Kurzbericht
BMBF-Förderung ENCOPIA

Thema
Verbundprojekt: Prüfbarer Schutz der Privatsphäre im Internet der Dinge
– ENCOPIA –
Teilvorhaben: Automatische Erkennung von Privacy-relevanter Funktionalität in kompilierter Software

Förderkennzeichen
16KIS1436

Laufzeit
01.06.2021 – 31.12.2024

Zuwendung
206.529,70 €

Langlauf Security Automation GmbH
Martin-Hülsemann-Straße 9, 59558 Lippstadt

Teil I – Kurzbericht

1. Ursprüngliche Aufgabenstellung

Im ENCOPIA-Projekt sollten Verfahren entwickelt werden, die eine automatisierte Prüfung von Datenschutzzielen in IoT-Diensten ermöglichen – von verankerten Sensorgeräten bis hin zu Cloud-Backends – über den gesamten Produktlebenszyklus hinweg. Angestrebt war ein System, das Entwickler während der Implementierung, beim Testing und nach der Markteinführung unterstützt, potenziell sensible Datenverarbeitung frühzeitig zu erkennen und zu bewerten.

2. Wissenschaftlicher und technischer Stand

Zu Projektbeginn existierten bereits etablierte Ansätze zur Binäranalyse (z. B. signaturbasierte Malware-Erkennung) sowie Tools für Informationsflussanalysen (z. B. Micro Execution, BAP). Allerdings fehlten skalierbare Methoden, um datenschutzrelevante Funktionen und Datenströme in (IoT-)Binaries automatisiert und mit hoher Präzision zu identifizieren. Insbesondere die Herausforderung, Funktionsnamen aus Binärdateien wiederherzustellen und komplexe Datenflussanalysen ohne falsche Positivmeldung durchzuführen, bildete den wissenschaftlichen Ausgangspunkt.

3. Ablauf des Vorhabens

Das Projekt war in vier Phasen gegliedert:

- **Arbeitspaket 1 (Grundlagen):** Identifikation und Bewertung existierender Verfahren zur Funktionserkennung und Informationsflussanalyse in Binär- und Bytecode.
- **Arbeitspaket 2 (Datenschutz-relevante Funktionalitäten):** Entwicklung eigener Methoden zur systematischen Erfassung privatsensitiver Funktionsnamen, automatischer Generierung von Signaturen, Datenbasierter Erkennung und Komponenteninventarisierung.
- **Arbeitspaket 3 (Server-seitige Analysen):** Kooperation mit dem französischen Partner SAP zur performanten Taint-Analyse auf Serverkomponenten.
- **Arbeitspakete 4–7 (Erweiterung & Demonstrator):** Integration in ein ganzheitliches Datenschutz-Framework, sowie Aufbau produktionsnaher Demonstratoren

4. Wesentliche Ergebnisse und Zusammenarbeit

- **Binäranalyse-PoC:** Konzept und Prototyp zur Wiederherstellung von Funktionsnamen in statisch gelinkten Binaries mittels signaturbasierter und datenbasierter Ansätze.
- **Skalierbare Signaturerzeugung:** Entwurf und Demonstration einer Infrastruktur zur massenhaften Generierung und Anwendung von Funktionssignaturen (Arbeitspaket 2.4).

- **Evaluation der Komponentenerkennung:** Automatische Identifizierung von OpenSSL- und Boost-Bibliotheken inklusive Versionseinschränkung über eine Kombination aus Signaturen und String-Analysen (Arbeitspaket 2.7).
- **Demonstratoren & Integration:** Mehrere Proof-of-Concept-Tools, darunter eine alternative Disassembler-Integration nach Lizenzänderung bei IDA Pro, sowie die Zusammenarbeit mit SAP bei serverseitigem Taint Tracking.

Die enge Kooperation mit akademischen Partnern (z. B. Universität zu Lübeck) und SAP ermöglichte den Abgleich theoretischer Forschungsergebnisse mit praxisnahen Anwendungsszenarien. Die entwickelten Methoden bilden die Grundlage für ein kommerzielles Privacy-Assessment-SaaS und sollen in Pilotprojekten bei KMU validiert werden

Langlauf Security Automation GmbH

Martin-Hülsemann-Straße 9, 59558 Lippstadt

E-Mail: hello@langlauf.io

Site: www.langlauf.io

langlauf
SECURITY AUTOMATION

Ausführlicher Schlussbericht
BMBF-Förderung ENCOPIA

Thema
Verbundprojekt: Prüfbarer Schutz der Privatsphäre im Internet der Dinge
– ENCOPIA –
Teilvorhaben: Automatische Erkennung von Privacy-relevanter Funktionalität in kompilierter Software

