Examensarbete

HÖGSKOLAN I SKÖVDE

Detektionsmetoder För Skadlig Kod I IoT-Baserat Smart Hem

En Systematisk Litteraturstudie

IoT-Malware Detection Methods In Smart Home

A Systematic Literature Review

Examensarbete för kandidatexamen med huvudområdet Informationsteknologi

Grundnivå 30 Högskolepoäng

Vårtermin 2023

Student: William Saxmark

Handledare: Dennis Modig
Examinator: Ali Padyab

ABSTRACT

IoT devices are being widely deployed within smart homes. Most of these devices are mass-produced at a low cost. As a result, due to the lack of security mechanisms, IoT devices become vulnerable to malware. As more IoT devices are connected to the internet, and given their inability to maintain robust security, these devices are at an increased risk of being infected with malware. Compromised IoT devices enhance the capabilities of cybercriminals and threat actors to perform attacks and distribute malware. To prevent this, proper detection mechanisms are needed. However, traditional malware detection approaches are often not feasible in an IoT environment. This study compiles current detection methods used to detect IoT-malware in smart homes. Existing malware detection solutions will be included to demonstrate the methods, usage, and effectiveness in a specific context. This was achieved by performing a qualitative systematic literature review of articles from two databases with high technological relevance.

In total, 12 articles were utilized for the study. The data from these articles were subject to a thematic analysis, yielding two main themes: method and placement. The “method” theme consists of four categories: anomaly detection, signature detection, statistical analysis, and combination of methods. The “placement” theme consists of two categories: device-based and network-based. The study results indicate that both standalone methods and a combination of multiple methods are being employed for the detection of IoT-malware in smart home environments. Based on the results, anomaly-based detection emerges as the most used method for detecting IoT-malware, both on the device and within the network.

Keywords: malware, botnet, detection, IoT, smart home, methods

Sammanfattning

IoT-enheter implementeras i allt större utsträckning inom smarta hem. Många av dessa enheter massproduceras till låg kostnad. Som ett resultat blir IoT-enheter, på grund av bristande säkerhetsmekanismer, sårbara för skadlig kod. När fler IoT-enheter ansluts till internet, och med tanke på deras oförmåga att upprätthålla god säkerhet, löper dessa enheter en ökad risk för att infekteras med skadlig kod. Infekterade IoT-enheter ökar förmågan hos cyberkriminella och hotaktörer att utföra attacker och sprida skadlig kod. För att förhindra detta krävs lämpliga detektionsmekanismer. Traditionella metoder för att detektera skadlig kod är ofta inte genomförbara i en IoT-miljö. Denna studie sammanställer aktuella detekteringsmetoder som används för att upptäcka skadlig kod som riktas mot IoT-enheter inom smarta hem. Existerande lösningar för att detektera skadlig kod inom smarta hem kommer att inkluderas för att demonstrera metoderna, användningen och effektiviteten i ett specifikt sammanhang. Detta uppnåddes genom att utföra en kvalitativ systematisk litteraturstudie av artiklar från två databaser med hög teknologisk relevans.

Totalt användes 12 artiklar för att utföra studien. Data från dessa artiklar analyserades med tematisk kodning, som resulterade i två huvudteman, metod och placering. Temat ”metod” består av fyra kategorier: anomalibaserad detektion, signaturbaserad detektion, statistisk analys och kombination av metoder. Temat ”placering” består av två kategorier: enhetsbaserad och nätverksbaserad. Resultatet från studien indikerar på att både självständiga metoder och en kombination av flera metoder används för att upptäcka skadlig kod riktat mot IoT-enheter inom smarta hem. Baserat på resultatet framträder anomalibaserad detektion som den vanligaste metoden för att detektera skadlig kod riktat mot IoT-enheter, både på enheten och inom nätverket.

Nyckelord: skadlig kod, botnät, detektion, IoT, smarta hem, metoder

Innehållsförteckning

1 Introduktion

Den tekniska utvecklingen har lett till en dramatisk ökning av uppkopplade enheter, ofta samlade under benämningen Internet of Things (IoT). Dessa enheter har revolutionerat vardagen genom att automatisera och effektivisera flertalet aspekter av det dagliga livet. IoT Analytics (2022) rapporterar att det förväntas finnas cirka 27 miljarder uppkopplade IoT-enheter 2025. Framväxten av smarta hem, som blivit en integrerad del av många människors vardag, illustrerar denna utveckling. De erbjuder en mängd olika tjänster och funktioner, exempelvis smarta kylskåp och avancerade övervakningssystem, som bidrar till att höja livskvaliteten.

Denna snabba tekniska utveckling och ökade uppkoppling har medfört nya risker. Förekomsten av skadlig kod riktad mot IoT-enheter har blivit alltmer frekvent. IoT-enheter, som ofta karaktäriseras av begränsade hårdvaruresurser, står inför utmaningen att implementera omfattande säkerhetsåtgärder. Denna begränsning gör enheterna sårbara för attacker (Nguyen et al., 2019). Dessutom förstärks riskbilden av det faktum at t många IoT-enheter ständigt är anslutna till internet, en situation som innebär en ökad exponering för potentiella hot. En IoT-enhet kan användas som en attackvektor för att exempelvis utöka rättigheter och angripa andra enheter inom samma nätverk (Markiewicz & Sgandurra, 2020). SonicWall (2022) rapporterar att första halvan av 2022 inleddes tufft, med totalt 57 miljoner attacker med skadlig kod mot IoT, en ökning på 77% jämfört med samma period året innan. Genom att upptäcka skadlig kod och förhindra dess förmåga att infektera en eller andra IoT-enheter, kommer hotaktörernas förmåga att sprida och attackera att hämmas (Martin et al., 2017).

Benkhelifa et al. (2018) undersökte vilka svårigheter som finns med att implementera intrångsdetektion (IDS) inom IoT och kom fram till fyra svårigheter. En variation av detektionsmetoder med varierande effektivitet finns och är ofta begränsade till att detektera specifika attacker. Bristande arbete på universella lösningar för att kunna detektera hot för alla domäner inom IoT. Antalet jämförbara dataset är lågt och detektionstekniker evalueras på fel sätt, exempelvis genom simulation. IoT-nätverk står inför svårigheter med att implementera traditionella intrångsdetektionstekniker, på grund av varierande teknologibehov och odefinierbar karaktär, av exempelvis placering och trafik. I ett smart hem kan det finnas många olika IoT-enheter, exempelvis kameror och lampor, dessa är svåra att karaktärisera då de kan ha olika förutsättningar när det kommer till hårdvara.

Denna studie studerar vilka detektionsmetoder används för att detektera skadlig kod riktat mot IoT-enheter inom smarta hem, samt dess effektivitet och förhållande. Detta uppnåddes genom att undersöka befintliga detektionssystem och detektionstekniker, och hur de presterar, med vilken metod och under vilka förhållanden, exempelvis placering och dataset. Placeringen av ett detektionssystem har en viss betydelse då det påverkar hur och vilken data som samlas in, kompatibilitet och skalbarhet. Studien kan förse användare inom smarta hem med värdefull information som kan underlätta selektion och implementation av ett detektionssystem för deras smarta hem. Utöver att underlätta för användare, förser studien forskare inom området med en sammanställning av befintliga detektionsmetoder och hur de appliceras och presterar för att detektera skadlig kod riktat mot IoT-enheter inom smarta hem.

I Kapitel 2 blir läsaren försedd med bakgrundsinformation om IoT, smarta hem och skadlig

kod, samt tidigare forskning. Kapitel 3 förklarar problemområdet, målet med studien och vilka avgränsningarna som är finns. I Kapitel 4 förklaras varje steg i metodologin och de riktlinjer som använts för att samla in artiklar. I Kapitel 5 implementeras metodologin i praktiken och de accepterade artiklarna presenteras. Kapitel 6 presenterar genomförandet av dataanalysen och dess resultat. I Kapitel 7 diskuterades resultatet i relation till tidigare forskning, metod och sociala och etiska aspekter, samt framtida arbete. I Kapitel 8 dras en slutsats baserat på resultatet från Kapitel 6.

2 Bakgrund

Det här kapitlet har som mål att ge läsaren en bättre förståelse för Internet of Things (IoT), generellt och för IoT inom smarta hem. Det kommer också att tas upp ett antal IoT-relaterade risker inom smarta hem. Det sista som kommer att tas upp är tidigare incidenter med skadlig kod och hur detektering i vissa fall hade kunnat förhindra incidenterna.

2.1 Vad är IoT?

Internet of things (IoT) kopplar samman flera miljarder enheter via internet. IoT kan delas upp i tre lager. Det första lagret ansvarar för insamlingen och kan bestå av exempelvis sensorer eller kameror. Det andra lagret, nätverkslagret, består av olika kommunikationskanaler, exempelvis internet och 3G. Nätverkslagret behandlar data från det första lagret och skickar det vidare till det tredje lagret, applikationslagret, som fungerar som gränssnitt mellan IoT och användaren. IoT används inom flera sektorer, exempelvis smarta hem, sjukvård, logistik och tillverkning (Chalasani & Alhamdani, 2021; Jie et al., 2013).

2.1.1 Smarta hem

Ett smart hem är ett helt vanligt hem som med IoT automatiserar funktioner inom hushållet, och ökar därför livskvalitén. Genom att koppla ihop enheter som belysning, säkerhetskameror och datorer, och sedan ansluta dem till internet, kan de centralt kontrolleras, exempelvis genom en telefon, oavsett tid och plats (Malche & Maheshwary, 2017; Hasan et al., 2018). För trådlös kommunikation är WiFi, Bluetooth och Zigbee, teknologier som ofta används inom ett lokalt nätverk. För att kunna nå omvärlden i det här scenariot, behöver en nätverksgateway finnas för att koppla nätverket till internet (Wenbo et al., 2015).

2.1.2 Säkerhetsläget för IoT

Ett växande antal tillverkare gör entré på IoT-marknaden och lanserar produkter i allt högre takt. Säkerhetsdesign nedprioriteras ofta för att påskynda produktutvecklingsprocessen, det kan leda till att slutprodukten innehåller säkerhetssårbarheter som kan utnyttjas av skadlig kod (Nguyen et al., 2019). IoT-enheter är ofta begränsade vad gäller beräkningskapacitet och minne, det innebär att säkerhetsmekanismer, exempelvis kryptering, inte alltid kan implementeras (Mogbil et al., 2020).

2.1.3 Risker relaterade till IoT

IoT erbjuder många möjligheter men det är inte helt riskfritt. Begreppet risk används med olika innebörd men definieras av NIST (2018) i NIST SP 800-37, revidering 2, med följande ”Ett mått på den omfattning till vilken en enhet är hotad av en potentiell omständighet eller händelse, och är typiskt en funktion av: (i) den negativa påverkan, eller omfattningen av skada, som skulle uppstå om omständigheten eller händelsen inträffar; och (ii) sannolikheten för förekomst” (s. 104).

