Bundesministerium
für Bildung
und Forschung

ZenSIM
Industrie 4.0

Zentrales Security Incident Management für KMU in Industrie 4.0 (ZenSIM4.0)

Teilvorhaben:

Simulation und Konzept eines Security Incident Management System Demonstrators mit integriertem Datenschutz

Schlussbericht

DOKUMENTINFORMATIONEN	
TYP	Bericht (Report)
TITEL	Abschlussbericht im Verbundprojekt ZenSIM4.0
LAUFZEIT DES VORHABENS	01. September 2021 bis 30. November 2024
ZUWENDUNGSEMPFÄNGER	Hochschule Bremen
ORT, DATUM	Bremen, den 31.01.2025
FÖRDERKENNZEICHEN	16KIS1504

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des BMBF unter dem Förderkennzeichen 16KIS1504 *gefördert. Die Verantwortung liegt bei den Autoren.*

KONTAKTINFORMATIONEN		
NAME	**ORGANISATION**	**E-MAIL**
Prof. Dr.-Ing. Evren Eren	Hochschule Bremen	evren.eren@hs-bremen.de
Sercan Catalkaya	Hochschule Bremen	sercan.catalkaya@hs-bremen.de

Inhalt

1 Kurzbericht (Teil I)

Zu jedem funktionierenden IT-Sicherheitsmanagement gehört ein Sicherheitsvorfallmanagement (Security Incident Management), das auch ein ständiges Schwachstellenmanagement (Vulnerability Management) beinhalten muss. Denn nur durch die regelmäßige Beobachtung und Reaktion auf neu hinzutretende IT-Sicherheitslücken sowie die Erkennung aktueller Angriffe oder sogar den Anzeichen einer erfolgreichen Kompromittierung kann ein vollständig geschlossener Prozess im Sinne eines PDCA-Zyklus (Plan, Do, Check, Act) gewährleistet werden. Dabei ist ein funktionierendes IT-Sicherheitsmanagement nicht nur eine Aufgabe, die Unternehmen von sich aus erfüllen müssen – ständig neu hinzutretende gesetzliche und normative Erfordernisse auf deutscher wie auf europäischer, für global tätige Unternehmen auch auf internationaler Ebene, machen IT-Sicherheit mehr denn je zu einer umfassenden Compliance-Aufgabe, die über technische Grenzen hinweg rechtliche sowie betriebswirtschaftliche und nicht zuletzt auch psychologische Fragestellungen adressiert.

Im IT-Sicherheitsmanagement sind Industrieumgebungen ein Spezialfall, denn die zunehmend komplexe Vernetzung von Komponenten steigert das Qualitätsrisiko. Außerdem ist ein hohes Niveau der IT-Sicherheit ein kritischer Erfolgsfaktor für Industrie 4.0 und Digitalisierung. Effektive Sicherheit kann nur gewährleistet werden, wenn neueste Angriffe sofort kommuniziert und die Reaktionszeiten zum Schließen von Sicherheitslücken reduziert werden können. So muss Sicherheit vornehmlich durch eine verbesserte Prävention bei der Systementwicklung (Security-by-Design) sowie durch eine möglichst schnelle Reaktion bei Bekanntwerden neuer Sicherheits-lücken oder Angriffsmöglichkeiten erreicht werden, um die Gefahr von Vorfällen (Security Incident) zu reduzieren. Hersteller von Komponenten und Systemintegratoren und Maschinenbauer der Industrie 4.0 sollten daher ebenso wie die Betreiber der Systeme ein profundes Sicherheits- und Risikomanagement vorweisen und belegen, dass sowohl ihre IT-Infrastruktur als auch ihre Produkte über ausreichende Schutzmechanismen verfügt. Dies bindet jedoch Personal und erfordert Know-how, das insbesondere bei KMUs oft nicht vorhanden ist. Eine weitere wichtige Basis für ein funktionierendes Schwachstellen-Management stellt die Komponenten-Inventarisierung (Asset Inventory) dar. Nur damit gelingt es, die schützenswerten Bestandteile (Assets) eines Produkts eindeutig zu identifizieren und mit relevanten Schwachstelleninformationen zu korrelieren. Diese Inventarlisten liegen bei vielen Unternehmen und gerade bei KMUs im Softwarebereich entweder gar nicht oder nur in sehr rudimentärer Form vor und verhindern somit ein effektives und regelmäßiges Überwachen der Produkte.

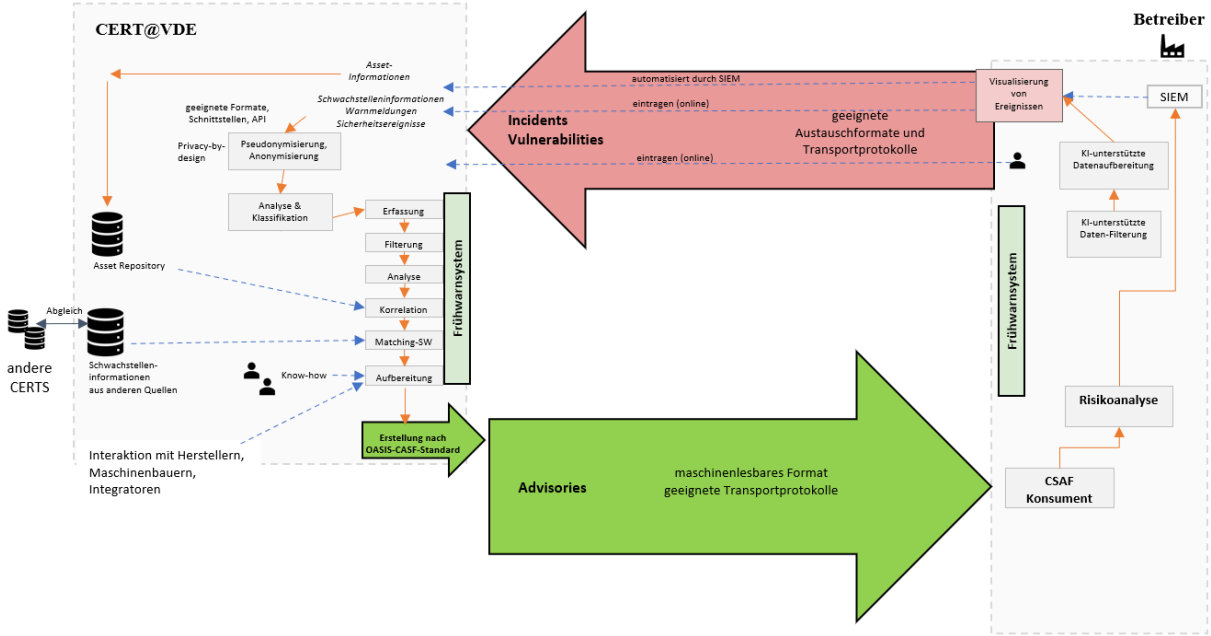

Abbildung 1: Informations- und Austauschplattform des ZenSIM4.0-Projekts

Das Projekt ZenSIM4.0 setzt an dieser Stelle an, indem hier keine gezielte Lösung zur Einrichtung eines Information Security Management Systems (ISMS) angeboten wurde, sondern Unterstützung zum selbstständigen Betrieb eines unternehmerischen „Security Incident Managements“ – also das Erkennen und den richtigen Umgang mit IT-Schwachstellen im Produkt und Produktionsumfeld für Hersteller, Maschinenbauer, Integratoren und Betreiber von Industrie4.0-Umgebungen. Dazu wurde im Rahmen des Projekts eine Plattform entwickelt, auf der Informationen zu der IT-/OT-Infrastruktur und den schützenswerten Informationen (Assets), der bereits verwendeten Sicherheitsmaßnahmen sowie bereits ereignete Sicherheitsvorfälle bereitgestellt werden (siehe Abbildung 1). Als Gegenleistung für ihre Teilnahme und die Wissensweitergabe erhalten KMUs fundierte Sicherheitsempfehlungen z.B. zu Exploits und Schwachstellen spezifisch für ihre IT-Infrastruktur und Assets. Zudem wird eine Vielzahl von Schwachstelleninformationen aus verschiedenen Quellen in die Plattform eingespeist, so dass ein KMU beispielsweise die Risikolage einschätzen kann. KMUs können so für sich fachlich relevante Warnungen individuell auswählen und konsumieren („Warenkorbansatz“), ohne dass hierfür ein eigenes CERT (Computer-Notfallteam) aufgebaut werden muss. Diese können bei größeren, insbesondere international agierenden, Firmen vorausgesetzt werden, die im Gegensatz zum Mittelstand die dafür nötigen Ressourcen aufwenden können.

Das ZenSIM4.0-Projekt hat daher eine Reihe technologischer Innovationen entwickelt, die für das industrielle „Security Incident Management“ für Betreiber, Hersteller und Anbieter über den gegenwärtigen Stand der Technik hinausgehen. Preisgabe, Speicherung, Aufbereitung und Austausch sind vertrauenswürdig und sicher sowie für die Teilnehmer transparent, d.h. verständlich und nachvollziehbar, gestaltet worden. In ZenSIM 4.0 wurde eine Plattform entwickelt, die Asset-Informationen verarbeitete, sowie Sicherheitsmaßnahmen und -vorfälle regelten. Hierbei wurde ein einheitlicher Zugriff auf Warnmeldungen ermöglicht und Assets sind automatisch abgeglichen worden innerhalb des Demonstrators. Die Plattform hat als „Single-Point-of-Contact“ Frühwarnung und konsolidierte Reaktion ermöglicht und damit präventive und reaktive Services bereitgestellt. Assets wurden aus unterschiedlichsten Quellen aggregiert und mit bereits vorhandenen Assets korreliert. Durch die Entwicklung eines automatisierten und einheitlichen Advisory-Systems entstand ein Frühwarnsystem als Demonstrator, das vor Schwachstellen-Bekanntmachung der Hersteller und konsolidierte Reaktion für KMUs zur Vermeidung von Hacking-Angriffen erlaubte. Damit konnten Schwachstelleninformationen datenschutzkonform und rechtssicher ausgetauscht

werden (u.a. durch Privacy-by-Design). Wesentliches Ziel war es, konkrete und verständliche Handlungsempfehlungen für KMUs ohne IT-Sicherheitspersonal generieren zu können. Dieses Ziel konnte anstandslos erreicht werden.

2 Eingehende Darstellung (Teil II)

Dieser Schlussbericht enthält eine Kurzbeschreibung aller Arbeiten und Ergebnisse, die seitens der Hochschule Bremen (HSB) im ZenSIM4.0-Projekt geleistet worden sind. Die Arbeiten sind ausführlich in verschiedenen AP-Berichten dokumentiert worden, die mit den Partnern innerhalb des Projektes angefertigt wurden:

- AP2-Bericht: Anforderungsanalyse und Systemarchitektur
- AP3/4-Bericht: Entwicklung der Plattform
- AP5-Bericht: Feldtests

Sie können nach Bedarf für eine Detailbetrachtung eingesehen werden.

2.1 Aufgabenstellung und wissenschaftlich/technischer Stand an den angeknüpft wurde

Die wissenschaftlichen und technischen Ziele der Hochschule Bremen im Projekt umfassten:

- **Datenformate und -quellen:** Untersuchung von Anforderungen für verschiedene Umgebungen (Office-IT und Industrie) und deren Integration in die spätere Konzeption.
- **High-Level-Architektur:** Definition einer Architektur basierend auf den Anforderungen, die beschreibt, welche Komponenten mit welchen Eigenschaften in einer typischen Infrastruktur implementiert werden sollen.
- **Simulationsplattform:** Validierung und Absicherung der Komponenten in einer Laborumgebung, einschließlich eines Security Assessments.
- **Integration der Komponenten:** Integration und Test der entwickelten Komponenten auf einer Testplattform für den Demonstrator.
- **Ende-zu-Ende Testszenarien:** Definition von Testprotokollen und Durchführung von Tests zur Bewertung der Anforderungen und der Nutzerstudie.
- **Analyse und Auswertung:** Bewertung und Dokumentation der Testergebnisse.
- **Datenschutz:** Proaktive Entwicklung von Datenschutzkonzepten für das Incident Management, sowohl für Komponenten als auch für den Demonstrator.
- **Rechtliche Anforderungen und Compliance:** Integration rechtlicher Anforderungen im Kontext einer Austausch- und Kooperationsplattform für sensible Informationen.

Projektziele

Die wesentlichen Projektziele des Teilvorhabens der HSB waren im ZenSIM4.0-Projekt:

- Schwachstelleninformationen und Asset Management
- Konzeption der Plattformarchitektur
- Simulation und Emulation wesentlicher Elemente bzw. Komponenten des Systems
- Datenschutz- und Rechtskonformität

Wie in Abbildung 2 dargestellt, sollte ein ganzheitliches "Security Incident Management" implementiert werden, speziell konzipiert für Betreiber, Hersteller und Anbieter. Durch diesen integrierten Ansatz sollte ZenSIM4.0 KMUs eine leistungsfähige Plattform zur Risikominimierung und effektiven Bewältigung von Sicherheitsherausforderungen bieten.

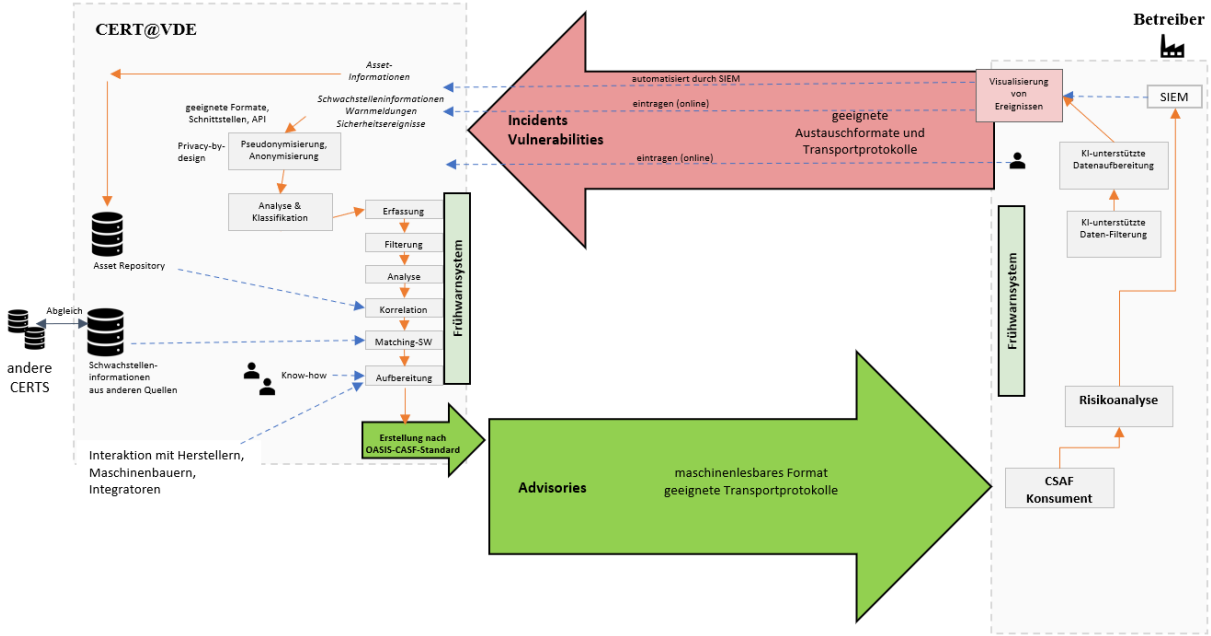

Abbildung 2: Informations- und Austauschplattform des ZenSIM4.0-Projekts

Zu Anfang des Projektes konnten Betreiber dem CERT@VDE Schwachstelleinformationen nur manuell melden, wahlweise per E-Mail oder über ein Kontaktformular. Die Meldungen wurden dabei unstrukturiert in Form eines Freitextes (Pflichtfelder: Name, E-Mail-Adresse, Betreff und Freitext). Teilweise wurden Informationen über neue Schwachstellen auch in Form von Videos (ohne weitere Erklärung) an das CERT@VDE übermittelt. Die Meldung der Schwachstelleninformation in einem unstrukturierten Format hat zur Folge, dass ein hoher manueller Aufwand zur Auswertung der übermittelten Informationen zu den Schwachstellen auf Seiten des CERT@VDE betrieben werden musste. Auch können hierbei wertvolle Schwachstelleninformationen verloren gehen. Auf Seiten CERT@VDE besteht die Anforderung, möglichst viele Informationen zu einer möglichen Schwachstelle zu erhalten, damit eine tiefgreifende Analyse möglich ist.

2.2 Ablauf des Vorhabens

Das ZenSIM4.0-Projekt gliederte sich in fünf eigenständige Arbeitspakete, wie in Tabelle 1 dargestellt. Diese Struktur ermöglichte eine effiziente Aufgabenverteilung unter den Projektpartnern. Jeder Partner übernahm die Hauptverantwortung für spezifische Arbeitspakete, wobei die HSB federführend für das Arbeitspaket AP3 zuständig war. Zusätzlich leistete die HSB unterstützende Beiträge zu den anderen Arbeitspaketen.

Projektstruktur und Verantwortlichkeiten

- Fünf distinkte Arbeitspakete: Bildeten die Grundlage der Projektorganisation.
- Spezialisierte Zuständigkeiten: Jeder Partner konzentrierte sich auf bestimmte Arbeitspakete.
- Rolle der Hochschule Bremen: Hauptverantwortlich für AP3 und unterstützende Funktion in anderen Arbeitspaketen.

Diese Aufteilung förderte eine zielgerichtete und effiziente Projektdurchführung, indem sie die Expertise jedes Partners optimal nutzte.