Förderkennzeichen
16KIS1436

Laufzeit
01.06.2021 – 31.12.2024

Zuwendung
206.529,70 €

Langlauf Security Automation GmbH
Martin-Hülsemann-Straße 9, 59558 Lippstadt

Teil II – Eingehende Darstellung *2*

Ursprüngliche Vorhabensbeschreibung **2**

Darstellung der durchgeführten Arbeiten **2**

Arbeitspaket 1: Identifizierung geeigneter Methoden für die datenschutzorientierte Binäranalyse 2

Arbeitspaket 1.1: Identifikation und Auswahl von Methoden zur Funktionserkennung und Analyse von Sicherheitsmaßnahmen 2

Arbeitspaket 1.2: Identifikation und Auswahl von Methoden zur Informationsflussanalyse in Binärdateien und Bytecode 3

Arbeitspaket 2: Erkennung von datenschutzrelevanten Funktionalitäten und Datenschutzverletzungen 4

Arbeitspaket 2.1: Systematische Erfassung datenschutzrelevanter Funktionsnamen 4

Arbeitspaket 2.2: Erzeugung von Test- und Trainingsdaten 4

Arbeitspaket 2.3: Günstige Methoden mit hoher Präzision 4

Arbeitspaket 2.4: Teure Methoden mit hoher Präzision (Signatur-basierte Erkennung) 5

Arbeitspaket 2.5: Erforschung & Entwicklung von Methoden zur abstrakten Funktionalitätserkennung 6

Arbeitspaket 2.6: Daten-basierte Erkennung datenschutzrelevanter Eigenschaften 6

Arbeitspaket 2.7: Erkennung von (Sub-)Komponenten mit bekannten und veröffentlichten datenschutzrelevanten Sicherheitsproblemen 7

Arbeitspaket 3: Server-seitige Informationsflussanalyse 7

Arbeitspaket 3.1: Performantes Taint Tracking 8

Arbeitspaket 3.2: Taint-kompatible Kommunikationsprotokolle und Persistierungsmechanismen 8

Arbeitspaket 3.3: Identifizierung von datenschutzrelevanten Datenquellen und -senken 8

Arbeitspaket 4: Analyse des Informationsflusses in Firmware für Embedded Devices 8

Arbeitspaket 5: Rehosting von Firmware mit automatischer Emulator-Erstellung 8

Arbeitspaket 6: Integration in das Framework zur ganzheitlichen Durchsetzung des Datenschutzes 8

Arbeitspaket 7: Auswertung und Demonstrator 8

Die wichtigsten Positionen des zahlenmäßigen Nachweises **8**

Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten **9**

Verwertbarkeit des Ergebnisses **10**

Während der Durchführung des Vorhabens dem Zuwendungsempfänger bekannt gewordenen Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen **10**

Erfolgte oder geplante Veröffentlichungen des Ergebnisses nach Nr. 5 der NKBF/NABF **10**

Teil II – Eingehende Darstellung

Ursprüngliche Vorhabensbeschreibung

Heutige IoT-Geräte sammeln riesige Mengen an Daten und teilen diese mit Cloud-Diensten. Da vernetzte Geräte mit ihren Sensoren ihre Umgebung durchgängig erfassen, ist es nicht mehr möglich, sich der Aufzeichnung von potenziell sensitiven Daten durch ein opt-out zu entziehen. Geräte wie autonome Fahrzeuge, interaktive Bildschirme und Smart-Glasses werden persönliche Informationen aller Passanten aufzeichnen. Hierdurch entsteht ein Konflikt zwischen dem Recht des Einzelnen auf Privatsphäre und dem Interesse der Gesellschaft, sich die Vorteile der Digitalisierung und Big-Data-Analysen nutzbar zu machen.

Das Ziel von ENCOPIA war es, Verfahren zu erforschen, die eine möglichst automatisierte Prüfung von Datenschutzzielen von IoT-Diensten - von Sensorgeräten bis hin zu den Cloud-Diensten - während der Entwicklung bis nach der Markteinführung -ermöglichen.

Die wissenschaftlichen und technischen Ziele umfassten abgeleitet davon die Erforschung und Erprobung von binären Analysemethoden, um Datenschutzziele für sensible Daten auf automatisierte Weise für den gesamten Lebenszyklus überprüfbar zu machen.

Dazu sollte mit einer Kombination aus Funktionalitätserkennung und der Verfolgung des Informationsflusses sichergestellt werden, dass die Nutzung von sensiblen Daten in (Binär-) Programmen überprüft werden kann.

Langlaufs Schwerpunkt als Spezialist für die Binäranalyse war insbesondere die Erforschung eines „Proof of Concepts" für die Wiederherstellung von Funktionsnamen samt Position im Code (für eine darauf basierende Abbildung auf die Funktionalität). Dafür sollten sowohl signaturbasierte als auch nichtsignaturbasierten Methoden erforscht werden, siehe Arbeitspaket 2 in der GVB.

