Sicherheit medizintechnischer Protokolle im Krankenhaus

Christoph Saatjohann[1] Fabian Ising[1] Matthias Gierlings[2] Dominik Noss[2] Sascha Schimmler[3] Alexander Klemm[4] Leif Grundmann[5] Tilman Frosch[3] Sebastian Schinzel[1]

Abstract: Medizinische Einrichtungen waren in den letzten Jahren immer wieder von Cyber-Angriffen betroffen. Auch wenn sich diese Angriffe derzeit auf die Office-IT-Infrastruktur der Einrichtungen konzentrieren, existiert mit medizinischen Systemen und Kommunikationsprotokollen eine weitere wenig beachtete Angriffsoberfläche.

In diesem Beitrag analysieren wir die weit verbreiteten medizintechnischen Kommunikations-Protokolle DICOM und HL7 sowie Protokoll-Implementierungen auf ihre IT-Sicherheit. Dafür präsentieren wir die Ergebnisse der Sicherheitsanalyse der DICOM- und HL7-Standards, einen Fuzzer "MedFUZZ" für diese Protokolle sowie einen Schwachstellenscanner "MedVAS", der Schwachstellen in medizintechnischen Produktivumgebungen auffinden kann.

Keywords: DICOM; HL7; TLS; Netzwerksegmentierung; Scanning; Fuzzing; Telematikinfrastruktur

1 Einleitung

Kürzlich bekannt gewordene Angriffe gegen medizinische Einrichtungen, z. B. gegen das Uniklinikum Düsseldorf, zielten auf die generische Office-IT-Infrastruktur ab. Jedoch sind auch medizinische Systeme, wie etwa bildgebende Systeme und Bildarchive, nicht vor Angriffen gefeit. So zeigte 2019 zum Beispiel die Firma Greenbone, dass zahlreiche DICOM-Systeme öffentlich zugänglich im Internet verfügbar waren und dort auch reale Patientendaten von Dritten heruntergeladen werden konnten [Gr19]. Obwohl diese Schwachstellen bekannt und medial aufbereitet wurden, konnten 2020 immer noch DICOM-Systeme sowie weitere Praxen- und Krankenhaus-Systeme in internetweiten Scans nachgewiesen werden [STB20].

In diesem Beitrag analysieren wir die gängigen medizintechnischen Protokolle DICOM und HL7 sowie zwei Implementierungen mit Fokus auf IT-Sicherheit.

Wissenschaftlicher Beitrag In Abschnitt 3 und 4 präsentieren wir Sicherheitsanalysen der DICOM- und HL7-Standards und beleuchten dabei die Aspekte Authentifizierung, Vertraulichkeit, Authentizität und Integrität. Wir betrachten außerdem Sicherheitslücken, die wir durch Fuzzing einer DICOM- und einer HL7-Implementierung aufgedeckt haben. Außerdem geben wir einen Überblick über in der Realität eingesetzte Sicherheitsmaßnahmen bei DICOM-Systemen. In Abschnitt 5 präsentieren wir unsere Plattform zum Testen

[1] FH Münster
[2] Ruhr-Universität Bochum
[3] G Data Advanced Analytics GmbH
[4] radprax Holding GmbH & Co.KG
[5] MedEcon Ruhr GmbH

doi:10.18420/sicherheit2022_09

medizinischer Software und Systeme. Hierbei liegt ein besonderes Augenmerk auf den beiden Hauptkomponenten, dem Fuzzer "MedFUZZ" und dem Scanner "MedVAS". Wir beschreiben insbesondere auch Erkenntnisse, die wir mit dem Scanner in realen Krankenhausumgebungen gemacht haben. In Abschnitt 6 fassen wir unsere Ergebnisse zusammen. Aufbauend geben wir in Anhang A konkrete Handlungsempfehlungen für Nutzende und Hersteller medizinischer Systeme und Vorschläge für die Weiterentwicklung der Standards.

Verwandte Arbeiten Klick et al. analysierten 2021 [KKB21] aus dem öffentlichen Internet aus erreichbare IP-Adressen deutscher Krankenhäuser und scannten dafür knapp 100 Ports pro IP-Adresse auf bekannte Services und Schwachstellen. Diese externen Scans zeigten bei 36% der Krankenhäuser bekannte Schwachstellen.

Eichelberg et al. untersuchten 2020 in einer Reihe von Papieren die Herausforderungen der IT-Sicherheit für die bildgebende Medizin [EKK20a], den aktuellen Literaturstand zur IT-Sicherheit in diesem Bereich [EKK20b] und die praktischen Probleme bei der Implementierung von IT-Sicherheit im Krankenhaus [EKK20c].

2018 befassten sich Dameff et al. [Da18] mit der Sicherheit von HL7. Die Autoren zeigten, dass es durch einfache TCP-MitM-Angriffe möglich ist, die Integrität von Patientendaten zu brechen. Anirudh Duggal zeigte 2017 ähnliche Angriffe [Du17]. Neben Denial-of-Service-Angriffen untersuchte der Autor hierbei auch mögliche Implementierungsfehler. 2019 untersuchten Wang et al. Fuzzing zum Testen von DICOM-Implementierungen [Wa19].

Danksagungen Dieser Beitrag wurde im Rahmen des Forschungsprojektes MITSicherheit.NRW (EFRE-0801419) vom Europäischen Fonds für regionale Entwicklung Nordrhein-Westfalen (EFRE.NRW) gefördert.

2 Grundlagen

DICOM Der Digital Imaging and Communications in Medicine (DICOM) Standard [Na21] definiert ein binäres Datenformat zur Speicherung und ein Kommunikationsprotokoll zum Austausch von medizinischen Bilddaten. Durch DICOM wird eine weltweite, herstellerunabhängige Interoperabilität der Daten von bilderzeugenden Geräten (Modalitäten, z.B. Computertomograph, Magnetresonanztomograph) und bildverarbeitenden, beziehungsweise -speichernden Systemen, z.B. dem Picture Archiving and Communication System (PACS), erreicht. Dadurch stellt es den De-facto-Standard für Bilddatenmanagement in der Medizin dar. Im Rahmen dieses Beitrags ist insbesondere das DICOM-Kommunikationsprotokoll von Interesse. Das DICOM Message Service Element (DIMSE)-Protokoll spezifiziert dabei den Austausch von Bilddaten und zugehörigen Daten. Nachrichten bestehen aus einem sogenannten "Command Set" und einem (optionalem) "Data Set". Einige relevante Kommandos des DIMSE-Protokolls sind in Tabelle 1 aufgelistet.