För att förstå vilka risker det finns för IoT inom smarta hem, behövs en grundkunskap om vilka sårbarheter som finns. NIST (2022) har lyft fram flera potentiella sårbarheter för IoT-enheter inom smarta hem, några exempel sammanfattas nedan.

- Obehörig åtkomst till IoT-enheten över internet, kan leda till vidare eskalering och obehöriga exekveringsmöjligheter.
- En okonfigurerad IoT-enhet, utan åtkomstkontroll, gör både IoT-enheten och resterande enheter på nätverket, sårbara. Det här kan leda till obehörig exekvering av

kod.

- Det är ofta som både användare och tillverkare använder ett bristfälligt lösenordsskydd. Det här ökar risken för att IoT-enheten används på ett oönskat sätt eller att den drabbas utav skadlig kod.

MSB (2020) grupperar risker för IoT utifrån deras konsekvenser, som har en negativ påverkan på ett eller fler skyddsvärden, vilka är konfidentialitet, tillgänglighet och riktighet. Nedan beskrivs fyra risker i relation till ett skyddsvärde.

- IoT-enheter kan användas för att stjäla information från IT-system eftersom de ofta är utrustade med mikrofoner och kameror, det gör det möjligt att inhämta information och spionera, och påverkar därför konfidentialiteten negativt.
- IoT-enheter kan tas över för att skapa ett botnät, som sedan kan användas för att utföra exempelvis överbelastningsattacker, och påverkar därför riktigheten negativt
- IoT-enheter kan göras obrukbara som följd av obehörig användning för skadliga syften, exempelvis för att skapa ett botnät. Eftersom IoT-enheten görs obrukbar, påverkar det tillgängligheten negativt.

2.1.4 Begrepp

För att få en grundlig förståelse för vad skadlig kod är, och relaterade begrepp, kommer några begrepp att förklaras nedan.

- Skadlig kod (Malware)

  Skadlig kod, ofta kallad malware, är en generell term som används för att beskriva olika typer av skadlig eller oönskad programvara, som kan skada eller manipulera en enhet (Sentor, 2021).

- Botnät

  Ett botnät består av en grupp med internetanslutna enheter som infekterats med skadlig kod, för att under kontroll av en extern aktör, utföra koordinerade aktiviteter utan ägarens vetskap (Malwarebytes, u.å).

2.1.5 Tidigare säkerhetsincidenter med skadlig kod inom smarta hem

Det har sedan starten av IoT skett flera säkerhetsincidenter där skadlig kod har använts för att göra IoT-enheter obrukbara eller använda dem för andra skadliga syften, exempelvis botnät. Nedan kommer tre exempel på säkerhetsincidenter där IoT inom smarta hem var inblandat.

- **Mirai**

  Mirai är en skadlig kod som riktar in sig på att infektera IoT-enheter för att göra dem till bots eller ”zombies” som det också kallas. En bot eller ”zombie” kan kontrolleras på distans och används ihop med andra bottar eller ”zombies” för att skapa ett botnät, som ofta används för att uträtta överbelastningsattacker (Cloudflare, u.å).

  Cloudflare skriver att Mirai riktade sig specifikt mot IoT-enheter som använder en ARC processor vilket kör en avskalad version av Linux. Mirai skannade igenom internet efter dess IoT-enheter och om inte standard användarnamnet och lösenordet hade blivit ändrat kunde Mirai infektera IoT-enheten. I oktober 2016 lyckades Mirai med hjälp av hundratusen kapade enheter, störa tjänsten hos Dyn, en domänregistrator.

I nya varianter av botnätet används sårbarheter som CVE-2020-10173 och CVE-2020-10987 för att få kontroll över routrar som sedan blir en del av ett botnät (CIS, u.å).

Den nyare varianten går att detektera med signaturbaserad detektion då det kan användas för att upptäcka sårbarheter direkt i nätverkstrafiken. Båda de gamla, och nyare varianterna går att upptäcka med reputation-baserad detektion, genom att analysera vart nätverkstrafik går. Om trafik går till och från IP-adresser eller domäner som är kända för att de är ”dåliga” (Sentor, 2021).

- **Torii**

  Torii fick sitt namn när den upptäcktes attackera en honey pot tillhörande VessOnSecurity som tweetade om det. Attackerna kom ifrån Tor exit noder, som är därför det döptes till ”Torii”. Efter Mirai kom det flera nya IoT botnet, men de skilde sig knappt från Mirai då de använde samma kodbas. Torii sticker ut då den är mer tystlåten av sig, och skapar sig ett bättre fotfäste på den infekterade enheten (Avast, 2018).

  Avast skriver att Torii har stöd för att infektera flera olika typer av arkitekturer som x64, x86, ARM, MIPS, SuperH, PowerPC med flera. Den kommer även med många funktioner för att kunna hämta och köra kommandon och program, samt exfiltrera information. Allt det här sker igenom krypterad kommunikation.

  För att detektera Torii, skulle ett network intension detection system (NIDS) fungera, eftersom det används för att övervaka inkommande och utgående trafik. Det kan även detektera skadlig och suspekt trafik (Fortinet, u.å). Det gör att en attack via Telnet kan detekteras.

- **BrickerBot**

  BrickerBot använder sig av ordlistor med standardlösenord för att få tillgång till IoT-enheter över Telnet. BrickerBot finns i två kända versioner varav den första riktar in sig på IoT-enheter som kör BusyBox, och använder SSH för att få tillgång till IoT-enheten, genom att använda en ordlista med standardlösenord. Den andra versionen använder sig utav Tor exit noder för att dölja sin aktivitet (Trend Micro, 2017). De skriver även att de har sett indikationer på att BrickerBot också använder sig utav remote code execution (RCE) sårbarheter hos routrar. En RCE sårbarhet, är när en angripare kan exekvera skadlig kod på offrets enhet (Cloudflare, u.å).

  BrickerBots mål är att göra permanent skada genom att använda en rad olika Linux kommandon som kan användas till skadliga ändamål. Det som kommandona gör är att misskonfigurera eller korruptera kernel parametrar och enhetens lagring. Det här kan göra enheten obrukbar då det kan hindra den från att ansluta till internet och manipulera enhetens prestanda, samt ta bort alla filer på enheten.

  Eftersom BrickBot angriper IoT-enheter och routrar via Telnet och med hjälp av sårbarheter, kan signaturbaserad detektion och network intrusion detection system (NIDS) vara bra. Signaturbaserad detektion kan upptäcka sårbarheter direkt i nätverkstrafiken (Sentor, 2021). NIDS kan användas för att detektera skadlig och suspekt trafik genom att övervaka inkommande och utgående trafik (Fortinet, u.å).

2.2 Framfarten av IoT malware

År 2025 förväntas det finnas cirka 27 miljarder uppkopplade IoT-enheter (IoT Analytics, 2022). Det innebär att i takt med att antalet IoT-enheter ökar, kommer attackytan att expandera (SonicWall, 2020). Enligt en IoT-rapport från Kaspersky Lab (2018) upptäcktes över 120 000 unika typer av skadlig kod under första halvåret 2018, det representerar en ökning på över 200% jämfört med hela föregående året. Enligt Ponemon Institute (som citeras i Li & Zhao, 2022) var tidigare okänd skadlig kod inblandad i 76% av alla lyckade attacker under 2018. IBM X-Force (2022) har identifierat en betydande ökning av aktivitet från skadlig kod inom IoT. Mellan tredje kvartalet 2019 och fjärde kvartalet 2020 observerades en ökning på nästan 3 000%. SonicWall (2022) rapporterar att första halvan av 2022 inleddes tufft, med totalt 57 miljoner attacker med skadlig kod mot IoT, en ökning på 77% jämfört med samma period året innan. Redan under januari månad hade antalet attacker med skadlig kod mot IoT fördubblats jämfört med föregående månad.

2.3 Tidigare arbete

Det finns några tidigare arbeten som identifierar vilka metoder som finns för att detektera skadlig kod inom IoT. Tidigare arbete som involverar att göra en sammanställning av metoder för att detektera skadlig kod är limiterat och förekommer knappt inom specifikt smarta hem. Majoriteten av det tidigare arbetet är taxanomier och en av studierna är en systematisk litteraturstudie. I den här delen kommer tre tidigare studier att sammanfattas.

Madan et al. (2022) presenterade i sin artikel flera detektionsmetoder som kan användas för att detektera Linux-baserad skadlig kod inom IoT. Madan et al. (2022) delar in metoder för analys av skadlig kod i tre olika kategorier, statisk analys, dynamisk analys och analys av minne, analysmetoderna kan användas för att detektera skadlig kod. Statisk analys innefattar exempelvis extrahering av egenskaper från filer och koder. Den här typen av metod behöver inte köra koden för att avgöra om den är skadlig eller ej. Den statiska analysen utförs genom att dekompilera den skadliga binären som i Linux är av typen ELF. De egenskaper som kan extraheras från en binär av typen ELF är exempelvis filomvandling, obfuskering, instruktionssekvenser och filsignaturer. Det som klassas som dynamisk analys omfattar exempelvis sandlådemiljöer och beteendebaserade egenskaper. De egenskaper som kan extraheras genom dynamisk analys är exempelvis processers beteende, information om filsystemet och nätverksaktivitet. Den sista kategorin, analys av minne är exempelvis egenskaper för exekvering av processer.

Auliar & Bekaroo (2021) studerade vilka detektionsmetoder som finns för att detektera Mirai inom smarta hem, genom att utföra en taxonomi. Två typer av metoder identifierades, anomalibaserad detektering och signaturbaserad detektering. De anomalibaserade detektionsmetoderna omfattade nätverksbaserad detektion, DNS baserad detektion och enhetsbaserad detektion. Nätverksbaserad detektion fokuserar på att analysera nätverkstrafik. DNS baserad detektion letar efter anomalier i DNS trafik som genereras av botnät. Inkommande DNS trafik filtreras med vit- och svartlistor och därefter analyseras trafikens egenskaper för att avgöra om den är associerad med ett botnät. Enhetsbaserad detektion omfattar detektering som sker direkt på en enhet i stället för att analysera hela nätverket. Den här typen av detektion kan exempelvis involvera blockkedjor och användning av verktyg för att söka efter sårbarheter. Signaturbaserade detektionsmetoder omfattar signaturbaserad IDS och användning av signaturbaserade binärer som baseras på skadlig kod. Signaturbaserad IDS använder regler för att detektera attacker i nätverkstrafik och signaturbaserade binärer används exempelvis av antivirusprogrammet ClamAV för att detektera skadlig kod.