AP	Name des Arbeitspaketes (AP)	Dauer	AP-Leiter
AP 1	***Projektmanagement***	**M01-M36**	**DECOIT**
AP 1.1	Koordination der Gremien- und Abstimmungsarbeiten	M01-M36	DECOIT
AP 1.2	Vernetzung und Kommunikation	M01-M36	DECOIT
AP 2	***Anforderungsanalyse und Entwurf der Plattformsystemarchitektur***	**M01-M12**	**CERT@VDE**
AP 2.1	Stakeholder- und Anforderungsanalyse	M01-M06	CERT@VDE
AP 2.2	Anwendungsszenario-Definition	M02-M04	HSB
AP 2.3	Spezifikation der Plattformarchitektur	M04-M08	DECOIT
AP 2.4	Spezifikation der Schnittstellen und Kommunikationsprotokolle	M04-M08	DECOIT
AP 2.5	Anforderungsanalyse zur Integration und Umsetzung eines maschinenlesbaren Advisory-Formats auf CSAF-Basis	M07-M12	CERT@VDE
AP 2.6	Anforderungsanalyse und Konzept zur Integration von Schwachstelleninformationen für SIEM-Hersteller	M07-M12	CERT@VDE
AP 3	***Entwicklung der Plattform***	**M08-M20**	**HSB**
AP 3.1	Matching Software Mock-up	M08-M09	HSB
AP 3.2	Simulationsumgebung	M08-M11	HSB
AP 3.3	Matching-Software	M10-M14	CERT@VDE
AP 3.4	Asset-Verarbeitung	M10-M14	CERT@VDE
AP 3.5	Konzept zur Visualisierung von Ereignissen	M10-M14	DECOIT
AP 3.6	Implementierung der Visualisierungskomponenten	M15-M20	DECOIT
AP 3.7	KI-unterstützte Datenfilterung	M15-M20	DECOIT
AP 3.8	KI-unterstützte Datenaufbereitung	M15-M20	DECOIT
AP 3.9	Demonstratoren zur automatisierten Aufnahme von OT-basierten SIEM-Meldungen	M10-M15	HSB
AP 3.10	Visualisierung von Ereignissen	M15-M20	DECOIT
AP 3.11	Implementierung von Schnittstellen zu Drittsystemen	M15-M20	CERT@VDE
AP 3.12	Teilsystemintegration und Vorvalidierung	M15-M20	CERT@VDE
AP 4	***Erprobung und Validierung***	**M21-M32**	**DECOIT**
AP 4.1	Einbindung unterschiedlicher Datenquellen	M21-M24	DECOIT
AP 4.2	Integration der Plattform	M21-M24	DECOIT
AP 4.3	Integrationstests	M24-M27	DECOIT
AP 4.4	System- und Funktionstests	M27-M30	DECOIT
AP 4.5	Evaluation des Demonstrators	M27-M32	HSB
AP 4.6	Untersuchung der Compliance	M28-M32	CERT@VDE
AP 5	***Feldtests und Standardisierung***	**M03-M36**	**CERT@VDE**
AP 5.1	Feldtests mit assoziierten Partnern	M24-M32	CERT@VDE
AP 5.2	Dokumentation und Bewertung der Testergebnisse	M28-M36	CERT@VDE
AP 5.3	Datenschutz und Rechtskonformität	M03-M36	HSB
AP 5.4	Abgleich mit den Standardisierungsaktivitäten	M03-M36	HSB

Tabelle 1: Übersicht über die Arbeitspakete des ZenSIM4.0-Projektes

In folgenden Arbeitspaketen hatte die HSB die AP-Leitung:

AP	Name des Arbeitspaketes (AP)
AP 2	***Anforderungsanalyse und Entwurf der Plattformsystemarchitektur***
AP 2.2	Anwendungsszenario-Definition
AP 3	***Entwicklung der Plattform***
AP 3.1	Matching Software Mock-up
AP 3.2	Simulationsumgebung
AP 3.9	Demonstratoren zur automatisierten Aufnahme von OT-basierten SIEM-Meldungen
AP 4	***Erprobung und Validierung***
AP 4.5	Evaluation des Demonstrators
AP 5	***Feldtests und Standardisierung***
AP 5.3	Datenschutz und Rechtskonformität
AP 5.4	Abgleich mit den Standardisierungsaktivitäten

Bei der HSB wurden zwei Masterarbeiten abgeschlossen:

- A. Lehmann, Analyse geeigneter Strukturen von Schwachstelleninformationen, Verfahren zur Asset-Verarbeitung sowie Implementierung von Schnittstellen für eine Incident-Management Plattform im Kontext des Forschungsprojektes ZenSIM 4.0, Masterarbeit an der Ruhr-Universität Bochum, August 2023, Bochum
- Meike Henschen-Bolte, Entwicklung einer Programmierschnittstelle zur Bereitstellung von Asset-Daten für ein SIEM im Rahmen des ZenSIM4.0-Forschungsprojekts, Masterarbeit an der Ruhr-University Bochum, März 2024, Bochum

Plattformarchitektur und Kommunikationsverbindungen (roter Pfeil)

Im Kontext der Plattformarchitektur waren die wesentlichen Kommunikationsverbindungen die in der Abbildung 2 illustrierten Pfeile. Insbesondere lag der Fokus bei der Verbindung mit dem roten Pfeil. Hierbei erfolgte seitens der HSB die Eignung der Architektur zur Erfüllung der Anforderungen bzgl. Datenaufnahme und -analyse. Dies hatte entsprechende Implikationen auf die Spezifikation der Schnittstellen und Kommunikationsprotokolle. Nach einer eingehenden Analyse von Prozessen und entsprechenden Datenflüssen wurden auf Basis des zuvor ausgewählten Frameworks die grundlegenden Komponenten der Architektur in einer Simulationsumgebung exemplarisch implementiert und die Schnittstellen definiert sowie geeignete Kommunikationsprotokolle untersucht. Da Betreiber Schwachstelleninformationen nur manuell, unstrukturiert und mit entsprechendem Aufwand melden konnten, war die wesentliche Aufgabe, diese Informationen direkt in einem Format abzufragen, oder zu übermitteln, welches automatisch oder halb automatisch auf Seiten CERT@VDE ausgewertet werden kann. Hierzu wurden alle Informationen aus dem SIEM beim Betreiber identifiziert. Flankiert wurde dies durch Befragungen von Betreibern. Das Ziel war es, Informationen in einem strukturierten (und menschenlesbaren) Format zu übermitteln, aber dabei die Komplexität zur Meldung einer Schwachstelle möglichst gering zu halten, damit eine mögliche Hemmschwelle zur Meldung nicht zu hoch ist, da sonst eventuell wertvolle Schwachstelleninformationen verloren gehen könnten.

Damit sowohl für den Betreiber der Aufwand zur Meldung von potentiellen Schwachstellen möglichst gering ist als auch dem CERT@VDE genug Informationen zur Auswertung von möglichen Schwachstellen vorliegen, musste ein geeignetes Maß und/oder Format zum Austausch der notwendigen Informationen gefunden werden. Insbesondere ist es für CERT@VDE notwendig, dass Meldungen zu potentiellen Schwachstellen mit so vielen Informationen wie möglich angereichert werden. Minimale Informationen sind: Allgemeine Informationen zum betroffenen Asset (angereichert mit Informationen wie beispielsweise Betriebssystem, Firmwareversion, Software-Patchstand, Port, IP-Adresse, MAC-Adresse sowie (falls möglich) das betroffene Protokoll.

Die rechte Seite der Abbildung beinhaltet die Sicht/Verantwortlichkeit des Betreibers. Hier wurden zunächst mit Hilfe der SIEM-Lösung des Projektpartners DECOIT Schwachstelleninformationen gesammelt, welche gefiltert und aufbereitet werden müssen. Ein Untersuchungsgegenstand war die geeignete Übertragung (roter Pfeil in der Abbildung) zum CERT@VDE, wofür geeignete Austauschformate sowie Transportprotokolle evaluiert und festgelegt werden mussten. Es wurde ein Proof-of-Concept (PoC) durch Implementierung der Schnittstellen auf der Seite des Betreibers sowie der Seite des CERT@VDE erarbeitet und durch Integration der Komponenten auf beiden Seiten in der Simulationsumgebung umgesetzt. Geeignete Austauschformate sowie Transportprotokolle vom Betreiber zum CERT@VDE wurden evaluiert. Es wurden Ansätze und Konzepte untersucht, Schnittstellen auf beiden Seiten zu implementieren, jedoch verworfen. Identifiziert wurden essentielle Informationen zur späteren vollständigen Analyse. Als Orientierung diente NIST mit einer Liste als Empfehlung zur Sammlung von Datenelementen (Basic Data

Elements) zu Vorfällen [10]. Es wurden Differenzierungen zwischen „Contact Information for the Incident Reporter and Handler“, „Incident Details“ und „General Comments“ vorgenommen.
Für die Übertragung von Ioc/IoA vom Betreiber zu CERT@VDE gibt es nun viele Möglichkeiten. Im Rahmen der Umsetzung innerhalb des ZenSIM4.0-Projektes wurden die in *Abbildung 3* dargelegten Transportwege implementiert und auf die Praxistauglichkeit untersucht.

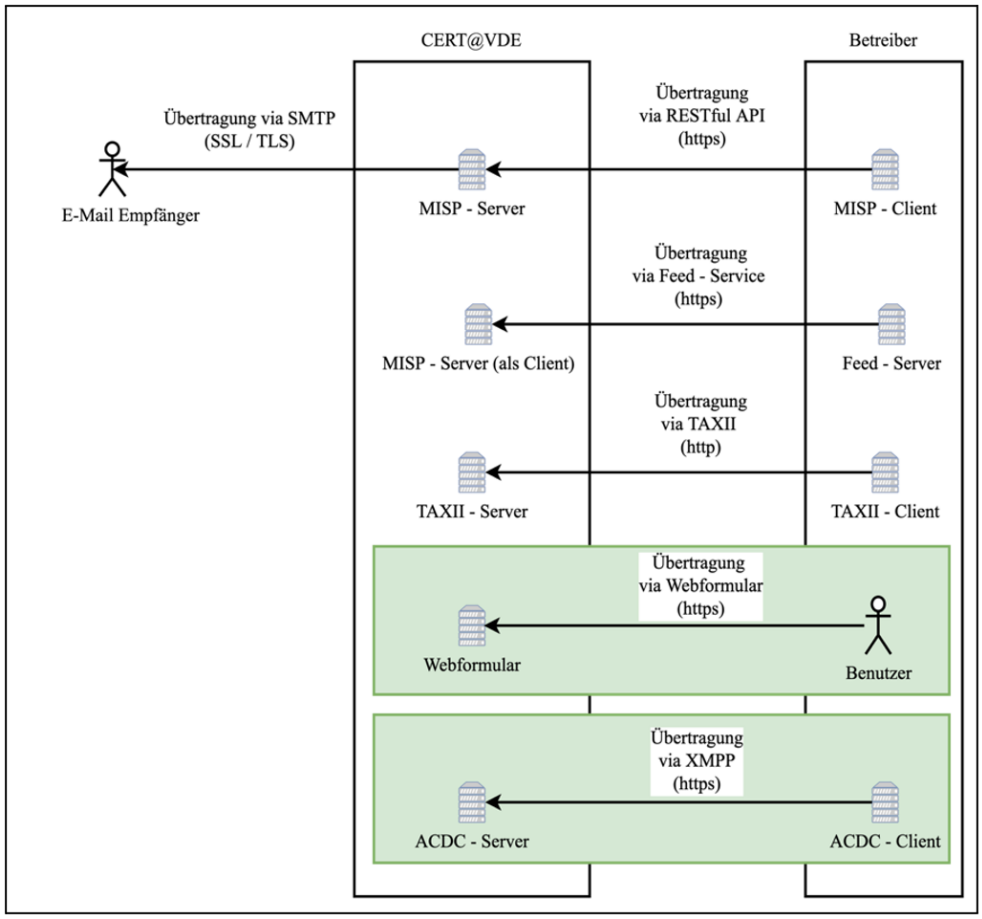

Abbildung 3: Transportwege

E-Mail:

- Von jedem System aus möglich, dass Emails versenden kann. Wenn das SIEM einen Mail-Server integriert hat, kann es auch vom SIEM-System aus erfolgen.
- Manuelle und halbautomatische Übermittlung von IoC/IoA möglich.
  - Manuell: Manuelles zusammentragen der Informationen aus dem SIEM-System.
  - Halbautomatisch: Das SIEM-System erkennt einen Vorfall und schlägt der autorisierten Person vor diese an das CERT@VDE zu übermitteln.
- Emails können wahlweise mit PGP verschlüsselt oder signiert werden, da das CERT@VDE ein Zertifikat dafür anbieten kann.

MISP:

- Betreiben eines MISP-Servers beim CERT@VDE, sodass ein MISP-Client beim Betreiber Vorfälle zum Server melden kann.
- Übertragung erfolgt über eine REST API.
- Verschlüsselung durch HTTPS.
- Es erfolgt eine E-Mail-Benachrichtigung bei neuen eingehenden Meldungen.

- MISP-Server kann auch auf der Betreiberseite betrieben werden, sodass neue IoC/IoA über E-Mail an das CERT@VDE gesendet werden.
- Der MISP-Client zur Ansteuerung der REST API wurde als Python Programm umgesetzt.
- Es ist auch möglich die IoC/IoA Daten über Feeds zu abonnieren und in regelmäßigen Intervallen automatisch abzurufen.
    - Der MISP-Client übernimmt die Rolle des Feed Servers und stellt die IoC/IoA Daten mittels Web Server Gateway Interface zur Verfügung.
    - Der MISP-Server bei der CERT@VDE Seite kann anschließend konfiguriert werden, um die Feeds dann per pull abzufragen. Der Datentransfer wird also vom MISP-Server initiiert.

TAXII:

- Übermittlung der IoC/IoA Daten im STIX-Format.
    - Die Daten müssen in das STIX-Format konvertiert werden.
- Übertragung erfolgt über HTTP.
- Setzt TAXII Server auf CERT@VDE Seite voraus.
    - TAXII Client wird mit einem Python Programm umgesetzt, welches http post und http get Methoden benutzt um die Daten zu senden und auszulesen.

Webformular:

- Mittels Webformular können Betreiber IoC/IoA direkt auf der Webseite (https://cert.vde.com/helper/reportvuln/) von CERT@VDE übermitteln.
- Als Freitext (unstrukturiertes Format).
- Übertragung erfolgt via HTTPS.
- Das CERT@VDE hat die Möglichkeit ergänzende Informationen von der meldenden Person anzufragen.
- Das Webformular zur Übermittlung von IoC/IoA wird zum jetzigen Zeitpunkt auf der Internetseite des CERT@VDE angeboten (https://cert.vde.com/helper/reportvuln/).

Grundlegend eignen sich alle Varianten zur Übermittlung der Daten von einem Betreiber zum CERT@VDE. Die Variante über den TAXII2-Server ist die bevorzugte, da diese gut zu implementieren ist und die STIX2/TAXII2-Standards eine große Akzeptanz innerhalb der Industrie aufweisen [11]. Des Weiteren ist bei dieser Variante der zu erwartende Overhead bei den zu übertragenen Daten gering, da die Dateigröße im Vergleich zu den anderen Formaten sehr klein ist. Die Implementierung umfasste das Aufsetzen eines PoC, um die Schnittstellen auf beiden Seiten der ZenSIM4.0-Plattform simulieren zu können. Hierbei wurden unterschiedliche Transportwege und Schnittstellen untersucht, praktisch implementiert und getestet. Für die Implementierung der Schnittstellen wurden verschiedene Python-Programme und Bash-Skripte (als Hilfsskripte) geschrieben. Darüber hinaus wurden einige Bibliotheken verwendet, welche die Verwendung und Steuerung (z.B. das Senden an den MISP-Server) vereinfachen, und bestehende Konfigurationsdateien angepasst sowie neue erstellt.

Advisories auf CSAF-Basis

Im Kontext der Anforderungsanalyse zur Integration und Umsetzung eines maschinenlesbaren Advisory-Formats auf CSAF-Basis wurden gemeinsam mit den Projektpartnern die Anforderungen an eine automatisierte Verarbeitung von Advisories auf Basis des ausgewählten CSAF-Formats identifiziert. Sowohl hier als auch im anschließenden Konzept zur Erzeugung von Schwachstelleninformationen aus der SIEM-Umgebung gab es eine Untersuchung von allgemeinen Strukturen von Schwachstelleninformationen.

Matching-SW Mock-up
Im Zuge der Untersuchungen des Projektpartners DECOIT im Bereich „Matching-SW Mock-up“ wurde ein selbstlernender KI-Algorithmus getestet, mit dem Ergebnis, dass viele „False Positives“ zusammenkamen. Vor diesem Hintergrund wurde bei der SIEM-Entwicklung der Fokus auf regelbasierte Verfahren gesetzt, sodass Mock-Up-Daten obsolet wurden.

Simulationsumgebung
Was die Simulationsumgebung anbetrifft, wurden die einzelnen Systembestandteile erarbeitet, die in der ersten Phase in einer Laborumgebung mittels Virtualisierungstechniken simuliert wurden. Zur Identifikation und Konfiguration der Systembestandteile und der Kommunikationstopologie hat sich die HSB an bestehende Systeme des CERT@VDE orientiert und überarbeitet. Die Simulation der Plattform beinhaltet die Möglichkeit, vorgenommene Überarbeitungen und Testimplementierungen durchführen zu können, ohne dass das Produktivsystem gestört wird. Bei der Entwicklung der Plattform wurden bereits einige Komponenten auf Basis der erarbeiteten Anforderungen auf der Anwendungsszenario-Definition in die Simulationsumgebung aufgenommen. Für die Simulationsumgebung wurden wesentliche und typische ICS-Komponenten, Sensoren und die Security-Assessment-Komponenten für das zu entwickelnde Gesamtsystem definiert und in die Topologie aufgenommen. Diese Implementierung ist eine hybride Verbindung (virtualisiert und physisch über das Internet). Darüber hinaus erfolgten Absicherung von Komponenten und Verbindungen. Im Laufe des Projektes fanden Sicherheitstests der Komponenten nach BSI und ISO 27001 statt. Dieses Security Assessment von Systemkomponenten sowie der Kommunikationsverbindungen diente einer kontinuierlichen Überwachung der Komponenten. Das Security Assessment der simulierten Komponenten und der Netztopologie hatte zum Ziel, die System- und Netzwerkkomponenten sowie die Verbindungen in der Kommunikationskette abzusichern. Für die Konfiguration der Systembestandteile und der Topologie wurden diverse Quellen wie z.B. das „ICS-Security-Kompendium“ [7] sowie das „IT-Grundschutz-Kompendium“ [8] des BSI herangezogen.

Die virtuelle Laborumgebung setzt sich im Wesentlichen KVM-basierter Virtualisierung und GNS3-basierter Emulation zusammen. Um die existierenden Standards umzusetzen, wurde in Anlehnung an das Purdue-Model ein 5-Level-Konzept erarbeitet und die Basisinfrastruktur um folgende Komponenten erweitert [9]:

- **Level 5:** Öffentliches Internet sowie externe Netze
  - Armin CERT, Kali 2023 Sliver, Win7 OpenVPN Client, Win7 IPSEC VPN Client
- **Level 4:** Unternehmensnetzwerk, Intranet bzw. Office-Netzwerk (Office-IT)
- **Level 4-3:** OPNsense Stage1 Firewall
- **Level 3:** Automatisierungsnetzwerk
  - Active Directory Server, Scada GRFICSv2, Siemens HMI, openHistorian, ScanBox®, IRMA
  - O.g. Systeme kommunizieren mit den Systemen in Level 4, in der Regel über eine entsprechende DMZ dazwischen. Eine direkte Kommunikation darf nicht erfolgen. Darüber hinaus dürfen Level 3-Systeme mit Systemen in Level 2 und 1 kommunizieren.
- **Level 3-2:** OPNsense Stage2-Firewall
- **Level 2:** Industrienetzwerk
  - GRFICSv2 Engineering Workstation, GRFICSv2 Chemicalplant, GRFICSv2 PLC, Siemens PLC, Siemens Engineering Workstation, ScanBox®, IRMA

  - O.g. Systeme dürfen mit Systemen in Level 1 kommunizieren. Darüber hinaus ist eine Kommunikation mit Systemen in Level 3 und 4, über dedizierte DMZ möglich.
- **Level 1:** Prozesssteuerungsnetzwerk
  - Es wird kein Level 1 verwendet.

