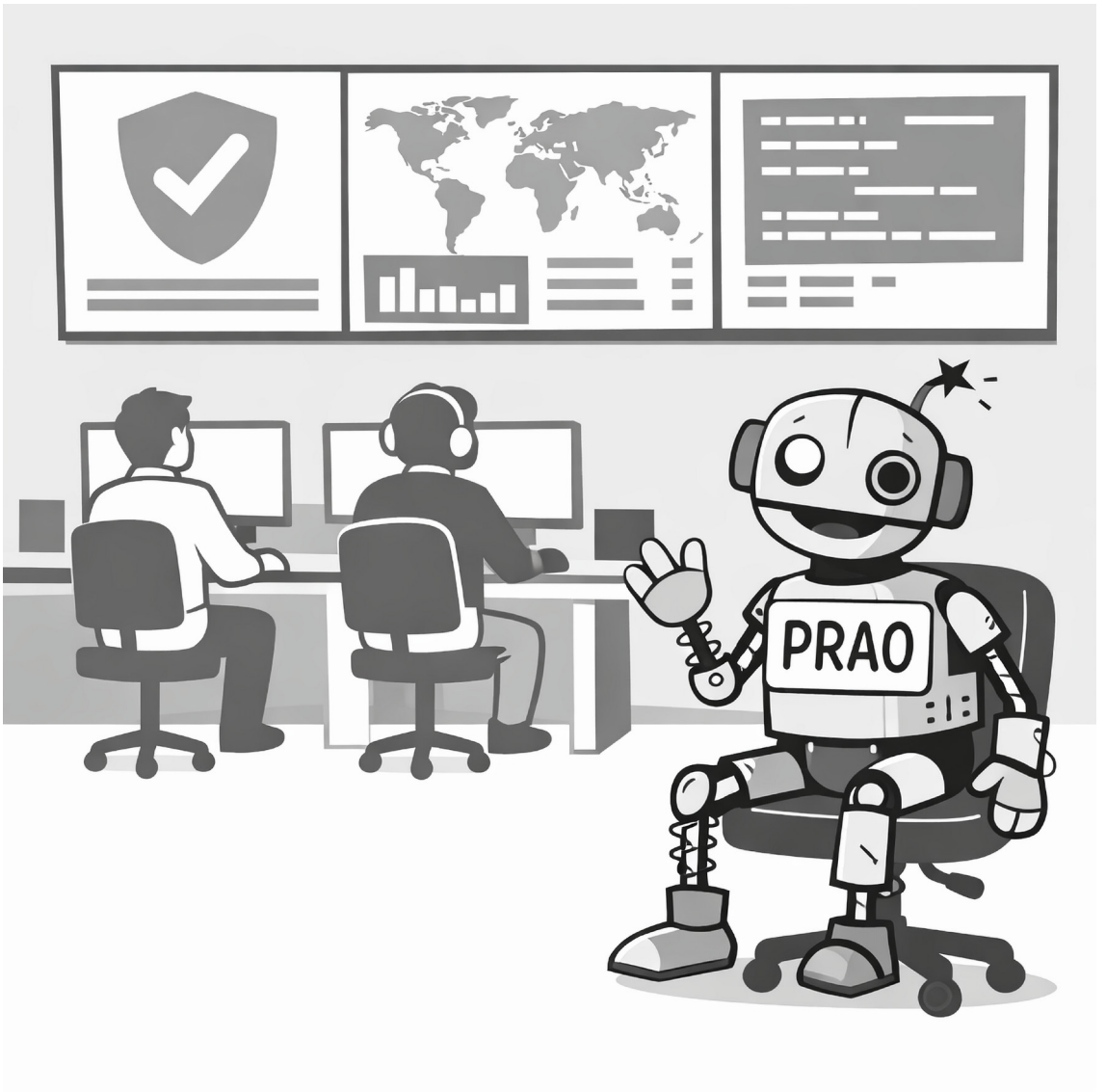
Automatisk incidenthantering

Erfarenheter från labbförsök

TEODOR SOMMESTAD, HENRIK KARLZÉN



FOI-R--5878--SE
ISSN 1650-1942 Februari 2026

Teodor Sommestad, Henrik Karlzén

Automatisk incidenthantering

Erfarenheter från labbförsök

Titel	Automatisk incidenthantering – Erfarenheter från labbförsök
Title	Automatic incident handling – Experiences from laboratory trials
Rapportnr/Report no	FOI-R--5878--SE
Månad/Month	Februari
Utgivningsår/Year	2026
Antal sidor/Pages	40
ISSN	1650-1942
Uppdragsgivare/Client	MSB
Forskningsområde	Cyberförsvar och cybersäkerhet
FoT-område	Inget FoT-område
Projektnr/Project no	B73012
Godkänd av/Approved by	Emil Hjalmarson
Ansvarig avdelning	Cyberförsvar och ledningsteknik

Bild/Cover: Shutterstock

Sammanfattning

FOI har deltagit i tre projekt som på olika sätt försökt automatisera incidenthanteringsprocessen. I projekten har ett stort antal tekniska lösningar för olika delsteg tagits fram ihop med förslag på hur de ska sättas samman. FOI har i alla projekt varit ansvariga för arbetet med att skapa testfall där fiktiva incidenter utspelar sig i en labbmiljö. Detta har använts för tester av enskilda tekniker och komponenter samt för mer omfattande demonstrationer där incidenter utspelar sig. Testerna och demonstrationerna pekar på att mycket återstår innan incidenthanteringsprocessen kan automatiseras i sin helhet. Dels finns praktiska hinder som forskningsprojekten med avsikt förenklat bort, dels fungerar få steg i de verktygskedjor som tagits fram.

Nyckelord: cyberförsvar, incidenthantering, cyber range, intrångsdetektion

Summary

FOI has participated in three projects that, in different ways, attempted to automate the incident handling process. In these projects, a large number of technical solutions for various subprocesses have been developed, along with proposals for how they should be integrated. In all projects, FOI has been responsible for creating test cases in which fictional incidents play out in a lab environment. These test cases have been used to test individual techniques and components, as well as in more comprehensive demonstrations where incidents play out. The tests and demonstrations indicate that much remains before the incident handling process can be fully automated. There are practical obstacles that have been intentionally simplified in the research projects, and only a few steps in the developed toolchains function as intended.

Keywords: cyber defence, incident handling, cyber range, intrusion detection

Innehållsförteckning

1 Inledning

Denna rapport handlar om erfarenheter från projekt som FOI deltagit i under perioden 2021–2025 och som rör automation av incidenthantering. I avsnitten nedan beskrivs bakgrunden till projekten, projekten som bedrivits och rapportens struktur.

1.1 Bakgrund

Denna rapport sammanställer kunskap som erhållits i tre projekt som syftade till att hel- eller halvautomatisera incidenthanteringsprocessen. I projekten klassificerades misstänksamma händelser i datornätverket automatiskt och utifrån detta fattades ett beslut om att exempelvis isolera vissa maskiner. Denna idé är på inget sätt ny. Kommersiell antivirusmjukvara har funnits i över 30 år och populära detektionssystem för nätverk har sedan länge gjort det möjligt att automatiskt utföra åtgärder när misstänkta saker observerats. Samtidigt sker mycket fortfarande manuellt i driftsatta system, vilket bland annat syns genom behovet av personal som övervakar systems cybersäkerhet. Det finns således utrymme för, och behov av, ytterligare automation.

FOI har sedan länge bedrivit forskning om incidenthantering i cyberdomänen. Ett exempel på tidig forskning är prototypen Panorama som samlade in loggar för central analys under cyberförsvarsövningar (Lundholm m.fl., 2011). Sedan denna tidiga forskning har detektionssystem och incidenthanteringsprocesser mognat och utvecklats betydligt. Exempelvis finns det idag flera väl utvecklade centraliserade logginsamlingslösningar och det börjar etableras standarder för hotbeskrivningar som kan användas för klassificeringar. Sådant var ovanligt eller mindre moget för tio år sedan. Denna ökade mognad och standardisering öppnar upp nya möjligheter att automatisera olika delar av incidenthanteringsprocessen, exempelvis för att dataformat och rutiner är mer genomtänkta.

Med anledning av att manuella moment dominerar i dagens incidenthantering, och att det gjorts tekniska framsteg som rimligtvis förenklar automationsansatser, valde FOI att runt år 2020 söka och delta i flera projekt som försöker automatisera delar av incidenthanteringsprocessen. I projekten fanns idéer om att det skulle gå att upptäcka angrepp genom att undersöka flera typer av loggar samtidigt och se samband mellan dessa. Det fanns också förhoppningar om att det skulle gå att automatisera beslut om åtgärder utifrån de loggar som samlas in och statisk information om nätverket.