Wazid et al. (2019) utförde en taxonomi med fokus på detektion, tillgångshantering och nyckelhantering inom IoT/IoMT, samt en jämförande studie med fokus att jämföra olika detektionssystem för skadlig kod inom IoT. Wazid et al. (2019) presenterade tre detektionsmetoder, anomalibaserad detektion, signaturbaserad detektering och specifikationsbaserad detektering. Den första metoden använder statistiska metoder för att detektera avvikelser i exempelvis nätverkstrafik genom att studera hur trafikflödet ser ut under normala och onormala omständigheter. Metoden presterarar bra vid detektion av okända attacker och andelen felaktiga klassifikationer är minimal. Den andra metoden, signaturbaserad detektion använder signaturer av kända attacker, exempelvis skadlig kod, för att detektera samma typ av attack. Den sista metoden som Wazid et al. (2019) presenterar är specifikationsbaserad detektion som använder de två tidigare metoderna tillsammans. Metoden kan upptäcka okända hot men kräver specificeringar och avgränsningar som tar mycket tid att skapa.

3 Problembeskrivning

Syftet med det här kapitlet är att ge läsaren en förståelse om problemområdet som studien berör. Det kommer att ges en förklaring på varför detektering inom smarta hem behövs. De följande kapitlen kommer att beskriva de mål och syften som finns, samt avgränsningar.

Som tidigare diskuterat i kapitlet 2.2 kommer det finnas ungefär 27 miljarder uppkopplade IoT-enheter runt om i världen (IoT Analytics, 2022). Parallellt med att antalet IoT-enheter ökar kommer även attackytan att öka (SonicWall, 2020). IoT-enheter är lättare att manipulera då de är åtkomstbara fysiskt. I och med det stora antalet IoT-enheter ökar risken för otillåten informationsinhämtning alternativt infektering med skadlig kod eller otillåten användning. Det här blir ännu lättare för angripare att utföra när tillverkare och användare använder lösenord som är svaga alternativt standardiserade (MSB, 2020).

För att detektera skadlig kod generellt finns det fyra grupper av metoder som används, signaturbaserad detektion, beteendebaserad detektion, heuristikbaserad detektion och molnbaserad detektion (Zeltser, 2016). Traditionella metoder som används för detektering av skadlig kod kan inte alltid användas inom IoT. Det har att göra med att vissa metoder inte är kostnadseffektiva nog för att användas inom IoT, exempelvis operationskodsbaserad analys (Wan et al., 2020).

Trots en stor variation av detektionsmetoder bland detektionssystemen finns det ingen sammanställning av vilka metoder som används inom smarta hem för att skydda IoT-enheter mot skadlig kod. Med många alternativ blir det svårt att avgöra vad som passar i en miljö utan att ha en sammanställning på de metoder som finns och i vilka sammanhang de tillämpas. Målet med studien är att skapa en sammanställning av de detektionsmetoder som finns för att detektera skadlig kod, och presentera hur metoderna tillämpas och presterar inom ett smart hem.

3.1 Mål och syfte

Målet studien är att ta reda på vilka detektionsmetoder det finns för att detektera skadlig kod riktat mot IoT-enheter inom smarta hem. Och sedan göra en sammanställning av de metoder som har hittats och vilka fördelar samt eventuella nackdelar respektive metod har. Studien har även som mål att ta reda på hur bra respektive metod fungerar och hur stort dataset som har använts, om det finns statistik tillgängligt.

För att mer tydligt visa vilka mål det finns med studien, finns följande punktlista:

- Metod och typ

  Målet är att beskriva vilken metod det är som används, exempelvis för inlärning eller analys och vilken typ metoden är, exempelvis signaturbaserad detektion eller anomalibaserad detektion.

- Statistik och dataset

  Målet är att ta reda på hur bra en metod fungerar genom att undersöka hur mycket en modell lyckats detektera, om statistik finns tillgängligt. Om det finns statistik är det intressant att veta hur stort dataset som har använts och vilken fördelning det är, om det finns tillgängligt.

Syftet med studien är att samla in information om existerande metoder och därefter göra en analys av informationen, som kommer att inhämtas från existerande forskning. Det här ska ge en bättre förståelse för vilka detektionsmetoder som finns för att detektera skalig kod riktat mot IoT-enheter inom smarta hem, och hur de fungerar. Det här kan hjälpa till för att förstå hur en IoT-enhet i ett smart hem ska skyddas, det är en fördel att veta vilka detektionsmöjligheter en IoT-enhet kan ha och hur externt skydd bör appliceras för skydda mot eventuella brister hos IoT-enheten. Den här studien ska kunna användas som en grund för framtida forskning inom det här forskningsområdet, exempelvis för att ge en överblick om vilka metoder som redan finns och hur bra respektive metod fungerar, om statistik finns.

Forskningsfrågan som binder målet och syftet samman är:

Vilka metoder finns det för att detektera skadlig kod riktat mot IoT-enheter inom smarta hem?

3.2 Avgränsningar

Denna studie kommer endast att fokusera på vilka metoder det finns för att detektera skadlig kod riktat mot IoT-enheter inom smarta hem. Det inkluderar olika analysmetoder och inlärningsmetoder. Studien förklarar hur respektive metod fungerar, men går inte in på exakta detaljer för hur ett detektionssystem är uppbyggd. Studien är inriktad på smarta hem och kommer endast inkludera metoder som går att härleda till smarta hem.

4 Metod

Målet i det här kapitlet är att ge en tydlig förståelse om den metodologi som använts för att genomföra den här studien. Det kommer att ges en motivering till varför just den här metodologin har använts.

Som tidigare nämnts i kapitel 3.1 är målet med studien att identifiera vilka metoder som finns för att detektera skadlig kod för IoT-enheter inom smarta hem och därefter göra en sammanställning av de metoder som har identifierats. Målet är även att identifiera fördelar och nackdelar med respektive metod och undersöka hur bra en metod fungerar, om det finns statistik tillgängligt. Om statistik finns, är det intressant att veta hur stort dataset som har använts vid testning. För att besvara forskningsfrågan, kommer en systematisk litteraturstudie att genomföras, eftersom det anses vara det mest lämpliga för det här arbetet. Befintliga vetenskapliga artiklar som är publicerade mellan 2013–2023 bör ge tillräckligt med data för att besvara forskningsfrågan. Det här forskningsområdet består av många experiment, och därför hade en fallstudie varit en alternativ metod för att besvara forskningsfrågan. Med en fallstudie skulle det vara möjligt att besvara forskningsfrågan och göra en jämförelse av de olika metoderna. En fallstudie skulle däremot bli svår att genomföra, då det är mycket som behöver återskapas från tidigare experiment. Det skulle resultera i mycket arbete och varje metod behöver testas under samma omständigheter för att ge ett korrekt resultat, det innebär exempelvis samma hårdvara och samma dataset. Genom att endast återanvända tidigare forskning kommer inte dessa svårigheter inte uppstå och därför är en systematisk litteraturstudie det bästa valet för den här forskningsfrågan.

4.1 Systematisk litteraturstudie

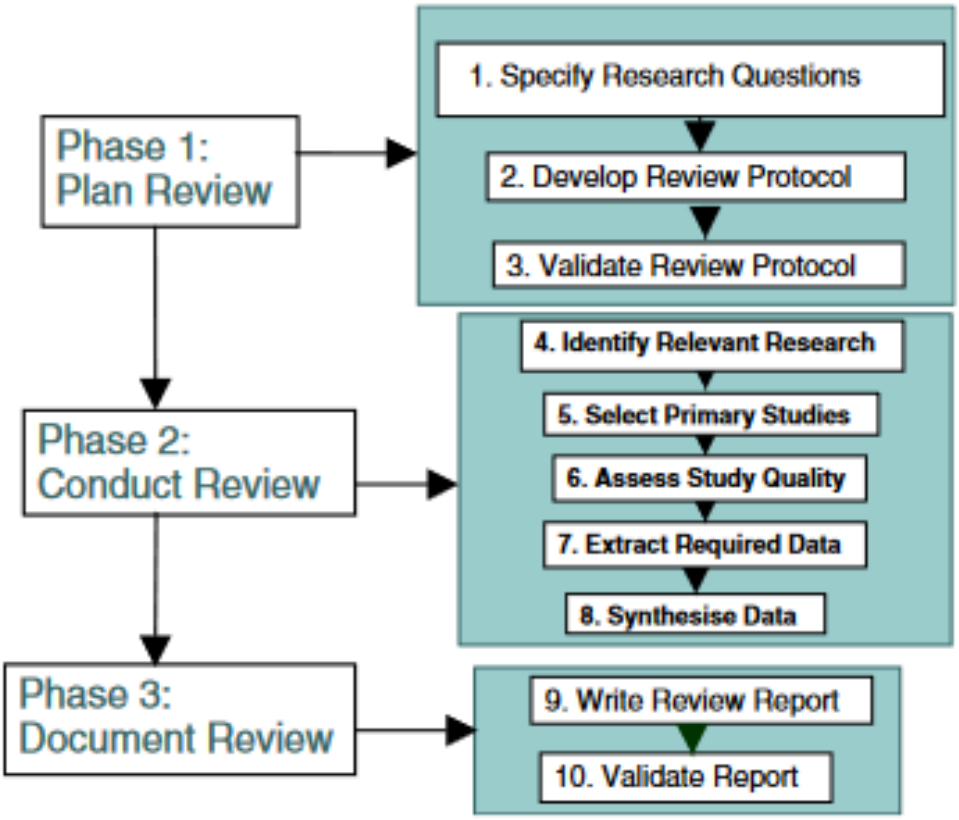
Det finns enligt Kitchenham & Charters (2007) flera anledningar till att välja en systematisk litteraturstudie, tre vanliga anledningar är följande:

- Kan användas för att summera existerande data.
- Kan användas för att hitta luckor i befintlig forskning och för att kunna föreslå områden för vidare undersökning.
- Kan användas för att erbjuda ett ramverk för ny forskning.

Kitchenham & Charters (2007) beskriver en systematisk litteraturstudie med följande:

> "En systematisk litteraturstudie (ofta kallad en systematisk översikt) är ett sätt att identifiera, utvärdera och tolka all tillgänglig forskning som är relevant för en specifik forskningsfråga, ämnesområde eller fenomen av intresse. Enskilda studier som bidrar till en systematisk översikt kallas primärstudier; en systematisk översikt är en form av sekundärstudie" (s. 3).

Den metod som kommer att användas i den här studien baseras på den process som Brereton et al. (2007) har tagit fram, vilket kan ses på Figur 1. Processen är baserad på de riktlinjer Kitchenham (2004) har skapat, och är uppdelad i tre faser: planering av litteraturstudie, genomförande av litteraturstudie och utvärdering av litteraturstudie.



Figur 1: Den process för en systematisk litteraturstudie som skapats av Brereton et al. (2007) vilket i sin tur är baserad på de riktlinjer som skapats av Kitchenham (2004).