Kommando	Beschreibung
A-ASSOCIATE-RQ	Verbindungsanfrage
C-STORE	Ablage von Informationen
C-GET	Abruf von Informationen
C-ECHO	Echo-Funktionalität für Verbindungstests

Tab. 1: Liste der für diesen Beitrag relevanten DICOM-Kommandos, nach [Na21, Part 7].

Bei der Kommunikation identifizieren sich verschiedene DICOM-Systeme durch die Nutzung von Application Entity Titles (AETs). Dies sind maximal 16 Zeichen lange, netzwerkweit eindeutige Kennungen, wobei ein System auch mehrere AETs für verschiedene Komponenten nutzen kann [Na21, Part 5, Part 8]. Während des DICOM-Verbindungsaufbaus werden durch zwei Felder Informationen über die verwendete Implementierung ausgetauscht: "Implementation Class UID" (notwendig) und "Implementation Version Name" (optional).

HL7 Health Level 7 (HL7) International beschreibt mit HL7 eine Menge von Standards für den Austausch klinischer und administrativer Daten zwischen medizinischen Anwendungen. Obwohl seit 2005 HL7 Version 3 [He17] als neuer Standard verfügbar ist, ist Version 2 immer noch deutlich verbreiteter [Jo18]. Auch die Bereitstellung von HL7 über Webschnittstellen, genannt Fast Healthcare Interoperability Resources (FHIR), ist erst wenig verbreitet.

Die Nachrichtenkodierung in HL7 V2 ist prinzipiell mit XML möglich, meist wird aber das ASCII-basierte, menschenlesbare Electronic Data Interchange (EDI) Datenformat verwendet (siehe Beispiel in List. 1). Neben den fest definierten Nachrichtenelementen unterstützt HL7 applikationsspezifische Segmente und verschiedene Nachrichtenprofile. Durch diese Flexibilität im Standard haben sich verschiedene, teilweise inkompatible, Profile und Formate in der Praxis durchgesetzt. Für diesen Beitrag relevante Felder sind die `Sending Facility` und `Sending Application` in welchen Informationen über die sendende Anwendung und Institution angegeben werden können. Der HL7 V2 Standard wurde seit der Version 2.0 im Jahre 1989 stetig weiterentwickelt bis zum jetzigen Standard V2.9 [He19].

Telematikinfrastruktur-Konnektor Seit 2018 sind alle Arzt- und Zahnarztpraxen verpflichtet, einen *Konnektor* in der Praxis zu installieren [Bu15]. In weiteren Ausbaustufen der Telematikinfrastruktur (TI) wurden, und werden, schrittweise weitere Institutionen, wie Apotheken und Krankenhäuser, mit in die Anschlusspflicht genommen. Dieser Konnektor verbindet, unter anderem, das Praxisnetzwerk über einen VPN-Tunnel mit dem zentralen TI-Netz. Damit Clients aus dem internen Netzwerk mit dem Konnektor kommunizieren können, wird auf der internen LAN-Schnittstelle per HTTP eine Konnektorbeschreibungsdatei, *connector.sds*, angeboten, welche Informationen über Versionsstand, Konfiguration bestimmter Parameter und verfügbare Dienste bereitstellt.

```
MSH|^~\&||<sender>|||<dateTime>||<messageType>|<messageID>|
     <processingStatus>|<syntaxVersion>
PID|||<patientID>^^^<source>^<IDType>||<familyName>^<givenName>||
     <dateOfBirth>|<sex>|||<address>
PV1|||<patientLocation>|||||<patientsGP>
OBR|||<accessionNumber>|<testCode>^<testName>^<codeType>|||
     <specimenDate>||||||||<specimenSource>|<requester>
OBX||<valueType>|<observableCode>^<observableName>|<subID>|<valueCode>^
     <valueText>^<valueCodeType>||||||<abnormalFlag>
```

List. 1: Aufbau einer HL7 Version 2 Nachricht: Jede Zeile definiert ein Segment der HL7 Nachricht. Die einzelnen Felder werden mit | getrennt, ^ trennt Unterfelder. Beim ersten Feld handelt es sich jeweils um den Segment Typ. Felder in <> enthalten Patientendaten. Quelle: [BG16]

Schwachstellen-Scanning Schwachstellen-Scanner werden genutzt, um in Netzwerken Schwachstellen in Software und Geräten aufzudecken. Dabei werden sowohl Fehlkonfigurationen als auch, gegebenenfalls veraltete, Software mit Sicherheitslücken entdeckt. Ein freier und weitverbreiteter Schwachstellen-Scanner ist der Open Vulnerability Assessment Scanner (OpenVAS) [Gr]. Scanning ist ein wichtiger Teil des Schwachstellenmanagements [Bu].

Schwachstellen-Scanner können üblicherweise nur bereits bekannte Schwachstellen finden, sind aber nicht in der Lage neue Sicherheitslücken aufzudecken. So müssen Scanner zum Beispiel für das Scannen medizinischer Produkte zuvor um Module für die speziellen Netzwerkprotokolle und damit verbundene bekannte Sicherheitslücken erweitert werden.