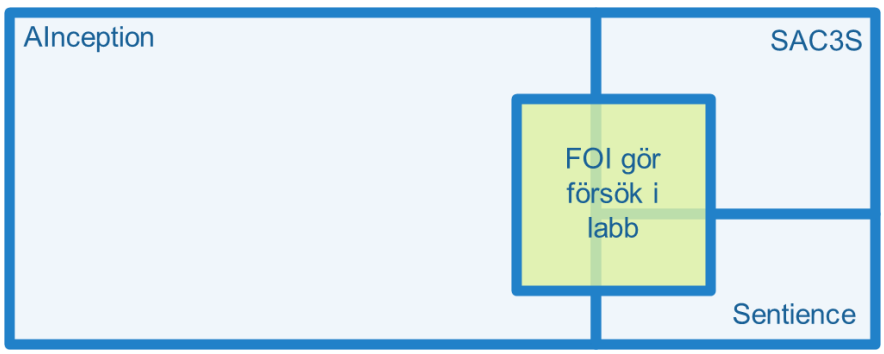
1.2 Projektverksamhet 2021–2025

Forskningen som legat till grund för denna rapport genomfördes under perioden 2021–2025 i tre projekt:

- Sentience – finansierat av Myndigheten för samhällsskydd och beredskap (MSB) och utfört tillsammans med Kungliga Tekniska högskolan (KTH).
- AInception – finansierat genom Europeiska försvarsfonden (EDF) och utfört tillsammans med ett tjugotal organisationer, bland andra norska Forsvarets forskningsinstitutt (FFI) och österrikiska Austrian Institute of Technology (AIT).
- SAC3S – finansierat genom Försvarsmaktens demonstratorprogram och utfört tillsammans med Försvarets materielverk (FMV) och KTH.

Samtliga projekt har tilldelats medel i konkurrens med andra sökande som presenterat alternativa forskningsprojekt.

Totalt omsluter de tre projekten över 100 Mkr under fyraårsperioden. FOI:s del har sammanslaget varit i storleksordningen 10 Mkr och i alla projekt varit liknande: att ta fram militärt relevanta testfall och testa tekniska lösningsförslag i ett labb. Figur 1 illustrerar de tre projektens storlek och hur stor del FOI varit inblandad i.



Figur 1. De tre projekten i ungefärlig storleksordning och FOI:s arbete med att skapa testfall, generera testdata och möjliggöra demonstrationer.

De automatiseringslösningar som beskrivs senare i rapporten har skapats av forskare utanför FOI. FOI har dock genom att delta i utvecklingen av testscenarion och övervaka både tester och demonstrationer fått god insyn i hur lösningar är konstruerade och hur de presterar. Denna rapport redovisar dessa lärdomar på en övergripande nivå.

1.3 Rapportstruktur

Denna rapport riktar sig till beslutsfattare inom cyberförsvaret och syftar till att på en övergripande nivå sammanfatta erfarenheterna från de projekt FOI varit involverade i. Författarna hoppas dessa erfarenheter kan informera om vad som är möjligt idag och vilka hinder som återstår innan hantering av cyberincidenter kan automatiseras.

Beskrivningen av projekten är i rapporten indelad i tre delar:

- I kapitel 2 beskrivs vad som testas. Det ges en beskrivning av problemet i stort och vilka tekniska lösningar som prövats i projekten.
- I kapitel 3 beskrivs hur det testats. Mer specifikt beskrivs hur FOI:s plattform Crate använts för att skapa datornätverk och sätta upp testfall.
- I kapitel 4 beskrivs hur det gick. Detta görs genom en översiktlig beskrivning av resultaten från tester och demonstrationer.

Sist, i kapitel 5, dras några slutsatser och ges några rekommendationer. Den som är intresserad av detaljer om de tekniska lösningarna kommer inte finna mycket av värde i denna rapport. Sådana läsare hänvisas istället till att söka efter rapporter och artiklar som producerats i projekten, av vilka flera citeras i denna rapport och refereras på projekts hemsidor[1].

[1] EU-projektet AInceptions hemsida finns på: https://www.ainception.eu; projektet Sentience hemsida nås på: https://www.kth.se/sv/cdis/forskning/sentience-simulerings-baserat-och-forstarkningsinlart-driftstodssystem-for-cybersakerhet-1.1316723.

2 Vad som testats

Som nämnts ovan är idén om automatiserad incidenthantering inte ny. Exempelvis diskuterades det redan för tjugo år sedan i Fuchsberger (2005) hur ”*intrusion prevention system”* var en ny term som användes i marknadsföring och att systemen var ”*essentially a combination of access control (firewall/router) and Intrusion Detection Systems”*. I andra artiklar från samma tid ges designförslag för mer kompletta lösningar som reagerar olika beroende på egenskaper hos miljön de är placerade i (Papadaki & Furnell, 2006). Trots att lösningar för automatisk incidenthantering marknadsförts och beskrivits länge sker ännu många steg i incidenthanteringsprocessen manuellt.

Nedan beskrivs först vilka delar som brukar ingå i incidenthanteringsprocessen. Därefter beskrivs vilka tekniska lösningar som ofta används i incidenthantering och vilka förslag som fanns innan projekten började. Sist beskrivs vilka ansatser som prövades i de tre projekten.

2.1 Incidenthanteringsprocessen

När ordet incidenthanteringsprocess används i denna rapport syftar det på liknande arbetsuppgifter som inkluderas i rollerna som NIST NICE[2] kallar *Defensive Cybersecurity* och *Incident Response*. Dessa två roller har ansvar för att analysera data som samlats in från diverse cybersäkerhetsskydd respektive att undersöka, analysera och svara på cybersäkerhetsincidenter i datornätverk. Tillsammans ska de utföra 65 uppgifter som listas i ramverket. Flera av uppgifterna är just sådant som projekten siktat på att helt eller delvis automatisera. Exempelvis har det funnits ambitioner att automatisera uppgifterna:

- *T1348: Distinguish between benign and potentially malicious cybersecurity attacks and intrusions*
- *T1388: Isolate malware*

Det finns också uppgifter i rollbeskrivningarna som, trots att de ingår i incidenthantering, inte har försökt automatiseras genom projektets ramar. Exempel på sådana uppgifter är:

- *T1539: Analyze organizational cybersecurity posture trends*

[2] Ett amerikanskt ramverk eller katalog som beskriver olika typer av uppgifter och roller inom cyberdomänen. https://www.nist.gov/itl/applied-cybersecurity/nice/nice-framework-resource-center/nice-framework-current-versions

- *T1618: Advise stakeholders on disaster recovery, contingency, and continuity of operations plans*

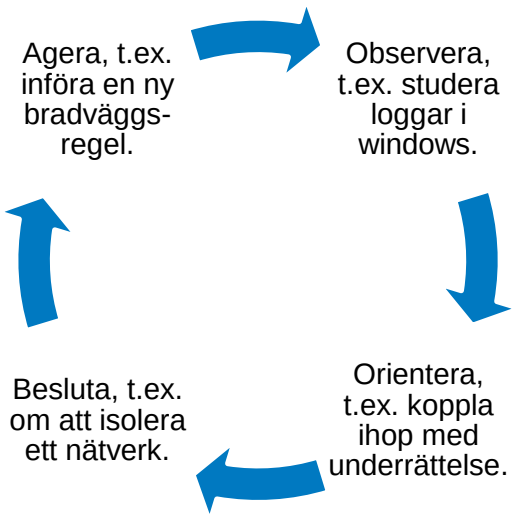
D3FEND[3] (Kaloroumakis & Smith, 2020) är ett annat sätt beskriva vad defensiva cyberoperationer och incidenthantering handlar om. D3FEND beskriver sju taktiker som tillsammans kopplas till över 200 tekniker. Vissa taktiker handlar om sådant som bör göras innan incidenter sker, t.ex. att modellera systemets behörighetskontroll och härda det med en stark lösenordspolicy. En taktik handlar om detektion av angrepp eller incidenter. Fyra taktiker handlar om åtgärder som kan göras under incidenthantering: isolera, vilseleda, avlägsna hot och återställa. Projekten som redovisas här täcker in tekniker som sorteras in under taktikerna modellering (t.ex. sårbarhetsinformation), detektion (t.ex. signaturer i nätverkstrafik) och ett fåtal under isolering (t.ex. blockering i brandväggar). I testerna görs antaganden om att de första stegen är förberedda och väl utförda. Exempelvis antas att kompletta nätverkskartor finns att tillgå eller att maskiners prioriteter är kvantifierade enligt skalor för riktighet, tillgänglighet och konfidentialitet. Dessa säkerhetsattribut förkortas ibland CIA, efter deras engelska motsvarigheter.

Inom Försvarsmakten har ett koncept för defensiva cyberoperationer kallat DCO-konceptet tagits fram (Ekstorm & Sethi, 2024). I detta beskrivs fyra operativa funktioner för defensiva cyberoperationer (DCO): sensoranalys, uppsökande verksamhet, incidenthantering och motåtgärder. Samma delar återkommer i doktrinansatsen för cyberförsvar (Cyberförsvarsledningen, 2024). Projekten i denna rapport fokuserar på incidenthanteringsfunktionen, men även problem inom funktionerna sensoranalys och motåtgärder berörs. Projekten har flera likheter med funktionsbeskrivningen i DCO-konceptet. Exempelvis står i DCO-konceptet att ”[v]erksamhetsägaren behöver kunna uttrycka vilka egenskaper (riktighet, tillgänglighet eller konfidentialitet) som är viktigast” (Ekstorm & Sethi, 2024, s. 64). Det är just sådant som förberetts med skalorna som nämndes i slutet av förra stycket. Det finns också skillnader. Exempelvis konstateras i DCO-konceptet att en incidenthanteringsgrupp kan behöva åtkomst som den normalt inte har. I forskningsprojekten har sådan tillgång getts och till och med förberetts så att de kan utföras med enkla förberedda API-anrop.

