1. Labeling Vorgehen

Der Labeling Prozess umfasst vier Schritte. Zu Beginn werden die Log-Dateien auf die weitere Analyse vorbereitet: Von Logrotate komprimierte Logs werden entpackt und Logs, die als Binärdateien gespeichert wurden, werden in raw-Format umgewandelt. Im zweiten Schritt transformiert der Parser die System Log-Dateien von un- in semi-strukturierte Form, sodass alle Tokens individuell referenziert und alle Einträge so in eine Datenbank gelesen werden können. Der Post-Prozessor kürzt die Logs auf den zu betrachtenden Simulations-Umfang und bereitet alle Regeln vor, indem die bis jetzt extrahierten Fakten in die Rule Templates eingesetzt werden. Iteriere nun im vierten und finalen Schritt über alle gerenderten Labeling Regeln und führe Anfragen auf der Datenbank mit den geparsten Log-Dateien aus, sodass alle passenden Events das entsprechende Label erhalten.

Es existieren vier verschiedene Labeling Rule Typen, die ihre Anwendung nur in dem Fall finden, in welchem eine Einteilung allein anhand von z.B. Zeitintervallen nicht verwendet werden kann.   
Die simpelste ist die „Query“, also Anfrage. Hier werden Fakten mit spezifischen Teilen von Log-Ereignissen verknüpft und die entsprechenden Labels zugewiesen. Dies ist nur anwendbar, wenn alle relevanten Logs mit der gegebenen Query übereinstimmen und zudem keine Log-Ereignisse geliefert werden, die benign wären und trotzdem mit dieser Anfrage übereinstimmen. Eine mögliche Anwendung wäre also das Zuweisen von Labels in der Log-Datei der DNS Verbindungen, wenn der Domainname des Angreifers in der Datenbank der Fakten bekannt ist.  
Der zweite Typ ist die „Sequence Rule“, also die Sequenz-Regel. Dies ist relevant, wenn ein Log gegeben ist, dessen Ereignisse nicht alle nötigen Felder besitzt, um sie mithilfe der „Query“ klar zu klassifizieren. Hier ist es nötig, verschiedene Log-Ereignisse untereinander zu verknüpfen, um herauszufinden, ob sie mit Verhalten des Angreifers in Verbindung stehen. Wird in einem Log ein malicious Ereignis erkannt, wird der Zeitraum des Auftretens in anderen Logs untersucht und mit neu aufgedeckten „Fakten“ verknüpft – zum Beispiel der jetzt bekannten Angreifer IP.   
Die „Sub Query Rule“ als auch die „Parent Query Rule“ verfolgen ein ähnliches Konzept: In beiden Fällen dient ein als malicious eingestuftes Ereignis im Bezug auf eine Anfrage („Query“) als Auslöser, ähnliche Einträge zu betrachten. Während bei der „Parent Query“ Regel eine allgemeinere Form der als verdächtig eingestuften Anfrage gestellt wird, um die daraus resultierenden Ergebnisse auf weitere potenzielle Angreifer-Spuren zu untersuchen, verwendet die „Sub Query“ Regel für den gleichen Zweck abgewandelte Versionen der gleichen Anfrage.

* + 1. Validität der Labels

Die Validität der zugewiesenen Labels auf den 8 simulierten Systemen des AIT Log Datensatzes V2.0 erfolgte durch eine Umfrage diverser Sicherheitsbeauftragter und IT Analytiker, welche die Labels mit gegebenem Kontext auf ihre Richtigkeit bewerten sollten.  
Insgesamt wurden die Daten von 8 Experten gesichtet, die im Voraus bezüglich ihrer Expertise ausgewählt wurden. Auch künstlich erzeugte „False Positives“, „True Negatives“ und Kontrollfragen wurden inkludiert, um die Aufmerksamkeit der Experten während der Begutachtung sicherzustellen.  
In fast allen Aspekten und Log-Dateien stimmte die Meinung der Experten mit den automatisch zugewiesenen Labels überein. Ausreißer waren durch Interpretationsspielraum, unklar formulierte Log-Ereignisse und fehlende Zeitstempel zu erklären.

* 1. Limitationen

Der Datensatz besitzt eine relativ simple Netzwerk Struktur und die Labels wurden automatisch erstellt. Hierbei ist zu erwähnen, dass eine manuelle Erstellung aufgrund des sehr großen Umfangs nicht realistisch umzusetzen wäre.  
Die Simulation von Nutzer-Aktivitäten ist aufgrund der automatisierten Erstellung limitiert, weshalb bei der Auswertung von IDS eine etwas niedrigere False Positive Quote zu erwarten ist im Vergleich zu ihrer Anwendung auf realen Daten.  
Die Logging Granularität ist gemäß herkömmlicher Richtlinien eingestellt, also werden zu „niedrige“ Alerts wie Debugging-Informationen in den meisten Fällen nicht aufgezeichnet.  
Die generierten Labels besitzen zudem keine „Label weights“, also keine numerische Werte, die eine Stärke der Assoziation zwischen Ereignissen und Labels angeben würden.