In Abbildung 4 ist die Gesamttopologie dargestellt. Hier finden sich die verschiedenen Level (Level 5 bis Level 2) eines ICS-Netzwerkes in einer Organisation. Der Screenshot ist aus dem GNS3-Client und stellt die Managementoberfläche des Topologie-Editors dar. Hier können alle virtuellen Maschinen und Appliances bedient und verwaltet werden. Eine direkte Interaktion mit den Betriebssystemen bzw. Benutzeroberflächen der VMs ist per VNC von außen oder direkt per Telnet-Session vom GNS3-Client aus möglich.

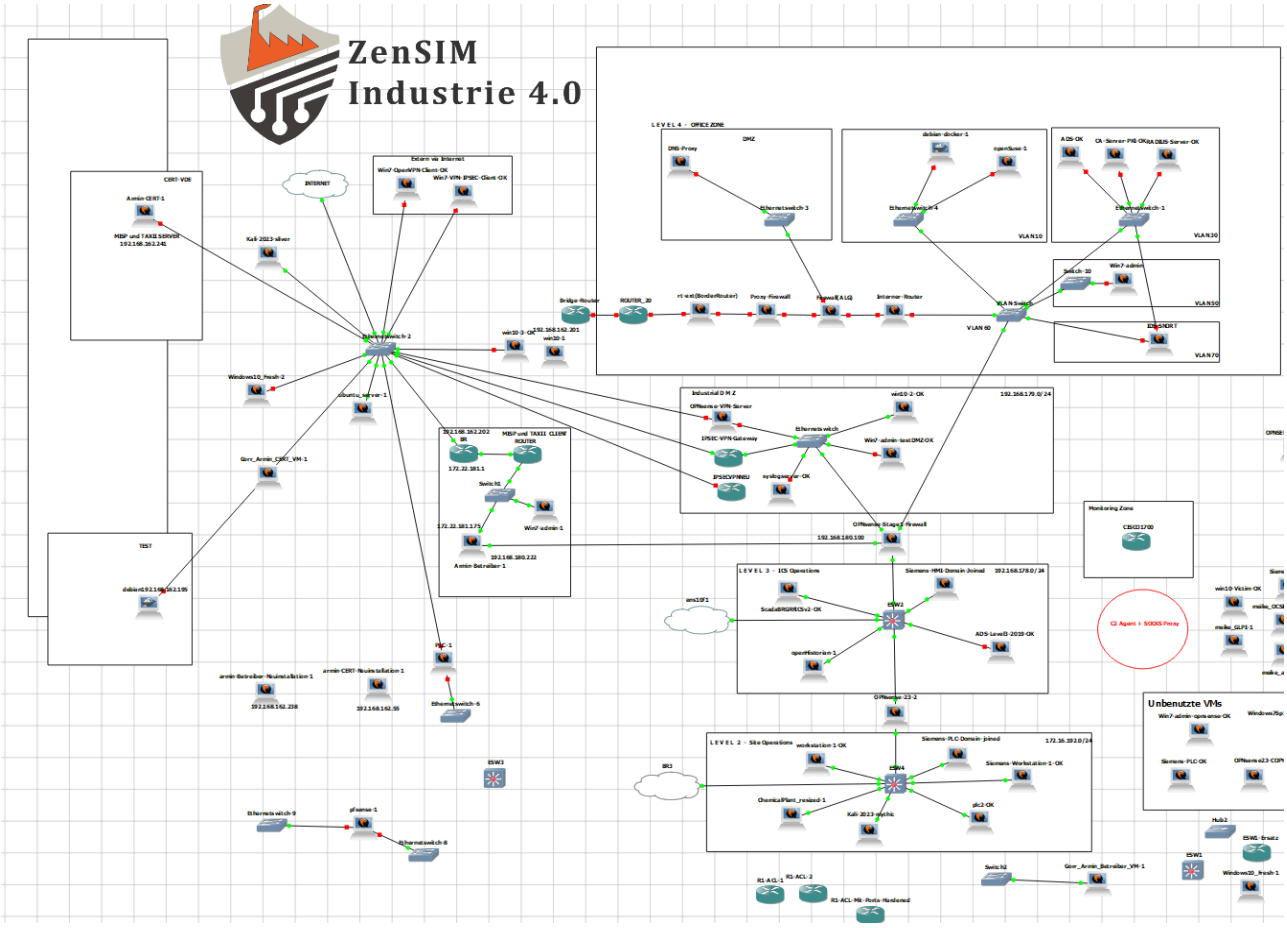

Abbildung 4: Netztopologie in GNS3

Ein weiterer Teil zur Simulation eines ICS-Netzwerks basiert auf GRFICSv2 (Fortiphyd Logic), was sich aus mehreren VMs zusammensetzt und eine Chemieanlage simuliert [6]. Hierbei kommt Modbus für die Kommunikation zwischen den ICS-Komponenten zum Einsatz:

- Simulation: Diese VM realisiert die Simulation der Chemieanlage. Sie simuliert mehrere Remote IO-Devices, die von der PLC-VM gesteuert werden. Außerdem bietet sie eine Web-Visualisierung der Anlage mittels Unity WebGL.
- PLC: Diese VM ist eine modifizierte Version von OpenPLC, welches eine veraltete libmodbus-Library benutzt, die für einen „Buffer Overflow“-Angriff ausgenutzt werden kann [6].

- HMI: Die HMI-Komponente wird mittels der ScadaBR realisiert. Diese verbindet sich mit der PLC und fragt die Prozessdaten ab, welche anschließend innerhalb der HMI visualisiert werden. Außerdem lässt sich der Prozess mittels eines „Start/Stopp“-Buttons von der HMI aus steuern.
- PfSense Firewall/Router: Die PfSense trennt die ICS-Komponenten von der DMZ. In diesem Fall würde sich die HMI in der DMZ befinden; die Simulation, PLC und Engineering Workstation residieren im ICS-Netzwerk.
- Engineering Workstation: Diese VM ist ein Ubuntu 16.04 Host zur Programmierung der PLC.

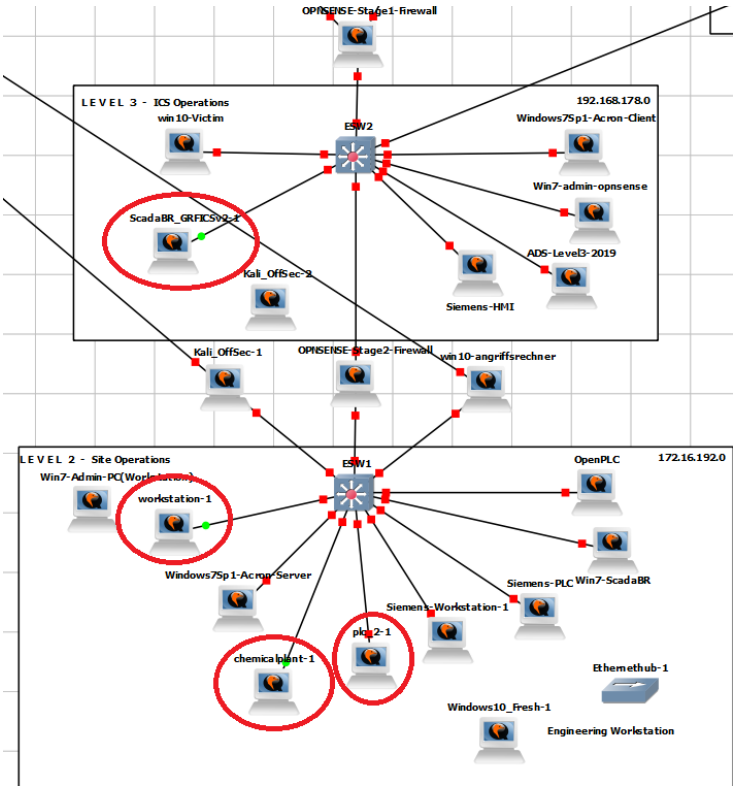

Abbildung 5: GRFICSv2-Appliances integriert in Level 2 und 3 der Simulationsumgebung

Des Weiteren wurde ein Historian realisiert. Es basiert auf OpenHistorian (GridProtectionAlliance), ein Backoffice-System zur Integration und Archivierung von Prozesssteuerungsdaten wie z.B. SCADA und Synchrophasor. OpenHistorian kann große Mengen an Zeitreihendaten, einschließlich hochauflösender Informationen im Sub-Sekundenbereich schnell und effizient speichern und abrufen [5]. OpenHistorian wurde in die Simulationsumgebung integriert und mit der GRFICSv2 PLC über das Modbus Protokoll. verbunden. Somit werden die Prozessdaten von der simulierten GRFICSv2 Chemieanlage archiviert.

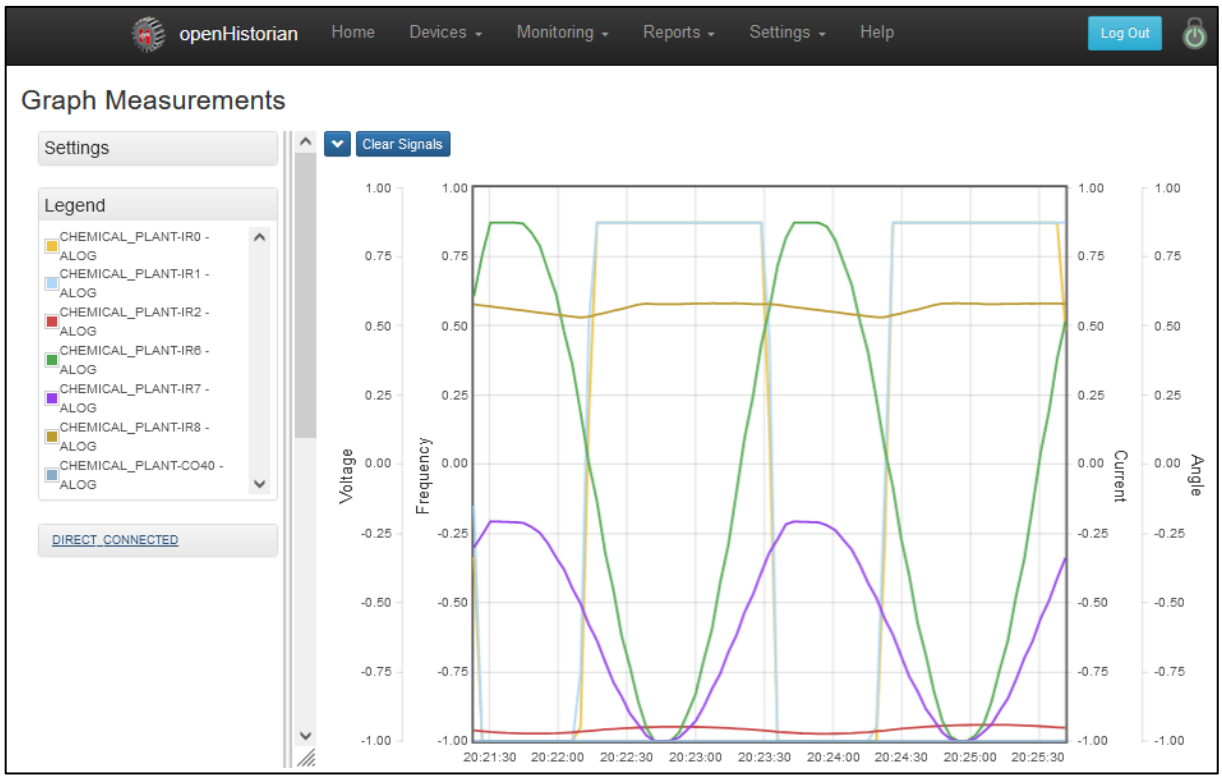

Abbildung 6: Grafische Darstellung von Modbus-Daten

Clearing House Services

Eine weitere Entwicklung sowie Implementierung adressierte die Realisierung einer externen Schnittstelle zu Clearing House Services. Im ZenSim-Projekt hat die HSB als Proof-of-Concept die Übermittlung von IoC-Daten und damit die Schnittstelle zu einem Clearing House implementiert. Hierzu wurde die Schnittstellendefinition des EU-Projektes Advanced Cyber Defence Centre (ACDC) [1] herangezogen. Dieses Projekt hatte das Ziel, die Zusammenarbeit gegen Cyberkriminalität zu fördern, um Botnetze frühzeitig zu erkennen, Werkzeuge zur Abwehr von Angriffen bereitzustellen und Best Practices zu entwickeln. Langfristig strebte das ACDC den Aufbau eines europaweiten Netzwerks von Cybersicherheitszentren an, die über ein sog. Centralised Data Clearing House (CCH) verbunden sind. Zur Erreichung dieser Ziele wurden Spezifikationen entwickelt. Dazu gehören definierte Datenformate, die leichtgewichtig und flexibel gestaltet sind, um Daten effizient zwischen technischen Sensoren, wie Honeypots oder Intrusion Detection Systems (IDS), und Benutzergruppen auszutauschen. Das ACDC nutzt außerdem bekannte Formate wie JSON, XML, sowie spezialisierte Standards wie X-ARF und STIX, um Sicherheitsvorfälle strukturiert darzustellen.

Für die Übermittlung von IoC/IoA innerhalb des ACDC-Projektes wurde sich für die Verwendung eines JSON-Formats entschieden – in Anlehnung an das Europäische Forschungsprojekt ACDC. Übermittelte IoC/IoA werden als *Report* bezeichnet, welche wiederum in *Reportkategorien* (Report categories) untergliedert werden. Die einzelnen *Reports* sind gemäß einem definierten JSON-Schema auf einem entsprechenden Abstraktionslevel definiert, so dass die auswertende Stelle (im Falle des ACDC-Projektes ein Central Clearing House (CCH)), diese verarbeiten und automatisch auswerten kann [1]. Abbildung 7 zeigt die im ACDC-Projekt verfügbaren *Reportkategorien* sowie deren *Unterkategorien* (subcategories). Anhand der Kategorie „eu.acdc.attack“ sowie der Unterkategorie „dos“ wird die vom ACDC-Projekt erwartete Struktur der zu übermittelten Daten vorgestellt.

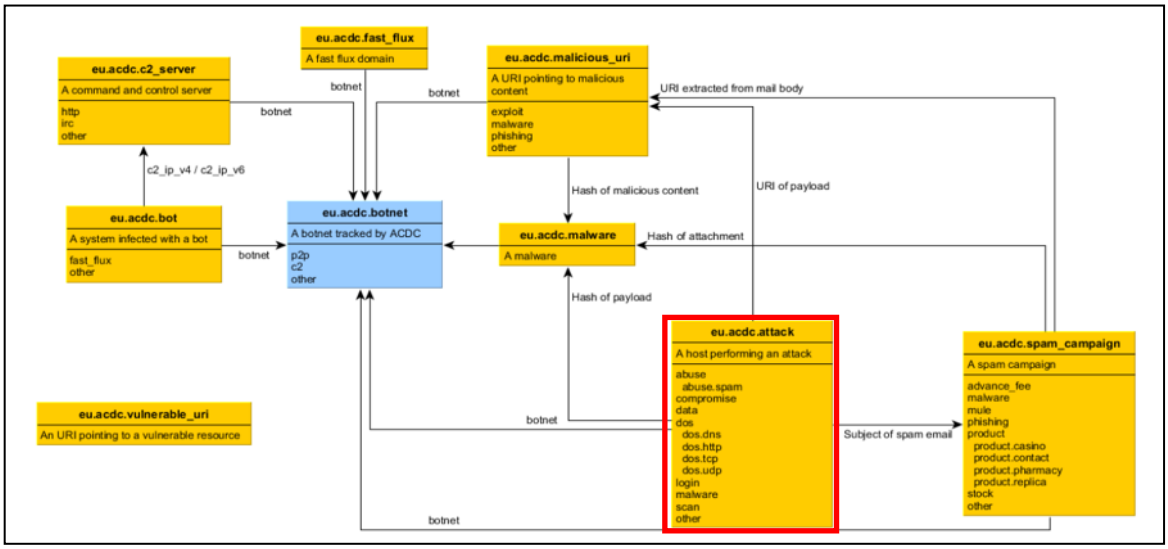

Abbildung 7: Reportkategorien innerhalb des ACDC-Projektes [1]

Das ACDC-Projekt definiert den Angriffsreport (Attack Report) als ein Szenario, in dem I-oC/IoA übermittelt werden, in welchem ein Host einen anderen Host angreift [1]. Das definierte Datenformat im ACDC-Projekt unterteilt sich in *Erforderliche Felder* (Required Fields), Optionale Felder (Optional Fields) und *Abhängigkeiten (Dependencies)*. Damit in diesem Fall vom Betreiber über das SIEM-Meldungen an das ACDC-Projekt übermittelt werden können, müssen die *Erforderlichen Felder* mindestens ausgefüllt werden. Da das ACDC-Projekt sehr gut definierte und klar strukturierte Beschreibungen der *Reportkategorien* sowie der dazugehörigen *Unterkategorien* besitzt, kann (sofern das SIEM des Betreibers die benötigten Daten liefert) ein Parser für das bzw. die notwendigen Datenfelder erstellt werden. Es gibt verschiedene Attack-Formate, denen ein jeweiliger Angriffstyp zugeordnet wird, um so einen strukturierten und schnellen Workflow zu gewährleisten. Zur Demonstration wurden die Typen „Attack“ und „Malware“ genutzt. Der Typ “Attack” wird dazu genutzt, um den Angriff eines Hosts auf den anderen zu melden. “Malware” ist eine Unterkategorie von diesem Typ und hat den Zweck, einen spezifischen Angriff zu dokumentieren, bei welchem der Angreifer versucht sein Ziel mit einer Schadsoftware zu infizieren (wie zum Beispiel einem “Wurm”). Weitere Formate können aus den Berichten[1] D1.7 Data Format Specification entnommen werden.

Die Implementierung umfasst sowohl die Client- als auch die Server-Seite. Als Programmiersprache wurde Python gewählt. Python bietet native Unterstützung für Datenformate wie JSON, XML und andere, die im Rahmen des ACDC-Projekts genutzt wurden. Python enthält außerdem eine Vielzahl an Libraries und eine Unterstützung für SQLite (z. B. das sqlite3-Modul), was eine reibungslose Integration mit der gewählten Datenbank ermöglicht, so wie Flask, welches eine leichtgewichtige Möglichkeit bietet Webanwendungen zu erstellen. Die SQLite-Datenbank ist in einer einzigen Datei gespeichert und lässt sich in Kombination mit Flask einfach auf andere Systeme übertragen und deployen.