Istället för att knyta an till ovanstående beskrivningar av defensivt cyberförsvarsarbete och incidenthantering kommer resterande del av denna rapport relatera till en schematisk bild av den beslutsprocess som sker under incidenthantering. Mer specifikt kommer beskrivningarna kopplas till den så kallade OODA-loopen (eng. observe, orient, decide, act; sv. observera, orientera, besluta, agera). OODA-loopen har utvecklats för att beskriva beslut i en militär

[3] Se https://d3fend.mitre.org.

kontext, ofta ihop med maxim av typen ”vi vill komma innanför motståndarens OODA-loop”. Det är inte ovanligt att loopen används för att beskriva beslut kopplat till cyberförsvar eller incidenthantering. Exempelvis användes OODA-loopen av Sawilla och Wiemer (2011) för att beskriva en planerad lösning för automatiskt cyberförsvar. Därtill beskrev Husák m.fl. (2022) ett rekommendationssystem för automatiskt cyberförsvar med hjälp av OODA-loopen. En förenkling[4] av vad OODA-loopen betyder i kontexten för denna rapport visas i figur 2.



Figur 2. OODA-loppens fyra steg och exempel på vad som sker i stegen.

Verktyg och automation är en naturlig del av processer kopplade till cyberförsvar och incidenthantering. Just den typ av automation som testats i projekten vi beskriver här pekas även ut i Försvarsmaktens DCO-koncept som ett exempel på när automation är passande (Ekstorm & Sethi, 2024, s. 33–34):

> ”*För skydd eller det defensiva området kan ett exempel på automatisering vara en verkansplattform som konsumerar flöden av tekniska indikatorer och sensordata och sedan agerar automatiskt genom att larma eller verka genom att ändra konfiguration direkt i relevanta komponenter i terrängen*.”

Det finns många hinder att övervinna för att framgångsrikt automatisera incidenthanteringsprocessen. Exempelvis är mycket av den data som finns

[4] Det är ofta svårt att bestämma var i OODA-loopen ett steg ligger. Exempelvis kan skapandet av säkerhetslarm placeras i antingen observera-steget eller i orientera-steget beroende på hur larmet skapas. Om det är ett larm som kommer från en enkel regel för ett känt filnamn kan det ses som observation; om det skapas genom att korrelera fler pågående händelser kan det ses som orientering.

tillgänglig ostrukturerad eller semistrukturerad i datamodeller som inte är kompatibla med varandra. I tabell 1 ges exempel på sådant som människor idag utför och en förklaring av varför det kan vara svårt att automatisera.

Tabell 1. Exempel på hinder för automation.

Fas	**Uppgift i NICE-ramverket**	**Aktiviteter som människor gör**	**Exempel på hinder för automation**
Observera	T1350: Perform continuous monitoring of system activity	De övervakar skärmar för att hitta allvarliga händelser eller suspekta mönster.	Att representera vad som är allvarligt i organisationen eller utgör suspekta mönster.
Orientera	T1299: Determine causes of network alerts	Ibland ringer de användare eller systemadministratörer för att kontrollera om de gjort något ovanligt.	Social interaktion som är flexibel och uppsökande.
Besluta	T1260: Perform real-time cyber defense incident handling	Fattar beslut, ofta med bristfällig information, om vad som kan isoleras på ett effektivt sätt utan att störa för mycket.	Det finns inget etablerat formellt språk för att beskriva hur en incident påverkar funktioner.
Agera	T1388: Isolate malware	Ibland ringer de en systemadministratör eller användare och ber dem rycka ut sladden.	Det kan saknas möjlighet att isolera från en central plats, t.ex. för att behörigheter saknas.

2.2 Automation som finns och föreslagits

Som i många teknikområden finns en skillnad mellan vilka tekniska lösningar som är driftsatta i verksamheter och den typ av tekniska lösningar som forskare arbetar med. En förenklad sammanfattning av skillnaderna ges i tabell 2.

Tabell 2. Förenklad beskrivning av skillnaden mellan driftsatta lösningar och de tekniska lösningar akademiker forskar på.

Steg	**Typisk driftsatt lösning i en större organisation**	**Typisk akademisk automationslösning**
Observera	SIEM-system med främst maskinloggar som behöver normaliseras och struktureras.	Fullpaketinspelningar eller flödesinformation från nätverksutrustning ihop med en nätverkskarta.
Orientera	Larm för känt elakartat beteende baserat på underrättelser. En prioritetslista över maskiner och tjänster och kunskap om normalbild.	Avancerad anomalidetektion med maskininlärning kombinerad med en modell över hot, t.ex. i form av en attackgraf.
Besluta	Människor som på egen hand eller i samarbete bestämmer vad som ska göras.	Matematisk modell som väger olika värden rankade 1–5 mot varandra för att avgöra om en aktion är värd att utföra.
Agera	Systemadministratörer som går in i system och ändrar inställningar, drar ur sladdar och liknande.	Isolerar maskiner med nätverksinställningar, stänger ner processer eller försöker vilseleda.

För praktiker är logginsamlingen ett centralt problem som kan uppta betydande resurser. I praktiken kräver detta bland annat hantering av sensorer av olika slag, korrekta konfigurationer på olika plattformar, meddelandeköer eller andra sätt att flytta data samt lagringsutrymme och enhetliga beskrivningar av loggformat. Central logginsamling har marknadsförts i cirka 20 år och system som benämns *Security Information And Event Management* (SIEM) började användas för cirka 15 år sedan. Denna typ av system är idag vanlig och utbredd. SIEM-system syftar till att lösa observera-steget i OODA-loopen, och i viss mån lösa ut orientera-steget genom exempelvis den korrelation som kan göras mellan loggar och tidpunkter. I analysföretaget Gartners *hype cycle* är SIEM-system på *plateau of productivity*, en fas där teknologin uppvisat fördelar i praktiken och blivit, eller börjar bli, utbrett använd (Nunez & Livingstone, 2025).

I akademiska sammanhang använder tester ofta nätverksloggar eller systemloggar för att automatisera delar av de två första stegen i OODA-loopen. De praktiska bestyren med att samla in sådan data förenklas dock ofta bort av akademiker. Inom steget observera finns tusentals akademiska studier för att lösa problemet med detektion, oftast med fokus på paketinspelningar från nätverkstrafik under optimistiska antaganden som att ingen trafik är krypterad. Flera sammanfattningar av sådan forskning finns, exempelvis den av Khraisat m.fl. (2019). Det finns också de som fokuserar på SIEM-system och syntes av loggar. Exempelvis har det presenterats en översikt över problem kopplade till SIEM-system med uppmaningar till forskare att arbeta med bland annat

datafusion och databasfrågor (Zuech m.fl., 2015). Därtill finns gott om forskning med fokus på orientera-steget i form av exempelvis korrelation av händelser (Kotenko m.fl., 2022), prioritering av larm (Jalalvand m.fl., 2024) och användning av hotunderrättelser (Sun m.fl., 2023).

De senare stegen i OODA-loopen har historiskt fått mindre fokus bland praktiker och på marknaden. För några år sedan var system som gick under benämningen *Security Orchestration, Automation, and Response* (SOAR) nya och sågs som framtiden. Sådana system gör det möjligt att utföra åtgärder i maskiner och nätverksutrustning på ett enkelt sätt genom att erbjuda ett standardiserat gränssnitt för omkonfigurationer etc. SOAR-system syftar alltså till att lösa ut agera-delen av OODA-loopen. Under 2025 dömde Gartners analys av security operations ut SOAR-system som föråldrade innan de nått bred användning och de ingår istället i den nyare tekniken *Extended Detection and Response* (XDR) (Nunez & Livingstone, 2025). XDR-lösningar syftar till att sätta ihop hela loopen, bland annat genom att kombinera detektion från exempelvis SIEM-system med hotunderrättelser innan de gör åtgärder i SOAR-system eller liknande. XDR bedömdes i samma Gartner-analys att vara i *trough of disillusionment*, där tekniken tidigare omfattats av överdrivna förväntningar, men nu nått en nivå det mediala intresset minskat eftersom förväntningarna inte kunde infrias. Flera andra relaterade tekniker tas upp i Gartners analys. Bland annat finns tekniker som visserligen har minst fem år till bred användning men som förknippas med uppåtgående förväntningar. Detta gäller *Cybersecurity Incident Response Management* (hantering av internkommunikation vid incidenter) och *AI SOC Agents* (för exempelvis sammanfattning av händelser). Det finns alltså en livlig marknad kring automatisering av incidenthanteringsverksamhet.

Även i forskning har de senare stegen av OODA-loopen historiskt getts mindre uppmärksamhet. Få studier har försökt bygga beslutsmodeller som avgör vad som behöver göras när något inträffar och lösa besluta-frågor. Ett exempel på forskning i detta spår är Shaked m.fl. (2023) som väger in verksamhetsmodeller i besluten. Det har också gjorts visst arbete med agera-steget. Exempelvis har en standard för SOAR-system kallad OpenC2 presenterats i en artikel (Mavroeidis & Brule, 2020) och specifika tekniker för vilseledning har studerats (Karlzén, 2021). Tydligt är dock att de första stegen i OODA-loopen har getts mer uppmärksamhet av forskare.