Darstellung der durchgeführten Arbeiten

Wir gehen im Folgenden auf die geleisteten Arbeiten ein und orientieren uns dabei an den in der TVB aufgeführten Punkten mit Langlauf Beteiligung. Wir gehen insbesondere auf Abweichungen im Vergleich zur ursprünglichen Vorhabenbeschreibung ein.

Arbeitspaket 1: Identifizierung geeigneter Methoden für die datenschutzorientierte Binäranalyse

Arbeitspaket 1.1: Identifikation und Auswahl von Methoden zur Funktionserkennung und Analyse von Sicherheitsmaßnahmen

Im Rahmen von Arbeitspaket 1.1 führte Langlauf eine Sichtung und Auswertung aktueller Verfahren zur Erkennung von Funktionen und Funktionalitäten in Binär- und Bytecode durch. Dazu erfolgte eine Recherche in aktueller wissenschaftlicher Literatur, die auch Verfahren zum Auffinden bekannter Sicherheitslücken in Binär- und Bytecode einbezog. Auf Grundlage dieser

Auswertung wurden besonders geeignete Verfahren, Methoden und Forschungsrichtungen identifiziert, die sich für die anschließende Forschung im Projekt ENCOPIA eigneten.

Als besonders relevant sind die folgenden Veröffentlichungen hervorzuheben:

- Rahimian, A., Nouh, L., Mouheb, D., & Huang, H. (2020). Binary Code Fingerprinting for Cybersecurity.
- Kim, Dongkwan, et al. "Revisiting binary code similarity analysis using interpretable feature engineering and lessons learned." arXiv preprint arXiv:2011.10749 (2020).
- Haq, Irfan Ul, and Juan Caballero. "A survey of binary code similarity." ACM Computing Surveys (CSUR) 54.3 (2021): 1-38.
- Khoo, W. M., Mycroft, A., & Anderson, R. (2013, May). Rendezvous: A search engine for binary code. In 2013 10th Working Conference on Mining Software Repositories (MSR) (pp. 329-338).IEEE.
- Zhang, D., Luo, P., Tang, W., & Zhou, M. (2020, December). OSLDetector: identifying open-source libraries through binary analysis. In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (pp. 1312-1315).

Eine interessante Erkenntnis war, dass sich Erfahrungen und Ansätze aus dem Bereich der „klassischen" Signatur-basierten Malware-Erkennung für unsere Zwecke nutzen lassen. Trotz der notwendigen Anpassungen an den Anwendungsfall „Goodware" liefern die folgenden Arbeiten wertvolle Impulse, insbesondere für einen Datenmarker-basierten Ansatz (siehe Arbeitspaket 2.6):

- Kephart, J. O. (1994). Automatic extraction of computer virus signatures. In Proc. 4th Virus Bulletin International Conference, Abingdon, England, 1994 (pp. 178-184).
- Griffin, K., Schneider, S., Hu, X., & Chiueh, T. C. (2009, September). Automatic Generation of String Signatures for Malware Detection. In RAID (Vol. 5758, pp. 101-120).

Arbeitspaket 1.2: Identifikation und Auswahl von Methoden zur Informationsflussanalyse in Binärdateien und Bytecode

Im Rahmen des Arbeitspakets 1.2 zur Identifikation und Auswahl von Methoden zur Informationsflussanalyse in Binärdateien und Bytecode hat Langlauf Recherchen zu geeigneten Werkzeugen und Methoden durchgeführt. Ziel war es, Verfahren zu identifizieren, die eine zuverlässige und idealerweise False-Positive-freie Datenflussanalyse ermöglichen.

Im Zuge dieser Recherchen wurden die Methode „Micro Execution" und die Binary Analysis Platform (BAP) als vielversprechender Ansatz bzw. Tool identifiziert.

Micro Execution ermöglicht die Ausführung beliebiger Codefragmente ohne die Notwendigkeit eines Testtreibers oder spezifischer Eingabedaten, indem es eine benutzerdefinierte Speicherpolitik verwendet, um Eingabewerte bereitzustellen, siehe

- Godefroid, P. (2014, May). Micro execution. In Proceedings of the 36th International Conference on Software Engineering (pp. 539-549).