Evaluation von Scanning-Tools für den Demonstrator

Zur Erhebung von Asset Informationen zum Matching von CSAF Advisories, wurden verschiedene Scanning Tools evaluiert. Da durch passives Monitoring nicht genaue Asset Informationen erhoben werden können, wurden insbesondere aktive Scanning Tools für den Demonstrator untersucht. Es kam auch zu einigen Gesprächen mit dem BSI, worin diese auch erwähnten, dass sie die Aussage, dass in ICS-Netzen nicht aktiv gescannt werden sollte, herausfordern. Im Folgenden wurden die

[1] https://www.acdc-project.eu/documents/acdc-deliverables/

Tools aus dem Paper „A Taxonomy for Contrasting Industrial Control Systems Asset Discovery Tools“ [2] untersucht.

		SIMATIC	Modscan	Nmap	Plcscan	Grassmarlin	NetworkMiner	Sophia	Lansweeper	SCADA-CIP	Wireshark	Nessus	OpenVAS	scada-tools	s7scan	Redpoint	ETTERCAP	OWASPNettacker	Unicornscan	nmap-scada	icsmaster	Modbusdiscover	scadascan	s7-info	plc-scanner	ICS-Hunter	ModbusScanner	ICSY	cyberlens
Specification	Bundled	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗
	Standalone	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✗	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓
	Commercial	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Open source	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Shareware	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Freeware	✗	✗	✗	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
	Single target	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗
	Wide target	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓
	Single	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✗
	Multiple	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✓
Execution	Passive	✗	✗	✗	✗	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
	Active	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
	Manual	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Automatic	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Interactive	✗	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗
	Point and click	✓	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓
	Offline	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
	Real-time	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Port scanning	✗	✓	✓	✓	✗	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
	ICMP scanning	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗
	ARP scanning	✗	✗	✓	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Banner grabbing	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Fingerprinting	✗	✗	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
	Automation protocols	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Internet protocols	✗	✗	✓	✗	✓	✓	✗	✓	✗	✓	✓	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Output/ Scanning Depth	1 Active IP address	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2 Listening ports	✓	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗
	3 Protocol and service identification	✓	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓
	4 Static device info	✓	✗	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✓
	5 Deployment specific info	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗
	6 Vulnerability identification	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

✓ –> applicable | ✗ –> non-applicable

Abbildung 8: Tabellenübersicht der Scanning Tools [2]

Die Tools wurden im virtuellen Testbed evaluiert. Dies setzt sich aus zwei Modbus Master/Slave paaren zusammen und aus einer virtuellen Siemens S7 1500er PLC, welches über S7comm-plus mit einer virtuellen HMI kommuniziert. In unseren Tests wurden fast alle Tools evaluiert. Die Simulationsumgebung beschränkte sich auf das Modbus on S7comm Protokoll. Es wurden jedoch auch Tools, die andere OT-Protokolle unterstützen untersucht, ob eventuell dessen Scanergebnisse nützliche Asset Informationen erheben können. In erster Hinsicht konnten die erwarteten Scanergebnisse mit den verschiedenen nmap Skripten von Redpoint und S7-info erzielt werden. Mit der Flexibilität von nmap, die Scanergebnisse auch im XML-Format exportieren zu können, und der anpassbaren Scangeschwindigkeit, eignete es sich in diesem Forschungsprojekt sehr gut für die Weiterverarbeitung der Daten im SIEM und dem Einsatz in OT-Umgebungen. S7scan würde auch als Python Tool in Frage kommen. Es liefert überwiegend dieselben Scanergebnisse wie S7-info für nmap, konnte jedoch zusätzlich noch den CPU-Typen auslesen. In den Meetings mit den Projektpartnern wurde besprochen, dass die nmap OS-Detection nicht präzise genug ist. Aus diesem Grund wurde eine alternative Lösung für die OS-Erkennung untersucht. Um genauere OS-Erkennungen zu erzielen, wurde Windows WMI (Windows Management Instrumentation) betrachtet. Ein großer Vorteil von WMI ist, dass keine Scans mehr notwendig sind, da sich die benötigten Informationen gezielt von den Hosts abfragen lassen. Dies reduziert auch den Netzwerk-Traffic des Demonstrators. Jedoch hat es den Nachteil, dass die Credentials vom Betreiber hierfür dann auf der Security Appliance gespeichert werden müssten. Da Betreiber ihre Credentials für eine externe Security Appliance nicht herausgeben, wurde sich entschieden bei nmap zu verbleiben.

Security Assessment

Da sich kein Industriepartner für die Feldtests finden ließ, wurden diese innerhalb der virtuellen Simulationsumgebung durchgeführt. Um das Verhalten eines Angreifers zu simulieren und Schaden zu verhindern, gibt es in der IT-Sicherheit die Methode des Penetrationstests, oder kurz Pentest. Das Verfahren wird zur Entdeckung und Analyse von Schwachstellen in Systemen eingesetzt. Hierbei werden die zu untersuchenden Systeme auf Schwachstellen geprüft und ausgenutzt. Für Pentests im deutschsprachigen Raum existiert ein Durchführungskonzept des Bundesamtes für Sicherheit in der Informationstechnik [3]. Nach Abschluss eines solchen Tests wird die geprüfte Entität über die Existenz der gefundenen Schwachstellen informiert, um ggf. Gegenmaßnahmen ergreifen zu können. In diesem Zusammenhang wurde der Demonstrator einem Security Assessment in der virtuellen Simulationsumgebung unterzogen. Da der Demonstrator zum derzeitigen Zeitpunkt keine OT-Protokoll interpretieren konnte, wurde für das Security Assessments die IRMA als Sensor in der Simulationsumgebung benutzt. Die IRMA ist eine passive Security Appliance von achtwerk, welches für Industrieumgebungen entwickelt wurde und somit auch OT-Protokolle unterstützt [4]. Demnach wurde die IRMA an den Mirror Ports in den ICS Netzsegmenten verbunden, wodurch sie somit die erhobenen Daten an den Demonstrator über eine API bereitstellt. Um zu testen, ob die Mirror Port Anbindung der IRMA sowie die Anomalie Erkennung funktioniert, wurden verschiedene Netzwerkscans durchgeführt, welche der Sensor erfolgreich erkennen konnte. Für das Security Assessment wurden zwei Angriffsszenarien ausgearbeitet und mit den Projektpartnern abgestimmt. Die Angriffsszenarien haben beide dasselbe Ziel, einen Denial-of-Service auszulösen, indem die PLC ausgeschaltet wird. Im ersten Angriffsszenario war der Angreifer direkt im ICS-Netz mit der PLC positioniert, siehe *Abbildung **9*** (rot markiert).

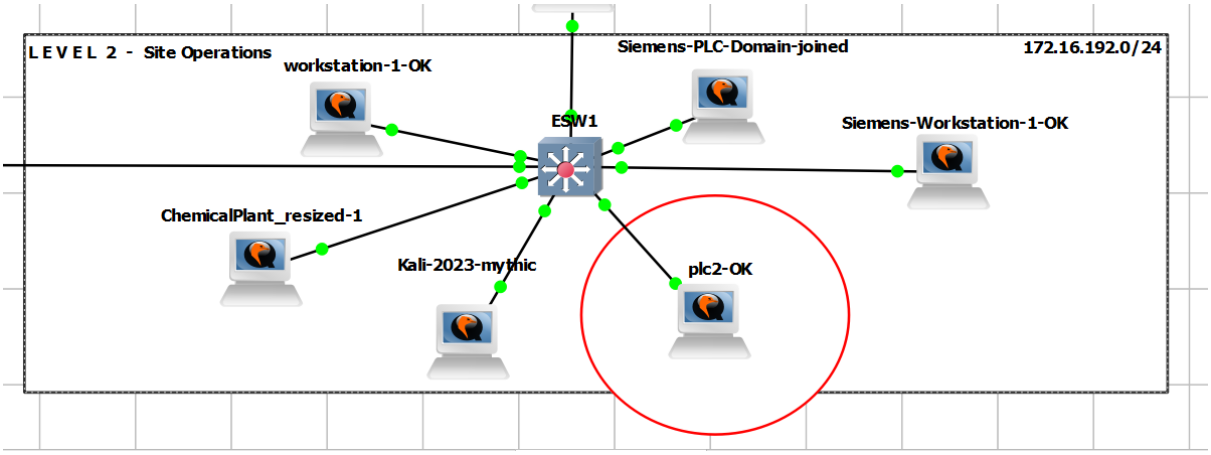

Abbildung 9: Simulationsumgebung – Level 2 Netzwerk

Die durchgeführten Angriffe im ersten Angriffsszenario:

- MITM-Angriff mittels ARP-Spoofing
- Auslesen der Modbus PLC-Adresse
- Ausschalten der Modbus PLC

Im zweiten Angriffsszenario wird im Wesentlichen derselbe Angriff behandelt, mit dem Unterschied, dass der Angreifer hier extern positioniert ist. In diesem Fall wird angenommen, dass der Angreifer bereits VPN Credentials kompromittiert hat. Die Firewall lässt ausschließlich RDP-Verbindungen vom Windows 10 Jump Host zur Windows 10 HMI in das „Level 3 ICS“-Netzwerk durch. Der Angriff fand in zwei Phasen statt. In der ersten Phase wurde das Pivoting vom Industrial DMZ in das „ICS-Level 3“-Netzwerk durchgeführt.

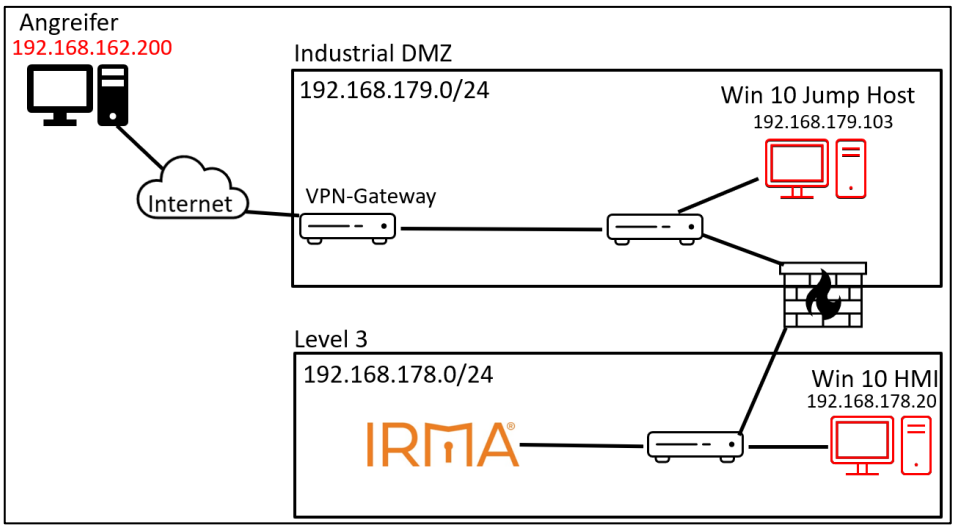

Abbildung 10: Darstellung – Der Externe Angreifer kompromittiert die Windows 10 HMI

Die zweite Phase war identisch mit dem ersten Angriffsszenario, worin ein Denial-of-Service durch das Ausschalten der PLC verursacht wurde. Da der Angreifer in diesem Szenario extern positioniert war, wurde das C2 Framework Sliver zum Verwalten der kompromittierten Systeme verwendet. Erzielt wurde das Ergebnis durch das Verwenden der Win10 HMI als Proxy für den Angriff auf die PLC.

Die durchgeführten Angriffe im zweiten Angriffsszenario:

- Pivoting in das Level 3 ICS-Netzwerk
- SOCKS Proxy zum Tunneln von Netzwerkverkehr des Angreifers in das ICS-Netzwerk
- Auslesen der Modbus PLC Adresse
- Ausschalten der Modbus PLC

Beide Angriffsszenarien konnten erfolgreich durchgeführt werden. Diese wurden in einem Bericht dokumentiert und den Projektpartnern für die Auswertung zur Verfügung gestellt.

Asset Management

Im Bereich „Asset Management“ wurden sowohl kommerzielle als auch „Open Source“-Asset Management-Lösungen bezüglich ihrer Funktionalität und Eignung für OT-Infrastrukturen analysiert. Dabei wurden Datenimport/-export-Möglichkeiten für potentielle Schnittstellen zum SIEM-System untersucht. Insbesondere wurden „Open Source“-Lösungen detailliert auf ihre Eignung für OT-Umgebungen getestet. Ein wichtiges Kriterium für Asset Management-Systeme in OT-Infrastrukturen ist minimale Belastung der OT-Infrastruktur bzw. des OT-Netzwerks. Es wurden hierzu diverse Open Source-Lösungen diesbezüglich untersucht. Dies umfasste die Prüfung der Auswirkungen von Inventarisierungsprozessen auf ICS-Netzwerke und erfasste Informationen sowie die Untersuchung der Netzwerkauslastung (-bzw. Last) während Scans, um Überlastungsrisiken für Systeme und Kommunikationswege zu bewerten. Da Tests in produktiven OT-Umgebungen nicht möglich waren, wurde Fortiphyd Virtual Training Grounds als virtuelle ICS-Versuchsumgebung genutzt. Fortiphyd Virtual Training Grounds besteht aus zwei Netzwerksegmenten: dem industriellen Bereich (ICS) und der DMZ. Der industrielle Bereich umfasst eine Workstation, eine PLC und eine VM für die Simulation einer Chemiefabrik. Die DMZ enthält ein SCADA-System sowie zusätzliche VMs für GLPI, OCS und Kali-Linux. Eine pfSense-Firewall überwacht den Datenverkehr zwischen den Segmenten. Diese Struktur ermöglichte realistische Tests und Sicherheitsanalysen in einer virtuellen ICS-Umgebung.

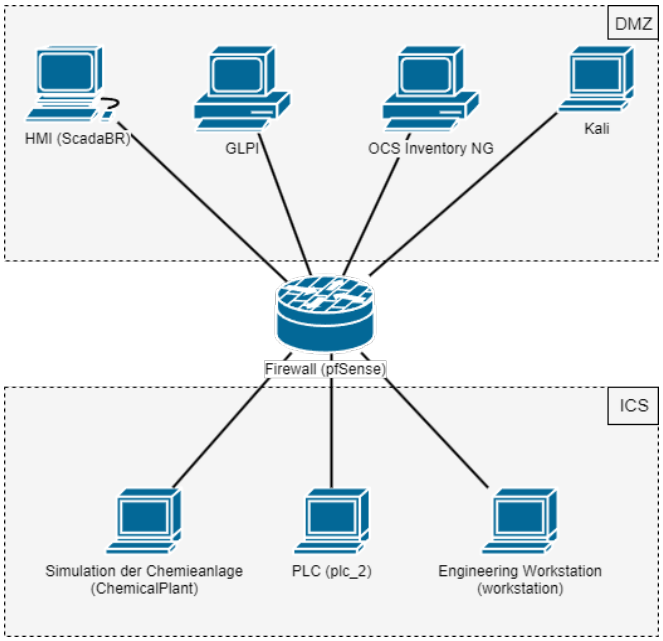

Abbildung 11: Aufbau der Testumgebung

System	Betriebssystem	Aufgabe	IP-Adresse
HMI (ScadaBR)	Linux ScadaBR Software (Debian 4.9)	Überwachen und Steuern der PLC	192.168.90.5
GLPI	Ubuntu 23.04	Bereitstellen der Asset Management-Lösung GLPI-Server und -Agent	192.168.90.114
OCS Inventory NG	Ubuntu 22.04	Bereitstellen der Asset Management-Lösung OCS-Server und -Agent	192.168.90.115
Kali	KaliLinux 2023.2	Nmap	192.168.90.116
Firewall (pfSense)	FreeBSD	Routing- und Firewall-Funktionen zwischen DMZ und ICS	192.168.95.1
Simulation der Chemieanlage (ChemicalPlant)	Ubuntu 22.04	Simulation einer chemischen Industrieanlage, gesteuert und überwacht von simulierten Remote-IO-Geräten	192.168.95.10 192.168.95.11 192.168.95.12 192.168.95.13 192.168.95.14 192.168.95.15

PLC (plc_2)	Ubuntu 16.04.4 LTS mit OpenPLC	Steuerung der ChemicalPlant	192.168.95.2
Engineering Workstation (workstation)	Ubuntu 16.04 LTS	Client-System	192.168.95.5

Tabelle 2: Komponenten der Testumgebung

Die Testumgebung wurde lokal in VirtualBox mit Host-only Adaptern aufgebaut, um isolierte Kommunikation zu gewährleisten. Das Host-System: Windows mit Intel Core i7-10750H, 32GB RAM. ICS-Kommunikation erfolgt über Modbus/TCP. Da direkte VM-Auslastungsmessungen nicht möglich waren, wurde die Prozessorzeit auf dem Host-System während Netzwerk-Scans gemessen. Zusätzlich wurden Privileged und User Processor Time sowie zugewiesener Arbeitsspeicher erfasst. Tests wurden durchgeführt für:

- OCS v2.11.1 (Varianten: "ping" und "nmap")
- GLPI v10.0.7 mit Agent v1.5-1/Toolbox-Plugin v1.1 (20 und 40 Threads)
- Nmap v7.94 (stellvertretend für NetBox und Open-AudIT)

Für die Netzwerkbelastungstests wurde die Versuchsumgebung in die virtuelle ZenSIM4.0-Laborumgebung der Hochschule Bremen überführt. Diese wurde mit GNS3, einer Open-Source-Software zur Netzwerksimulation, modelliert. GNS3 bietet eine grafische Benutzeroberfläche und ermöglicht das Testen von Netzwerktopologien ohne physische Hardware. Als Virtualisierungslösung diente ein Typ-1-Hypervisor (KVM).