Trots att de senare stegen i OODA-loopen inte varit i fokus för forskare har en hel del forskning försökt automatisera hela processen eller flera steg i processen. FOI gjorde själva en systematisk genomgång av den forskning som publicerats sedan år 2000 (Karlzén & Sommestad, 2023). Denna genomgång tittade på vilken indata och utdata som kom ut ur automationslösningarna. Slutsatserna var att vissa indata dominerar och föredras av forskare. Exempelvis är nätverkskartor och nätverkstrafik vanliga som indata och många akademiker fokuserar på att isolera hot med hjälp av nätverkskonfigurationer. Forskarnas förslag sattes i

relation till publika kataloger över åtgärder som finns i kommersiella SOAR-system. Här märktes tydliga skillnader. Dels handlade cirka en tredjedel av alla åtgärder i SOAR-systemens kataloger om att sammanföra eller kontrollera information, exempelvis se om en IP-adress finns i ett system. Dels är det betydligt vanligare med indata som rör exempelvis IP-adresser eller identifierare som checksummor från hotunderrättelserapporter. Forskares problembeskrivning är alltså inte helt i linje med vad praktiker arbetar med, vilket tabell 2 illustrerade.

Ett naturligt steg på vägen mot automation är att automatisera del av analysen och snarare ge mänskliga operatörer rekommendationer eller färdiga kommandon att exekvera. Även här finns en hel del kvar innan forskningen är mogen för användning i driftsatta system. En översikt av rekommendationssystem för incidenthantering har gjorts av Husák & Cermak (2022). De landade i slutsatsen att nästan ingen undersöker incidenthanteringsproblemet i sin helhet utan istället görs djupdykningar på enskilda moment. Samtidigt noterar de att området fått ökad uppmärksamhet av forskare och att det troligtvis kommer fler studier.

2.3 Ansatser prövade i projekten

Forskningen som gjorts i de tre projekten var snarlika och ligger i linje med vad tidigare forskning gjort och rekommenderat.

I projektet AInception lades cirka hälften av projektets budget på att ta fram scenarion för test och demonstration. Den andra hälften lades på tekniska arbetspaket som fokuserade på detektion, kontextualisering av larm, värdering av alternativ och utförande av åtgärder. Av dessa lades mest resurser på den första delen av OODA-loopen (observera-orientera) där en OpenSearch-baserad databas över loggar var grunden för arbetet. Totalt togs över 30 tekniker och verktyg fram. Några av de mer framträdande var:

- IAuditGraph som använder *k-nearest neighbors* på maskinloggar i linux (auditd) med hög upplösning.
- AMiner som skapar regler med gränsvärden för acceptabla värden för maskinloggar (Landauer m.fl., 2023) (Wurzenberger m.fl., 2024).
- Maskininlärning för att se anomalier i processer för hur fartyg färdas och drönare flyger.
- D-Bello som använder *k-nearest neighbors* på netflow-data.
- SAGE-IDS som använder GraphSAGE-algoritmen på netflow-data.
- Telosian (Maldonado m.fl., 2025) som gör tidsserieanalys av netflow-data.
- Blend-IDS som använder grafbaserade neurala nätverk på netflow-data.
- Extrahering av information som indikatorer för angrepp (eng. *indicator of compromise*) ur hotunderrättelser i textformat.

- Neurosymbolisk analys av larm för att koppla ihop larm med varandra och med annan information från exempelvis hotunderrättelser (Eckhoff m.fl., 2025).
- Visualisering av en angreppssekvens (eng. *kill chain*) genom att försöka pussla ihop grupper av larm.
- Kunskapsgrafer för fusion eller korrelation av händelser med annan information (Eckhoff m.fl., 2026).
- Simulering av angrepp i ett separat labb (en förenklad digital tvilling) för att lära sig vad som är bra att göra (Jaber, 2024) (Jaber m.fl., 2025).
- Ett verktyg som förklarar vad som ligger bakom en rekommendation genom att räkna ut hur viktiga olika underlag (t.ex. en CVE-kod) varit.
- En planerarare och simulator som med symbolisk analys skapar förslag på åtgärder, bl.a. baserat på nätverkets struktur.
- COATI som översätter förslag för att exempelvis konfigurera om en brandvägg till konkreta tekniska instruktioner i standardformat som Collaborative Automated Course of Action Operations (CACAO).
- SOARCA[5] kan utföra åtgärderna som behövs och klarar att skicka kommandon till bland annat brandväggar med http-anrop, ssh-anrop eller OpenC2-anrop.

Främst testades teknikerna offline mot tillrättalagda data producerad i labb. De flesta delar av projektet nådde således *technology readiness level* (TRL) 3 (*experimental proof of concept*). Cirka hälften av teknikerna integrerades med hjälp av utvalda komponenter via en meddelandekö (mjukvaran Kafka) där det skickas information i olika kanaler mellan komponenterna. Denna användes i slutdemonstrationen där exempelvis AMiner och Telosian skrev in meddelanden i kanaler som lästes när det skulle skapas kunskapsgrafer. Med en generös tolkning har de delar som ingått i slutdemonstrationen nått TRL 4 (*technology validated in lab*).

Det svenska projektet SAC3S hade mer fokus på integration och demonstration av ett koherent verktyg som hanterar hela OODA-loopen. Projektgruppen var också betydlig mindre och sammanhållen. Målet var i planeringen att utveckla en prototyp på TRL 5, dvs. en prototyp demonstrerad i relevant miljö, såsom ett datornät i Försvarsmakten. Det justerades i början av projektet till TRL-nivå 4 och en prototyp demonstrerad i labb. Demonstrationen syftade till att visa hur hotmodeller uttryckta i modelleringsspråket coreLang (Katsikeas m.fl., 2024) kan användas för att analysera risker med larm från SIEM-systemet Wazuh. SIEM-systemet och datainsamlingen var tillrättalagda på så sätt att relevant data samlades in, signaturer för angreppen fanns på plats och åtgärderna kunde utföras med anrop till programmeringsgränssnitt. Baserat på bland annat värderingar av maskiners säkerhetsvärden utryckta i skalan 1–5 fattas beslut om åtgärder som

[5] https://github.com/COSSAS/SOARCA

realiseras med agenterna som SIEM-systemet använder för logginsamling. En modell byggd på maskininlärning med grafbaserade neurala nätverk jämfördes med en enkel modell vilken direkt isolerade maskiner som gav larm om misstänkta hot.

Projektet Sentience har ett betydande personalöverlapp med SAC3S och använder samma modelleringsspråk. Fokus i projektet är dock maskininlärning för att fatta bra beslut om åtgärder. Projektet har alltså en tonvikt på just *besluta*-steget i OODA-loopen. Detta arbete görs genom att skapa grafbaserade modeller över tänkbara angrepp och med förstärkningsinlärning identifiera vad som bör göras vid olika angrepp. En tidig variant av detta ingick i det SAC3S demonstrerade. Arbetet och tester i Sentience pågår när denna rapport skrivs.

Projekten som sammanfattas i denna rapport är förhållandevis typiska för forskningsprojekt och lika de ansatser som tagits tidigare. Den indata som användes i projekten var den som var vanligast i tidigare forskning och de åtgärder som utförs hör till de vanligaste i tidigare forskning. Mycket handlar också om detektion baserat på nätverkstrafik, precis som mycket av tidigare forskning. Det nya är i första hand att delar pusslas ihop till enhetliga system som hanterar flera steg i OODA-loopen. En översikt över vad projekten fokuserat på ges i tabell 3.

Tabell 3. Översikt över ansatser i EU-projektet (AInception) och de nationella projekten (SAC3S och Sentience).

Steg	**EU-projektet**	**Nationella projekten**
Observera	Många olika loggtyper insamlade till OpenSearch. Främst netflow. Larm skapas i första hand med statistiska modeller för avvikelser.	Larm från ett öppet SIEM-system (Wazuh) konfigurerat för att vara lik industristandard, t.ex. med signaturer från Sigma[6].
Orientera	Bland annat kunskapsgrafer, angreppsmodeller och larmaggregering.	Angreppsmodeller och beskrivning av nätverk med värden för olika tillgångar.
Besluta	Flera olika ansatser men framförallt modeller tränade i ett separat enklare labb.	Grafneurala nätverk i förstärkningsinlärning och enkla if-satser.
Agera	Ett eget verktyg kallat SOARCA som är kompatibelt med OpenC2.	Egna varianter på de aktiva åtgärder som exekveras med Wazuhs agenter.

[6] Sigma underhåller en publik databas med pseudokod för signaturer som skapats för vanliga angrepp. https://github.com/SigmaHQ/sigma

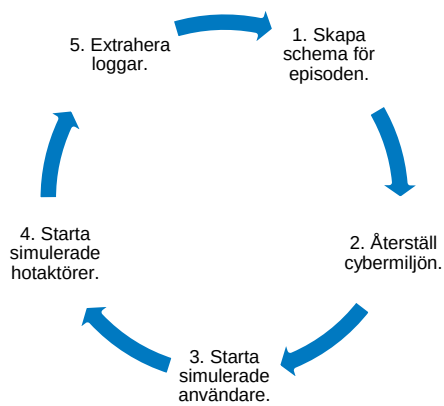
3 Hur testerna gjordes

Samtliga projekt har använt plattformen Crate för att skapa testfall. I de nationella projekten gjordes tester och slutdemonstrationen i olika nätverk skapade i Crate. I EU-projektet gjordes tester i Crate och tre andra plattformar hos andra organisationer, men endast Crate användes som plattform för slutdemonstrationen.