BAP ist ein von der Carnegie Mellon University entwickeltes Werkzeug zur Analyse von Binärprogrammen, das verschiedene Architekturen wie x86, ARM, MIPS und PowerPC

unterstützt. Es bietet eine Vielzahl von Analysen, darunter einen Micro-Execution-Interpreter und einen symbolischen Exekutor, siehe

- https://github.com/BinaryAnalysisPlatform/bap

BAP ist in der Programmiersprache Ocaml geschrieben und verwendet eine eigene domänenspezifische Sprache namens Primus Lisp zur Implementierung von Analysen und zur Interaktion mit dem Micro Execution Framework. Eine gründliche technische Evaluierung von BAP erwies sich insbesondere durch diesen eher unüblichen Technologiestapel als sehr zeitintensiv und musste angesichts der begrenzten Personenmonate in diesem Arbeitspaket abgebrochen werden.

Arbeitspaket 2: Erkennung von datenschutzrelevanten Funktionalitäten und Datenschutzverletzungen

Das Arbeitspaket 2 war der Schwerpunkt von Langlaufs Forschungsarbeiten und wurde daher auch durch Langlauf geleitet.

Arbeitspaket 2.1: Systematische Erfassung datenschutzrelevanter Funktionsnamen

Im Rahmen von Arbeitspaket 2.1 wurde ein System zur systematischen Sammlung von datenschutzrelevanten Funktionsnamen entworfen und implementiert, sowie eine entsprechende Datenbank mit Bibliotheks- und Funktionsnamen erstellt.

Arbeitspaket 2.2: Erzeugung von Test- und Trainingsdaten

Ferner wurde in Arbeitspaket 2.2 unter anderem ein Verfahren entwickelt, um basierend auf günstigen AWS "Spot Instanzen" große Mengen von verschiedenen Varianten einer bestimmten Bibliothek zu kompilieren.

Arbeitspaket 2.3: Günstige Methoden mit hoher Präzision

Hintergrund: Für die technische Erkennung von (datenschutzrelevanten) Bibliotheksfunktionen ist es entscheidend, ob die zugehörige Bibliothek statisch oder dynamisch eingebunden wird. Der Code aus statisch gelinkten Bibliotheken wird beim Kompilieren in das Binary kopiert. Im Binary werden dann nur die auf Adressen basierenden Referenzen angepasst. Die Namen der Bibliotheksfunktionen müssen daher nicht unbedingt im finalen Binary enthalten sein. Unter Windows ist dies z.B. der Standardfall. Die Erkennung solcher „namenlosen" Funktionen wird in Arbeitspaket 2.4 behandelt. Im Gegensatz zu statisch gelinkten Bibliotheken verbleibt der Code von dynamisch gelinkten Bibliotheken in der Bibliotheksdatei, d.h. der DLL unter Windows bzw. dem shared object unter Linux. Referenzen von Funktionen erfolgen meistens (indirekt) über den Funktionsnamen. Die Namen sind folglich sehr häufig im Binary vorhanden und können auch nicht entfernt werden. Dies vereinfacht die Erkennung, ob eine Funktion im Binary enthalten ist.

Wir haben im Rahmen dieses Arbeitspaketes ein automatisiertes Verfahren entwickelt, dass zum einen etwaige vorhandene Funktionsnamen ausliest und darüber hinaus in vielen Fällen auch Aufrufe dieser Funktionen erfasst. Hier sind vor allem lange bzw. auf dynamischen Werten basierende Aufrufketten ein Problem und erfordern letztendlich eine umfassende

Daten- und Kontrollflussanalyse. Die Erkennung, ob eine Bibliotheksfunktion vom finalen Binary aufgerufen wird, war - wie erwartet - eine große technische Herausforderung, die zusätzlich noch vom Binärformat und auch der Prozessorarchitektur abhängt.

Das System wurde im Anschluss um die automatische Erfassung von dynamisch gelinkten Funktionen erweitert. Diese Erweiterung stellte sich als einfacher als erwartet heraus, da diese importierten Funktionen sowohl bei Windows PE Binaries als auch bei Linux ELF Binaries in dedizierten Tabellen in den Header-Meta-Daten aufgeführt sind.

Arbeitspaket 2.4: Teure Methoden mit hoher Präzision (Signatur-basierte Erkennung)

Wie oben beschrieben, ist der Code von statisch gelinkten Funktionen in vielen Fällen ohne das zugehörige Symbol, also ohne den Funktionsnamen, im finalen Binary. Das derzeitig verbreitete Verfahren zur Erkennung von statisch gelinkten Bibliotheks-Funktionen nutzt sogenannte Funktionssignaturen. Eine Funktionssignatur besteht aus einer Art Fingerabdruck, basierend auf den zur Funktion gehörenden Assembler-Instruktionen bzw. Opcodes. Dieser Fingerabdruck wird mit dem Funktionsnamen verknüpft gespeichert. Das bedeutet, dass zur Erstellung der Signatur die zu erkennende Funktion in kompilierter Form mit Funktionsnamen vorliegen muss. Hier ist zu beachten, dass eine Funktion im Quellcode zu praktisch unendlich vielen unterschiedlichen Instruktionen umgewandelt werden kann – abhängig unter anderem vom Compiler, der Compiler-Version und den Compiler-Flags.