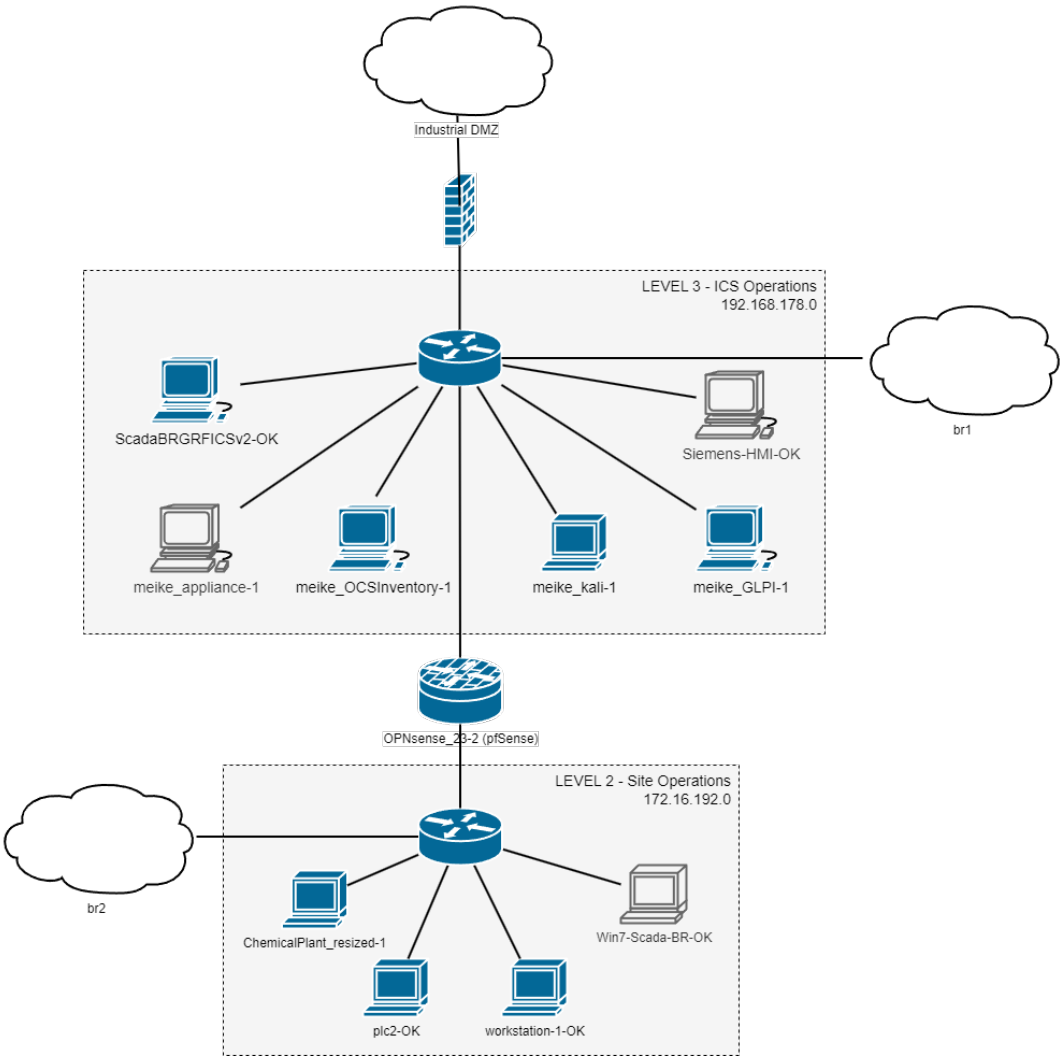

Abbildung 12: Ausschnitt der Versuchsumgebung

System	Betriebssystem	Aufgabe	IP-Adresse
HMI (ScadaBRGRFICSv2-OK)	Linux ScadaBR Software (Debian 4.9)	Überwachen und steuern der PLC	192.168.178.30
GLPI (meike-GLPI-1)	Ubuntu 23.04	Bereitstellen der Asset Management-Lösung GLPI-Server und -Agent	192.168. 178.121
OCS (meike-OCSInventory-1)	Ubuntu 22.04	Bereitstellen der Asset Management-Lösung OCS-Server und -Agent	192.168.178.122
Kali (meike-kali-1)	KaliLinux 2023.2	Nmap	192.168.178.123

System	Betriebssystem	Aufgabe	IP-Adresse
Firewall OPNsense (pfSense)	FreeBSD	Routing- und Firewall-Funktionen zwischen Level 2 und Level 3	172.16.192.100

Tabelle 3: Ausschnitt der Versuchsumgebung - Komponenten

Die Netzwerkauslastung wurde mit iPerf3, einem Werkzeug zur Messung der Netzwerkleistung, ermittelt. iPerf3 wird vornehmlich von ESnet bzw. dem Lawrence Berkeley National Laboratory entwickelt. Es ist unter der Open Source-Lizenz BSD veröffentlicht.

iPerf3 liefert folgende Messwerte:

- Datenmenge (Transfer)
- Bandbreite (Bandwidth)
- Re-Transmissions (Retr)
- Congestion Window (Cwnd)

Dabei wurden Datenmenge und Bandbreite Server- und Clientseitig gemessen, Re-Transmissions und Congestion Window nur Clientseitig.

Die Ergebnisse der Untersuchungen sind in *Tabelle **4*** zusammengefasst.

Anwendung		Open Source				Kommerziell	
		OCS	**GLPI**	**Open-AudIT**	**NetBox**	**Lansweeper OT**	**Claroty CTD**
Strategie	Lösung ist speziell an OT-Infrastrukturen angepasst	Nein	Nein	Nein	Nein	Ja[2]	Ja[2]
	Agenten werden eingesetzt	Ja	Ja	Nein	Nein	Nein	Nein
	Primäre Scan-Methode	Client-System passiv, Netzwerk aktiv	Client-System passiv, Netzwerk aktiv	Aktiv	Kein Scan vorgesehen	Selective Probing	Passiv
Asset-Inventarisierung	Erhebung erfolgt nach Möglichkeit rückwirkungsfrei	Teilweise	Eher nicht	Nein	Keine Asset-Inventarisierung	Ja[2]	Ja[2]
	Aktive Scans erfolgen über spezifische Protokolle	Nein	Nein	Nein		Ja[2]	Ja[2]
	Aktive Scans gehen besonders vorsichtig vor, um kein System oder das Netzwerk zu überlasten	Teilweise	Eher nicht	Nein		Ja[2]	Ja[2]

[2] Herstellerangabe

Anwendung		Open Source				Kommerziell	
		OCS	GLPI	Open-AudIT	NetBox	Lansweeper OT	Claroty CTD
	Unterschiedliche Methoden und Datenquellen werden verwendet	Ja	Ja	Ja		Ja[2]	Ja[2]
	Im- bzw. Export von Daten unter Verwendung eines Standardformats ist möglich	Ja	Ja	Ja		Ja[2]	Ja[2]
	Spezifische Protokolle aus OT-Infrastrukturen sind verfügbar	Nein	Nein	Nein		Ja[2]	Ja[2]
	Verteilte Infrastrukturen werden abgedeckt	Falls ein geeignetes System für einen Agenten existiert		Ja		Ja[2]	Ja[2]

Tabelle 4: Überprüfung der Asset Management Lösungen gegen spezifische OT-Anforderungen

Kommerzielle Lösungen bevorzugen passive Netzwerk-Scans mit gezieltem Einsatz aktiver Scans. Open Source-Lösungen nutzen passive Scans nur für Client-Systeme mit Agenten und sind nicht für OT-Infrastrukturen konzipiert. Tests zur Systembelastung waren nicht aussagekräftig. Netzwerkbelastungstests zeigten, dass GLPI-Scans unnötige Anfragen senden, während OCS effizienter arbeitet. Nmap erwies sich als besonders effizient mit hoher Datenqualität bei geringer Netzwerkbelastung. Empfehlungen:

- GLPI: Prüfen, ob SNMP- und NBNS-Scans optimiert werden können.
- Open-AudIT: Möglichkeit der Anpassung für OT-Infrastrukturen untersuchen.
- NetBox: Erfüllt viele Kriterien nicht, da es nicht als vollständige Asset Management-Lösung konzipiert ist.

Darüber hinaus wurde ein API zur Bereitstellung von Asset-Daten für ein SIEM konzeptioniert und entwickelt. Die Schnittstelle sollte grundsätzlich für das SIEM-System ScanBox® des Herstellers DECOIT entwickelt werden.

Hierzu wurden folgende technische Anforderungen definiert:

- Verfügbarkeit als Service
- Implementierung als REST-API
- Möglichkeit der Integration in ein beliebiges SIEM-System, unabhängig von der ScanBox®
- Erreichbarkeit über eine Web-GUI
- Programmierung in Python oder Java

Mittels der Schnittstelle sollten Asset-Daten zum Import in die Software ScanBox® bereitgestellt werden. Die Schnittstelle gibt die Daten im JSON-Format zurück, das in ScanBox® importiert

werden kann. Die Komponente sollte als Service verfügbar sein. Dazu wurde eine Appliance mit der verfügbaren API bereitgestellt, die in ein bestehendes Netzwerk integriert werden kann und durch die ScanBox® erreichbar ist. Die Programmierschnittstelle sollte als REST-API vorgelegt werden. Die Ausgestaltung der Programmierschnittstelle als Service gewährleistet, dass diese in ein beliebiges SIEM-System integrierbar ist. Das Ausgabeformat der Schnittstelle ist allerdings an die ScanBox® angepasst. Neben dem direkten Zugriff auf die API sollte ein Zugriff über eine Web-GUI in einem Webbrowser möglich sein. Für den direkten Zugriff wurde in der Entwicklung cURL (Clients for URLs) verwendet, ein Kommandozeilenprogramm zur Übertragung von Dateien in Rechnernetzen [15]. Zusätzlich ist eine Web-GUI implementiert, die mit Hilfe eines Webservers bereitgestellt wird. Für die Implementierung der Programmierschnittstelle wurde die Programmiersprache Python3 gewählt. Die Hauptanwendungen dieser vielseitigen und effizienten Programmiersprache sind Data Science, Webentwicklung und Skripterstellung. Python3 selbst stellt bereits eine sehr umfassende Standardbibliothek bereit.

Seitens DECOIT wurden bestimmte Informationen definiert, die für ScanBox® möglichst bereitgestellt werden sollen. Diese sind aus den vorhandenen Inventarlisten zu extrahieren. Es wurden Informationen beschrieben, die für ein effizientes Asset Management interessant sind. Es besteht die Möglichkeit, auch diese Informationen mit der hier entwickelten Schnittstelle zu extrahieren. *Tabelle **5*** gibt einen Überblick über alle Informationen, die durch die Schnittstelle aus Inventarlisten extrahiert werden können. Für den Import in ScanBox® werden Daten in einem definierten JSON-Format benötigt.

Da Asset-Inventare, insbesondere in KMU, häufig in manuell gepflegten Listen geführt werden, liegen sie meist als individuelle Datenbanken, Excel-Dateien[3] oder CSV-Dateien vor. Aus softwaregestützten Asset Management-Lösungen können die dort erfassten Daten üblicherweise in CSV-Dateien exportiert werden. Inventardateien sind aufgrund ihrer verschiedenen Quellen in Anordnung und Art der dokumentierten Daten sehr unterschiedlich aufgebaut. Sie können in jedem Unternehmen eine individuelle Ausprägung annehmen. Standardisierte Formatvorgaben auf Feld-Ebene gibt es nur für universelle Informationen wie MAC-Adressen, IP-Adressen und Datumsangaben. Alle weiteren Felder können Freitexte enthalten. Das automatisierte Erfassen oder direkte Parsen der Informationen ist also nicht ohne weiteres möglich. Vielmehr besteht die Notwendigkeit, ein Mapping von Ort und Informationen an die Programmierschnittstelle zu übergeben. Das bedeutet, dass benannt werden muss, welche Information sich in welcher Spalte eines Inventars befindet bzw. ab welcher Zeile Inhalte zu erwarten sind. In der Schnittstelle wird die Zuordnung von Datenfeldern zu den gesuchten Informationen als JSON-String in der Anfrage übergegeben. Der einem JSON-Feld zugeordnete Wert entspricht der Spaltenüberschrift der Inventartabelle. Alternativ können Spalten auch über Indexe identifiziert werden.

[3] Alternativ Formate anderer Tabellenkalkulations-Software, z.B. OpenDocument-Spreadsheet

Name	Beschreibung	Anforderung von ScanBox®	Beispiel	Anmerkung
updated_at	Datum der letzten Aktualisierung des Repositories	Ja	2023-05-24 oder 2023-04-13 11:02:00	Information kann optional im Request angegeben werden, String
Mac	MAC-Adresse	Ja	08:3A:88:5A:A0:93	String
Ip	IP-Adresse	Ja	10.0.30.1	String
name	Name des Assets	Ja	PLC	String
description	Beschreibung des Assets, Verantwortliche, Kritikalität und weitere Informationen	Nein		String
vendor	Herstellername	Ja	RockwellAutomation	Hardware, String
serial_number	Seriennummer des Assets	Ja	00987DBF	Hardware, String
model_number	Modell-ID des Assets	Ja	1756-ENBT/A	Hardware, String
product_name	Software-Produktname / Firmware	Ja	V6.006	OS / Firmware, String
product_version	Produktversion bzw. Patch-Level	Ja	SP1	Version oder Patch-Level, String
Purl	Persistent Uniform Resource Locator	Ja	/W3C/RDF	String
Cpe	Common Plattform Enumeration (Industriestandard)	Ja	cpe:/o:redhat:enterprise_linux:3:ga:desktop	String
hashes		Ja	43E56DC298B6A7682D74	String
sbom_urls		Ja		SBOM[4] in XML oder JSON, String
hostnames		Ja	H10_L1STA20_2	String
hostnames_order		Ja	1	String
Port	Portnummer	Ja	2000 oder 44818/tcp	String
protocol	Protokollname	Ja	Modbus RTU	String
connection	Informationen zur verwendeten Netzwerkschnittstelle	Nein	LAN	String
location	Informationen zum Standort, Rack	Nein	Halle 2, Quadrant A1	String

Tabelle 5: Übersicht der zu extrahierenden Informationen

Folgende Dateiformate werden für die Eingabe unterstützt:

- Excel-Arbeitsmappe, Dateiendung: xlsx
- Comma-Separated Values, Dateiendung: csv
- Excel-Arbeitsmappe 97-2003, Dateiendung: xls
- OpenDocument-Kalkulationstabelle, Dateiendung: ods

[4] Software Bill of Materials (Software-Bestandsliste) bezeichnet eine Liste aller Komponenten und Abhängigkeiten, die in einer Softwareanwendung enthalten sind, ähnlich einer Auflistung der Inhaltsstoffe auf einem Lebensmittel

Für das Datenformat CSV kann optional angegeben werden, welches Trennzeichen in der Datei verwendet wird (Defaultwert: ",“). Über das Mapping wird definiert, welche Informationen aus dem Inventar gewonnen werden sollen. Das hier definierte Mapping mit allen Schlüsseln zeigt *Abbildung **13*** die zugeordneten Werte sind Beispiele. Das Mapping kann angepasst werden, Schlüssel-Wert-Paare können weggelassen werden. Es muss aber mindestens ein Wert für die Schlüssel „mac“ oder „ip“ angegeben werden. Es muss berücksichtigt werden, dass ggf. mehrere Zeilen als Überschrift verwendet werden. Daher ist eine optionale Angabe möglich, in welcher Zeile das erste Asset zu finden ist, wie viele Zeilen also zu Beginn der Datei übersprungen werden müssen. Der Standardwert ist hier 0 (keine Zeile). Ein optionaler, von DECOIT gewünschter Wert ist das Datum der letzten Aktualisierung. Dieser Wert soll einmalig zum Upload für alle Assets angegeben werden. Er kann für jedes Asset auch aus dem Inventar übernommen werden, falls dies im Mapping angegeben wird.
Inventardateien sollten in ein standardisiertes Format überführt werden, um sie in andere Asset Management-Lösungen oder in ein SIEM importieren zu können. Dazu eignen sich die aktuellen, populären Formate XML und JSON. Für die ScanBox® soll JSON für den Import genutzt werden, daher werden die ermittelten Daten in einer JSON-Liste ausgegeben. Falls angegeben wird je Asset das Datum der letzten Aktualisierung bereitgestellt. Das JSON-Schema der Rückgabe wurde auf Basis der Anforderungen von DECOIT festgelegt.


```
{
            "updated_at": "2023-05-24",
            "asset_name": "PLC",
            "asset_description": "string",
            "mac": [
                  "08:3A:88:5A:A0:93",
                  "F0:77:C3:94:7B:9F"
            ],
            "ip": [
                  "192.178.0.1"
            ],
            "vendor": "RockwellAutomation",
            "serial_number": "00987DBF",
            "model_number": "1756-ENBT/A",
            "product_name": "V6.006",
            "product_version": "SP1",
            "purl": "/W3C/RDF",
            "cpe": "cpe:/o:redhat:enterprise_linux:3:ga:desktop",
            "hashes": "sha256:8d3ac3489996423f53d6087c811800hjgsdf7ceb8759",
            "sbom_urls": "string",
            "hostnames": ["Host10_A20_1, Host10_A20_2"],
            "hostnames_order": ["1, 2"],
            "port": {
                        "portnumber": "44818",
                        "transport": "tcp"
            },
            "protocol": "Modbus RTU",
            "connection": "LAN",
            "location": "Halle 2, Quadrant A1"
      }
```


Abbildung 13: Beispielhafte Rückgabe der API für ein Asset mit allen möglichen Schlüssel-Wert-Paaren

Ein wichtiger Aspekt bei der technischen Umsetzung ist die Zustandslosigkeit der Programmierschnittstelle. Eine Anfrage vom Client an den Server übergibt alle nötigen

Informationen, die für die Verarbeitung der Anfrage erforderlich sind. Der Server speichert keinen Zustand über den Client und erfragt keine weiteren Daten.

Darüber hinaus werden durch die Programmierschnittstelle keine Daten persistent vorgehalten. Eine eingereichte Datei mit einem Asset-Inventar wird im Arbeitsspeicher des Servers verarbeitet und in das neue Format überführt. Da nicht zwingend notwendig, werden keine Daten auf dem Filesystem des Servers gespeichert oder in einer Datenbank abgelegt. Somit ist die Gefahr eines Datenlecks oder unbefugten Zugriffs auf langfristig gespeicherte Daten nicht gegeben. Daten, die nicht persistiert sind, gehen spätestens verloren, wenn der Server heruntergefahren wird.[5] Die potenziell sensiblen Daten sind nur für die Dauer ihrer Verarbeitung im RAM[6] sowie während ihrer Übertragung zwischen Client und Server in der Verantwortung der Schnittstelle. Somit ist die Gefahr der Verletzung von Vertraulichkeit, Integrität und Authentizität eingegrenzt. Der Zugriff auf den RAM eines Servers durch einen Angreifer erfordert fortgeschrittene Kenntnisse sowie spezielle Angriffsmethoden und ist deshalb nicht einfach. Die Schnittstelle ist außerdem nur in der isolierten Umgebung eines Unternehmensnetzwerks erreichbar. Soll trotzdem eine weitere Sicherheitsebene eingeführt werden, so ist es leicht möglich, einen Verzeichnisschutz mit Basic Authentication zu implementieren. [14] Um die Möglichkeit von Denial-of-Service-Angriffen einzuschränken, wird die Größe der Dateien, die hochgeladen werden können, beschränkt [16].
Um Angriffsformen wie Cross-Site-Scriptig (XSS) und UI Redressing (Clickjacking) abzuwehren, verwendet das Web-Frontend eine Content Security Policy (CSP) (siehe Abbildung 14) [14].

```python
def start_application():
    app = FastAPI(title=settings.PROJECT_NAME, version=settings.PROJECT_VERSION)

    @app.middleware("http")
    async def modify_request_csp(request: Request, call_next):
        response = await call_next(request)
        csp = "default-src 'none';script-src 'self';style-src 'self';img-src 'self';frame-ancestors 'self'"
        if isinstance(response, StreamingResponse):
            response.headers["Content-Security-Policy"] = csp
        return response

    app.add_middleware(HTTPSRedirectMiddleware)
    app.mount("/static", StaticFiles(directory="static"), name="static")
    include_router(app)
    return app
```

Abbildung 14: Implementierung der Content Security Policy und des erzwungenen HTTPS-Redirect durch Middleware-Komponenten [17]

Bei der Übertragung der Daten zwischen Client und Server ist die Vertraulichkeit und Integrität durch eine zertifizierte HTTPS[7]-Verbindung sichergestellt (siehe Abbildung 14). Die Daten, die zwischen Client und Server übertragen werden und über die gesamte Infrastruktur eines Unternehmens Aufschluss geben können, werden bei der Übertragung verschlüsselt. Dies verhindert passive Angriffe (Lauschangriffe), bei denen ein Angreifer den Datenverkehr abhört und sensible Informationen abgreift. HTTPS setzt verschiedene Integritätsprüfungen um, so dass die Daten auch vor Veränderung geschützt sind. Die Authentizität wird durch die Verwendung von digitalen Zertifikaten sichergestellt, die die Identität des Servers und bestenfalls auch des Clients verifizieren [14].