Figur 3 visar processen som använts för att köra ”episoder” som involverar en återställning av miljö, simulering av händelser samt extrahering av loggar. Processen har automatiserats och utförts återkommande med nya kodbibliotek och skript enligt följande:

1. Först skapas ett schema med tider och eventuellt slumpade förutsättningar för episoden. Exempelvis rör det olika scenarion för angrepp som väljs ur en lista.
2. När tiden för start inträffar sätts nätet i ett känt och önskat tillstånd med anrop till det programmeringsgränssnitt som Crate erbjuder.
3. Efter ytterligare lite tid startas simulerade användare med ett anrop till verktyget SVED:s programmeringsgränssnitt.
4. Vid viss tid startas det simulerade angreppet med ett anrop till verktyget Lores programmeringsgränssnitt.
5. Några minuter innan nästa episod stängs hotaktören och simulerade användare av. Sist extraheras loggar som kan behövas för analys med diverse programmeringsgränssnitt. Exempelvis hämtas loggen från angreppssimuleringen och loggar från SIEM-system.

I avsnitten nedan beskrivs steg 2, 3 och 4. Sist ges en beskrivning av realismen och relevansen som åstadkoms.



Figur 3. Episoderna som körts mot Crate för utveckling och test.

3.1 Cybermiljöer

I alla projekt görs simuleringar i en datorhall i Linköping där Crate finns installerad. Det finns olika typer av plattformar för att skapa *cyber ranges* för test och övning. Kampourakis m.fl. (2025) skiljer exempelvis på de som använder molnlösningar, de som använder virtualisering och de som använder containrar. I Crate skapas datornätverk med virtualisering. Detta görs genom att först använda en datamodell för att specificera vilka maskiner som ska ingå och sedan starta en sekvens av utrullningssteg där skript och externa verktyg används för att konstruera det som anges i specifikationen.

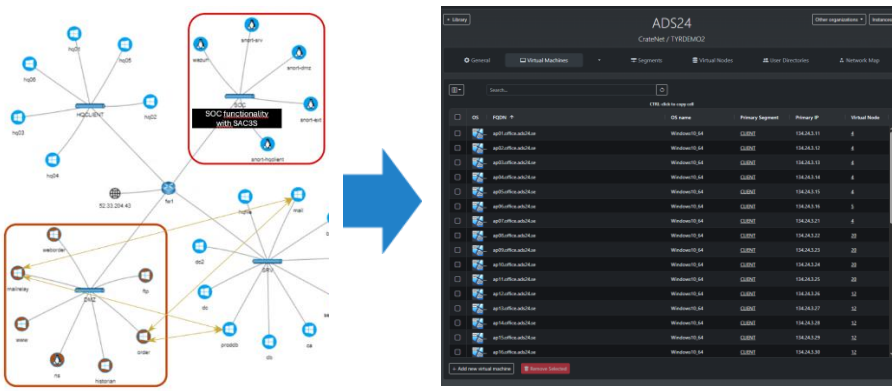
Grafiska gränssnitt och logiska regler hjälper den som bygger nät att göra rimliga val och sätta parametrar med rimliga värden. Exempelvis finns det stöd för att sätta IP-adresser på maskiner och skapa listor med användare och deras lösenord. Stora delar av arbetet med specifikationerna kräver dock kunskap om datornätverk, kunskap om cybersäkerhet eller kunskap om hur Crate fungerar. Under projekten har ett flertal nätverk i olika versioner använts. Processen för att skapa cybermiljöer har varit i stil med detta:

1. Projektdeltagare ritar på en whiteboard eller beskriver ett nätverk i ett textdokument med bilder.
2. Baserat på de installationsskript och licenser som finns i plattformen föreslås olika lösningsalternativ.
3. En första specifikation skapas i webbgränssnittet, ofta genom att kopiera och modifiera ett existerande nät som använts tidigare.
4. Specifikationen rullas ut. Diverse fel inträffar då, exempelvis för att parametrar satts fel. Processen görs om några gånger tills det fungerar.
5. Det görs manuell handpåläggning för sådant som inte hanteras väl av installationsskripten. Exempelvis brandväggsregler.
6. Funktionen verifieras manuellt och en ögonblicksbild tas av nätverket för att kunna återvända till det rena tillståndet som föregår angrepp mm.

Dessa sex steg är en förenkling och processen har krävt flera personmånader av arbete för varje scenario där flera omtag ofta behövts, exempelvis för att något i en konfiguration visat sig vara fel i steg 4.

Nätverken som byggts i Crate har skapats utifrån specifikationer som tagits fram för att vara militärt relevanta av bland andra nederländska domänexperter samt personal på FMV. Textbeskrivningarna och diagrammen som beskrivit de önskade nätverken har varit schematiska som i figur 4 medan specifikationerna för en enskild maskin i Crates databas kan innehålla hundratals tekniska parametrar. Exempelvis resulterade en wordfil på cirka 10 sidor med den nätverkskarta som illustreras i figur 4 i en specifikation med cirka 4000 parametrar som skickades för utrullning. Vissa av dessa parametrar är då fritextfält med konfigurationsinput till sådant som logginsamlingsverktyg där många parametrar kan väljas. Steg 1–6 kräver således att det görs en rad

antaganden utifrån personens domänkunskap och kunskap om försöken. Detta blir inte alltid som de som önskat scenariot har tänkt sig. Ibland har det krävts en korrigering av steg 3 och ny utrullning; ibland har det räckt med en manuell korrigering och att ny ögonblicksbild tas.



Figur 4. Nätverksdiagram över nätverk skapad av FMV och bild från Crates core-webb på maskinerna som definieras i nätverket.

I samtliga projekt uppstod situationen att installationsskript och mjukvaror som behövdes i steg 2 inte fanns färdiga på förhand. Under projekten har därför installationsskript skrivits för:

- Arton olika mjukvaror som haft önskvärda sårbarheter som passat de scenarion som anges. Exempelvis har installationsskript skapats för servermjukvaror som är sårbara för CVE-2023-46604 och CVE-2020-35476.
- En handfull fejkade IT-system kopplade till verksamheten som ska simuleras. Exempelvis simuleras trafik i ett stridsledningssystem med pythonskript som konfigureras via Crate.
- Installation av logginsamlingslösningar som använts i projekten. En var baserad på Elastic/OpenSearch och en var baserad på Wazuh. Till detta har flera färdiga ”profiler” för logginsamling skapats, exempelvis så att vanliga Sysmon-konfigurationer kan väljas i rullgardinslistor.

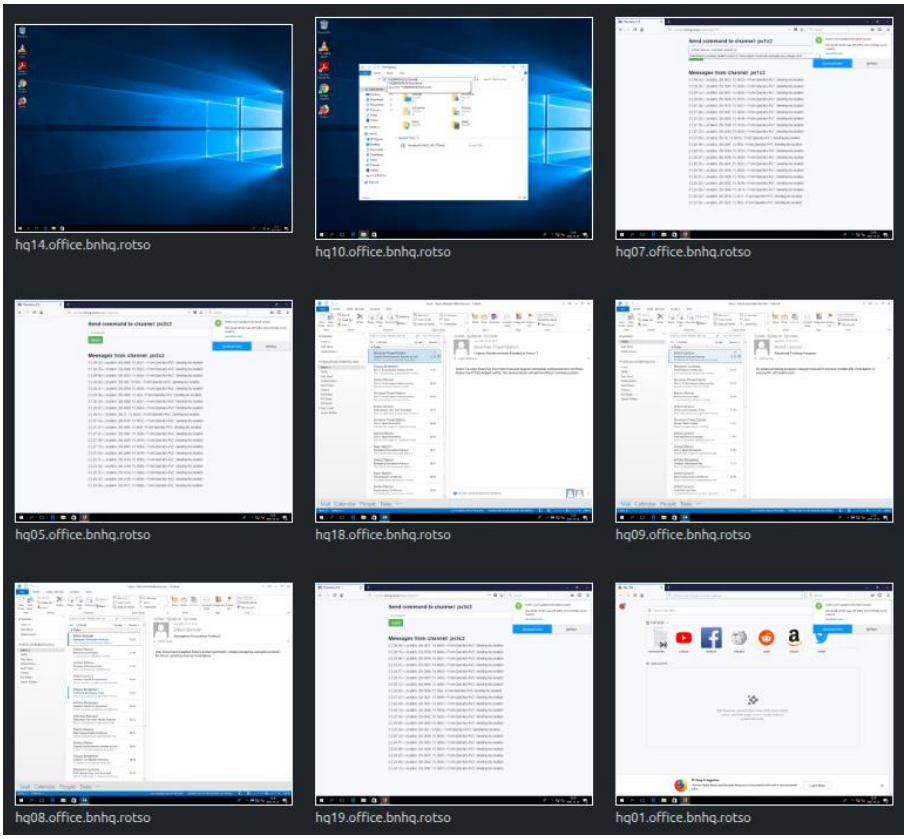
Det sätt som installationsskript numera skrivs på tillkom under projekten och försöksverksamheten har därmed fungerat som en kravställare och testare av funktioner och design i Crate. Ovanpå detta har behoven av att hantera episoder med skript, och att sätta parametrar för simulerade användare, inneburit att programmeringsgränssnittet för Crate använts på nya sätt och behövt utvecklas.

Ännu finns det stora möjligheter att förbättra hur plattformen Crate stödjer tester och experiment. För den typ av experiment som vi utfört i de tre projekten är förmodligen den viktigaste förbättringen att skapa större variation i nätverk och sårbarheter. Det finns begränsad variation i nätverken som skapas och de standardkonfigurationer som återkommer är inte heller validerade som typiska för driftsatta nätverk. Dessa begränsningar i realism och variation gör att det är osäkert om det går att överföra resultat och tekniker från Crate-nätverk till driftsatta nätverk. Som grundläggande testfall för prototyper har dock de stereotypa nätverken i Crate bedömts vara fullt tillräckliga.