I Figur 1 visas den process som är skapad av Brerton et al. (2007) för att genomföra en strukturerad litteraturstudie. Det första steget i processen att specificera en forskningsfråga. Enligt (Kitchenham & Charters, 2007) är forskningsfrågan den viktigaste delen i en systematisk översikt. Detta eftersom hela den systematiska översiktens metodik är beroende av forskningsfrågan. Nästa steg är att skapa ett protokoll för studien. Genom att definiera i protokollet hur den systematiska översikten ska utföras för att uppnå målet med minimal partiskhet. I protokollet finns en detaljerad plan för den systematiska översikten och specificerar vilken process som ska följas, även eventuella villkor som ska tillämpas vid exempelvis urval av primärstudier, det är även rekommenderat att själva protokollet blir granskat och validerat (Brerton et al., 2007). I den andra fasen, genomförande av litteraturstudie, är det första steget enligt (Kitchenham & Charters, 2007) att med en opartisk strategi hitta ett stort antal primärstudier som kan relateras till forskningsfrågan. De inhämtade primärstudierna behöver sedan analyseras för att försäkra att dem är relevanta med forskningsfrågan. Det bör även göras en bedömning av kvalitén på primärstudierna, utöver inkludering- och uteslutningskriterierna, för att exempelvis skapa mer detaljerade inkludering- och uteslutningskriterier och för att kunna väga betydelsen av en enskild primärstudie när resultatet ska sammanställas. Med det här kommer ett bibliotek att skapas som data därefter kan extraheras och sammanfattas ifrån för att kunna besvara forskningsfrågan. Det sista steget av processen är att dokumentera alla resultat och utifrån resultaten dra en slutsats (Kitchenham & Charters, 2007).

4.2 Databaser

För att samla in vetenskapliga artiklar kommer två databaser att användas. Den första databasen är IEEE Xplore vilket är en databas som innehåller bland annat vetenskapliga artiklar och konferenspublikationer inom datavetenskap, elektroteknik, informationsteknologi, teknik och elektronik. Databasen erbjuder avancerade sökfunktioner som användning av jokertecken, som gör att exempelvis ”*” kan användas för att matcha allt, ett exempel på hur * är ”smart home*” vilket matchar exempelvis ”smart home” och ”smart homes”. Även användning av booleska operatorer stödjs av databasen, vilket gör att ”AND”, ”OR” och ”NOT” kan användas. Användningen av jokertecken och booleska operatorer underlättar sökningen efter artiklar, eftersom det gör det möjligt att göra väldigt specifika sökningar som kan matcha ord i både singular och plural. Den andra databasen är ACM Digital Library som innehåller bland annat tidskrifter och konferenspublikationer inom datavetenskap. Även den här databasen erbjuder avancerade sökfunktioner som användning av jokertecken och booleska operatorer. Anledningen till att dessa databaser har valts är för att de är väldigt relevanta inom IT och de är tillgängliga genom Högskolan i Skövdes bibliotek.

4.3 Söktermer

För att hitta artiklar och konferenspublikationer som är relevanta med forskningsfrågan, skapades en söksträng som inkluderar både jokertecken och booleska operatorer. Söksträngen är på engelska eftersom de flesta artiklar i IEEE Xplore och ACM Digital Library är på engelska. Ett tidsspann på tio år applicerades på sökningen för att filtrera bort artiklar som är äldre än tio år. För båda databaserna används samma söktermer men söksträngarna skiljer sig åt. Söksträngen för IEEE Xplore innehåller inga krav på vart ett sökord behöver finnas, vilket söksträngen för ACM Digital Library har då det annars kommer upp flera resultat som inte har med detektering av skadlig kod att göra, men då sökordet finns i texten blir det ändå en träff. Därför måste ”detection” och ”malware” eller ”IoT-malware” förekomma i abstraktet för artikeln. Nedan är de söksträngar som har använts, den första söksträngen användes för IEEE Xplore och den andra för ACM Digital Library.

- detection AND smart home* AND (malware OR IoT-malware)
- [All: "smart home*"] AND [Abstract: detection] AND [[Abstract: malware] OR [Abstract: iot-malware]] AND [E-Publication Date: (04/01/2013 TO 04/30/2023)]

Söksträngen för ACM Digital Library inkluderar en period för när artikeln ska publicerad. För IEEE Xplorer appliceras inte en period i söksträngen, det ställs in manuellt efter sökningen. Söksträngen måste vara i ett annat format för att återskapas, utan att manuellt återskapa söksträngen med ACMs sökverktyg. För att få samma resultat utan datum kan AllField:("smart home*") AND Abstract:(detection) AND Abstract:(malware OR IoT-malware) användas.

4.4 Inkludering- och uteslutningskriterier

Genom att använda inkludering- och uteslutningskriterier, går det att filtrera bort de artiklar från föregående kapitel som inte är relevanta för studien genom att applicera kriterier. I Tabell 1 listas inkludering- och uteslutningskriterierna upp i respektive kolumn och under tabellen kommer alla kriterier att förklaras.

Inkluderingskriterier	Uteslutningskriterier
• Innehåller forskning som är relevant med forskningsfrågan • Skriven på svenska eller engelska. • Publicerad mellan 2013 och 2023. • Tillgänglig genom Högskolan i Skövdes bibliotek. • Vetenskaplig artikel som är publicerad i en journal eller konferens. • Genomgått kollegial granskning.	• Uppfyller inte inkluderingskriterier. • Kräver betalning för åtkomst. • Dubblett av redan inhämtad artikel.

Tabell 1: Inkludering- och uteslutningskriterier.

Eftersom målet är att göra en sammanställning av de metoder som finns för att detektera skadlig kod för IoT-enheter inom smarta hem, behöver artiklarna handla om detektering av skadlig kod för IoT-enheter och metoden eller metoderna som föreslås ska gå att applicera inom smarta hem. Eftersom teknik konstant utvecklas och IoT är relativt nytt, kan äldre forskning vara opålitlig och därför används en tidsperiod på tio år. Artikeln måste vara skriven på svenska eller engelska för att författaren ska kunna tolka den. Artikeln måste ha genomgått en kollegial granskning och ska vara publicerad i en journal eller konferens, och den måste vara tillgänglig genom Högskolan i Skövdes bibliotek. För att en artikel ska vara relevant för studien måste alla inkluderingskriterier uppfyllas, det ska inte krävas en betalning för att få åtkomst till artikeln och det ska inte vara en dubblett av en artikel som redan är inhämtad.

4.5 Metod för dataanalys

När ett flertal artiklar har genomgått inkludering- och uteslutningskriterierna och blivit godkända, behöver de analyseras. Den insamlade datan är av kvalitativ karaktär, och därför kommer tematisk kodning att användas som analysmetod. Enligt Braun & Clarke (2006) kan teman i data identifieras på två sätt: genom induktion och deduktion. Med ett induktivt tillvägagångssätt kodas data utan att använda fördefinierade kategorier. De teman som identifierats är direkt kopplade till datan, det gör att denna form av tematisk kodning är datadriven. Ett deduktivt tillvägagångssätt är mer analytikerdrivet, eftersom det i större utsträckning drivs av forskarens teoretiska eller analytiska intresse för området. I denna studie kommer det induktiva tillvägagångssättet att användas, eftersom det inte finns några fördefinierade kategorier.

Enligt Braun & Clarke (2006) används koder för att identifiera data av intresse. Koderna sorteras sedan in i teman som delas upp i huvudteman och subteman. Vissa koder kan eventuellt avfärdas.

4.6 Validitetshot

För att en litteraturstudie ska vara pålitlig, krävs det att den är spårbar. Processen för den systematiska litteraturstudien måste vara väl dokumenterad, för att andra forskare ska kunna följa samma steg och komma fram till samma slutsats (Xiao & Watson, 2019). Partiskhet, som ofta uppstår oavsiktligt, kan påverka sammanställningen av data, exempelvis genom förutfattade meningar och åsikter. Selektionspartiskhet kan uppstå när data som valts ut på ett partiskt sätt inkluderas, exempelvis genom att göra en sammanställning baserad på ett avsiktligt urval av artiklar. Det kan även hända genom inkludering av databaser som inte representerar forskningsområdet på ett adekvat sätt. Sökströngarnas roll är också betydande En otillräcklig söksträng, som antingen saknar tillräckligt med söktermer eller är för restriktiv, kan medföra att relevant och gynnsam forskning missas (Haddaway et al., 2015).

Haddaway et al. (2015) föreslår att noggrant utformade söksträngar används och att samma kriterier tillämpas för att bestämma relevansen av alla sökresultat. Sökningarna bör dessutom utföras i flera databaser.

I denna studie har varje fas i genomförandeprocessen noggrant dokumenterats, exempelvis hur artiklar blev accepterade och hur varje process går till. Det har gjorts för att säkerställa fullständig spårbarhet och minimera eventuella negativa effekter på studiens validitet. Inkludering- och uteslutningskriterier har tillämpats konsekvent och korrekt på relevanta artiklar från två noggrant utvalda databaser, som valdes på grund av deras höga teknologiska relevans. Med flera databaser minskar risken att relevanta artiklar förbises. Trots noggrann selektion av flera databaser, kan en avvikelse från metodologin hota studiens validitet.

5 Implementation av metodologi

I det här kapitlet beskrivs implementeringen av den metodologi som diskuterades i föregående kapitel. Det här kapitlet markerar starten på den andra fasen, genomförandet av litteraturstudien, i processen som definierats av Brerton et al. (2007). Baserat på de databaser och söksträngar som valdes ut i föregående kapitel har vetenskapliga artiklar samlats in. Dessa har sedan genomgått en urvalsprocess baserat på de inkludering- och uteslutningskriterierna som även de specificerades i föregående kapitel. De artiklar som har klarat urvalsprocessen bedöms vara relevanta för forskningsfrågan och har därför blivit accepterade. Först lästes abstraktet för alla resultat i databassökningarna, de som var relevanta lästes igenom helt för att bedöma om de är relevanta och ska inkluderas i studien.

5.1 Resultat av artikelinsamling

I Tabell 2 nedan presenteras information och resultat av sökningarna i respektive databas. Informationen i tabellen innefattar vilken söksträng som har använts, vilket tidsspann som har applicerats i kombination med söksträngen, när sökningen har utförts, hur många resultat som sökningen har gett och slutligen hur många resultat som efter urvalsprocessen har blivit accepterade. I den nedersta raden visas det totala antalet accepterade artiklar.

Databas	Hämtad	Resultat	Accepterade	Tidsspann	Söksträng
IEEE Xplore	2023-04-13	33	6	10 år	detection AND smart home* AND (malware OR IoT-malware)
ACM Digital Library	2023-04-30	148	6	10 år	AllField:("smart home*") AND Abstract:(detection) AND Abstract:(malware OR IoT-malware)
Totalt antal accepterade artiklar: 12					

Tabell 2: Antalet resultat från respektive databas och antalet accepterade artiklar efter urvalsprocessen.