Fuzzing Als Fuzzing oder Fuzz-Testing wird eine Technik zum automatisierten Auffinden von Sicherheitslücken in Software bezeichnet. Hierbei wird die zu testende Software mit, für diese, unerwarteten Eingaben getestet. Dieser Prozess wird dabei automatisch in hoher Frequenz durchgeführt. Ein Fuzzer generiert dabei die Eingabedaten üblicherweise aus existierenden Daten (mutation-based fuzzing) oder komplett neu auf Basis des verwendeten Protokolls oder Dateiformats oder komplett zufällig (generation-based fuzzing). [SGA07]

Es wird beim Fuzzing allgemein zwischen dem Blackbox-Fuzzing, bei dem eine als Binärdatei vorliegende Software ohne zusätzliche Informationen gefuzzt wird, und dem Guided Fuzzing unterschieden. Beim Guided Fuzzing werden zusätzliche Informationen genutzt, um bessere Eingabedaten zu generieren, die einen größeren Bereich der Software testen. Diese zusätzlichen Informationen werden üblicherweise durch Instrumentierung der Binärdatei, häufig während des Kompilierens, gewonnen. Diese Instrumentierung erlaubt etwa nachzuvollziehen, welche Pfade im Programm genommen werden. [Ma19]

Klassischerweise deckt Fuzzing Speicherprobleme und undefiniertes Verhalten in nicht speichersicheren Sprachen (zum Beispiel C oder C++) auf. Es kann aber auch nach anderen Fehlern, etwa nach unauthentifizierten Datenzugriffen oder Exceptions, gesucht werden.

3 Sicherheitsanalyse DICOM

Bei der Sicherheitsanalyse des DICOM-Standards betrachten wir die Sicherheitseigenschaften Authentifizierung, Verschlüsselung, Authentizität und Sicherheit ruhender Daten. Zuletzt legen wir dar, woran die Nutzung von Sicherheitsmechanismen in der Praxis scheitert.

Authentifizierung Die "User Identity Negotiation" [Na21, Part 7] erlaubt die Authentifizierung von Benutzenden mittels Benutzername, Benutzername und Passwort, Kerberos Tickets, SAML oder JSON Web Tokens (JWTs). Die Authentifizierung wird hierbei beim Aufbau einer Verbindung mit der `A-ASSOCIATE-RQ`-Nachricht durchgeführt. Hierbei ist ausschließlich eine Authentifizierung des Anfragenden, nicht des Servers vorgesehen.

Auch wenn AETs nicht explizit für die Authentifizierung vorgesehen sind, so erlauben viele reale Systeme nur den Zugriff mit freigegebenen AETs. Da diese aber nur maximal 16 Zeichen lang sind, häufig sprechende Namen wie Abteilungskürzel haben, und üblicherweise im Klartext ausgetauscht werden, stellt dies keine effektive Authentifizierung dar.

Transportverschlüsselung Der DICOM-Standard definiert die Kommunikation über TLS-verschlüsselte Verbindungen im optionalen "BCP 195 TLS Secure Transport Connection Profile" [Na21, Part 15], das den Vorgaben der Empfehlungen von BCP 195 [SHSA15] folgt. Dadurch wird die Verwendung von TLS 1.2 und 1.3 empfohlen, vor TLS 1.1 und 1.0 wird gewarnt. Strikter ist hier das optionale "Non-Downgrading BCP 195 TLS Secure Transport Connection Profile", das die Verwendung von TLS 1.0 und 1.1 verbietet und Ciphersuites mit kurzlebigen Schlüsseln vorschreibt. Kritisch ist hier, dass das DICOM Komitee die Verteilung und Generierung von Zertifikaten als außerhalb des Geltungsbereichs definiert, was die Implementierung in der Praxis stark erschwert. Gespräche mit IT-Verantwortlichen medizinischer Einrichtungen ergaben, dass TLS bei DICOM praktisch nicht eingesetzt wird.

Authentizität Um die Authentizitätsprüfung von DICOM-Daten zu ermöglichen, definiert der DICOM-Standard "Digital Signature Profiles" [Na21, Part 15], die beschreiben, wie ein Hash ausgewählter DICOM Daten signiert werden kann.[6] Dabei wird ausschließlich die Signatur mithilfe von EMSA-PKCS1-v1_5 [KS98] und auch explizit nur die Verwendung von RSA-Schlüsseln definiert. Die weiteren Signaturprofile definieren dabei, welche Datenelemente spezifischer DICOM-Objekte signiert werden sollten.

Bei den Profilen ist es problematisch, dass neben akzeptablen Hashalgorithmen wie SHA256, SHA384 und SHA512 und RIPEMD160 auch noch MD5 und SHA1 erlaubt sind, welche aufgrund praktischer Angriffe auf ihre Kollisionsresistenz als unsicher gelten [WY05, St17].

[6] Kurioserweise spricht der Standard hier von "MAC" (Message Authentication Code), obwohl hier mutmaßlich nur normale Hashes gemeint sind.

Weiterhin ist auch bei der Definition von TLS für Signaturen die Verteilung von Zertifikaten nicht näher spezifiziert, sondern wird als "site-specific" definiert. Auch die Einschränkung auf RSA-Zertifikate ist nicht mehr zeitgemäß. Am kritischsten ist jedoch, dass Signaturen ohne weiteres durch einen Angreifer entfernt werden können, ohne dass dies erkennbar wäre. Nur wenn ein System die Verwendung von Signaturen erzwingt, würde dies auffallen.

Sicherheit ruhender Daten Neben den Sicherheitsprofilen für den Austausch von DICOM-Daten, definiert der DICOM-Standard auch Sicherheitsprofile für ruhende Daten, die "Media Storage Security Profiles" [Na21, Part 15]. Darunter ist ein spezifisches Profil für die Kapselung von DICOM Dateien in "Secure DICOM Files" definiert. Hierbei werden die Daten mit AES oder Triple-DES verschlüsselt, und können entweder mit einer RSA-Signatur oder einem Hash geschützt werden. Die DICOM-Objekte werden dazu mithilfe der Cryptographic Message Syntax (CMS) verpackt. Zum Schutz des Content Encryption Keys kann entweder RSA oder die PBKDF2 genutzt werden.