Nachdem eine Signatur erstellt ist, kann diese bei einer kompilierten Software mit fehlenden Funktionsnamen auf die namenlosen Funktionen angewendet werden. „Trifft" (matcht) die Signatur, wird der Funktionsname aus der Signatur übernommen, die Funktion also „erkannt". Die Erforschung der Übertragung dieses Ansatzes auf eine automatische und großskalige Anwendung war Gegenstand dieses Arbeitspakets.

Dazu wurde als erster Schritt eine konzeptuelle und praktische Evaluation, sowie Gegenüberstellung aktueller Verfahren zur Signatur-basierten Funktionserkennung (u.a. IDA Flirt, Radare2 signatures, Ghidra Function IDs) mit Blick auf Skalierbarkeit, Performance und Robustheit durchgeführt. Die Robustheit bezog sich dabei nicht nur auf die eigentlich Erkennung, sondern auch auf den gesamten notwendigen Technologie-Stapel.

Es zeigte sich, dass sich die oben genannten existierenden Verfahren konzeptuell stark ähneln, inklusive der Datenstrukturen der Signaturen. Eine wesentliche Gemeinsamkeit ist auch, dass die Umsetzung des Signatur-Matchings für den Anwendungsfall „Disassembler" ausgelegt ist. Das bedeutet, dass versucht wird die Signaturen sequenziell auf die Bytes des im Disassembler geladenen Binaries zu matchen.

Dieses Verfahren ist für die manuelle Analyse einzelner Binaries mit Hilfe eines Disassembler passend, aber nicht für eine von uns angestrebte skalierbare und voll automatisierte Signatur-basierten Funktionserkennung. In unserem Anwendungsfall gibt es perspektivisch Millionen verschiedene Funktionen in jeweils Hunderten von Varianten, also Matching-Vorgängen im Milliardenbereich. Dafür stellten sich die die existierenden Verfahren als ungeeignet heraus.

Wir haben daher einen gänzlich anderen Matching-Ansatz erforscht der – basierend auf unseren derzeitigen Experimenten - der Schlüssel für eine wirklich skalierbare Anwendung von Funktionssignaturen ist.

Darüber hinaus wurde ein Demonstrator entwickelt, um „unsere" Signaturen großskalig zu generieren. Eine wichtige Erkenntnis war, dass der von uns initial gewählte Technologiestapel rückblickend keine gute Entscheidung war. Denn widererwartend erwies sich Docker Swarm als nicht ausreichend robust und das Projekt wird überdies auch praktisch nicht mehr weiterentwickelt.

Die Evaluation mit Hilfe der in Arbeitspaket 2.2 erzeugten Test-Samples konnte zwar durchgeführt werden. Für eine weitere produktivere Nutzung des Systems sind jedoch wesentliche Entwicklungsarbeiten notwendig, um Docker Swarm aus dem Technologiestapel zu entfernen und gegen eine andere, robustere Technologie auszutauschen. Für die positive Demonstrierung des Ansatzes war dies jedoch nicht entscheidend.

Arbeitspaket 2.5: Erforschung & Entwicklung von Methoden zur abstrakten Funktionalitätserkennung

In diesem Arbeitspaket hat Langlauf vorwiegend unterstützend gewirkt. Dies gilt sowohl für die Recherche der verschiedenen existierenden Verfahren als auch für die praktischen Evaluation von bereits existierenden Tools basierend auf den in Arbeitspaket 2.2 erzeugten Test-Samples.

Arbeitspaket 2.6: Daten-basierte Erkennung datenschutzrelevanter Eigenschaften

In den Arbeitspaketen 2.1 bis 2.4 basiert die Erkennung von datenschutzrechtlich relevanter Funktionalität vorwiegend auf der Erkennung von im Binary enthaltenen Funktionen. Ein weiterer Ansatz zur Erfassung von relevanter Funktionalität bzw. relevanten Eigenschaften ist die Erfassung und Auswertung von im Binary enthaltenen Daten. Dazu zählen u.a. bestimmte konstante Werte, Strings, aber auch Ressourcen und Binary-Metadaten. Die Identifikation relevanter Daten sowie ihre automatisierte Erfassung und Auswertung wurden in diesem Arbeitspaket gebündelt.

Wir haben das Problem bei der Daten-basierten Erkennung in zwei Unterprobleme aufgeteilt: Die Erkennung von (1) Zeichenketten / „Strings" und die Erkennung von (2) Daten. Der Begriff Daten umfasst hier rohe Bytes mit Werten, die nicht als ASCII-Zeichen repräsentiert werden können.