Die Beschreibung der Schnittstelle im OpenAPI Format (Version 3.0.2) ist im folgenden Listing zu finden.

[5] Die Entscheidung keine Daten zu persistieren kann auch die Einhaltung von Datenschutzvorschriften wie der Datenschutz-Grundverordnung (DSGVO) erleichtern. Daten, die nicht gespeichert werden, unterliegen weniger strengen Datenschutzbestimmungen.

[6] Ab einer bestimmten Größe wird eine Datei nicht direkt im Arbeitsspeicher verarbeitet, sondern zuvor temporär gespeichert. Pythons interne Mechanismen löschen die Datei nach Beendigung der Verarbeitung automatisiert.

[7] HTTPS verwendet HTTP, fügt jedoch eine Verschlüsselungsschicht basierend auf SSL/TLS hinzu.

```yaml
openapi: 3.0.2
info:
  title: Frollein's Inventory Converter
  description: >-
    This API converts a list of asset inventories and all information about
    the individual assets, which is available in the formats Excel, CSV or
    OpenDocument into a JSON format.
  version: 1.0.0
  contact:
    email: api@frollein-web.de
servers:
  - url: /v1
tags:
  - name: inventories
    description: Convert an asset inventory list into JSON.
paths:
  /repository/json/:
    post:
      tags:
        - inventories
      description: Provides an asset-inventory in JSON-format.
      requestBody:
        required: true
        content:
          multipart/form-data:
            schema:
              $ref: '#/components/schemas/data'
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/asset'
        '400':
          description: The provided parameters are incomplete or incorrect.
          content:
            text/html:
              schema:
                $ref: '#/components/schemas/errorMessage'
              examples:
                incorrectFile:
                  value: >-
                    Error while processing the file: The file format is not
                    permitted. Please provide a CSV-, Excel- or OpenDocument-
                    file.
                incorrectMapping:
                  value: >-
                    Fatal error: The mapping was not provided in a valid JSON
                    format.
  /repository/html/:
    post:
      tags:
        - inventories
      description: Displays the converted inventory in HTML including
      messages.
      requestBody:
        required: true
        content:
          multipart/form-data:
            schema:
              $ref: '#/components/schemas/data'
      responses:
        '200':
          description: OK
          content:
            application/html:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/asset'
        '400':
          description: The provided parameters are incomplete or incorrect.
          content:
            text/html:
              schema:
```

```
              $ref: '#/components/schemas/errorMessage'
            examples:
              incorrectFile:
                value: >-
                  Error while processing the file: The file format is not
                  permitted. Please provide a CSV-, Excel- or OpenDocument-
                  file.
              incorrectMapping:
                value: >-
                  Fatal error: The mapping was not provided in a valid JSON
                  format.
components:
  schemas:
    data:
      description: Information to convert the inventory
      type: object
      required:
        - file
        - mapping
        - output
      properties:
        file:
          description: >-
            Path and filename of the inventory.
            Possible file formats are Excel (xlsx and xls), CSV and
            OpenDocument (odb).
          type: string
        mapping:
          description: >-
            Set of mapping information indicating which column contains which
            information. For Excel files the column name, for CSV the column
            number should be assigned. All columns are optional. Specify only
            the columns that are included in your inventory file or that you
            want to receive in your output.
          type: object
          properties:
            asset_name:
              type: string
            asset_description:
              type: string
            mac:
              type: string
            ip:
              type: string
            manufacturer:
              type: string
            serial_number:
              type: string
            model_number:
              type: string
            product_name:
              type: string
            product_version:
              type: string
            purl:
              type: string
            cpe:
              type: string
            hashes:
              type: string
            sbom_urls:
              type: string
            hostnames:
              type: string
            hostnames_order:
              type: string
            port:
              type: string
            protocol:
              type: string
            connection:
              type: string
            location:
              type: string
            update:
              type: string
        updated_at:
          description: Date of the last update of the asset inventory in the
```

```
        format yyyy-mm-dd
      type: string
    separator:
      description: For CSV-files, select the used separator
      type: string
    start_at:
      description: If the head line ist not the first row, number of rows
        to skip at the beginning (0-indexed)
      type: integer
    sheet_name:
      description: If the file contains more than one sheets, give the
        sheet number which contains the information (0-indexed)
      type: integer
    output:
      description: (Only web frontend) select the output-type (HTML or
        JSON)
      type: string
asset:
  type: object
  description: Information of an asset
  properties:
    updated_at:
      type: string
      example: "2023-05-24"
    asset_name:
      type: string
      example: "PLC"
    asset_description:
      type: string
    mac:
      type: string
      example: "08:3A:88:5A:A0:93"
    ip:
      type: string
      example: "192.178.0.1"
    vendor:
      type: string
      example: "RockwellAutomation"
    serial_number:
      type: string
      example: "00987DBF"
    model_number:
      type: string
      example: "1756-ENBT/A"
    product_name:
      type: string
      example: "V6.006"
    product_version:
      type: string
      example: "SP1"
    purl:
      type: string
      example: "/W3C/RDF"
    cpe:
      type: string
      example: "cpe:/o:redhat:enterprise_linux:3:ga:desktop"
    hashes:
      type: string
      example: "sha256:8d3ac3489996423f53d6087c81180006263b79f206d3fdec9e66f0e27ceb8759"
    sbom_urls:
        type: string
    hostnames:
        type: array
        items:
          type: string
          example: "Host10_A20_2, Host10_A20_2"
    hostnames_order:
        type: array
        items:
          type: integer
          example: "1, 2"
    port:
      type: object
      properties:
        portnumber:
          type: string
          example: "44818"
        transport:
```

```
            type: string
            example: "tcp"
        protocol:
          type: string
          example: "Modbus RTU"
        connection:
          type: string
          example: "LAN"
        location:
          type: string
          example: "Halle 2, Quadrant A1"
    errorMessage:
      type: string
```

Die Implementierung erlaubt den Zugriff auf die Programmierschnittstelle auf zwei Wegen. Über ein Web-Frontend wird ein Formular bereitgestellt, das alle notwendigen und optionalen Eingabemöglichkeiten zur Verfügung stellt (siehe *Abbildung 15*). Es enthält ebenfalls Erläuterungen und Hinweise zu Eingabeformaten und -optionen sowie eine Auswahl zur Bestimmung des gewünschten Endpunkts (URI). Beim Absenden des Formulars werden die eingegebenen Daten validiert, an die API übertragen und bei dem zuvor ausgewählten URI eingeliefert.

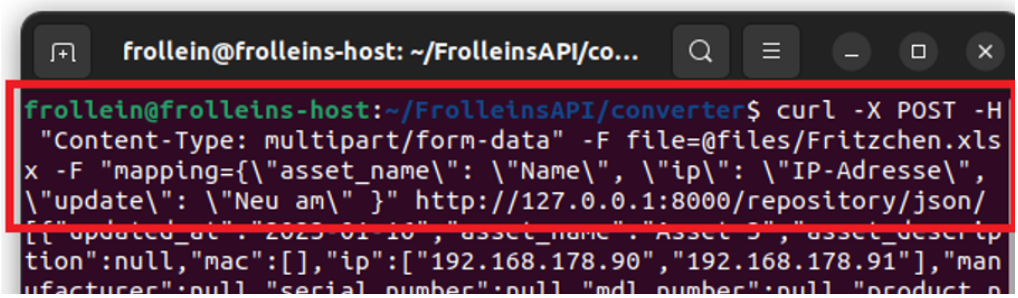

Abbildung 15: Eingabeformular (Web-Frontend)

Außerdem ist der direkte Zugriff auf die API möglich, z.B. mittels cURL (siehe *Abbildung 16*).

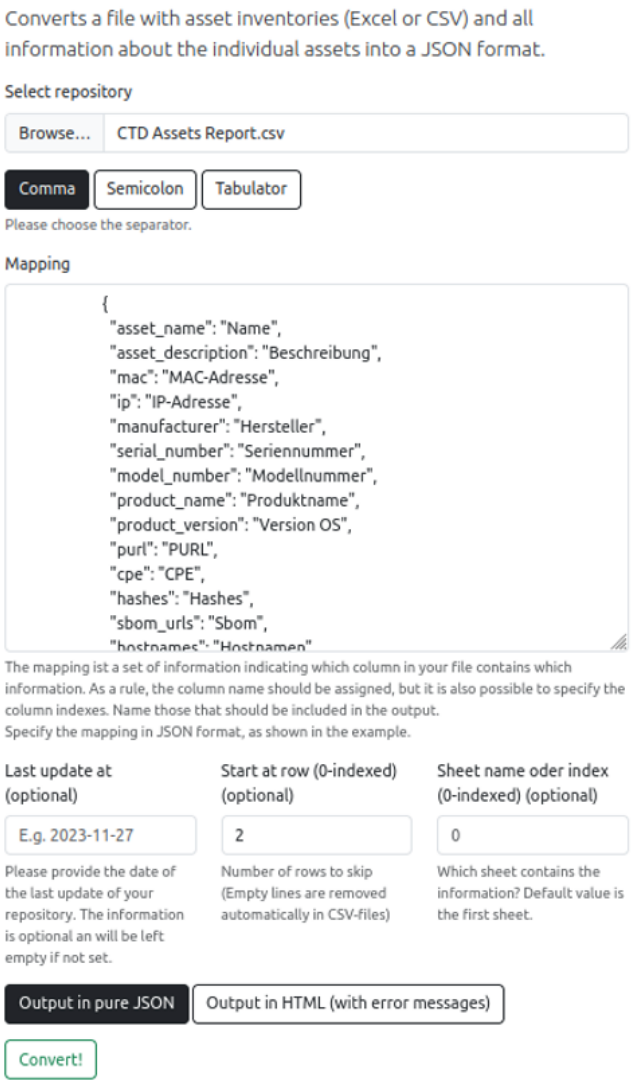

Abbildung 16: Eingabe mit cURL

Zur Verarbeitung einer Eingabe existieren zwei Endpunkte, spezifiziert über eindeutige URI. Zudem ist der Aufruf der Startseite des Web-Frontend über die Root-URL möglich. Der Endpunkt /repository/json wandelt die eingelieferten Daten in die spezifizierte JSON-Ausgabe um und liefert diese aus. Die JSON-Ausgabe enthält die erfolgreich umgewandelten Asset-Informationen aus dem Inventar. Während der Verarbeitung ggf. generierte Fehlermeldungen bzgl. einzelner Einträge sind darin nicht enthalten. Ist eine Verarbeitung generell nicht möglich, z.B. wenn ein Request oder Teile davon nicht in einem validen Format eingeliefert werden oder unvorhergesehene Ereignisse auftreten, die die Verarbeitung unmöglich machen, enthält die JSON-Ausgabe eine Fehlermeldung anstatt eines Ergebnisses. Der Endpunkt /repository/html liefert eine Ausgabe in HTML. Diese stellt das umgewandelte Inventar sowie Informationen und Fehlermeldungen, die im Umwandlungsprozess generiert wurden, dar. Diese ausführlichen Informationen ermöglichen es dem Anwender auf Fehlersuche zu gehen, falls die Schnittstelle nicht die gewünschten Ergebnisse liefert. Ist eine Verarbeitung generell nicht möglich, wird der Anwender auf die Startseite des Web-Frontend zurückgeleitet, wo die entsprechende Fehlermeldung angezeigt wird.

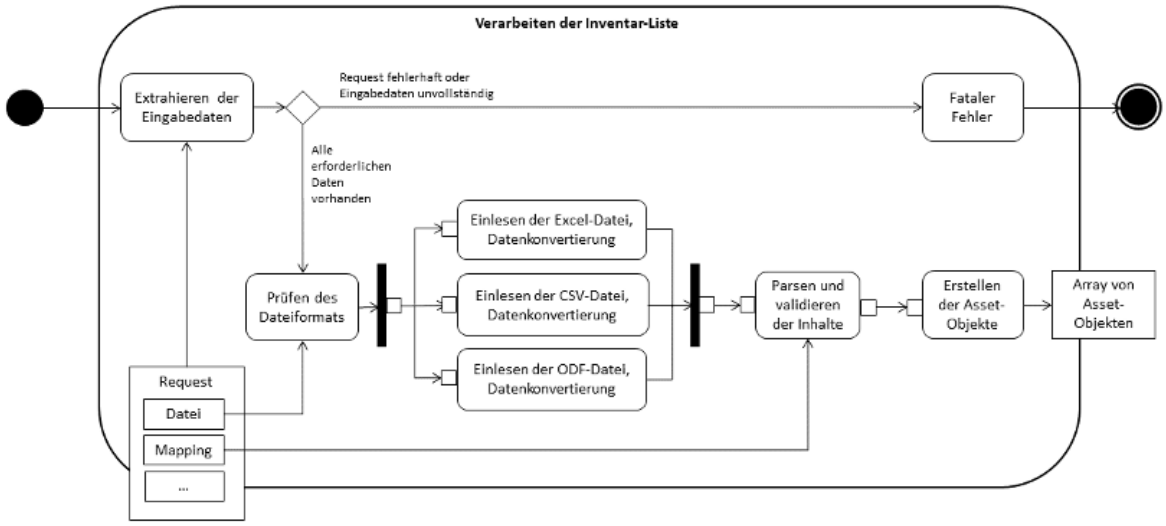

Abbildung 17: Aktivitätsdiagramm zum Aufruf und zur Ausführung der Programmierschnittstelle

Auf der Weboberfläche wird der gewünschte Endpunkt über einen Radiobutton ausgewählt. Beim direkten Zugriff auf die API muss der gewünschte Endpunkt als URI in der Anfrage angegeben werden. Der Ablauf der Verarbeitung der Daten unterscheidet sich nach der Annahme am Endpunkt nur hinsichtlich seiner Ausgabe. Die Schritte zur Umwandlung der Eingabe sind für beide Endpunkte identisch. Nach der Annahme des Request an den Endpunkten werden alle Formulardaten an eine Methode weitergereicht, die die Daten entsprechend den angegebenen Einstellungen und Informationen verarbeitet. Aus dem Rückgabewert der Methode kann die gewünschte Ausgabe generiert werden. Abbildung 17 skizziert den Ablauf.

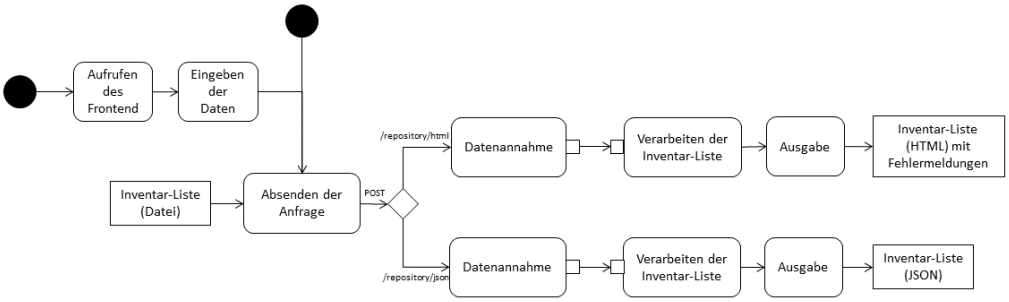

Abbildung 18: Aktivitätsdiagramm zur Verarbeitung der Eingabedaten

Den Schritt „Verarbeiten der Inventar-Liste“ aus Abbildung 17 stellt Abbildung 18 detailliert dar.

Alle Inhalte der HTTP-Anfrage POST werden als Parameter an die Methode übergeben. Der Request liefert die Datei, die die Inventarisierungsdaten enthält. Um die Datei in einem POST zu übertragen, muss der Request dem „multipart/form-data“-Content-Type entsprechen. Neben der Datei wird das Mapping übertragen. Dieses definiert, welche Daten aus der Inventarisierungs-Datei extrahiert werden sollen und in welcher Spalte diese jeweils zu finden sind. Für Excel-Dateien mit mehreren Sheets kann angegeben werden, welche Sheets verwendet werden sollen. Enthält eine Datei mehr als eine Zeilen-Überschrift, kann bestimmt werden, ab welcher Zeile die Inventarisierungs-Liste beginnt. Des Weiteren kann das Datum der letzten Aktualisierung als optionaler Parameter für alle Datensätze hinterlegt werden.

Falls nicht alle notwendigen Eingabeparameter vorhanden sind oder ein Format nicht den Vorgaben entspricht, wird die Verarbeitung abgebrochen und eine Fehlermeldung (Fataler Fehler) zurückgegeben. Wurden die Eingabedaten erfolgreich validiert, wird das Dateiformat anhand der Dateiendung identifiziert. Dann wird die dem Format entsprechende Methode angewandt, um die Daten aus der Datei in einen für Python3 lesbaren Datentyp umzuwandeln („Datenkonvertierung“ in Abbildung 18). Anschließend können die Daten einheitlich weiterverarbeitet werden.

Mit Hilfe des Mapping werden die ausgewählten Daten aus der Datenstruktur gefiltert. Falls möglich, sollten Daten hinsichtlich ihres Formats validiert werden. Dies ist allerdings nur für MAC- und IP-Adressen umsetzbar, da nur für diese Werte standardisierte Formatvorgaben gegeben sind. Je Asset muss mindestens eine MAC- oder IP-Adresse angeben sein, um Assets eindeutig zu identifizieren zu können. Alle anderen Werte werden als String-Werte übernommen, diese dürfen auch leer sein. Treten bei der Validierung Fehler für einzelne Datenfelder auf oder kann eine Angabe nicht richtig zugeordnet werden, werden entsprechende Fehlermeldungen erzeugt und in einem global definierten Array hinterlegt. Für alle Datensätze, die erfolgreich validiert und umgewandelt wurden, wird ein Asset-Objekt erstellt. Alle Asset-Objekte werden in einer Liste gespeichert. Diese Liste ist nach der Abarbeitung der Daten der Rückgabewert der Methode.