Fragwürdig ist hierbei, dass der als unsicher geltende [BL16] Triple DES Algorithmus noch immer erlaubt ist, von dessen Nutzung bereits seit einigen Jahren abgeraten wird [NI17]. Die verwendbaren Hashalgorithmen sind das unsichere SHA1 und die aktuell als sicher eingestuften SHA256, SHA384, SHA512. Für Signaturen wird erneut nur RSA erlaubt.

DICOM in der Praxis In der Praxis scheitert die Implementierung der Sicherheitsmechanismen, neben fehlender Unterstützung durch die Produkte, an dem Fehlen von Zertifikaten. Da der Standard keine Vorgaben zur Erstellung und Verteilung von Zertifikaten macht und in medizinischen Praxen keine Strukturen dafür bestehen, werden nach unserer Erfahrung weder TLS noch digitale Signaturen eingesetzt. Auch Eichelberg et al. stellten 2020 das Zertifikatsmanagement als eine zentrale Herausforderung in diesem Bereich dar [EKK20c].

4 Sicherheitsanalyse HL7

Im aktuellen HL7 2.9-Standard liegt die Datensicherheit, konkret genannt werden die Verschlüsselung und Authentizität von Medizindaten, explizit nicht im Geltungsbereich des Standards [He19, Kapitel 1.8.2]. Die Verantwortung für die rechtliche und technische Einschätzung wird explizit den HL7-Implementierenden überlassen.

Während der Standard weder Sicherheitsziele noch -maßnahmen vorgibt, wurde 1999 ein sogenannter *non-balloted Standard Guide* veröffentlicht [Bl99], welcher technisch sehr detailliert Sicherheitsziele wie Vertraulichkeit, Authentizität und Integrität erklärt. Für die Erreichung dieser Ziele wird der Einsatz von TLS zwischen den Kommunikationspartnern empfohlen. Zu kritisieren ist, dass der Einsatz von TLS nur optional ist, und auch bis heute nicht vom Standard verpflichtend vorgegeben wurde. Auch der Standard Guide wurde seit Veröffentlichung nicht aktualisiert und enthält entsprechend veraltete Empfehlungen.

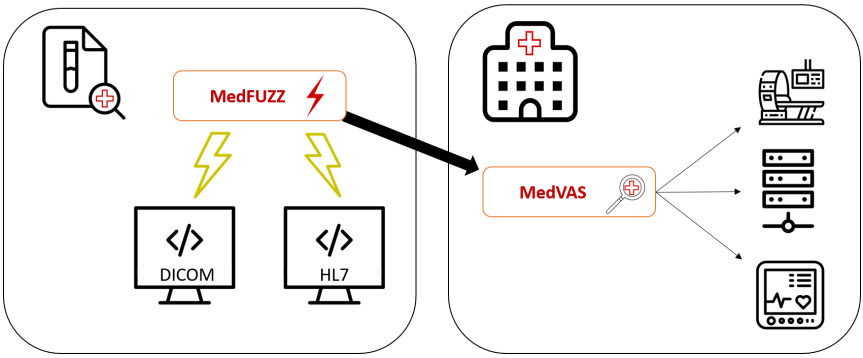

Abb. 1: Plattform zum Testen medizinischer Software und Systeme.

Beispielsweise werden im Zusammenhang mit TLS die unsicheren Algorithmen SHA1 und MD5 als “secure hash functions” genannt. Da es zum Veröffentlichungszeitpunkt noch keinen Advanced Encryption Standard (AES) gab, enthalten die Auflistungen der zu nutzenden Verschlüsselungen weitere unsichere Algorithmen wie RC2, RC4 oder DES.

Eine HL7-Implementierung, die sich allein auf diesen Leitfaden verlässt, und nicht eigenständig eine Aktualisierung auf den Stand der Technik durchführt, wird als Folge dessen mit großer Wahrscheinlichkeit verwundbar gegen bekannte Angriffe sein.

Das HL7 HCS Dokument aus dem Jahr 2014 [Se14] beschreibt ein interoperables Format für Sicherheits-Klassifizierung von HL7-Objekten. Während der Standard fünf verschiedene Sicherheits-Einstufungen definiert, bleibt die konkrete Umsetzung der Sicherheitsstufen den jeweiligen Softwareherstellern und Systembetreibern überlassen. Da keine weiteren technischen Erläuterungen verfügbar sind, und auch die verlinkten Pilotprojekte nicht mehr existieren, gehen wir davon aus, dass dieser Standard in der Praxis wenig relevant ist.

5 Softwareplattform zum Testen medizinischer Software und Systeme

Unsere Plattform zum Testen medizinischer Software und Systeme besteht aus zwei Komponenten, die in Abbildung 1 dargestellt sind. In einem ersten Schritt wurden in einer Laborumgebung Implementierungsfehler in DICOM- und HL7-Software durch den Fuzzer “MedFUZZ” identifiziert. Informationen über Software mit diesen Fehlern wurden in den auf OpenVAS basierenden Scanner “MedVAS” integriert, sodass dieser neben den üblichen Schwachstellen, z.B. in der Office-Infrastruktur, auch verwundbare medizinische Software erkennen kann. Im Folgenden beschreiben wir die beiden Komponenten genauer.

5.1 MedFUZZ - Fuzzer für medizinische Software

Zum Testen von DICOM- und HL7-Implementierungen haben wir den Fuzzer **MedFUZZ** auf AFL- [Za] und SharpFuzz-Basis [Mi19] entwickelt, mit dem mehrere Probleme aufgedeckt werden konnten. Neben der Durchführung des Fuzzings unterstützt die Fuzzing-Plattform Nutzende auch bei der Einrichtung der Umgebung, dem Bau der zu testenden Software und der Ergebnisauswertung (insb. Deduplizierung) mit Hilfe von Skripten. Außerdem enthalten sind Werkzeuge zur Übertragung der Ergebnisse auf Closed-Source-Software.