3.2 Användare och legitima händelser

I näten som simulerades sker en del icke-trivial maskin-till-maskin-kommunikation när cybermiljöerna startas upp, exempelvis fiktiv trafik från ett simulerat militärt ledningssystem. Utöver detta var det önskvärt att simulera användare för att skapa realistiska och relevanta förutsättningar för testerna. Cyberförsvar av en miljö utan aktiva legitima användare är i de flesta fall enklare än i en där det finns aktiva användare. Framförallt är observation och orientering enklare när merparten av den ovanliga aktiviteten kan antas vara från hotaktörer. Men även att besluta eller utföra nedstängning av system är enklare utan användare som har behov eller lägger sig i. Vad användarna gör i systemet spelar således roll.

I testerna och demonstrationerna utfördes en handfull simulerade aktiviteter och beteendet var förhållandevis repetitivt. De simulerade kontorsanvändarna skickade e-post inbördes, öppnade e-post, öppnade filer och besökte interna webbsidor. I simuleringarna fanns också simulerade systemadministratörer som ibland körde kommandon eller gjorde andra aktiviteter med högre privilegier i systemet. Allt detta sker via det grafiska gränssnittet på maskinerna för att säkerställa att det ser realistiskt ut i loggar. Figur 5 visar hur det ser ut på skrivborden för några simulerade användare som är igång.



Figur 5. Skärmbilder från simulerade användare i arbete.

Processen och verktygen för att simulera användare i Crate har utvecklats betydligt under projekten. Instrumenteringen av simuleringsverktygen kräver att det skapas instruktionslistor som anger när simulerade användare ska göra olika aktioner, exempelvis när de ska skicka e-post eller öppna en viss fil. För att göra detta behöver det dels skapas en lista på aktiviteter över tid (t.ex. när e-post ska skickas), dels sättas parametrar för aktioner (t.ex. vem e-post ska skickas till). Under projekten har dessa verktyg utvecklats så att det finns en uppsättning olika sätt att schemalägga aktioner på och så att parametrar sätts automatiskt med hjälp av anrop mot Crates programmeringsgränssnitt. Instruktionslistorna används för att skicka enskilda instruktioner till virtuella maskiner där en exekverbar kod lyssnar och utför aktioner som att skicka e-post. Denna kod var tidigare skriven i skriptspråket AutoIt där knapptryckningar användes för att simulera användaraktivitet. På grund av brister i tillståndshanteringen var det inte ovanligt att de simulerade användarna fastnade i dialoger en bit in i försöken och inte lyckades gör det instruktionerna angav. Koden är numera skriven i C#, använder även musen, kan utföra fler aktiviteter och har tydliga datastrukturer för

hantering av instruktioner. Numera kan de simulerade användarna köra i veckor i sträck utan att hamna i tillstånd de inte kan hantera.

Även om verktygen för simulering av användare utvecklats betydligt under projekten är simuleringsförmågan långt ifrån tillräcklig för att representera alla problem som finns i verkligheten. Som en del av forskningen i projektet gjordes en enkätstudie (Landauer m.fl., 2025) ihop med kollegor från Austrian Institute of Technology (AIT) för att kartlägga behoven av användarsimulering. I enkäten beskrevs händelser som kan ske i IT-system och praktiserande experter inom logganalys frågades om händelserna var sådana som kan förväxlas med hotaktörers aktivitet. Studien identifierade sjutton typer av händelser som skapar problem i logganalysarbete. Det var stor variation mellan vilka aktiviteter som experterna hade problem med och vilka händelser som orsakar falsklarm tycks skilja mycket mellan organisationer.

Endast två av de sjutton typerna av aktiviteter som identifierades i studien har använts för att försvåra cyberförsvarsarbetet i testerna under projekten: systemadministratörer som kör kommandon på maskiner och systemadministratörer som öppnar fjärrmaskiner. Användarsimuleringarna domineras av mer reguljärt brus där det skickas e-post, öppnas filer och surfas på webbsidor. Dessa kan förvisso verka suspekta i loggar ibland, exempelvis för att en wordfil med makro öppnas, men är i de flesta fall något som inte träffar signaturer i detektionslösningar. Arbete med att simulera fler sådana svårbedömda beteenden pågår. Detta gäller bland annat legitimt användande av verktyg som Certutil och SSH (Daniel, 2025).

3.3 Hotaktörer och angrepp

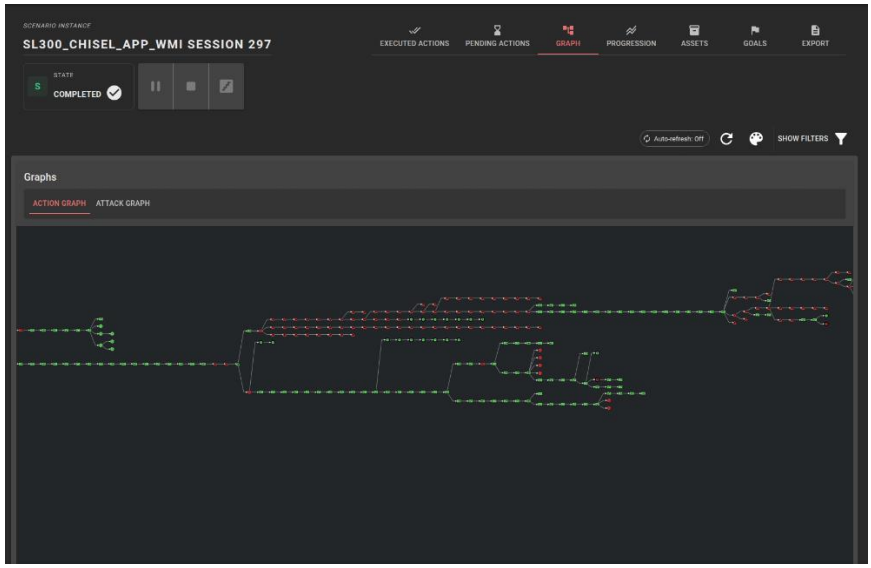
Cyberangreppen är centrala i tester och demonstrationer av hur cyberangrepp kan hanteras. Likt simuleringen av användare finns här möjlighet att påverka hur svårt incidenthanteringsarbetet ska vara, exempelvis genom att välja angrepp som kan förväxlas med användares aktioner eller syns dåligt i loggar. I projekten har verktyget Lore använts för att simulera hotaktörer (Holm, 2023) (Holm & Sommestad, 2025). Lore använder två konfigurationsfiler: en generell profil för hotaktören (t.ex. aktioner den föredrar) och en scenariobeskrivning för hur hotaktören ska användas i en viss simulering (t.ex. vilka IP-adresser som ska prioriteras som mål i angrepp). I projekten har processen för att skapa simuleringarna varit i stil med detta:

1. Projektmedlemmar som ska säkerställa relevans producerar en beskrivning av angreppet på papper eller på tavla i möte. Denna kan exempelvis bestå av hopp som ska ske i nätverket med olika attackteknikter.

2. Baserat på de alternativ som Lore erbjuder, eller enkelt kan utökas med, presenteras lösningsförslag. Förslaget kan exempelvis vara att en viss nätverkssårbarhet introduceras i nätverket och prioriteras av Lore.
3. Justeringar görs av Lores konfiguration och av nätverket i Crate. Exempelvis kan ett nytt installationsskript skapas i Crate och ett visst angrepp i Lore-profilen kan prioriteras. Ibland integreras nya verktyg i Lore för att täcka upp för saknad förmåga.
4. Människor kontrollerar att angreppet fungerar som det var tänkt och om det ger önskat avtryck i loggar mm. Om svaret är nej startas processen om för att justera.

Lore var ett kapabelt verktyg redan innan dessa projekt började. Cyberangrepp är dock invecklade och Lore använder en uppsjö av verktyg av varierande kvalitet. Det har därför ändå uppstått en rad problem under projektet som inneburit att problem i processen som beskrivits ovan kan ta timmar eller dagar att felsöka och korrigera. Exempelvis har projektet inneburit nya angrepp som involverar pivotering i nätverk med restriktiva brandväggar, vilka krävt tillägg till den verktygsuppsättning Lore använder. Återkommande handhavandefel har också resulterat i förbättrade sätt att konfigurera Lore.

Eftersom Lore ger autentiska angreppskedjor har det varit värt den tid det tagit att använda och justera verktyget. Under en typisk exekvering på två timmar gör Lore cirka 300 aktioner (t.ex. skanningar, nätverksangrepp eller skalkommandon för att samla information). En illustration över ”aktionsgrafen” i Lore från en körning visas i figur 6. I denna är varje grön prick en lyckad aktion och varje röd är en misslyckad aktion. Att manuellt skapa skript för dessa aktioner skulle ta väldigt lång tid. Dessutom ger Lores dynamiska beslutsförmåga möjlighet att se hur försvarsåtgärder påverkar angreppens progress. Det skulle inte vara meningsfullt om alla angreppssteg var förutbestämda.



Figur 6. Vy från Lore som visar aktioner som utförs i olika trådar under en körning.