6 Resultat

I det här kapitlet kommer dataanalysprocessen från Kapitel 4.5 att genomföras på de 12 artiklar som har genomgått urvalsprocessen och blivit accepterade. De 12 artiklar som blivit accepterade under urvalsprocessen i det förra kapitlet, finns att se i Tabell 3. Artiklarna som klarade sig igenom urvalsprocessen är i snitt relativt nya vilket indikerar på att forskningsområdet utvecklas. Av de accepterade artiklarna är majoriteten publicerade 2020 eller senare och den tidigast publicerade artikeln är från 2017. De accepterade artiklarna kommer att analyseras och tematisk kodning från Kapitel 4.5 kommer att användas för att skapa teman för metodtyp och placering av detekteringssystemet, vilket används för att kategorisera artiklarna. När alla artiklar har kategoriserats kommer subkategorier att skapas. De identifierade kategorierna presenteras i delkapitel 6.1. Om statistik gällande detektion av skadlig kod, resursanvändning och information om dataset finns tillgängligt, kommer det att presenteras i samband med resultatet från kodningsprocessen. Statistik och information om dataset kommer bara presenteras om det finns tillgängligt.

ID	Artikel	Författare
A01	Smart Home Security Analysis System Based on The Internet of Things	Yu et al., 2021
A02	Detection and Analysis of P2P Malware Detection in IoT Smart Home Applications	Chalasani & Alhamdani, 2021
A03	Security in IoT-based Smart Homes: A Taxonomy Study of Detection Methods of Mirai Malware and Countermeasures	Auliar & Bekaroo, 2021
A04	Anomalous Communications Detection in IoT Networks Using Sparse Autoencoders	Shahid et al., 2019
A05	DÏoT: A Federated Self-learning Anomaly Detection System for IoT	Nguyen et al., 2019
A06	TRIS: A Three-Rings IoT Sentinel to Protect Against Cyber-Threats	Pelaez et al., 2018
A07	Clust-IT: Clustering-Based Intrusion Detection in IoT Environments	Markiewicz & Sgandurra, 2020
A08	Fending off IoT-Hunting Attacks at Home Networks	Martin et al., 2017
A09	ThingNet: A Lightweight Real-Time Mirai IoT Variants Hunter through CPU Power Fingerprinting	Li & Zhao, 2022
A10	Leveraging Side-Channel Information for Disassembly and Security	Park et al., 2019
A11	Sentinel: A Robust Intrusion Detection System for IoT Networks Using Kernel-Level System Information	Cosson et al., 2021
A12	BOND: Efficient and Frugal DL Model Co-Design for Botnet Detection on IoT Gateways	Gandhi et al., 2021

Tabell 3: Accepterade artiklar.

6.1 Dataanalys

Under dataanalysprocessen av de accepterade artiklarna identifierades två teman. Det första temat baserar sig på vilken metod som används för att detektera skadlig kod för IoT inom smarta hem. De identifierade detektionsmetoderna har delats upp i fyra olika kategorier som presenteras med en beskrivning i Tabell 4. Kategorin *kombination av metoder* betyder att en modell för att detektera skadlig kod använder minst två av de detektionsmetoder som redan är kategoriserade.

Metod	Beskrivning	Artiklar
Anomalibaserad detektering	Identifierar avvikelser i beteende.	A03, A04, A05, A07, A09, A10, A11, A12
Signaturbaserad detektering	Identifierar fördefinierade signaturer.	A03
Statistisk analys	Statistiska metoder används för att analysera data.	A02
Kombination av metoder	En kombination av en eller fler detektionsmetoder.	A01, A06, A08

Tabell 4: Kategorier tillhörande temat metod.

Det andra temat baserar sig på placeringen av detektionssystemet. De identifierade placeringarna har delats upp i två kategorier som presenteras med en beskrivning i Tabell 5. Detektionssystemet kan antingen vara placerat direkt på enheten eller någonstans inom det lokala nätverket. En nätverksbaserad placering kan inkludera användning av molnmiljöer, exempelvis för att matcha signaturer.

Placering	Beskrivning	Artiklar
Enhetsbaserad	Skyddet är enhetsbaserat, placerat på enheten.	A09, A10
Nätverksbaserad	Skyddet är nätverksbaserat, placerat i nätverket.	A01, A02, A03, A04, A05, A06, A07, A08, A11, A12

Tabell 5: Kategorier tillhörande temat placering.

6.1.1 Anomalibaserad detektion

Anomalibaserad detektion används för att identifiera avvikelser från en tidigare skapad baslinje för normalt beteende. Bland de olika detektionsmetoderna som identifierats vid analys av artiklarna är anomalibaserad detektering identifierat som den kategori som är allra vanligast för att detektera skadlig kod. Anomalibaserad detektering har styrkan att kunna identifiera hot som är okända, vilket ger metoden möjligheten att upptäcka nya former av skadlig kod.

Detektering av skadlig kod genom anomalier i nätverkstrafik används av A03, A04, A05, A07 och A12. BOND, ett detektionssystem som med en sparsam djupinlärningsmodell kan genom nätverkstrafik detektera skadlig kod som innan detektion varit okänd, presenteras i A12. BOND jämförs främst med Kitsune som är den primära konkurrenten till detektionssystemet. Under evalueringsprocessen testades BOND och jämförelse med Kitsune på 27 dataset, innehållande olika botnät. BOND ut presterade Kitsune med ett F1-värde, 35% högre än Kitsunes F1-värde och BOND hade 5% högre träffsäkerhet på tidigare okänd skadlig

kod än Kitsune. BOND klassificerade trafik som skadlig eller normal två gånger snabbare än andra detektionssystem under evalueringsprocessen och den har en 4% högre träffsäkerhet på tidigare okänd skadlig kod. Utöver högre träffsäkerhet och bättre prestanda var minnesanvändningen 25% lägre än minnesanvändningen hos konkurrerande detektionssystem.

Med anomalibaserad detektion kan det uppstå många falska alarm i IoT-miljöer, då variationen på IoT-enheter gör det svårt att träna detektionsmetoden på att kunna identifiera avvikelser med hög precision i varje IoT-enhets beteende. I A04, A05, A07 och A12 implementeras inlärningsmetoder för att reducera antalet falska alarm vid användnings av anomalibaserad detektering.

I A04 tränades ett anomalibaserat detektionssystem med hjälp utav SAE som står för sparse autoencoders. Automatisk kodare är ett begrepp för ett artificiellt neuralt nätverks som utan mänskliginteraktion, kan lära sig att kopiera indata till dess utdata under vissa begränsningar som adderas i ett gömt lager. Begränsningarna används för att hjälpa det neurala nätverket att lära sig hur korrekt indata ser ut på ett effektivt sätt. Sparsity står för gleshet och är en begränsning som används i samband med en automatisk kodaren och tillsammans blir de SAE. Gleshet används för att minska antalet aktiva neuroner i det gömda lagret. SAE testades på ett dataset som innehåller legitim trafik från fyra IoT-enheter från ett smart hem, och den skadlig trafiken kom ifrån IoTPOT. Datasetet har totalt 46,796 olika TCP strömmar som innehåller både legitim och skadlig trafik, med en majoritet av skadlig trafik. När det anomalibaserade detektionssystemet testades mot datasetet uppvisade det en träffsäkerhet på mellan 86,9% till 91,2% och mängden falska alarm låg på 0,1% till 0,5%.

I A05 användes ett självlärande detektionssystem som använder en federerad inlärningsalgoritm som går ut på att samla in data lokalt i nätverket för att träna lokala modeller som sätts ihop till en global modell, genom att använda all data som finns tillgänglig ökas träffsäkerheten för den anomalibaserade detektionen. Detektionssystemet som kallas för DïoT testades på ett dataset med 33 IoT-enheter där en skadlig kod kallad Mirai användes mot 5 sårbara IoT-enheter. Detektionsstatistiken delades upp i fem faser, den första fasen är när Mirai inväntar kommando, den andra fasen är före infektion med Mirai, den tredje fasen är infektion med Mirai, den fjärde fasen är när den infekterade IoT-enheten skannar andra enheter och det femte och sista fasen är när den infekterade IoT-enheten utför en DoS attack. Den första fasen har en TPR på 33,33% och en responstid på 4,051,889ms, TPR står för mängden korrekta träffar. Och de fyra resterande faserna har en sammanlagd TPR på 95,60% utan falska alarm med en responstid på 257ms, infektionsfasen med Mirai har en TPR på 93,45%, med en responstid på 272ms.

I A07 presenteras Clust-IT, ett självlärt anomalibaserat detektionssystem som använder algoritmen OPTICS (Ordering Points To Identify the Clustering Structure) för att skapa ordningen i klustren i en klusteranalys, vilket är en metod inom maskininlärning som används för att gruppera data i ett dataset baserat på likheter. OPTICS har fördelen gentemot andra densitetbaserade klusteranalys tekniker, att klusterdensiteten anpassningsbar, beroende på IoT-miljön. Clust-IT testades mot dataseten Kitsune och BOT-IoT, båda dataseten innehåller olika attacker, för BOT-IoT är attackerna inte separerade till olika dataset medan Kitsune har separerat dataseten till Mirai, fuzzing och OS Scan. För Kitsune är endast Mirai relevant då det är direkt kopplat till skadlig kod. Clust-IT detekterade totalt 93,6% av alla attacker i Kitsune: Mirai med 5,3% falska alarm och i BOT-IoT detekterades 91,2% av alla attacker och mängden falska alarm låg på 3,9%.

A09 och A10 har i den här studien klassats som enhetsbaserade, vilket eliminerar faktorn med att lära in flera IoT-enheters beteende. Gemensamt för A09 och A10 är att de använder sidokanaler för att analysera IoT-enhetens beteende. Sidokanaler kan exponera information som strömförbrukning, temperatur och elektromagnetisk strålning, i A09 och A10 ligger fokuset på att upptäcka avvikelser i strömförbrukningen. I A10 används extern övervakning för att läsa av information från sido-kanalerna. Extern monitorering kan appliceras på eller kopplas direkt till IoT-enheten.

I A09 används processorn eller mikrokontrollerna och den inbyggda strömsensorn på IoT-enheten för att läsa av sidokanalerna, vilket används i A09 för att få information om processorns strömförbrukning. I A09 används en djup maskininlärning som kallas ThingNet, för att detektera en infektion med skadlig kod i realtid på en IoT-enhet. ThingNet har fördelen att det fungerar bra i miljöer där störningar förekommer, exempelvis processorer där flera applikationer körs samtidigt. ThingNet testades på en Odroid-XU3 som körde olika applikationer exempelvis ett smart hemsystem av OpenHAB och de skadliga koderna Mirai, Satori, Okiru och Masuta användes, alla tillhörande Mirai familjen och de tre sistnämnda skadliga koderna är okända. ThingNet detekterade under testen skadlig kod med en träffsäkerhet på 99,1% och andelen falska alarm låg på 0,21%. Vid detektion av tidigare okänd skadlig kod hade ThingNet en träffsäkerhet på 96,3%. ThingNet använder ungefär 1,8 kilobyte minne och detekterar skadlig kod med en hastighet på ungefär 3 millisekunder. I A10 presenterades resultat från olika där sido-kanaler har använts för olika syften, exempelvis detektering av skadlig kod. I studien användes en sido-kanals disassembler (SCD) med alla instruktionskoder för två IoT-enheter tillhörande kategorin smarta hem. Författarna av A10 har tidigare gjort ett experiment med den SCD som presenteras i A10 men då låg fokuset på att demontera skadlig kod men det nämns att samma metod har potential att detektera sofistikerade typer av skadlig kod.