Als DICOM-Implementierung haben wir das weit verbreitete, und oft als Basis für kommerzielle Software genutzte, Open-Source, in ANSI C und C++ geschriebene DICOM-Toolkit (DCMTK) [Of] untersucht. In unser Testumgebung wird als Zielanwendung der DCMTK DIMSE-Kommunikation MedFUZZ genutzt, welcher statt standardkonformer DICOM-Nachrichten Testvektoren für das Fuzzing sendet. Die Kommunikation erfolgt aus Performancegründen nicht über ein physisches Netzwerk, sondern über ein Loopback-Interface.

Für die Untersuchung von HL7-Implementierungen wurde die ebenfalls weit verbreitete .NET Bibliothek NHapi [Co] ausgewählt. Als Einstiegspunkt wurde die Parsingroutine `PipeParser` gewählt, die für die Verarbeitung jeder eingehenden HL7-Nachricht aufgerufen werden muss, und in unserer Umgebung direkt MedFUZZ-Testvektoren entgegennimmt.

Wir haben zum Fuzzing ein zweistufiges Verfahren gewählt: Zuerst wurde ein Durchlauf ohne Instrumentierung zur Suche von Speicherfehlern durchgeführt, um möglichst schnell Testvektoren mit hoher Testabdeckung zu generieren. In einem zweiten Durchlauf haben wir die generierten Testvektoren als Basiseingaben (Seed) für den Fuzzer genutzt, um damit in den instrumentierten Binärdateien Speicherfehler aufzudecken. Durch dieses Verfahren kann trotz langsamerer instrumentierter Binärdateien schnell eine hohe Testabdeckung erreicht werden, ohne dabei Fehler in schwer zu erreichenden Code-Pfaden zu übersehen.

Zur tieferen Analyse der Testvektoren, die zum Absturz bzw. Einfrieren der Applikation geführt haben, wurden diese manuell in einem Debug-Build geprüft und die zugrundeliegenden Fehler identifiziert und an die Entwickler gemeldet. So wurden mehrere Memory Leaks und zwei Out-of-Bound-Reads im DCMTK aufgedeckt. Weiterhin wurden 62 Probleme in NHapi aufgedeckt, die unter anderem zu einem Denial-of-Service führen können.

Durch eine Kooperation konnten wir die DICOM Testvektoren auf einem DCMTK-basierten Closed-Source PACS evaluieren und die beiden gefundenen Bugs auch hier bestätigen.

5.2 MedVAS - Scanner für medizinische Systeme

Das Scannen von digitaler Krankenhausinfrastruktur ist zum einen durch die heterogene IT-Landschaft, und zum anderen durch die hohe Kritikalität risikoreicher als Schwachstellen-Scans in reinen Office-IT-Umgebungen. Für **MedVAS** (Medical Vulnerability Assess-

ment Scanner) haben wir OpenVAS als Basis genommen und für den Einsatz im laufenden Krankenhausbetrieb erweitert. Ein wichtiger Aspekt ist die Reduzierung der Scan-Geschwindigkeit, genauer die vom Scanner pro Sekunde verschickten Pakete. Diese Änderung ist notwendig da einige Medizingeräte äußerst fragil auf aggressive Scans reagieren [Ag19]. In eigenen Versuchen mit verschiedenen netzwerkfähigen Medizingeräten haben wir festgestellt, dass mindestens ein produktiv eingesetztes Gerät eine, vom Hersteller bestätigte, Intrusion-Detection-Funktion hat, welche bei einem Portscan die Funktion komplett einstellt und erst nach einem Werksreset wieder konfiguriert und genutzt werden kann.

Um neben OpenVAS bekannten Geräten auch Medizingeräte in Krankenhausinfrastrukturen zu identifizieren haben wir folgende Module entwickelt und in den Scanner integriert:

- DICOM-Server-Identifizierung
- HL7-Server-Identifizierung
- TI-Konnektor-Identifizierung
- TLS-Scanner-Integration

Das DICOM-Modul schickt an die zu testende IP-Adresse und Port eine DICOM-`C-ECHO`-Nachricht und wertet die ggf. empfangene Antwortnachricht aus. Hiermit kann neben der Existenz des DICOM-Servers in den meisten Fällen anhand der Antwort die eingesetzte Software samt Versionsnummer aus den Protokollfeldern `Implementation Version Name` und `Implementation Class UID` identifiziert werden.

Das HL7-Identifizierungs-Modul schickt, analog zum DICOM-Modul, eine HL7-Nachricht und überprüft die empfangene Nachricht auf HL7-Konformität und gibt, falls mitgeschickt, die `Sending Facility` und die `Sending Application` als Identifizierungsmerkmal mit an.

Das dritte Modul, die TI-Konnektor-Identifizierung, versucht von dem zu scannenden Gerät die XML-basierte Konnektor-Beschreibungsdatei `connector.sds` vom Port 80 und 443 mittels HTTP GET zu laden. Heruntergeladene Dateien werden auf Übereinstimmung mit der Konnektor-Spezifikation überprüft. Im Falle einer Übereinstimmung, werden die Konfigurationsfelder `ANCL_TLS_MANDATORY` und `ANCL_CAUT_MANDATORY` ausgegeben. Diese beiden Konfigurationsparameter sollten für einen sicheren Betrieb, und für die Nutzung zukünftiger Anwendungen wie beispielsweise der Komfortsignatur, auf `TRUE` stehen um die Verwendung der TLS-Transportverschlüsselung zwischen dem Konnektor und den Clients sowie die Authentifizierung beim Konnektor zu erzwingen [ge21].

Um gefundene TLS-unterstützte Systeme zu evaluieren wurde der TLS-Scanner [So16] in den Scanner integriert. Dieser evaluiert alle unterstützten TLS-Konfigurationen der gefundenen Server auf bekannte TLS-Schwachstellen.

Für MedVAS wurde auf das Client-Server-Konzept zurückgegriffen. Dadurch kann ein Dienstleister remote Überprüfungen verfolgen und auswerten. Der Scan-Client wird dazu in einem dafür vorgesehenen Netz mit den erforderlichen Zugriffsrechten installiert. Der Dienstleister kann den Client über den Server starten und die Ergebnisse auslesen und auswerten. So kann die krankenhausinterne IT-Abteilung sinnvoll entlastet werden, gerade falls keine ausreichenden Ressourcen für regelmäßige Schwachstellenscans vorhanden sind.