Utöver upptäckta och åtgärdade begränsningar i Lores förmåga att göra angrepp har arbete med diverse anpassningar och utökningar av Lore behövts i projekten. Ett sådant arbete är att vidareutveckla koden som väljer aktioner så att de blir lika hotaktören APT29 (Granstedt, 2024). Ett annat är försök att producera rapporter om Lores aktivitet för att förbättra analysmöjligheterna kopplat till intrångsdetektion. Ett tredje är att försöka automatiskt jämföra loggarna från Lore med de från SIEM-system och koppla ihop dem baserat på olika attribut. Alla dessa är arbeten som i olika omfattning och kontext kommer att fortsätta efter projekten.

3.4 Simuleringarnas realism och relevans

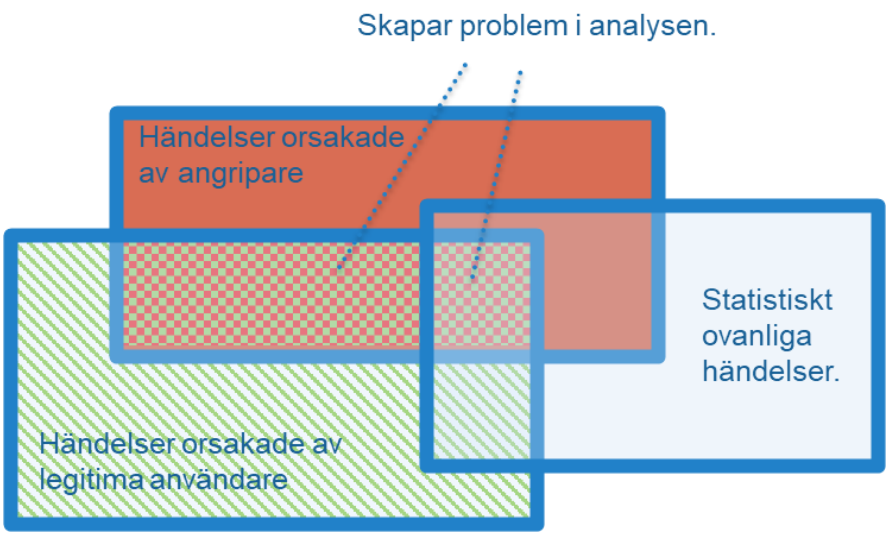
Föga förvånande har felen i olika delar av experimenten inneburit att många episoder kraschat eller resulterat i missvisande resultat. Utöver felen som nämnts ovan har problem uppstått med andra delar av episodexekveringen som har potential att ge missvisande resultat. Exempelvis har diskar blivit fulla med temporära filer över tid varefter filkopieringen i loggextraheringen har slutat fungera. Vår bedömning är att sådana misstag upptäckts och inte färgat resultatet annat än att de försenat forskningen.

Under projektet har felhanteringen successivt förbättrats och numera hanteras fel i de flesta delar med omstarter och liknande för att fel i en episod inte ska följa med till nästkommande episoder. Vår till höst 2025 har över 1000 episoder körts, där de flesta är 3 timmar långa och resterande dagar eller veckor. Cirka hälften

av dessa är kompletta. Mot slutet av projekten har det körts episoder gång på gång under veckor utan synbara problem.

Även om körningen av episoder i Crate mot slutet av projekten varit smärtfri så finns det stor förbättringspotential i vad som simuleras. Det har samtidigt gjorts en rad medvetna avkall på realism för att kunna simulera angrepp med publikt tillgängliga verktyg och inte behöva sekretessklassa arbetet. Exempelvis har antivirus stängts av eftersom de annars hade stoppat den Meterpreter-bakdörr som användes för att köra kod på övertagna maskiner. Det har också kompromissats med äktheten i flera delar av nätverken. Bland annat körs inget riktigt ledningssystem i EU-projektets militära cybermiljö. Istället har motsvarande dataflöden skapats med pythonkod och en webbapplikation. Vissa delar är alltså konceptuellt rimliga men tekniskt annorlunda än angrepp som kan förväntas i verkliga militära cybermiljöer.

Utöver dessa avkall på realism har en del beslut tagits för att skapa lämplig svårighetsnivå i scenariona. Som diskuterats ovan, och illustreras i figur 7, kan incidenthanteringsuppgiften göras mer eller mindre svår genom att justera vilka aktioner som utförs i simuleringarna. I alla projekt har det gjorts intrimningsarbete för att säkerställa att händelserna som angriparen ger upphov till syns i loggar. I de nationella projekten har det också säkerställts att det finns tydliga larm för sådant som endast angriparen gör. Därtill är de simulerade användarna rudimentära och utmanar inte logganalysarbetet på samma sätt som verkliga systemanvändare eller systemadministratörer kan göra. Scenariona i simuleringarna är alltså enklare än vad som kan förväntas i verkliga nätverk.



Figur 7. Svårighetsnivån i analysen och vilka händelser som simuleras.

I allmänhet är dataset som används i cybersäkerhetsforskning är behäftade med flera väl kända problem (Landauer m.fl., 2024)(Liu m.fl., 2025)(Flood m.fl., 2024). Vi bedömer att den data som producerats fört forskningen framåt och står sig väl i jämförelse med den data som typiskt används i publicerade studier. Simuleringarna är på vissa sätt bättre än de som vanligtvis används:

- De innehåller angrepp som sker i flera sammanhängande steg.
- Både maskinloggar och nätverksloggar sparas.
- Användare simuleras på ett ovanligt autentiskt och genomtänkt sätt.
- Försvarslösningar har kunnat prövas mot dynamiska angrepp.

På andra sätt är de sämre. Framförallt har de dataset som producerats i projekten en oprecis markering av vilka loggar som orsakats av angrepp respektive legitim aktivitet. Detta beror på att det är svårt att kartlägga vilka händelser som orsakas av angrepp när angreppen är mer invecklade och loggarna mer heterogena än i andra simuleringar samt när simulerade användare finns.

Som test av försvarslösningar finns också begränsningar. Eftersom simuleringen gjort flera avkall på realism, lagt mycket till rätta för incidenthanteringen samt skapat ett förhållandevis enkelt cyberförsvarsscenario, kan episoderna ses som ett ensidigt test av tekniska lösningar för automatisk incidenthantering. De som misslyckas med att hantera incidenterna i våra simulerade scenarion kommer definitivt inte klara verkliga cyberangrepp i driftsatta miljöer. De som klarar våra testscenarion behöver däremot inte fungera i driftsatta miljöer.

4 Hur det gick

När denna rapport skrivs har inte alla projekt avslutats och vissa tester återstår. På en övergripande nivå är det dock tydligt vilka lösningar som har potential att fungera och vilka som inte kommer fungera i komplexa driftsatta system. I tabell 4 ges en översikt med OODA-loopen som utgångspunkt. Detaljer ges i avsnitten som följer.

Tabell 4. Sammanfattning av resultat som FOI tagit fram eller tagit del av i projekten.

Steg	**Lösning som provats**	**Resultat av tester**
Observera	Logginsamling enligt industristandard med olika profiler för sensorer etc.	Fungerar bra och tillfredsställer behoven som funnits i projekten.
	Anomalidetektion för data med nätverksflöden.	Fungerar dåligt i realtid under demonstrationer.
	Anomalidetektion genom regler intrimmade med en normalbild.	Lovande och larmar på uppenbara avvikelser med acceptabel precision.
	Insamling av statisk information om terrängen via logginsamlingskedjan.	Fungerade acceptabelt och gav exempelvis grundläggande nätverksinformation.
Orientera	Modeller över tänkbara angrepp utryckta i coreLang.	Ger schematiska resultat om tänkbara framtida steg i angreppet baserat på information om nätverket.
	Manuellt angivna prioriteter för olika maskiner uttryckta i skala 1–5 på säkerhetsattributen CIA.	Går ofta att uttrycka och fungerar som indata för maskininlärningsmodeller om beslut.
	Kunskapsgrafer som binder ihop observationer med bl.a. nätverksinformation.	Fungerar delvis under demonstrationer.
	Gruppering av liknande observationer.	Tycks fungera offline men klarar inte att producera resultat i realtid.
Besluta	Träning på modeller över angrepp med grafbaserade neurala nätverk.	Fungerar ganska bra och lär sig ofta att stänga av övertagna maskiner.
	Alltid isolera det som ser övertaget ut.	Fungerar som förväntat, dvs. ganska bra.
	Träning i digital tvilling i annan enklare testbädd för prediktion.	Svårbedömt då det är dyrt och ej har demonstrerats.
Agera	Ett OpenC2-kompatibelt system för att utföra åtgärder.	Har fungerat bra men saknar avancerade åtgärder för exempelvis vilseledning.
	Justeringar av Wazuhs inbyggda aktiva åtgärder.	Har fungerat bra med grundläggande åtgärder som att stänga av maskiner, men kan bli överbelastat vid angrepp.

4.1 Observationer

Inget av projekten har försökt göra framsteg inom logginsamlingssystem eller kedjor för att samla in loggar. Inte heller har försök gjorts att förbättra konfigurationer för verktyg som producerar loggar. Istället har industristandard varit utgångspunkten. Ibland har vissa kalibreringar gjorts för att passa analyserna i senare steg. För FOI har detta resulterat i mer kunskap om hur logginsamling går till i praktiken, vad som är svårt med det samt en uppsjö återanvändningsbara installationsskript för logginsamlingslösningar.