Während es im Fall von Strings eingängige Vorarbeiten für eine (naive) Erfassung gibt, hat unsere Recherche nur sehr wenige Vorarbeiten zur Erfassung von Daten in Binärdateien gefunden. Ein Grund ist vermutlich, dass es ist im Binary nicht unmittelbar ersichtlich ist, wo ein Datum startet und endet. Auch ist nicht klar, ob die genauen zu einem Datum gehörigen Bytes abhängig vom Compiler, Compiler-Version und/oder Compiler-Flags sind.

Wir haben einen vielversprechenden Ansatz konzipiert, der das Problem für bestimmte Arten von Daten in ELF Binaries löst.

Arbeitspaket 2.7: Erkennung von (Sub-)Komponenten mit bekannten und veröffentlichten datenschutzrelevanten Sicherheitsproblemen

Ziel dieses Arbeitspaketes war die Erforschung von Verfahren zur Erkennung von Software und (Sub-)Komponenten, die bekannte bzw. veröffentlichte, und für den Datenschutz relevante, Schwachstellen enthalten. Die Idee war, nach einer möglichst präzisen Identifizierung einer Software bzw. einer in dieser Software enthaltenen

Komponente, die NVD Schwachstellendatenbank nach veröffentlichten Schwachstellen für diese Software(-komponente) zu befragen.

Für dieses Ziel hat Langlauf an automatischen, aber dennoch ausreichend präzisen und generischen Identifikationsverfahren geforscht. Ein erster Ansatzpunkt, neben etwaig vorhandenen Meta-Daten, war die Verwendung der in Arbeitspaket 2.2 erzeugten Funktions-Signaturen. Dies wurde um weitere, vorab zu erfassende, Informationen wie statische Daten und Strings, siehe Arbeitspaket 2.6, erweitert.

Es wurden Experimente mit einer String-basierten Komponentenidentifikation durchgeführt und ein Demonstrator gebaut, der die OpenSSL Applikation, aber auch die OpenSSL Bibliotheken in statisch gelinkten Linux ELF Binaries erkennt. Die Ergebnisse waren deutlich besser als gedacht, sogar bei der Identifizierung der statisch gelinkten Bibliotheken: Die Bibliothek wurde in fast allen Fällen erfolgreich identifiziert. Viel erstaunlicher war jedoch, dass selbst die Version in den meisten Fällen auf wenige, oftmals eine Version, eingegrenzt werden konnte.

Die ausschließlich positiven Ergebnisse wurden allerdings bei Experimenten mit einer weiteren Bibliothek relativiert. Die Erkennung der sehr populären, aber auch sehr umfangreichen und komplexen C++ boost Bibliothek lieferte nur durchwachsene Ergebnisse. Wir konnten dafür unter anderem folgende Gründe identifizieren:

- Die Bibliothek boost ist weniger spezialisiert als openssl und enthält daher mehr Strings, die auch in anderen Bibliotheken vorkommen. Extreme Beispiele für Strings dieser Kategorie sind „True", „None", „Error" usw."
- Sofern boost mit einer Microsoft Toolchain gebaut wird, besteht eine bidirektionale Abhängigkeit mit der Microsoft STL Bibliothek. D.h. boost inkludiert die Microsoft STL, aber die STL selbst enthält wiederum auch Teile von Boost (u.a. Boost Math).

Der ersten Herausforderung konnten wir mit einer intelligenten Nachbereitung der Auswertung begegnen. Ideen zur Lösung der Abhängigkeitsprobleme konnten aus zeitlichen Gründen nicht vollständig umgesetzt werden.

Arbeitspaket 3: Server-seitige Informationsflussanalyse

Arbeitspaket 3 wurde in großen Teilen vom französischen Partner SAP bearbeitet.

Langlauf hat vor allem in den Bereichen unterstützt, in denen Überschneidungen mit Arbeitspaket 2 bestanden.

Arbeitspaket 3.1: Performantes Taint Tracking

Langlauf unterstütze SAP bei der Umsetzung des erforschten performance-optimierten Taint Tracking Verfahren, insbesondere durch die in Arbeitspaket 1.1 und 1.2 gewonnen Erkenntnisse.

Arbeitspaket 3.2: Taint-kompatible Kommunikationsprotokolle und Persistierungsmechanismen

Keine Beteiligung

Arbeitspaket 3.3: Identifizierung von datenschutzrelevanten Datenquellen und -senken

Langlauf unterstützte bei der Entwicklung von Techniken zur automatischen Identifikation von clientseitigen Quellen sensibler Daten, insbesondre durch die Wiederherstellung von Funktionsnamen, siehe Arbeitspaket 2.4.