Evaluation im Krankenhaus Für die praktische Evaluation wurde MedVAS in einem großen radiologischen Versorgungszentrum, sowie in zwei Krankenhäusern während des Produktivbetriebs getestet. Die zu scannenden Netze wurden von der dortigen IT-Abteilung vorgegeben. Durch die defensive Scangeschwindigkeit von MedVAS dauerten die Scans teils über mehrere Tage. In diesen stichprobenartigen Tests funktionierte der Scanner zuverlässig und beeinflusste den medizinischen Betrieb nicht. Die häufigsten gefundenen Probleme waren Software außerhalb des Supportzeitraums sowie unsichere Standardpasswörter. In einem der Krankenhäuser erkannte MedVAS einen MIRTH-HL7-Server in einem Netzbereich in dem, laut Dokumentation, keine HL7-Server betrieben werden. Die vollständigen Berichte wurden den Krankenhäusern für die weitere Bearbeitung zur Verfügung gestellt.

In der Evaluation in der echten Krankenhausinfrastruktur haben wir mehrere Grenzen des Scanners festgestellt. Ein mögliches Hindernis für eine erfolgreiche Ausführung ist, dass der Scan nicht komplett vollautomatisch funktioniert. Das heißt insbesondere, dass die jeweilige IT-Abteilung eine gut gepflegte Dokumentation der genutzten IP-Adressen und Ports benötigt. Gerade bei Protokollen, welche in der Praxis keine standardisierten TCP-Ports verwenden, beispielsweise HL7, ist die Konfiguration der zu scannenden Ports essenziell für einen aussagekräftigen Bericht.

In unseren Praxistests konnten wir keine DICOM-Server identifizieren. Nach Rücksprache mit den IT-Abteilungen konnten wir dies darauf zurückführen, dass die PACSs nur mit zuvor fest konfigurierten IP-Adressen kommunizieren. Aus IT-Sicherheits-Sicht ist diese Konfiguration grundsätzlich positiv zu bewerten. Allerdings kann unser Scanner dadurch keine veraltete, oder sogar verwundbare, DICOM-Software identifizieren. Falls ein solcher Scanner dauerhaft genutzt werden soll, wäre die Vergabe einer entsprechend konfigurierten IP-Adresse eine Möglichkeit auch DICOM-Systeme zu identifizieren.

6 Fazit

Mit der immer weitergehenden Digitalisierung in der Gesundheitsbranche steigen auch die IT-Sicherheits-Risiken. Wir haben gezeigt, dass die aktuell gängigen Protokolle im klinischen Alltag einen unzureichenden Schutz gegen IT-Angriffe bieten und langfristig gegen moderne Verfahren ausgetauscht werden sollten.

Da solche Migrationen im stark regulierten Krankenhausumfeld langwierige Prozesse bedeuten, zeigen wir in Anhang A Handlungsempfehlungen auf, welche von Krankenhäusern und Medizingeräteherstellern kurzfristig in Erwägung gezogen werden sollten, sowie für die Erstellung von neuen Medizinstandards evaluiert werden sollten.

Des Weiteren wurde ein nicht-invasiver Schwachstellen-Scanner implementiert und evaluiert, welche die besonderen Umstände medizinischer Infrastruktur mit einbezieht und im produktiven laufenden Umfeld genutzt werden kann.

Literaturverzeichnis

[Ag19] Agnew, Joe: Medical Device Security, Part 1: How to Scan Devices Without Letting Safety Flatline. Bericht, Rapid7, April 2019. https://www.rapid7.com/blog/post/2019/04/29/medical-device-security-how-to-scan-devices-without-letting-safety-flatline.

[BG16] Benson, Tim; Grieve, Graham: Principles of Health Interoperability. Springer Verlag London, 3. Auflage, 2016.

[Bl99] Blobel, Bernd; Spiegel, Volker; Pharow, Peter; Engel, Kjeld; Krohn, Rolf: Standard Guide for EDI (HL7) Communication Security. 1999. https://www.hl7.org/implement/standards/product_brief.cfm?product_id=238.

[BL16] Bhargavan, Karthikeyan; Leurent, Gaëtan: On the Practical (In-)Security of 64-Bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS '16, Association for Computing Machinery, New York, NY, USA, S. 456–467, 2016.

[Bu] Bundesamt für Sicherheit in der Informationstechnik (BSI): Open Vulnerability Assessment System (OpenVAS). https://www.bsi.bund.de/EN/Topics/Industry_CI/ICS/Tools/OpenVAS/OpenVAS_node.html, abgerufen am 20.09.2021.

[Bu15] Bundesgesetzblatt Jahrgang 2015: Gesetz für sichere digitale Kommunikation und Anwendungen im Gesundheitswesen sowie zur Änderung weiterer Gesetze. Bundesanzeiger Verlag, Kapitel Teil 1, Nr. 54, S. 2408–2423, Dezember 2015. http://www.bgbl.de/xaver/bgbl/start.xav?startbk=Bundesanzeiger_BGBl&jumpTo=bgbl115s2408.pdf.

[Co] NHapi. http://nhapi.sourceforge.net/home.php, abgerufen am: 09.11.2021.

[Da18] Dameff, Christian; Bland, Maxwell; Levchenko, Kirill; Tully, Jeff: Pestilential Protocol: How Unsecure HL7 Messages Threaten Patient Lives. In: Blackhat USA 2018. August 2018. https://i.blackhat.com/us-18/Thu-August-9/us-18-Dameff-Pestilential-Protocol-How-Unsecure-HL7-Messages-Threaten-Patient-Lives-wp.pdf.

[Du17] Duggal, Anirudh: HL7 2.X Security. In: HITBSecConf 2017. April 2017. https://paper.bobylive.com/Meeting_Papers/HITB/2017/D2T2---Anirudh-Duggal---Hacking-Medical-Devices-and-Healthcare-Infrastructure.pdf.