Detektering av skadlig kod och insamling av data behöver inte ske på samma enhet. I A11 presenteras Sentinel, en central lösning som samlar in data på kärnnivå från IoT-enheter med en kärnmodul som installeras direkt på IoT-enheten. Sentinel använder olika maskininlärningsmetoder för att detektera anomalier i den insamlade informationen. Sentinel testades på flera olika IoT-plattformar med en simulering av Mirai. Sentinel visade hög träffsäkerhet i testen och fick ett F-värde på över 96% vid användning av maskininlärningsmodellerna beslutsträd och slumpmässig skog.

I A3 nämns en detekteringsmetod som använder Ethereum-blockkedjan. Likt A11 utför inte IoT-enheten detekteringen själv, ett autonomt system används för att sköta kommunikationen mellan IoT-enheten och blockkedjan som innehåller listor på exempelvis enheter infekterade med Mirai och svartlistade IP adresser. Blockkedjan upptäcker själv infekterade enheter med hjälp utav ett tröskelvärde, om antalet paket som skickas till en IoT-enhet överstiger tröskelvärdet kommer IP adressen bli svartlistad. Svartlistan delas sedan ut mellan de autonoma systemen.

En studie gällande detektering av skadlig kod baserat på DNS trafik studeras i A03 och metodens effektivitet valideras med ett experiment där flera olika metoder används, exempelvis klusteranalys. Metoden består utav fem steg varav de två första stegen omfattar inhämtning av inkommande DNS trafik som matchas med listor innehållande vitlistade och svartlistade domäner. Steg tre identifierar om ett botnät förekommer i nätverket genom att analysera DNS trafikens egenskaper. Steg fyra och fem letar reda på och isolerar den infekterade enheten, samt analyserar egenskapsvektorer. Sammanlagt användes 23 dataset med 16 804 prover av DNS trafik från IoT-enheter som inte är infekterade och 15 611 prover

av DNS trafik från infekterade IoT-enheter. Resultatet från experimentet visade en träffsäkerhet på 96,06% till 98,01% och andelen falska alarm låg på 0,015% till 0,31%.

6.1.2 Signaturbaserad detektion

Signaturbaserad detektion är bland de analyserade artiklarna knappt förekommande som en heltäckande lösning för att detektera skadlig kod. Ett detektionssystem som använder signaturbaserad detektion använder signaturer av kända hot för att detektera exempelvis skadlig kod eller trafik. Då signaturer är kopplade till redan kända hot kan inte nya okända hot detekteras. I A3 presenteras två tillvägagångssätt för hur en signaturbaserad detektionslösning med ClamAV och Snort IDS kan implementeras för att detektera den skaliga koden, Mirai. ClamAV använder regler, skapade med verktyget Yara för att identifiera exempelvis specifika strängar, byte-mönster och instruktionssekvenser, inom skadlig kod. Snort kan använda regler för att detektera signaturer i nätverkstrafik. Med signaturbaserade regler inom IDS kan infektionsförsök med skadlig kod förhindras innan de når enheten. Ett exempel på hur ett infektionsförsök med den skadliga koden Mirai kan detekteras är att skapa regler som upptäcker otillåtna sessioner till eller från IoT-enheter över protokollet telnet.

6.1.3 Statistisk analys

Statistisk analys är en ovanlig detektionsmetod bland de analyserade artiklarna och förekommer bara en gång. Statistisk analys är likt anomalibaserad detektering men använder mer statistiska metoder som att studera trender, samband och mönster i data. I A02 testades statistisk analys på TCP-flaggor för att detektera skadlig kod från botnät i nätverkstrafik. Testet utfördes på datasetet CTU-13 som innehåller trafik som skapats av botnät. För att analysera datasetet användes Python och Google Colab platform. De TCP-flaggor som metoden riktar in sig på är TCP-SYN, TCP SYN/ACK, TCP FIN, TCP XMAS och TCP NULL och flaggorna kan indikera på olika attacktyper. TCP SYN och TCP SYN/ACK kan användas för att detektera en DDoS attack, TCP FIN kan användas av angripare för att förbipassera brandväggar genom att ändra flaggans bit till 1, vanliga IoT-enheter ändrar inte flaggans bit till 1. De sista två flaggorna är ovanliga i ett vanligt P2P nätverk och kan indikera på skadlig kod.

6.1.4 Kombination av metoder

Att kombinera flera detektionsmetoder innebär att två eller fler metoder används ihop för att detektera hot som skadlig kod. Bland de analyserade artiklarna använder A01, A06 och A08 flera detektionsmetoder för att detektera hot. A06 och A08 använder signaturbaserad detektion och anomalibaserad detektion och A01 använder statistisk analys och signaturbaserad detektion.

Detektionssystemet som presenteras i A01 använder fyra olika moduler. De första tre modulerna, trafikinhämtning, trafikanalys och systemdetektion används för att detektera hot som skadlig kod. Den fjärde modulen används för att varna om detekterade hot. Trafikanalysmodulen analyserar trafiken som samlats in av trafikinhämtningsmodulen med statistisk analys. Trafikanalysmodulen extraherar även signaturer och jämför dem med signaturer i en databas. Systemdetektionsmodulen detekterar skadligkod genom att övervaka processers CPU användning och hur ofta processer startas om.

TRIS, ett detektionssystem som presenteras i A06, består av tre ringar för att detektera och skydda sig emot hot. Varje ring har en dedikerad modul. Den första ringen använder Yara regler för att matcha signaturer i misstänka filer och prover som kommer från de ringar som har hand om monitorering. Den andra ringen använder maskininlärningsmodellen

slumpmässig skog, för att avgöra om det som skickats till ring två är skadligt. Den tredje och sista ringen består av en extern analysator som skickar misstänkta filer och prover till en extern server som VirusTotal, för att se om den externa servern detekterar något misstänkt. Om inget identifieras i de tre ringarna, klassas det som harmlöst. TRIS har flera andra moduler som kan användas vid behov. TRIS testades med ett dataset på 3000 skadliga koder och detekterade 100% av alla skadliga koder. 8% detekterades i den första ringen med Yara, 75% detekterades i den andra ringen med maskininlärning och de sista 17% detekterades med VirusTotal. De skadliga koderna delades in i storlekskategorier för att mäta hastigheten för varje ring. För skadliga koder på 100KB låg ring 1 på under 0,1s, ring 2 under 0,3s och ring 3 på ungefär 0,5s. För 5MB och mer låg ring 1 på lite mer än 0,4s, ring 2 på nästan 1,6s och ring 3 på lite över 1s.

I A08 används djup paketinspektion i kombination med Pot2DPI för att detektera skadlig kod i nätverkstrafik. Pot2DPI körs direkt på routern och består av fyra olika komponenter som tillsammans detekterar och förhindrar skadlig kod. Den första och andra komponenten skapar signaturer för exempelvis skadlig kod. Den tredje och fjärde komponenten filtrerar trafik och försvårar för angripare, exempelvis med djup paketinspektion. Klusteranalys som är en anomalibaserad detektionsmetod, används för att skapa högkvalitativa signaturer, även vid flera parallella attacker. För att evaluera Pot2DPI användes fyra olika dataset med skadlig kod och varje för varje dataset mäts antalet skickade paket, antalet detekterade paket, antalet anslutningar och antalet detekterade anslutningar. Det sammanlagda detektionsresultatet för Pot2DPI är 1 262 detekterade paket av totalt 1 286 paket och 1 170 detekterade anslutningar av totalt 1 175 anslutningar.

6.1.5 Enhetsbaserad

Bland de analyserade artiklarna är det A09 och A10 som använder enhetsbaserade detektionssystem, där både informationsinhämtning och detektion sker. Både A09 och A10 använde sidokanaler för att detektera anomalier i exempelvis energiförbrukning. En enhetsbaserad lösning med en disassembler som i A10, kan detektera sofistikerad skadlig kod genom att analysera strömförbrukning i realtid även om den skadliga koden är väldigt lik harmlös kod. En disassembler kan även detektera skadlig kod genom att analysera OP-koder i maskinkod för aktiva program. Utöver möjligheten att detektera sofistikerad skadlig kod, visar det hårdvarubaserade detektionssystemet i A09 en hög träffsäkerhet på 96,3% när det kommer till att detektera okända versioner av skadlig kod och med en detektionstid på 3ms.

6.1.6 Nätverksbaserad

Majoriteten av de analyserade artiklarna använde ett nätverksbaserat detektionssystem för att detektera skadlig kod. Alla artiklar med en nätverksbaserad lösning inkluderade analys av nätverkstrafik i sitt detektionssystem. Det här ger en nätverksbaserad lösning möjligheten att samla och analysera trafik från alla IoT-enheter i ett lokalt nätverk. Detektionssystemet kan vara nätverksbaserat men samtidigt hämta information lokalt från en IoT-enhet, som i A11 där en kärnmodul som skickar data till det nätverksbaserade detektionssystemet, installeras på IoT-enheten. Användning av lockbeten inom det lokala nätverket förekommer i A03 och A08, med syftet att angriparen ska försöka infektera lockbeten och utifrån infektionen eller infektionsförsöket skapas signaturer för att detektera samma attack bland övriga IoT-enheter. Användningen av en nätverksbaserad lösning motiveras i A05 där de limiterade systemresurserna gör det svårt att utföra detektering direkt på en IoT-enhet.

7 Diskussion

I det här kapitlet kommer resultatet att analyseras i relation med tidigare forskning, planering och implementation av metodologi och sociala och etiska aspekter. Resultatet kommer att diskuteras med fokus på hur det förhåller sig till tidigare forskning, om valet av metodologi haft en påverkan, om det är applicerbart, vilken betydelse det har för detektionssystem inom smarta hem och inverkan på sociala och etiska aspekter.

7.1 Tidigare forskning

Den här studien har skapat en tydligare bild på vilka detektionsmetoder som används inom smarta hem för att detektera skadlig kod. Utöver att identifiera detektionsmetoder har studien presenterat flera detektionssystem och hur systemen använder metoderna för att detektera skadlig kod riktat mot IoT-enheter inom smarta hem. Statistik för träffsäkerhet och effektivitet och det dataset som använts under evalueringen av ett detektionssystem förekommer, har det presenterats. Bland de tidigare studierna var det endast studien av Madan et al. (2022) som riktade in sig primärt mot smarta hem, men hade endast en skadlig kod, Mirai, i fokus. Den här studien fokuserar på smarta hem och alla skadliga koder.