Die Umsetzung der Programmierschnittstelle in Python3 (Version 3.10.12) erfolgte auf Basis des freien Framework FastAPI[8] (verwendete Version 0.103.1). FastAPI ist ein dediziertes und schlankes Python3-Framework zur Implementierung von REST API. Es basiert auf den offenen Standards OpenAPI und JSON und unterstützt die automatische Validierung aller ein- und ausgehenden JSON-Daten. Gemeinsam mit FastAPI wurde das Python3-Modul Pydantic[9] verwendet. Dieses erlaubt individuelle Definitionen von Datenstrukturen sowie deren individuelle Validierungen. Vorteilhaft ist, dass aus Pydantic-Modellen direkt JSON erzeugt werden kann.

Als Webserver für die Schnittstelle wurde die in der Dokumentation von FastAPI vorgeschlagene ASGI[10] Webserver-Implementierung Uvicorn[11] (Version 0.23.2) eingesetzt. Alternativ kann jeder andere Python3-fähige Webserver verwendet werden. Zur Erstellung der Templates für die Web-GUI wurde auf die Bibliothek Jinja2 (Version 3.1.2) zurückgegriffen, die ebenfalls von FastAPI empfohlen wird. Zur Verarbeitung der Multipart-Objekte über die Schnittstelle wurde die Bibliothek python-multipart[12] (Version 0.0.6) benötigt, ein Streaming-Multipart-Parser für Python3.

Die Bibliothek Pandas[13] (verwendete Version 2.1.1) stellt Funktionen zur Datenanalyse und -manipulation bereit. Unter anderem ermöglicht sie das einfache Lesen und Schreiben von Datenstrukturen wie CSV- und Excel-Dateien. Die Bibliothek Numpy[14] (verwendete Version 1.26.0) unterstützt bei der Handhabung von mehrdimensionalen Arrays.

Zum Lesen der unterschiedlichen Datenformate bei der Eingabe werden weitere Bibliotheken benötigt, openpyxl (verwendete Version 3.1.2) zur Verarbeitung von Excel 2010-Dateien (xlsx/xlsm), xlrd (verwendete Version 2.0.1) zur Verarbeitung von Excel 97-2003-Dateien (xls) und odfpy (verwendete Version 1.4.1) zur Verarbeitung von OpenDocument-Dateien.
Um Abhängigkeiten auf Versionsebene zu verwalten, wurde im Entwicklungsprozess Poetry eingesetzt. Poetry ist ein modernes Werkzeug zur Verwaltung von Abhängigkeiten und Erstellung

[8] https://fastapi.tiangolo.com
[9] https://docs.pydantic.dev/latest/
[10] Asynchronous Server Gateway Interface
[11] https://www.uvicorn.org
[12] https://andrew-d.github.io/python-multipart/index.html
[13] https://pandas.pydata.org
[14] https://numpy.org

von Projekten in Python3. Es wurde entwickelt, um den Prozess der Verwaltung von Python3-Projekten zu vereinfachen. Abhängigkeiten werden direkt in der pyproject.toml-Datei spezifiziert.

Über die Open-Source Plattform snyk[15] ist es möglich, die Qualität von Python-Bibliotheken zu überprüfen. Das Projekt bewertet Python-Bibliotheken anhand der Kriterien Sicherheit, Popularität, Wartungszustand und Größe der Community. Die Bewertung erfolgt über den sogenannten Health Score, der maximal bei 100 Punkten liegen kann. Die verwendeten Bibliotheken sind mit folgenden Health Scores bewertet:

- FastAPI: 88
- Pydantic: 89
- Uvicorn: 97
- python-multipart: 85
- Pandas: 87
- Numpy: 89
- openpyxl: 72
- xlrd: 80
- odfpy: 64
- Poetry: 91

Zur Validierung wurden Testdaten selbst erzeugt werden. Hierzu wurde zunächst ein eigenes Asset-Inventar manuell erzeugt und mit Daten gefüllt. Ein weiteres Inventar wurde zum Test mit den Daten in Excel selbst erstellt. Dieses Inventar enthält die folgenden Daten: MAC-Adresse, DHCP, IP-Adresse, Netzwerk, Name, Beschreibung, Version OS, Produktname, Hersteller, Seriennummer, Modellnummer, PURL, CPE, Hashes, Sbom, Host-Name, Host-Reihenfolge, Port, Protokoll, und Standort. Das Dokument wurde auch in den Dateiformaten CSV, OpenDocument und Excel-Arbeitsmappe 97-2003 gespeichert. In den Testdaten wurden bewusst Werte für IP- und MAC-Adresse hinterlegt, die nicht den geforderten standardisierten Formaten entsprechen, um Fehlermeldungen zu provozieren. Von der r-tec GmbH wurde aus einer Test-Installation von CTD ein Template für den Import von Daten im CSV-Format (CTD Assets Report.csv) zur Verfügung gestellt. Dieses enthielt Testdaten. Weiterhin konnte ein Template (Excel-Datei) eines Unternehmens aus der Elektronikbranche (elektronikhersteller.xlsx) zum Test verwendet werden. Abschließend konnten alle Tests erfolgreich durchgeführt werden. Alle in den getesteten Dateien enthaltenen Daten wurden gemäß der Spezifikation fehlerfrei in JSON umgewandelt. Für die bewusst fehlerhaft hinterlegten IP- und MAC-Adressen wurden Fehlermeldungen erzeugt.

Um die Schnittstelle zu härten und sprechende Fehlermeldungen für die Anwender zu hinterlegen, wurden ausgiebige Tests mit falschen Daten (fehlendes oder fehlerhaftes Mapping, falsche Angaben zur Kopfzeile, fehlerhafte Dateien oder falsche Dateiformate etc.), die eine Verarbeitung generell unmöglich machen, durchgeführt. Alle Tests wurden sowohl über das Web-Frontend als auch mit dem Kommandozeilen-Tool cURL im direkten Zugriff durchgeführt.

Um einen Test zur Auslastung durchzuführen, wurden zwei Excel-Dateien auf Basis der Vorlage Fritzchen-Extended.xlsx erstellt. Eine Datei umfasste 6000 Assets, die andere 10000 Assets.[16] Für jedes Asset wurden vollständige Daten angegeben. Für die Umwandlung der 424 Kbyte und 695

[15] https://snyk.io/advisor/python

[16] Es wurden 6.000 und 10.000 Assets gewählt, weil unterstellt wurde, dass dies die maximale Anzahl der in mittelgroßen bzw. großen KMU vorhandenen Assets realitätsnah abbildet

Kbyte großen Dateien benötigte die Schnittstelle nur jeweils wenige Sekunden (424 Kb: ca. 5s, 695 Kbyte: ca. 10s)

Zur Vermeidung von Denial-of-Service-Angriffen sollte die Größe der Dateien, die hochgeladen werden können, beschränkt werden. Deshalb wurde die maximale Dateigröße auf 800 Kbyte festgelegt. Auf Basis der Test-Dateien wurde ersichtlich, dass diese Dateigröße für ein Inventar von mehr als 10000 Assets genügt. Diese Menge ist für die praktische Verwendung in KMU ausreichend.

Feldtests

Wie bereits erwähnt, konnte für einen Feldtest leider kein Industriepartner gefunden werden, um das System zu testen. Die virtuelle Simulationsumgebung, welche unter anderem diverse OT-Komponenten simuliert, diente als Ersatz, um den Demonstrator zu testen.

Datenschutz- und Rechtskonformität

Das Arbeitspaket 5.3 zielte darauf ab, Datenschutz- und Rechtskonformität im Rahmen des Projekts sicherzustellen. Basierend auf dem „Privacy by Design"-Ansatz wurden die datenschutzrechtlichen Anforderungen bei der Verarbeitung personenbezogener Daten identifiziert und berücksichtigt. Dabei wurden zunächst die Anforderungen an den Anwendungsbereich der DS-GVO definiert, einschließlich der sachlichen Geltung sowie der Personenbeziehbarkeit von Daten. Insbesondere wurden die rechtlichen Vorgaben für anonyme und pseudonyme Datenverarbeitungen analysiert, wobei das Prinzip der Datensparsamkeit im Fokus stand. Zudem wurden verschiedene Ansätze der datenschutzrechtlichen Bewertung, sowohl aus rechtswissenschaftlicher als auch aus aufsichtsbehördlicher Perspektive, untersucht.

Die datenschutzrechtlichen Verarbeitungsprinzipien, wie Rechtmäßigkeit, Zweckbindung, Datenminimierung, Speicherbegrenzung und Transparenz, wurden im spezifischen Projektkontext systematisch geprüft und auf ihre praktische Anwendung hin konkretisiert. Darüber hinaus erfolgte eine Analyse des rechtlichen Rahmens zur Legitimation der personenbezogenen Datenverarbeitung. Hierbei wurden die datenschutzrechtliche Einwilligung sowie gesetzliche Erlaubnistatbestände, insbesondere die Interessenabwägung gemäß Art. 6 Abs. 1 lit. f DS-GVO, betrachtet. Der Erwägungsgrund 49 der DS-GVO erkennt IT-Sicherheit als berechtigtes Interesse an, sofern die Datenschutzrechte der Betroffenen nicht überwiegen. Im Projektkontext wurden keine gegensätzlichen Datenschutzinteressen festgestellt, die eine solche Verarbeitung unzulässig machen könnten.

Ein weiterer Schwerpunkt lag auf der Prüfung von Datentransfers außerhalb der EU. Hier wurden die rechtlichen Rahmenbedingungen analysiert und mögliche Compliance-Risiken identifiziert, die für den weiteren Projektverlauf relevant sind. Bisher wurden jedoch keine Übermittlungen in Drittstaaten mit unzureichendem Datenschutzniveau festgestellt.
Zur weiteren Konkretisierung der Anforderungen wurde ein Gutachten erstellt, das den sachlichen Anwendungsbereich der DS-GVO klärt und die Abgrenzung zwischen personenbezogenen, anonymisierten und pseudonymisierten Daten untersucht. Zudem wurden die datenschutzrechtlichen Verarbeitungsgrundsätze, einschließlich der Pflicht zur Information der Betroffenen, in den Projektkontext übertragen.

Im Ergebnis zeigt die Prüfung, dass die Plattformentwicklung den Prinzipien der Datensparsamkeit genügt und der gegenwärtige Entwicklungsstand datenschutzrechtlichen Prinzipien entspricht. Eine fortlaufende Überprüfung ist jedoch erforderlich, um die Einhaltung der rechtlichen Vorgaben auch in zukünftigen Entwicklungsphasen und technischen Iterationen sicherzustellen.

Zusammenfassendes Vorgehen entlang der Arbeitspakete

AP2: Anforderungsanalyse und Entwurf der Plattformsystemarchitektur

Zunächst erfolgte eine umfassende Stakeholder- und Anforderungsanalyse, um die Anforderungen für den Demonstrator zu dokumentieren. Auf Basis dieser Analyse wurden Anwendungsszenarien abgeleitet und die Basisfunktionen für ein Incident Management definiert. Anschließend wurde die High-Level-Architektur der Plattform spezifiziert und die Eignung verschiedener Frameworks untersucht. Parallel dazu wurden die Schnittstellen, Kommunikationsprotokolle und Austauschformate für die Plattform konzipiert. Weitere Arbeitsschritte beinhalteten die Entwicklung von Anforderungen für ein maschinenlesbares Advisory-Format auf CSAF-Basis und ein Konzept zur Integration von Schwachstelleninformationen für SIEM-Hersteller.

AP3: Entwicklung der Plattform

Die Entwicklung umfasste die Erstellung eines Matching-SW Mockups, um die Anforderungen für den Abgleich von Assets und Schwachstellen zu klassifizieren und zu prüfen. Darauf folgte die Simulationsumgebung, in der Systemkomponenten identifiziert, konfiguriert und simuliert sowie einem Security Assessment unterzogen wurden. Zusätzlich wurde die DECOIT-Basisfunktionalität in die Simulationsumgebung integriert und auf ihre Eignung geprüft. Die Asset-Verarbeitung wurde hinsichtlich der Heterogenität von Asset-Informationen untersucht und entsprechende Verfahren zur Verarbeitung definiert. Zudem wurden Visualisierungsansätze für Ereignisse bewertet und Testdaten aggregiert. Unterstützung wurde bei der Integration der Visualisierungs-komponenten geleistet. Weitere Arbeitsschritte beinhalteten die KI-unterstützte Datenfilterung und Datenaufbereitung, die das Eliminieren redundanter Informationen und das Anreichern von Daten umfassten. Der Demonstrator für SIEM-Meldungen wurde entwickelt, indem Schnittstellen konzipiert und implementiert wurden. Des Weiteren wurden Schnittstellen zu Drittsystemen implementiert und getestet. Schließlich wurde eine erste Teilsystemintegration durchgeführt und vorvalidiert.

AP4: Erprobung und Validierung

In der Erprobungsphase wurden System- und Funktionstests durchgeführt, um die Qualität der Daten und die Anomalie-Detektion zu prüfen. Nach der Integration der Komponenten erfolgte eine schrittweise Integration und Test der Plattform, wobei etwaige Fehler korrigiert wurden. Nach vollständiger Integration wurde das Gesamtsystem umfassend getestet. In den darauffolgenden System- und Funktionstests wurden mögliche Fehler korrigiert und die Funktionalität der entwickelten Komponenten geprüft. Abschließend erfolgte eine Evaluation des Demonstrators, bei der das Zusammenspiel der Systeme untersucht und Optimierungspotenziale identifiziert wurden. Zudem wurde ein Anwenderhandbuch erstellt.

AP5: Feldtests und Standardisierung

Nach den Tests wurden die Testergebnisse dokumentiert und bewertet. Zudem erfolgte ein Abgleich mit Standardisierungsaktivitäten, bei dem die Ergebnisse dem BSI kommuniziert wurden, um Anforderungen abzugleichen und Rückmeldungen für die Weiterentwicklung der Plattform zu erhalten. In einer engen Kommunikation mit dem BSI wurde der CSAF-Demonstrator entwickelt. Das BSI, welches zusammen mit nationalen und internationalen Partnern an der Verbreitung von CSAF arbeitet, um Anwendern das Auffinden sowie die Bewertung und Umsetzung von Security Advisorys (SA) zu erleichtern, zeigte großes Interesse. Durch den regen Austausch während der Feldtestphase konnten die Tests optimiert und das Gesamtergebnis verbessert werden.

Erreichte Ziele
Die ZenSIM4.0-Plattform bietet eine Sicherheitslösung für KMU, die sowohl präventive als auch reaktive Maßnahmen umfasst:

- Frühwarnsystem: Ermöglicht die frühzeitige Erkennung potenzieller Sicherheitsrisiken.
- Konsolidierte Reaktion: Bietet koordinierte Antworten auf Sicherheitsvorfälle
- Asset-Management: KMU können ihre Assets in der Plattform registrieren. Automatischer Abgleich von Assets und Schwachstelleninformationen.
- Datenintegration und -korrelation: Die Plattform sammelt Informationen aus verschiedenen Quellen. Diese werden mit den registrierten Assets in Beziehung gesetzt, um ein umfassendes Sicherheitsbild zu erstellen.
- Security Incident Management

2.3 Zusammenarbeit mit anderen Forschungseinrichtungen

Eine Zusammenarbeit mit externen Industriepartnern gab es seitens der HSB nicht. Jedoch wurden die in ZenSIM4.0 erbrachten Ergebnisse im Rahmen des Verwertungsplans in das Netzwerk des cyberintelligence.institute (CII) eingebracht. CII versteht sich als Digital Hub zu allen Fragestellungen der digitalen Resilienz mit Sitz in Frankfurt am Main. Die Ergebnisse wurden und werden in Zukunft vor allem weiteren Partnern aus der Industrie und den Unternehmen zugänglich gemacht werden. Hierzu ist geplant, eine Arbeitsgruppe zum Cybersecurity Incident Management mit Wirtschaftspartnern einzurichten, innerhalb derer die Projektergebnisse eingebracht werden sollen.

2.4 Die wichtigsten Positionen des zahlenmäßigen Nachweises

Im ZenSIM4.0-Projekt wurde die Ressourcenplanung mit einem Projektmitarbeiter kalkuliert, was sich auch als zutreffend herausstellte.

Es gestaltete sich äußerst schwierig, geeignetes Personal in Form von studentischen Mitarbeitern zu finden, was die notwendigen Grundlagen und Kenntnisse in den Bereichen Rechnernetze, Virtualisierung und insbesondere IT-Sicherheit mitbringt. Besonders im Themenkomplex OT-Security gibt es kein Personal. Vor diesem Hintergrund konnten die ursprünglich eingeplanten Personalmittel für Beschäftigungsentgelte nicht komplett verwendet werden.

Die spätere kostenneutrale Verlängerung war notwendig, weil der Mitarbeiter erst spät ins Projekt eingebunden werden konnte, wodurch die Feldtests länger als ursprünglich geplant durchgeführt werden konnten.

Da es nicht möglich war, wissenschaftliches Personal (mit Masterabschluss) finden konnten und einen Projektmitarbeiter mit Bachelorabschluss besetzt haben, konnten die Mittel für Personal nicht im vollen Umfang ausgeschöpft werden.

Da die zur Antragsphase veranschlagten Kosten für die benötigte Hardware später im Projektzeitraum nicht mehr realistisch waren (höhere Preise), wurde die Differenz durch Umwidmung von Beschäftigungsentgelten von studentischen Hilfskräften kompensiert.
Auch wurde später festgestellt, dass die Simulationsumgebung weitere Hardware erforderte, was eine weitere Umwidmung von Beschäftigungsentgelten von studentischen Hilfskräften erforderte.

Die Reisekosten weichen deutlich von den ursprünglichen Schätzungen ab, was durch die Corona-Pandemie bedingt ist. Diese führte zu mehr virtuellen Meetings und dem Ausfall von Konferenzen und Messen. Der normale Betrieb begann erst Mitte 2022, und die Konsortialtreffen fanden bei DECOIT statt, wodurch keine zusätzlichen Reisekosten entstanden.

2.5 Die Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten

Die Projektergebnisse decken sich vollständig mit den ursprünglich geplanten Zielen.

Das ZenSIM4.0-Projekt hat eine innovative Plattform-Architektur hervorgebracht, die als Drehscheibe für kritische IT-Sicherheitsinformationen dient. Diese Plattform ermöglicht es, detaillierte Daten über IT-/OT-Infrastrukturen, wertvolle Assets, bestehende Sicherheitsmaßnahmen und vergangene Sicherheitsvorfälle zu sammeln und zu teilen. KMUs profitieren von dieser Wissensbasis, indem sie im Gegenzug für ihre Teilnahme maßgeschneiderte Sicherheitsempfehlungen erhalten. Diese Empfehlungen adressieren spezifisch potenzielle Schwachstellen und Exploits in ihrer individuellen IT-Umgebung. Ein Schlüsselelement dieser Plattform ist das Common Security Advisory Framework (CSAF). Dieses offene, standardisierte System ermöglicht die effiziente, automatisierte Verbreitung von maschinenlesbaren Sicherheitsinformationen und Schwachstellenmeldungen. Die Zusammenarbeit mit dem Bundesamt für Sicherheit in der Informationstechnik (BSI) im Rahmen des Projekts hat die Bedeutung und Reichweite von CSAF weiter verstärkt. Das BSI arbeitet mit verschiedenen Partnern daran, CSAF zu einem unverzichtbaren Werkzeug für Anwender zu machen, um Sicherheitshinweise leichter zu finden, zu bewerten und umzusetzen.