[EKK20a] Eichelberg, Marco; Kleber, Klaus; Kämmerer, Marc: Cybersecurity Challenges for PACS and Medical Imaging. Academic Radiology, 27(8):1126–1139, 2020.

[EKK20b] Eichelberg, Marco; Kleber, Klaus; Kämmerer, Marc: Cybersecurity in PACS and Medical Imaging: an Overview. Journal of Digital Imaging, 2020.

[EKK20c] Eichelberg, Marco; Kleber, Klaus; Kämmerer, Marc: Cybersecurity Protection for PACS and Medical Imaging: Deployment Considerations and Practical Problems. Academic Radiology, 2020.

[ge21] gematik: Spezifikation Konnektor, Version 5.14.0. September 2021. https://fachportal.gematik.de/fachportal-import/files/gemSpec_Kon_V5.14.0.pdf.

[Gr] OpenVAS – Open Vulnerability Assessment Scanner. https://openvas.org, abgerufen am 09.11.2021.

[Gr19] Greenbone Networks GmbH: Sicherheitsbericht - Ungeschützte Patientendaten im Internet. September 2019. https://www.greenbone.net/wp-content/uploads/CyberResilienceReport_DE.pdf.

[He17] Health Level Seven International: Health Level Seven Version 3. Januar 2017. https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186.

[He19] Health Level Seven International: HL7 Messaging Standard Version 2.9. Dezember 2019. https://www.hl7.org/implement/standards/product_brief.cfm?product_id=516.

[Jo18] Joyia, Gulraiz Javaid; Akram, Muhammad Usman; Akbar, Chaudary Naeem; Maqsood, Muhammad Furqan: Evolution of Health Level-7: A Survey. In: Proceedings of the 2018 International Conference on Software Engineering and Information Management. ICSIM2018, Association for Computing Machinery, New York, NY, USA, S. 118–123, 2018.

[KKB21] Klick, Johannes; Koch, Robert; Brandstetter, Thomas: Epidemic? The Attack Surface of German Hospitals during the COVID-19 Pandemic. In: 2021 13th International Conference on Cyber Conflict (CyCon). S. 73–94, 2021.

[KS98] Kaliski, B.; Staddon, J.: PKCS #1: RSA Cryptography Specifications Version 2.0. Internet Engineering Task Force (IETF), Oktober 1998. https://datatracker.ietf.org/doc/html/rfc2437.

[Ma19] Manès, Valentin Jean Marie; Han, HyungSeok; Han, Choongwoo; Cha, Sang Kil; Egele, Manuel; Schwartz, Edward J.; Woo, Maverick: The Art, Science, and Engineering of Fuzzing: A Survey. IEEE Transactions on Software Engineering, S. 1–1, 2019.

[Mi19] SharpFuzz: Bringing the power of afl-fuzz to .NET platform. https://mijailovic.net/2019/01/03/sharpfuzz/, abgerufen am: 09.11.2021.

[Na21] National Electrical Manufacturers Association (NEMA): , The DICOM Standard. http://dicom.nema.org/medical/dicom/2021d/, 2021. Stand 2021d, abgerufen am 05.11.2021.

[NI17] NIST: Update to Current Use and Deprecation of TDEA. Juli 2017. https://csrc.nist.gov/News/2017/Update-to-Current-Use-and-Deprecation-of-TDEA.

[Of] DICOM-Toolkit (DCMTK). https://dcmtk.org/, abgerufen am: 09.11.2021.

[Se14] Security Work Group: HL7 Healthcare Privacy and Security Classification System (HCS). Bericht, HL7 International, August 2014. https://www.hl7.org/implement/standards/product_brief.cfm?product_id=345.

[SGA07] Sutton, Michael; Greene, Adam; Amini, Pedram: Fuzzing - Brute Force Vulnerability Discovery. Pearson Education Inc., 2007.

[SHSA15] Sheffer, Y.; Holz, R.; Saint-Andre, P.: Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). Internet Engineering Task Force (IETF), Mai 2015. https://tools.ietf.org/pdf/bcp195.pdf.

[So16] Somorovsky, Juraj: Systematic Fuzzing and Testing of TLS Libraries. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS '16, Association for Computing Machinery, New York, NY, USA, S. 1492–1504, 2016.

[St17] Stevens, Marc; Bursztein, Elie; Karpman, Pierre; Albertini, Ange; Markov, Yarik: The first collision for full SHA-1. In: Annual international cryptology conference. Springer, S. 570–596, 2017.

[STB20] Saatjohann, Christoph; Tschirsich, Martin; Brodowski, Christian: Tut mal kurz weh – Neues aus der Gesundheits-IT. In: Remote Chaos Experience (rC3). Dezember 2020. https://media.ccc.de/v/rc3-11342-tut_mal_kurz_weh_neues_aus_der_gesundheits-it.

[Wa19] Wang, Zhiqiang; Li, Quanqi; Wang, Yazhe; Liu, Biao; Zhang, Jianyi; Liu, Qixu: Medical Protocol Security: DICOM Vulnerability Mining Based on Fuzzing Technology. In: The 2019 ACM SIGSAC Conference. S. 2549–2551, 11 2019.

[WY05] Wang, Xiaoyun; Yu, Hongbo: How to Break MD5 and Other Hash Functions. In (Cramer, Ronald, Hrsg.): Advances in Cryptology – EUROCRYPT 2005. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 19–35, 2005.

[Za] american fuzzy lop. https://lcamtuf.coredump.cx/afl/, abgerufen am: 09.11.2021.

A Handlungsempfehlungen

Basierend auf unseren Ergebnissen geben wir kurzfristig umsetzbare Handlungsempfehlungen für Krankenhäuser und Hersteller digitaler Medizintechnik.