Två av de tidigare studierna, av Wazid et al. (2019) och Auliar & Bekaroo (2021), har kommit fram till liknande resultat när det kommer till vilka metoder som finns. Detn metod som de tidigare studierna inte har identifierat men som den här studien presenterar är statistisk analys. Den tredje studien av Madan et al. (2022) skiljer sig från de andra studierna genom att gruppera metoderna annorlunda, men ger ändå ett liknande resultat. Minnes analys har identifierats som en metod, inte förekommande i de andra två studierna eller den den här studien. Och signaturbaserad analys klassas som statisk analys och anomalibaserad analys klassas som dynamisk analys. Bland de tre tidigare studierna är det bara Wazid et al. (2019) som presenterat en metod som kombinerar två metoder, i det här fallet anomalibaserad detektion och signaturbaserad detektion.

Den här studien inkluderar dataset och statistik, för att visa detektionssystemet hur metoderna presterar och passar i olika sammanhang. Med dataset och statistik blir det lättare att förstå hur bra en metod fungerar i ett sammanhang.

Detektionssystemen grupperas också in i vart det placeras, nätverksbaserat eller enhetsbaserat. Bara i studien av Auilar & Bekaroo (2021) nämns enhetsbaserad och nätverksbaserade detektionssystem och statistik förekommer endast för ett detektionssystem.

7.2 Metod, implementation och resultat

Studien genomfördes med en kvalitativ systematisk litteraturstudie och utgår från de riktlinjer som definierats i Kapitel 4. Den kvalitativa informationen analyserades med tematisk kodning, en metodologi som använder teman för att på ett flexibelt och effektivt sätt kategorisera data. För att försäkra att alla studier håller en hög nivå har endast artiklar publicerade i journaler och konferenser, som är kollegialt granskade, inkluderats. Konsekvensen med att exkludera resten är att det kan finnas detektionssystem som använder andra metoder.

Databaserna IEEE Xplore och ACM Digital Library valdes för att förse studien med högkvalitativa studier. Under urvalsprocessen var det väldigt få artiklar som handlade om detektering av skadlig kod inom just smarta hem och det accepterade antalet blev därför lågt. Med en eller fler databaser hade fler artiklar kunnat inkluderas i urvalsprocessen och

eventuellt lett till fler accepterade artiklar för att stärka resultatet och kanske bidra med en annan detektionsmetod.

Genom att begränsa sökorden ”malware”, ”iot-malware” och ”detection” till abstraktet i ACM Digital Library, kan relevanta artiklar ha uteslutits. Utan att begränsa sökorden resulterade sökningen i ett stort antal artiklar utan relevans, av den orsaken applicerades en begränsning. Användningen av jokertecken hade kunnat öka antalet träffar och kanske resulterat i fler artiklar, exempelvis ”detect*”.

Flera studier specificerade inte området inom IoT som var i fokus och därför kunde det inte bekräftas om de föreslagna detektionssystemen eller metoderna är applicerbara inom smarta hem, av den orsaken blev de exkluderade.

Resultatet från studien kan användas för att skapa en bredare förståelse för hur och när respektive metod bör appliceras, genom att ge exempel på befintliga detektionssystem och hur de använder detektionsmetoderna. Resultatet kan användas för att förbättra befintliga detektionssystem genom att presentera befintliga detektionssystem och hur bra de presterar.

7.3 Sociala och etiska aspekter

Detektionsmetoder kan användas för att upptäcka skadlig kod inom smarta hem och varna användaren om dess existens. Det kan förhindra att skadlig kod, utan användarens vetskap, kan utföra illasinnade handlingar, exempelvis övervakning. Utöver att användaren varnas kan i vissa fall den skadliga koden isoleras, alternativt tas bort, utan användarinteraktion. Genom att detektera skadlig kod kan åtgärder vidtas och användarens integritet kan skyddas. Den här studien kan bidra till att säkra upp smarta hem genom att kartlägga de befintliga detektionsmetoderna för skadlig kod och hur metoderna kan appliceras inom ett smart hem. Kartläggningen kan hjälpa tillverkare och användare att förstå hur respektive detektionsmetod kan användas inom smarta hem och när respektive metod passar bäst. Studien bidrar till mål 16 av UNs hållbarhetsmål genom att bidra till människors digitala säkerhet.

Studien ger en sammanfattning på vilka metoder som finns och hur de används, och kan därför bidra till utvecklingen av nya metoder och detektionssystem, genom att göra information mer lättillgänglig. Studien inkluderar statistik för de detektionssystem där det finns dokumenterat, och med det kan en metods effektivitet i ett specifikt sammanhang utvärderas. Det här bidrar till mål 9 av UNs hållbarhetsmål genom att kartlägga detektionsmetoder för skadlig kod mot IoT-enheter inom smarta hem, som gynnar innovation inom området.

7.4 Framtida arbete

Det här arbetet fokuserar på att sammanställa detektionsmetoder för att skydda IoT-enheter inom smarta hem mot skadlig kod. Vad som inte inkluderas är vilka detektionsmetoder som används inom andra IoT-områden, och hur de används. Genom att sammanställa detektionsmetoderna för skadlig kod inom fler IoT-områden, kan forskning utföras för att identifiera likheter och skillnader mellan hur skadlig kod detekteras inom de olika IoT-områdena. Och det kan sedan studeras genom exempelvis en fallstudie, för att utforska om och hur metoderna fungerar inom andra IoT-områden.

Ett annat forskningsförslag är att utföra en fallstudie för att jämföra hur respektive detektionsmetod presterar i olika sammanhang, exempelvis enhetsbaserad, nätverksbaserad och med olika CPU arkitekturer, och med samma förutsättningar, exempelvis samma dataset och indata.

7.5 Limiteringar

Studiens limiteringar är antalet databaser och att många studier inte specificerar om metoden de använder kan detektera skadlig kod eller ej. Vad gäller databaser, är det antalet som är för lågt, det skulle ha behövts fler databaser för att eventuellt få fler relevanta artiklar. Många artiklar som har verkar vara relevanta har identifierats, men då det inte har framgått om de verkligen detekterar skadlig kod eller endast skadlig nätverkstrafik, har de behövts exkluderas, för att försäkra att resultatet är högkvalitativt. Faktorer som tidsbegränsning och resursbegränsning limiterar studien. Mer tid skulle kunna leda till ett bättre resultat genom att mer arbete utförs. Resurser kan vara begränsade genom att databaser exempelvis kräver betalning för åtkomst.

8 Slutsats

I den här studien har två databaser med fokus på teknologi använts för att samla in artiklar som är relevanta för forskningsfrågan. Målet var att göra en sammanställning på vilka metoder det finns för att detektera skadlig kod, vilka för- och nackdelar respektive metod har, hur bra detektionssystem för respektive metod fungerar, om statistik finns tillgängligt och vilket och hur stort dataset som använts.

Totalt 12 artiklar inkluderades i studien. För att analysera artiklarna användes tematisk kodning med ett induktivt tillvägagångssätt. Totalt identifierades två teman, varav det ena hade fyra kategorier och det andra hade två kategorier. Från varje artikel delades relevanta stycken in i respektive matchande kategori som sedan användes för att få fram ett resultat.

De flesta artiklarna använde anomalibaserad detektering för att detektera skadlig kod och angrepp. Anomalibaserad detektering kan identifiera avvikelser i ett beteende genom att jämföra beteendet med en tidigare skapad baslinje för normalt beteende. En stor fördel med metoden är att den kan upptäcka skadlig kod som tidigare varit okänd, och därför kan metoden upptäcka nya former av skadlig kod. Hur anomalibaserad detektering implementerats i artiklarna skiljer sig mycket och därmed skiljer sig respektive detektionssystem i effektivitet. Detektionssystemen använder olika metoder för att detektera anomalier och olika inlärningsmodeller för att träna detektionssystemen. Exempelvis användes maskininlärning med olika algoritmer, en federerad inlärningsalgoritm, automatiska kodare och djupinlärning. Dataseten i artiklarna skiljer sig mellan varandra i både storlek och typ. Flera dataset inkluderade mer än bara skadlig kod, exempelvis när den infekterade IoT-enheten utför överbelastningsattacker och skannar efter potentiella offer för vidare infektion. Nästan alla detektionssystem hade en träffsäkerhet på över 90% och högst hade ThingNet med en träffsäkerhet på 99,1%.

Att endast använda signaturbaserad detektion i ett detektionssystem förekom inte i artiklarna men två exempel på tillvägagångssätt för en signaturbaseradlösning föreslogs. Signaturbaserad detektion använder signaturer av kända hot, exempelvis skadlig kod, eftersom signaturerna är baserade på kända hot kan inte okända hot upptäckas. De föreslagna lösningarna är ClamAV och Snort IDS, den första använder regler skapade med Yara för att identifiera exempelvis byte-mönster, instruktionssekvenser och specifika strängar, i skadlig kod. Snort IDS använder också regler för att detektera signaturer, fast i nätverkstrafik.

Statistisk analys förekom i en artikel och använder statistik för att studera trender, samband och mönster i data. Att bara förlita sig på statistisk analys förekom inte i ett detektionssystem, däremot användes det för att detektera olika botnät och skadlig kod i ett dataset CTU-13 genom att analysera TCP-flaggor. Genom att analysera ett flertal olika TCP-flaggor kan olika attacker detekteras, varav flaggorna TCP XMAS och TCP NULL kan användas för att detektera skadlig kod.

Tre av artiklarna använde en kombination av detektionsmetoder för att identifiera skadlig kod. I två av artiklarna användes signatur- och anomalibaserad detektion och i en av artiklarna användes statistisk analys och signaturbaserad detektion. Gemensamt bland de tre detektionssystem är att de använder flera moduler eller lager som har olika funktioner. Detektionssystemet som använde signaturbaserad detektion och statistisk analys analyserade nätverkstrafik med signaturer och statistik för att hitta skadlig kod. Detektionssystemet använde också statistisk analys för att bedöma om en IoT-enhet är infekterad med skadlig

kod genom att kolla på CPU användning och processer. De detektionssystem som är anomali- och signaturbaserade använder signaturer för att detektera skadlig kod i nätverkstrafik och filter. Det ena detektionssystemet använder anomalibaserad detektion för att skapa bättre signaturer och det andra detektionssystemet använder anomalibaserad detektion för att bara detektera skadlig kod. Det sistnämnda detektionssystemet hade en träffsäkerhet på 100% med ett dataset på 3000 skadliga koder.