Arbeitspaket 4: Analyse des Informationsflusses in Firmware für Embedded Devices

Keine Beteiligung

Arbeitspaket 5: Rehosting von Firmware mit automatischer Emulator-Erstellung

Keine Beteiligung

Arbeitspaket 6: Integration in das Framework zur ganzheitlichen Durchsetzung des Datenschutzes

Langlauf unterstütze bei der Integration der durch Methoden aus Arbeitspaket 2 gewonnenen Quellen-Informationen.

Arbeitspaket 7: Auswertung und Demonstrator

Langlauf unterstützte bei der Entwicklung von Proof-of-Concept Analysekomponenten, insbesondere basierend auf den sich aus AP2.4 ergebenen Ergebnissen.

Eine Herausforderung für Langlauf war, dass die Firma Hex-Rays das Lizenzmodell des für den automatisierten Service genutzten IDA Pro Disassemblers geändert hat. Zwar waren die gekauften Lizenzen weiterhin gültig und die geplante Forschung und Experimente konnten durchgeführt werden. Dennoch musste für den Demonstrator auf die Nutzung eines anderen Disassemblers umgestellt werden, um den lizenzbedingten Einschränkungen auszuweichen (u.a. eine nicht erlaubte Bereitstellung des Systems als Software as a Service). Dies erforderte zusätzlichen Integrationsaufwand.

Die wichtigsten Positionen des zahlenmäßigen Nachweises

Wie aus Abbildung 1 ersichtlich ist, wurden 96,7 % der Mittel für Personalausgaben aufgewendet. Auf ‚vorhabenspezifische Abschreibungen' entfielen 1 %, auf Reisekosten 0,4 % und auf ‚sonstige unmittelbare Vorhabenkosten' 1,9 % der insgesamt eingesetzten Mittel.

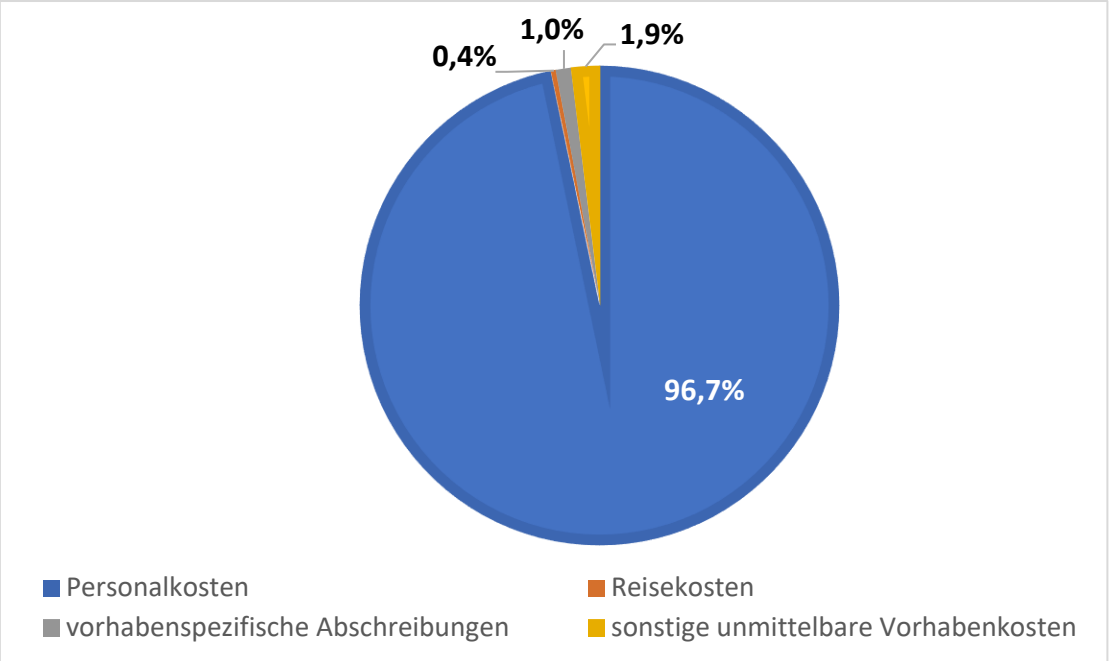

Abbildung 1: Ausgaben nach Kostenpositionen in Prozent

Abbildung 2 zeigt die Ausgaben nach Kostenpositionen in absoluten Euro-Beträgen:

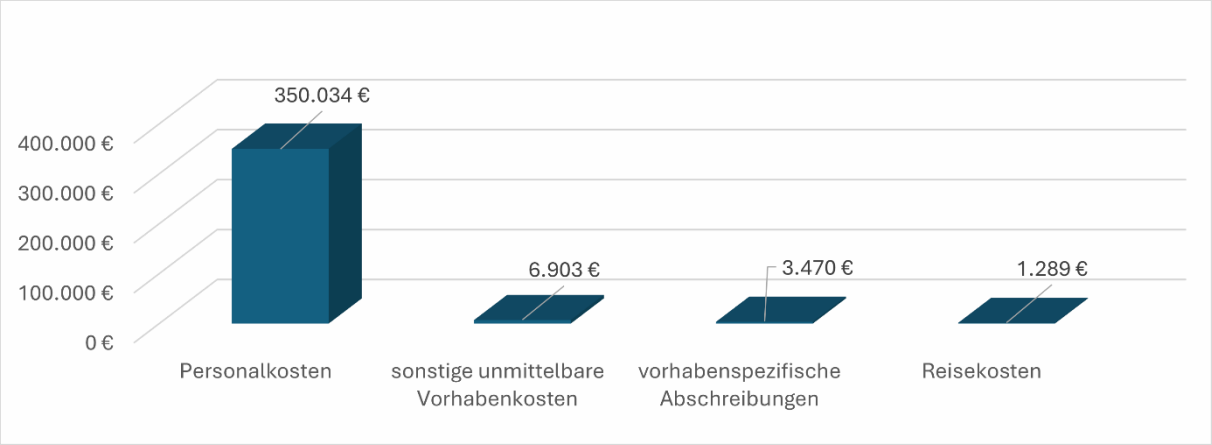

Abbildung 2: Ausgaben nach Kostenpositionen in Euro

Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten

Die im Rahmen des ENCOPIA-Projekts durchgeführten Arbeiten waren notwendig, um die in der ursprünglichen Vorhabenbeschreibung definierten Forschungsziele systematisch zu bearbeiten und technisch zu validieren.

Die Auswahl ressourcenoptimierter Technologien und die Umsetzung mehrerer Proof-of-Concept-Demonstratoren stellten sicher, dass der projektbezogene Aufwand in einem angemessenen Verhältnis zu den erzielten Ergebnissen stand und eine praxisnahe Validierung der Konzepte ermöglichte.

Verwertbarkeit des Ergebnisses

Die im Projekt entwickelten Verfahren bilden die Basis für die Umsetzung in ein kommerzielles Privacy-Assessment-SaaS-Produkt für kleine und mittlere Unternehmen sowie ein ergänzendes Tool zur Unterstützung manueller Datenschutzaudits. In den nächsten sechs Monaten ist die Durchführung von Pilotprojekten mit fünf ausgewählten Kunden geplant, um die Lösung in der Praxis zu validieren, gefolgt von einem kommerziellen Rollout in Deutschland und Österreich im ersten Jahr nach Projektende und einer anschließenden Ausweitung auf den übrigen EU-Markt. Mittelfristig wird ein Umsatzwachstum von bis zu 50 % innerhalb von zwei Jahren angestrebt, mit dem Ziel, spätestens nach vier Jahren einen Jahresumsatz von über 700.000 € zu erzielen. Erste Gespräche zeigten, dass es überdies eine ggf. noch stabilere und planbarere Nachfrage nach einer Komponentenerkennung zur SBOM-Erstellung gibt, die nicht primär datenschutzgetrieben, sondern vor allem in den neuen Security Compliance Vorgaben wie dem CRA (The Cyber Resilience Act) ihren Ursprung hat.

Während der Durchführung des Vorhabens dem Zuwendungsempfänger bekannt gewordenen Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Langlauf wurden keine wesentlichen Fortschritte auf dem Gebiet der *präzisen* Funktionserkennung bzw. Komponentenerkennung bekannt. Zwar wurden zahlreiche Paper zur Verwendung von KI und LLMs veröffentlicht. Diese Ansätze sind aber derzeit weder präzise noch stichhaltig genug und zielen daher eher auf andere Anwendungsfälle wie z.B. Code Clone Detection ab, in denen sogar bewusst eine gewisse Varianz erkannt werden soll.

Ein für Langlauf relevanter Fortschritt ist hingegen die signifikante Weiterentwicklung und Reifung des Open Source Projektes „Dependency Track". Dependency Track erlaubt bei vorliegender Software Bill of Materials, also einer Liste von Komponenten-IDs mit zugehörigen Versionen, den Abgleich mit mehreren führenden Schwachstellendatenbanken wie z.B. der NVD oder Google OSV.

Erfolgte oder geplante Veröffentlichungen des Ergebnisses nach Nr. 5 der NKBF/NABF

Als Industriepartner haben wir keine eigenen wissenschaftlichen Veröffentlichungen getätigt. Es wurden aber die akademischen Partner, insbesondere die Universität zu Lübeck, bei diversen Veröffentlichungen durch Daten und Feedback unterstützt.

Langlauf Security Automation GmbH

Martin-Hülsemann-Straße 9, 59558 Lippstadt

E-Mail: hello@langlauf.io

Site: www.langlauf.io