Mit konzeptionellen Arbeiten an der Plattformsystemarchitektur, der Realisierung eines Simulationssystem für eine OT-Umgebung, durch System- und Funktionstests sowie der Untersuchung der datenschutzrechtlichen Anforderungen beim Aufbau des Systems konnte die Hochschule Bremen einen wesentlichen Beitrag leisten.

Die Nachfrage nach fortschrittlichen Konzepten zur Aggregation und Analyse sicherheitsrelevanter Netzwerkdaten sowie deren automatischer und effizienter Verarbeitung steigt gerade in KMU an. Die Ergebnisse des ZenSim4.0-Projekts bieten aufgrund der hohen automatisierten Erkennung eine Chance für die Entwicklung einer Software-Lösung, die später weite Verbreitung in diesem Bereich finden kann. Durch die anspruchsvollen konzeptionellen Fragestellungen war Zusammenarbeit mit der Hochschule Bremen sinnvoll und notwendig. Die Hochschule hat unterschiedliche Methoden und Expertise aus verschiedenen Anwendungsfeldern eingebracht, die sich auf spezifischen Fragestellungen der OT-Sicherheit übertragen lassen konnten.

Angesichts neuer regulatorischer Anforderungen wie NIS2 für KMUs wird der Bedarf an solchen Lösungen voraussichtlich steigen. KMUs, die oft keine eigenen IT-Sicherheitsexperten beschäftigen können, benötigen Unterstützung, um zeitnah auf sich ändernde Bedrohungslagen zu reagieren. Zusammenfassend lässt sich sagen, dass die Ergebnisse des ZenSIM4.0-Projekts das Potenzial haben, einen signifikanten Beitrag zur Verbesserung der IT-Sicherheitslandschaft in Deutschland zu leisten.

2.6 Voraussichtlicher Nutzen und Verwertbarkeit

Bei der Verwertung der Ergebnisse des Projektes besteht kein wirtschaftlicher Nutzen. Die Hochschule Bremen wird die Projektergebnisse für das LISA-Labor im Sinne eines sog. „Show Room“ verwerten. Interessierten wird das Zusammenspiel von Anwendungen, Komponenten in einer OT-Infrastruktur präsentiert.

Auch werden die Ergebnisse des ZenSIM4.0-Projektes direkt in die Lehre einfließen. In die Lehre für folgende Module/Veranstaltungen:

- IT-Sicherheitsarchitekturen (ITSARCH) - BSc. Technische Informatik - 4. Semester
- Rechnernetze (RNETZE) - BSc. Technische Informatik - 2. Semester

- Virtualisierung und Cloud Computing (VCC) - MSc. Komplexe Softwaresysteme – Master 3. Semester

Der Projektleiter der HSB (Prof. Dr.-Ing. Evren Eren) wird Erkenntnisse und Inhalte aus dem ZenSIM4.0-Projekt in die Vorlesungen einfließen lassen. Auf diese Weise können Studierende durch direkten Kontakt mit diesen neuen Technologien ihr Wissen erweitern.

Außerdem sollen Studierende in Studienprojekten sowie in Abschlussarbeiten (Bachelor und Master) viele Teile der Simulationsumgebung modular nutzen können.

Die HSB plant auf Basis der Ergebnisse des Vorhabens weitere Forschungs- und Entwicklungsprojekte im nationalen Umfeld mit entsprechenden Partnern aus dem Hochschul- und Industrieumfeld. Die HSB verfolgt eine Verwertung der Ergebnisse in Form der Integration in weitere Entwicklungsprojekte, die zusammen mit verschiedenen Industriepartnern durchgeführt werden sollen. Hierdurch wird das satzungsgemäße Ziel des Wissenstransfers erfüllt.

Überdies wurden die in ZenSIM4.0 erbrachten Ergebnisse im Rahmen des Verwertungsplans in das Netzwerk des cyberintelligence.institute (CII) eingebracht. Hierbei handelt es sich um einen Digital Hub zu allen Fragestellungen der digitalen Resilienz mit Sitz in Frankfurt am Main. Die Ergebnisse sollen hier vor allem weiteren Partnern aus der Industrie und den Unternehmen zugänglich gemacht werden. Hierzu ist geplant, eine Arbeitsgruppe zum Cybersecurity Incident Management mit Wirtschaftspartnern einzurichten, innerhalb derer die Projektergebnisse eingebracht werden sollen.

2.7 Bekannt gewordener Fortschritt auf dem Gebiet des Vorhabens

Die IT-Sicherheit ist heutzutage wichtiger denn je, mit Blick auf die Zunahme von Cyberangriffen in Deutschland. Laut BSI-Lagebricht aus dem Jahr 2024 [13] ist die Bedrohungslage in Deutschland weiterhin von einer hohen Dynamik geprägt. Dabei bleibt Ransomware die größte Bedrohung für Unternehmen: die Lösegelder weltweit, die von Ransomware-Gruppen erbeutet wurden stieg in 2023 auf 1,1 Mrd. US $. (siehe Abbildung 19). Die Zahl der Ransomware-Angriffe stieg damit deutlich an. Viele Angreifer nutzen dafür Zero-Day-Schwachstellen in den Software-Produkten aus. Pro Tag werden 78 neue Schwachstellen in Software-Produkten bekannt, weshalb es den Angreifern auch weiterhin zu leicht gemacht wird. Opfer waren überwiegend Klein- und Mittelständische Unternehmen (KMU). Aber auch IT-Dienstleister werden zunehmend angegriffen sowie Kommunen.

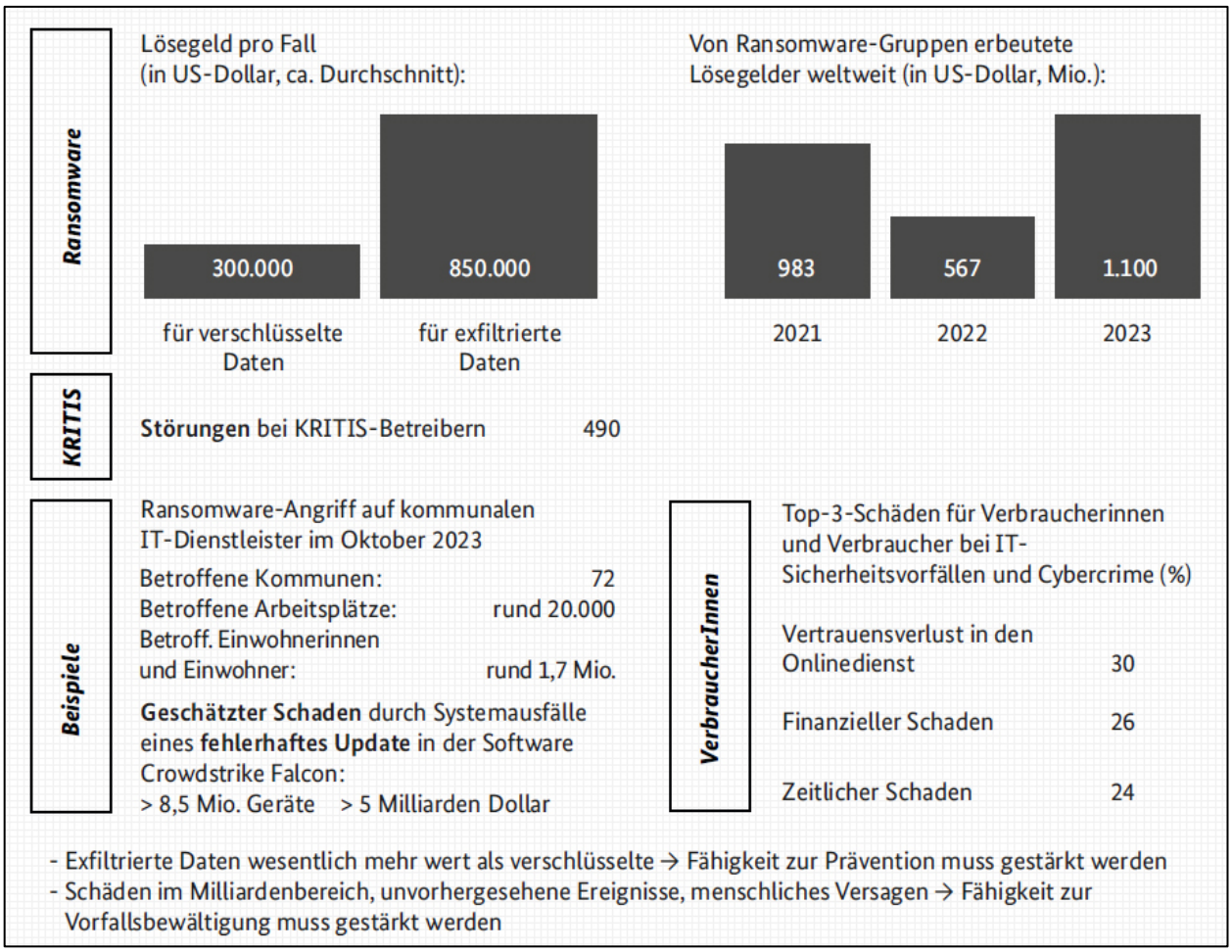

Abbildung 19: Zusammenfassung des BSI-Lageberichts von 2024

Der Schaden für die deutsche Wirtschaft ist hierbei enorm: laut einer Bitkom-Umfrage entstehen 223 Milliarden Euro Schaden pro Jahr [12]. Neun von zehn Unternehmen waren von Cyberangriffen betroffen. Haupttreiber des enormen Anstiegs sind Erpressungsvorfälle, verbunden mit dem Ausfall von Informations- und Produktionssystemen sowie der Störung von Betriebsabläufen. Sie sind meist unmittelbare Folge von Ransomware-Angriffen. Durch sie werden Computer und andere Systeme blockiert, anschließend werden die Betreiber erpresst. Die Bitkom-Studie kommt damit zu dem gleichen Ergebnis wie das BSI. Das BSI spricht inzwischen von besorgniserregenden Zuständen.

Den Markt beeinflusst auch die Gesetzeslage. So ist das IT-Sicherheitsgesetz 2.0 (IT-SiG2.0) im Mai 2023 für KRITIS-Betreiber wie Energieversorger in Kraft getreten. Sie werden damit angehalten Anomalie-Erkennungssysteme einzusetzen und Vorfälle an das Bundesamt für Sicherheit in der Informationstechnik (BSI) zu melden. Diese Vorschriften werden sich durch das NIS2-Gesetz ab Oktober 2024 auch auf Unternehmen kleinerer Größenordnung (ab 50 Mitarbeitern oder 10 Mio. Umsatz) ausweiten. Dadurch wird sich die Nachfrage nach SIEM-Systemen deutlich erhöhen.

2.8 Erfolgte oder geplante Veröffentlichungen

Während des Projektes wurden Veröffentlichungen für Messen, Konferenzen und in Fachzeitschriften von und mit der Hochschule Bremen geschrieben:

1. S. Daneshgadeh, K.-O. Detken, S. Çatalkaya, E. Eren: *Central Security Incident Management Platform in Industry 4.0 with Threat Intelligence Interface.* The 12th IEEE

International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 7-9 September, 2023, Dortmund, Germany

2. Evren Eren, Kai-Oliver Detken, Andreas Harner, Dennis-Kenji Kipker: *Präsentation des ZenSIM4.0-Projekts*. IIA-Kolloquium, 30.06.2023, Hochschule Bremen, Bremen 2023

3. A. Lehmann, Analyse geeigneter Strukturen von Schwachstelleninformationen, Verfahren zur Asset-Verarbeitung sowie Implementierung von Schnittstellen für eine Incident-Management Plattform im Kontext des Forschungsprojektes ZenSIM 4.0, Masterarbeit an der Ruhr-Universität Bochum, August 2023, Bochum

4. Meike Henschen-Bolte, Entwicklung einer Programmierschnittstelle zur Bereitstellung von Asset-Daten für ein SIEM im Rahmen des ZenSIM4.0-Forschungsprojekts, Masterarbeit an der Ruhr-University Bochum, März 2024, Bochum

Eine komplette Veröffentlichungsliste ist auf der ZenSIM4.0-Projektwebseite einsehbar. Hier sind auch relevante Veröffentlichungen der Projektpartner zu sehen. Aktuelle Neuigkeiten wurden unter News laufend geschrieben und eingepflegt. Die Webseite wird auch nach Projektablauf weiter bei der DECOIT® gehostet werden, um spätere Kundenanfragen bedienen zu können.

2.9 Wesentliche Erkenntnisse

Das Projekt ZenSIM4.0 zielte darauf ab, eine Plattform für kleine und mittlere Unternehmen (KMU) im Bereich Industrie 4.0 zu entwickeln, die ein effektives "Security Incident Management" ohne eigenes CERT ermöglicht. Statt ein vollständiges ISMS anzubieten, konzentrierte sich das Projekt auf die Unterstützung beim Erkennen und Bewältigen von IT-Schwachstellen in Produkt- und Produktionsumgebungen. Die entwickelte Plattform ermöglicht es KMUs, Informationen über ihre IT/OT-Infrastruktur, Assets und Sicherheitsmaßnahmen sowie vergangene Sicherheitsvorfälle zu teilen. Im Gegenzug erhalten die teilnehmenden Unternehmen maßgeschneiderte Sicherheitsempfehlungen für ihre spezifische IT-Infrastruktur und Assets. Ein zentrales Feature der Plattform ist die Integration verschiedener Schwachstelleninformationen, die es KMUs ermöglicht, ihre Risikolage besser einzuschätzen. Durch einen "Warenkorbansatz" können Unternehmen relevante Warnungen individuell auswählen und nutzen, ohne ein eigenes CERT aufbauen zu müssen. Dieses Ziel wurde im Rahmen des ZenSIM4.0-Projekts erfolgreich umgesetzt.

Alle Meilensteine wurden daher im Projekt erfüllt:

a. **MS1 (Phase 1):** Ist-Analyse und Entwurf der Plattformsystemarchitektur (Monat 12): die Anforderungen von Integratoren, Herstellern, Maschinenbauern und Betreibern konnten vom VDE CERT eingebracht werden. Szenarien wurden von den Partnern entwickelt und konnten dadurch auf ihre Eignung überprüft werden. Eine Gesamtsystemarchitektur wurde mit den Partnern zusammen entwickelt.
b. **MS2 (Phase 2):** Entwicklung und Implementierung (Monat 20): die notwendigen Systemkomponenten (Matching-Software, KI-unterstützte Datenfilterung/-aufbereitung, automatisierte Aufnahme von OT-Meldungen, Event-Visualisierung) wurden zusammengestellt, entwickelt und integriert. Zusätzliche Schnittstellen (wie z.B. CSAF) wurden geschaffen. Eine Simulationsumgebung hat die Integration wie geplant begleitet.
c. **MS3 (Phase 3):** Erprobung und Validierung (Monat 32): das neu entwickelte Gesamtsystem wurde auf Demonstrator-Ebene diversen Systemtests unterzogen. Nach erfolgreichem Systemtest erfolgte die Entwicklung des Datenformats für die Warnmeldungen.
d. **MS4 (Phase 4):** Feldtests und Standardisierung (Monat 36): abschließend erfolgte ein Nachweis der Einsetzbarkeit, indem entsprechende Feldtests durchgeführt und ausgewertet

werden. Dies konnte leider nur in der Simulationsumgebung der Hochschule Bremen umgesetzt werden, ließ aber die Anforderungen des Datenschutzes und der Rechtskonformität leichter einhalten. Letztere waren auch ein Grund, warum sich kein assoziierter Partner fand, um an den Feldtests mitzuwirken.

3 Anhang

Im Anhang wird auf den Erfolgskontrollbericht, die verwiesene Literatur, die Abbildungen und das Abkürzungsverzeichnis eingegangen.

3.1 Literaturverweise

[1] ACDC, D1.1.2 – Overall Software Description. https://acdc-project.eu/wp-content/uploads/2015/12/ACDC_D1.1.2_Overall_Software_Architecture.pdf, 2015, abgerufen am 15.08.2023.

[2] E. Samanis, J. Gardiner und A. Rashid: A Taxonomy for Contrasting Industrial Control Systems Asset Discovery Tools, https://arxiv.org/pdf/2202.01604.pdf

[3] Bundesamt für Sicherheit in der Informationstechnik: Durchführungskonzept für Penetrationtests, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Penetrationstest/penetrationstest.pdf, 2003

[4] ACHTWERK GmbH & Co KG, IRMA IIoT – Für den Maschinen und Anlagenbau, https://irma-security.de/wp-content/uploads/2022/09/IRMA_IOT_05_2021_DE.pdf, 2021

[5] GridProtectionAlliance, openHIstorian, Version 2.8.157, https://github.com/GridProtectionAlliance/openHistorian, 2022

[6] Fortiphyd, GRFICSv2 - Graphical Realism Framework for Industrial Control Simulation, https://github.com/Fortiphyd/GRFICSv2, 2020

[7] Bundesamt für Sicherheit in der Informationstechnik, ICS Security Kompendium https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/ICS/ICS-Security_kompendium_pdf.html, 2024

[8] Bundesamt für Sicherheit in der Informationstechnik, IT-Grundschutz-Kompendium (Edition 2023), https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/IT-GS-Kompendium/IT_Grundschutz_Kompendium_Edition2023.html, 2023

[9] T. J. Williams, The Purdue Enterprise Reference Architecture and Methodology (PERA). In: Arturo Molina (Hg.): Handbook of life cycle engineering. Concepts, models and technologies. Dordrecht: Kluwer Academic Publishers, 1998

[10] P. Cichonski, T. Millar, T. Grance, K. Scarfone. *Computer security incident handling guide*, NIST Special Publication 800(61) Rev. 2, 2012

[11] D. M. Shackleford, R. M. Lee., Cyber threat intelligence uses, successes and failures: The sans 2017 ctisurvey, In: SANS Institute, 2017

[12] bitkom, Angriffsziel Deutschland, 2021, https://www.bitkom.org/Presse/Presseinformation/Angriffsziel-deutsche-Wirtschaft-mehr-als-220-Milliarden-Euro-Schaden-pro-Jahr#_

[13] BSI, Die Lage der IT-Sicherheit in Deutschland 2024, Bonn 2024, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2024.html

[14] Jörg Schwenk: Sicherheit und Kryptographie im Internet (5. Auflage 2020)

[15] curl.1 the man page, aufgerufen im Januar 2024. URL: https://curl.se/docs/manpage.html

[16] Petar Halachev: Web application with Python and security of the information system. In: Security & Future, 2020, S. 103-106

[17] Princewill Inyang (Semaphore): Building Custom Middleware in FastAPI, Oktober 2023, aufgerufen im Januar 2024, URL: https://semaphoreci.com/blog/custom-middleware-fastapi

3.2 Abbildungsverzeichnis

3.3 Tabellenverzeichnis