Större delen av alla artiklarna använde lösningar som var nätverksbaserade, endast två var enhetsbaserade. Gemensamt för de metoder som var enhetsbaserade är att de använder sido-kanaler för att detektera skadlig kod. De lösningar som räknas som enhetsbaserade är de som utför detektion direkt på enheten. Några nätverksbaserade lösningar använde moduler på IoT-enheten för att samla in data, och utförde analysen externt. De nätverksbaserade detektionssystemen presterar i snitt lägre än det enhetsbaserade detektionssystemet som hade en träffsäkerhet på 99,1% och 96,3% för tidigare okände hot. Bland de nätverksbaserade detektionssystemen, uppnådde ett detektionssystem en träffsäkerhet på 100%.

Referenser

AL MOGBIL, R., AL ASQAH, M., EL KHEDIRI, S. (2020). IoT: Security Challenges and Issues of Smart Homes/Cities. 2020 International Conference on Computing and Information Technology (ICCIT-1441), 1-6, doi:10.1109/ICCIT-144147971.2020.9213827 [Hämtad 2023-05-14]

Auliar, R. B. & Bekaroo, G. (2021). Security in IoT-based Smart Homes: A Taxonomy Study of Detection Methods of Mirai Malware and Countermeasures. 2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 1-6, doi:10.1109/ICECCME52200.2021.9590841 [Hämtad 2023-04-30]

Avast. (27 september 2018). Torii botnet - Not another Mirai variant. https://blog.avast.com/new-torii-botnet-threat-research [Hämtad 2023-03-22]

Benkhelifa, E., Welsh, T., Hamouda, W. (2018). A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems. IEEE Communications Surveys & Tutorials. 20, 3496-3509, doi:10.1109/COMST.2018.2844742 [Hämtad 2023-05-14]

Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research In Psychology, 3, 77-101, doi: 10.1191/1478088706qp063oa [Hämtad 2023-03-30]

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software Volume 80, Issue 4, 571-583, https://doi.org/10.1016/j.jss.2006.07.009 [Hämtad 2023-03-29]

Chalasani, V. & Alhamdani, W. (2021). Detection and Analysis of P2P Malware Detection in IoT Smart Home Applications. 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), 1-6, doi:10.1109/CSITSS54238.2021.9683299 [Hämtad 2023-04-30]

CIS. (u.å). The Mirai Botnet – Threats and Mitigations. https://www.cisecurity.org/insights/blog/the-mirai-botnet-threats-and-mitigations [Hämtad 2023-03-17]

Cloudflare. (u.å). What is remote code execution? https://www.cloudflare.com/learning/security/what-is-remote-code-execution/ [Hämtad 2023-03-23]

Cloudflare. (u.å). What is the Mirai Botnet? https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/ [Hämtad 2023-03-17]

Cosson, A., Sikder, A. K., Babun, L., Celik, Z. B., McDaniel, P., Uluagac, A. S. (2021). Sentinel: A Robust Intrusion Detection System for IoT Networks Using Kernel-Level System Information. Proceedings of the International Conference on Internet-of-Things Design and Implementation, 53-66, doi:10.1145/3450268.3453533 [Hämtad 2023-04-30]

Fortinet. (u.å). Intrusion Detection System (IDS). https://www.fortinet.com/resources/cyberglossary/intrusion-detection-system [Hämtad 2023-03-23]

Gandhi, H., Mehra, M., Ribeiro, V. (2021). BOND: Efficient and Frugal DL Model Co-Design for Botnet Detection on IoT Gateways. The First International Conference on AI-ML-Systems, 1-7, doi:10.1145/3486001.3486237 [Hämtad 2023-04-30]

Haddaway, N. R., Woodcock, P., Macura, B., Collins, A. (Juni 2015). Making literature reviews more reliable through application of lessons from systematic reviews. Conservation Biology, 29, doi: 10.1111/cobi.12541 [Hämtad 2023-03-30]

Hasan, M. (18 maj 2022). State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally. https://iot-analytics.com/number-connected-iot-devices/ [Hämtad 2023-03-24]

Hasan, M., Biswas, P., Bilash, M. T. I., Dipto, M. A. Z. (2018). Smart Home Systems: Overview and Comparative Analysis. 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), 264-268, doi:10.1109/ICRCICN.2018.8718722 [Hämtad 2023-05-14]

IBM. (2022). X-Force Threat Intelligence Index 2022. https://www.ibm.com/downloads/cas/ADLMYLAZ [Hämtad 2023-05-12]

Jie, Y., Pei, J. Y., Jun, L., Yun, G., Wei, X. (2013). Smart Home System Based on IOT Technologies. 2013 International Conference on Computational and Information Sciences, 1789-1791, doi:10.1109/ICCIS.2013.468 [Hämtad 2023-05-14]

Kaspersky. (18 september 2018). New IoT-malware grew three-fold in H1 2018. https://www.kaspersky.com/about/press-releases/2018_new-iot-malware-grew-three-fold-in-h1-2018 [Hämtad 2023-05-12]

Kitchenham, B. & Charters, S., (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering [Hämtad 2023-03-29]

Kitchenham, B. (Juli 2004). Procedures for Performing Systematic Reviews. https://www.researchgate.net/publication/228756057_Procedures_for_Performing_Systematic_Reviews [Hämtad 2023-03-29]

Li, Z. & Zhao, D. (2022). ThingNet: A Lightweight Real-Time Mirai IoT Variants Hunter through CPU Power Fingerprinting. Proceedings of the 2022 Conference & Exhibition on Design, Automation & Test in Europe, 310-315, doi:10.5555/3539845.3539923 [Hämtad 2023-04-30]

Madan, S., Sofat, S., Bansal, D. (2022). Tools and Techniques for Collection and Analysis of Internet-of-Things malware: A systematic state-of-art review. Journal of King Saud University - Computer and Information Sciences. 34, 9867-9888, doi: https://doi.org/10.1016/j.jksuci.2021.12.016 [Hämtad 2023-05-11]

Malche, T. & Maheshwary, P. (2017). Internet of Things (IoT) for building smart home system. 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 65-70, doi:10.1109/I-SMAC.2017.8058258 [Hämtad 2023-05-14]

Malwarebytes. (u.å). WHAT IS A BOTNET. https://www.malwarebytes.com/botnet [Hämtad 2023-06-13]

Markiewicz, R. P. & Sgandurra, D. (2020). Clust-IT: Clustering-Based Intrusion Detection in IoT Environments. Proceedings of the 15th International Conference on Availability, Reliability and Security, 1-9, doi:10.1145/3407023.3409201 [Hämtad 2023-04-30]

Martin, V., Cao, Q., Benson, T. (2017). Fending off IoT-Hunting Attacks at Home Networks. Proceedings of the 2nd Workshop on Cloud-Assisted Networking, 67-72, doi:10.1145/3155921.3160640 [Hämtad 2023-04-30]

MSB. (mars 2020). IoT-relaterade risker, uppdaterad version. https://rib.msb.se/filer/pdf/29057.pdf [Hämtad 2023-03-10]

Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., Sadeghi, A. (2019). DÏoT: A Federated Self-learning Anomaly Detection System for IoT. 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 756-767, doi: 10.1109/ICDCS.2019.00080 [Hämtad 2023-03-26]

NIST. (4 februrari 2022). Recommended Criteria for Cybersecurity Labeling for Consumer Internet of Things (IoT) Products. https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.02042022-2.pdf [Hämtad 2023-05-14]

NIST. (december 2018). Risk Management Framework for Information Systems and Organizations. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-37r2.pdf [Hämtad 2023-05-14]

Park, J., Rahman, F., Vassilev, A., Forte, D., Tehranipoor, M. (2019). Leveraging Side-Channel Information for Disassembly and Security. J. Emerg. Technol. Comput. Syst. 16, 1-21, doi:10.1145/3359621 [Hämtad 2023-04-30]

Pelaez, D. U., Diaz Lopez, D., Nespoli, P., Marmol, F. G. (2018). TRIS: A Three-Rings IoT Sentinel to Protect Against Cyber-Threats. 2018 Fifth International Conference on Internet of Things: Systems, Management and Security, 123-130, doi:10.1109/IoTSMS.2018.8554432 [Hämtad 2023-04-30]

Sentor. (29 mars 2021). Vad innebär termerna IDS och IPS? https://www.sentor.se/artikel/vad-innebar-termerna-ids-och-ips/ [Hämtad 2023-03-17]

Sentor. (29 mars 2021). Vad är malware? https://www.sentor.se/artikel/malware/ [Hämtad 2023-06-13]

Shahid, M. R., Blanc, G., Zhang, Z., Debar, H. (2019). Anomalous Communications Detection in IoT Networks Using Sparse Autoencoders. 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA), 1-5, doi:10.1109/NCA.2019.8935007 [Hämtad 2023-04-30]

SonicWall. (2022). 2022 SONICWALL CYBER THREAT REPORT. https://www.sonicwall.com/medialibrary/en/white-paper/mid-year-2022-cyber-threat-report.pdf [Hämtad 2023-03-25]

SonicWall. (juli, 2020). 2020 SONICWALL CYBER THREAT REPORT. https://www.sonicwall.com/resources/2020-cyber-threat-report-mid-year-update-pdf/ [Hämtad 2023-03-25]

Trend Micro. (19 april 2017). BrickerBot Malware Emerges, Permanently Bricks IoT Devices. https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/brickerbot-malware-permanently-bricks-iot-devices [Hämtad 2023-03-23]

Wan, T., Ban, T., Lee, Y., Cheng, S., Isawa, R., Takahashi, T., Inoue, D. (2020). IoT-Malware Detection Based on Byte Sequences of Executable Files. 2020 15th Asia Joint Conference on Information Security (AsiaJCIS), 143-150, doi: 10.1109/AsiaJCIS50894.2020.00033 [Hämtad 2023-03-27]

Wazid, M., Das, A. K., Rodrigues, J. J. P. C., Shetty, S., Park, Y. (2019). IoMT Malware Detection Approaches: Analysis and Research Challenges. IEEE Access. 7, 182459-182476, doi:10.1109/ACCESS.2019.2960412 [Hämtad 2023-05-11]

Wenbo, Y., Quanyu, W., Zhenwei, G. (2015). Smart home implementation based on Internet and WiFi technology. 2015 34th Chinese Control Conference (CCC), 9072-9077, doi:10.1109/ChiCC.2015.7261075 [Hämtad 2023-05-14]

Xiao, Y. & Watson, M. (2019). Guidance on Conducting a Systematic Literature Review. Journal of Planning Education and Research, 39, 93-112, doi: 0.1177/0739456X17723971 [Hämtad 2023-03-30]

Yu, R., Zhang, X., Zhang, M. (2021). Smart Home Security Analysis System Based on The Internet of Things. 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 596-599, doi:10.1109/ICBAIE52039.2021.9389849 [Hämtad 2023-04-30]

Zeltser, L. (10 februari, 2016). How Antivirus Software Works: 4 Detection Techniques. https://zeltser.com/how-antivirus-software-works/# [Hämtad 2023-03-28]