A.1 Krankenhäuser

Für Krankenhäuser gibt es neben den allgemein gültigen IT-Sicherheits-Empfehlungen spezielle Empfehlungen und Leitfäden. Für Krankenhäuser welchen den Vorgaben der Kritische Infrastrukturen (KRITIS) unterliegen, in der Regel ab 30.000 vollstationären Fällen pro Jahr, ist der Branchenspezifische Sicherheitsstandard (B3S) für die Gesundheitsversorgung im Krankenhaus verpflichtend umzusetzen. Auch kleinere Häuser sind durch §75c im SGB Vverpflichtet IT-Sicherheitsmaßnahmen nach dem *Stand der Technik* zu implementieren. Die hier empfohlenen Maßnahmen können die Sicherheit im Krankenhaus unmittelbar wirksam erhöhen, sie ersetzen allerdings nicht eine langfristige IT-Sicherheits-Strategie, welche beispielhafte Themen wie Backup-Systeme, Asset- und Patch-Management sowie Incident-Response-Prozesse betrifft.

Netzwerk- und Sicherheitsarchitektur Unsere internen Scans in realen Krankenhäusern und externe Scans haben gezeigt, dass oftmals Firewall- und Server-Konfigurationen nicht sicher umgesetzt sind. Nicht benötigte Portweiterleitungen und -freigaben für Anwendungen sollten entfernt werden, das betrifft insbesondere erlaubte Verbindungen aus dem Internet auf interne Systeme wie DICOM- oder HL7-Server. Es sollten generell nur Dienste freigegeben werden, die tatsächlich verwendet werden.

Weiterhin sollte das Netzwerk so segmentiert werden, dass lediglich Administratoren auf sicherheitskritische Systeme und Dienste zugreifen können. Dazu bietet sich die bereits vorhandene Unterteilung des Netzwerks in VLANs an. Es sollte evaluiert werden, ob die Implementierung einer Network Access Control (NAC)-Lösung sinnvoll ist, um unbekannten Geräten den Zugriff auf das Netzwerk zu verwehren.

Nicht verwendete Netzwerk-Ports innerhalb des Gebäudes sollten zudem deaktiviert werden, um sicherzustellen, dass frei zugängliche Ports nicht verwendet werden können.

Nutzung vorhandener Sicherheitsmechanismen Sicherheitsmaßnahmen, welche von den eingesetzten Geräten von Haus aus unterstützt werden, sollten standardmäßig genutzt werden. Das umfasst beispielsweise sowohl die verpflichtende Authentifizierung und Nutzung des TLS-Protokolls bei TI-Konnektoren als auch SSH-basierte Remoteverbindungen für Administrationszwecke, welche ausschließlich über eine Public-Key-Authentifizierung mit einem passwortgeschützten Schlüssel gewährt werden sollten.

Unsere Evaluation hat gezeigt, dass einige DICOM-basierte Systeme zwar TLS-Unterstützung anbieten, es sich im praktischen Betrieb allerdings schwierig gestaltet diese zu nutzen, da nicht alle im Krankenhaus genutzten Geräte TLS bzw. eine interoperable Konfiguration unterstützen. Auch das benötigte Zertifikatsmanagement ist oftmals nicht im Krankenhaus vorhanden. Aus dieses Grund empfehlen wir für DICOM-basierte Systeme als Basisschutz einen verpflichtenden individuellen und nicht erratbaren AET einzusetzen, und die erlaubten Verbindungen durch Nutzung einer *Allowlist*, welche nur explizit benötigte Verbindungen erlaubt, einzuschränken. Da diese Konfiguration keinem echten Passwortschutz entspricht, und viele System ein Durchprobieren des AET erlauben, kann der Schutz durch Netzwerk-Logging und automatisches Blocken bei zu vielen neuen Verbindungen auf die DICOM-Systeme weiter erhöht werden.

Regelmäßige Audits Es sollten regelmäßige IT-Sicherheitsüberprüfungen (z.B. Penetrationstests) der internen und externen IT-Infrastruktur durchgeführt werden. Dadurch können Sicherheitslücken und Schwachstellen aus technischer und organisatorischer Sicht identifiziert und behoben werden. Hierbei kann insbesondere MedVAS eingesetzt werden um automatisiert die interne Netzwerkinfrastruktur zu überprüfen. Regelmäßige Tests schärfen zudem das Bewusstsein der IT-Mitarbeiter für typische Fallstricke, was die IT-Sicherheit auch nachhaltig verbessert.

Prozesse Neben den technischen Aspekten sollten alle Prozesse in die IT-Sicherheitsüberlegungen mit einbezogen werden. Diese fangen bei dem Einkauf neuer Medizingeräte an, in welchen IT-Sicherheitsfunktionalitäten wie bspw. die Verwendung von Zwei-Faktor-Authentisierung (2FA) oder die Verwendung von Transportverschlüsselung als verpflichtend betrachtet werden sollten.

A.2 Hersteller und Standardisierungsorganisationen

Forcierung von IT-Sicherheitsfunktionalitäten in Geräten und Standards Wie wir in Abschnitt 3 gezeigt haben, ist es in der Praxis oftmals nicht möglich vom Standard unterstützte Sicherheitsmerkmale zu nutzen. Hier liegt es bei den Herstellern und deren Verbänden einheitliche interoperable Konfigurationen festzulegen welche regelmäßig auf sogenannten *Plugfests* evaluiert werden.

Neue Standardversionen sollten Verfahren, welche nicht mehr dem Stand der Technik entsprechen, konsequent als obsolet kennzeichnen und nach einer Karenzzeit als nicht mehr standardkonform behandeln. Weiterhin sollten Neu- und Weiterentwicklung von medizinischen Standards und Protokollen eine verpflichtende Nutzung von aktuellen Sicherheitsmaßnahmen wie Verschlüsselung, Authentizität und Integrität der Daten vorsehen.

Sichere Entwicklungs-Prozesse Durch unsere Fuzz-Tests auf DICOM- und HL7-Software haben wir mehrere sicherheitskritische Bugs gefunden und den Herstellern mitgeteilt. Fuzz-Tests werden in der modernen Softwareentwicklung immer mehr zum Standard und sollten dementsprechend auch in der Entwicklung von Medizinsoftware eingesetzt werden.