Det vanligaste sättet att skapa raffinerade observationer i AInception har varit att försöka använda trafikflöden och observera anomalier. Detta har gjorts genom att först observera en normalbild och sedan under tester med angrepp skapa loggar om något avviker från normalbilden. Ett problem med denna ansats är att det i driftsatta system kan vara svårt att veta om normalbilden innehåller angrepp. Bortsett från detta är resultaten som demonstrerats i projektet svaga för denna typ av lösningar. Vissa lösningar är så analystunga att de inte kan köras i realtid och inga lösningar tycks ge larm för sådant som ingår i angreppsscenariot trots att scenariot lägger till IP-adresser och portar som inte ingår i normalbilden. Det kan här noteras att forskarna som utvecklat verktygen rapporterat en precision på över 50 procent när de använder tidserieanalyser och över 85 procent för neurala nätverk som tittar på nätverksflöden. Det kan dock inte visas i realtid under demonstrationerna trots att det är samma typ av scenarion som utspelar sig. Kanske är detta ett resultat av att det går väldigt mycket trafik i nätverket och att sådant som varierar mellan episoder förvirrar algoritmerna.

En annan teknik som testats i AInception är den som används av AMiner (Landauer m.fl., 2023), där acceptabla gränsvärden sätts för olika typer av systemhändelser. Detta kan exempelvis inkludera listor på användare som bör förekomma i relevanta loggar, listor på IP-adresser som bör synas i loggar samt rimliga frekvenser av förfrågningar mot webbtjänster. Att skapa sådana regler kräver visst arbete. När forskarna i projektet trimmat in det på träningsdata nås cirka 85 procent precision och cirka 40 procent av angreppsstegen ger ett larm. Det har också fungerat i demonstrationer där det exempelvis larmas på fjärrstyrningstrafik eftersom den involverar nya IP-adresser och liknande.

I de nationella projekten söktes inga direkta framsteg i observationsfasen. Istället användes och kalibrerades vanliga signaturer. I slutändan genererade scenariona cirka 20 typer av larm för observationer och tester. En förhållandevis innovativ lösning användes för att ge observationer av terrängen som behövdes: verktyget Osquery konfigurerades för att regelbundet skicka information om tillståndet i maskiner (t.ex. IP-adresser och användare), medan sårbarhetsskannrar som kom med Wazuh användes för att ge information om sårbarheter och mjukvaror.

4.2 Orientering

Orienteringen i projekten görs med hjälp av observationer (t.ex. larm om säkerhetshändelser), information som finns på förhand (t.ex. topologikartor) samt regler och antaganden (t.ex. från maskininlärningsmodeller). Som en del av detta behöver det avgöras vad en observation rör för tillgångar och om observationen ska ses som ett riktigt hot eller ett falsklarm.

En ansats som prövats i alla projekten, och som var huvudnumret i SAC3S, är att använda en variant av modelleringsspråket coreLang för att skapa modeller över tänkbara angrepp. Denna ansats bygger på att det finns den nätverksinformation som modelleringsspråket använder för att göra antaganden om angrepp. Till exempel har modellerna använts för att prediktera vad angriparen kan nå från olika punkter i nätverken och vad möjliga nästa steg kan vara när en viss maskin har ansetts komprometterad. Detta kan då, som i SAC3S, illustreras i en nätverkskarta. Modellerna som skapats är på övergripande nivå och de analyser som gjorts tycks bli korrekta under förutsättning att indata är korrekt. För att visa vilka skyddsvärden som står på spel används manuellt angivna prioriteter för konfidentialitet, riktighet och tillgänglighet för maskiner i nätverket. Detta relativt enkla och vedertagna sätt att utrycka säkerhetskrav har visat sig tillräckligt för att träna maskininlärningsmodeller så att de kan fatta beslut som upplevs rimliga.

I AInception har orsaksanalyser till larm och händelser varit det centrala och fokus har varit på att sammanställa information som kan presenteras för en mänsklig operatör. Konceptet *kunskapsgraf* har använts för detta. En kunskapsgraf kopplar ihop entiteter genom relationer. I projektet har kunskapsgraferna bland annat visat att ett larm pekar på en dator och hänger ihop med en underrättelserapport. Utöver detta har verktyg skapats för att förklara larm som inträffat av anomalidetektionslösningar:

- Det har skapats verktyg som visar vilken data (t.ex. loggposter) som orsakat avvikelselarm.
- Det har prövats att ge förklaringar med stora språkmodeller.
- Det har gjorts statistiska analyser för att gruppera larm eller händelser som är lika.

Gruppering av händelser och larm har bland annat jämfört inkommande larm med sådana som är förknippade med kända angreppssekvenser (t.ex. att en webbtjänst skannas). Detta sätt att aggregera och korrelera strömmen av larm tycks lovande men har tyvärr inte fungerat i realtid när det testats.

I AInception har det funnits flera interaktioner mellan partners som illustrerar deltagarnas skiftande förväntningar på vad som är möjligt och rimligt. Exempelvis uttrycktes det klagomål på den aggregering av larm som erhölls med hjälp av *Graph Matching Network*. De som skulle använda resultatet av

aggregeringen i kunskapsgrafer ifrågasatte om det fungerade eftersom det fanns över tusen larmgrupper i data. De som gjort aggregeringen konstaterade att det fanns miljoner loggposter innan deras modul aggregerat loggposterna till grupper.

Att tolka händelser och förstå deras säkerhetsimplikationer är erkänt svårt. Det finns ansatser i projekten som har potential men ingen har producerat en lösning som löser alla delar på ett enhetligt sätt. En reflektion från deltagare på demonstrationer har varit att det var en trevlig visualisering som gjordes av larm kopplat till en nätverkskarta. Sådana relativt enkla sätt att kontextualisera händelser kan vara värda att gå vidare med om det är en mänsklig operatör som ska fatta beslut. De ansatser som gjorts med att gruppera larm och loggar tycks också meningsfulla och lovande.

4.3 Beslut

I alla projekt har beslutsrymden varit förhållandevis enkel. Förutom att inte göra någonting har följande alternativ varit aktuella: skicka e-post, stänga av maskin, isolera maskin i brandväggar, låsa ute användare och stänga av enskild process.

De nationella projekten har lagt mycket av tiden på att skapa modeller som kan fatta beslut om åtgärder. Dessa har utgått från de attackgrafer som skapas av modellerna i coreLang och skapat grafneurala nätverk som för-tränats på olika scenarier. Modellerna har alltså använts för att simulera hur en angripare skulle kunna ta sig runt i nätverket (t.ex. ta sig mellan maskiner) och hur olika åtgärder (t.ex. avstängning av maskiner) skulle påverka angriparens progress och de säkerhetsvärden som är uttryckta på skalan 1–5 i tre dimensioner. Att träna modellerna på ett nätverk av den storlek som användes i testerna (ca 30 maskiner) tar några dagar på en kraftfull dator. Efter träning kan det ställas frågor om vad det bästa beslutet är givet nuvarande tillstånd (t.ex. övertagna maskiner). Som svar ges åtgärder och deras förväntade värde om de utförs. Detta värde kan vara negativt eller positivt och kopplar till värderingen av de säkerhetsvärden som uttryckts. De som är mest positiva ska enligt modellen vara fördelaktiga. I tester och demonstrationer har denna maskininlärningslösning ställts mot en beslutsmodell benämnd *whack-a-mole*, där en enkel beslutsregel används: stäng av eller isolera maskiner som verkar övertagna. De framräknade värdena över vad som är optimalt överensstämmer ofta med denna enkla beslutsregel. Det är i sig inte särskilt förvånande. I NIST:s gamla guide för incidenthantering står exempelvis (Cichonski m.fl., 2012, s. 35):

> Most incidents require containment, so that is an important consideration early in the course of handling each incident. Containment provides time for developing a tailored remediation strategy. An essential part of containment is decision-making (e.g., shut down a system, disconnect it from a network, disable certain functions).

I AInception har fokus varit på att presentera information i ett webbaserat gränssnitt och inte att automatiskt utföra den fullständiga OODA-loopen. Försök har gjorts att simulera cyberangrepp i ett labb på förhand för att sedan använda förstärkningsinlärning för att testa vilka åtgärder som stoppar olika angrepp (Jaber m.fl., 2025). Inlärningsprocessen kräver dock omfattande förberedelser och den har inte kunnat demonstreras live i verktygskedjan. Författarnas bedömning är att metoden är opraktisk och inte användbar i driftsatta nätverk. En annan ansats som gjort i EU-projektet är att skapa önskemål med satslogik som bland annat resonerar om huruvida en åtgärd stoppar en aktion. Denna logik var i demonstrationer bristfällig och det kan exempelvis ges förslag om att isolera maskiner från varandra trots att det inte finns skydd mellan dem som kan utföra isoleringen.

4.4 Agerande

När åtgärder beslutats behöver de också utföras. I verktygen som skapats finns både färdiga rekommendationer (beslut) som en användare kan omsätta och sådant som beslutas och utförs automatiskt utan interaktion med användare.

I EU-projektet har en nederländsk forskargrupp utvecklat ett verktyg kallat SOARCA och gjort det publikt tillgängligt. Detta verktyg skapar konkreta instruktioner (så kallade *playbooks*) enligt två standarder: CACAO och OpenC2. Båda är gjorda för att konsumeras av SOAR-system. Projektet har demonstrerat hur SOARCA skapar instruktioner för följande D3FEND-tekniker:

- DNS Allowlisting
- Executable Denylisting
- File Removal
- Network Traffic Filtering
- Outbound Traffic Filtering
- Process Termination
- Restore Software
- Software Update

Av dessa har isolering via brandvägg (Network Traffic Filtering) visats i demonstrationer. Dessutom finns det stöd för att automatiskt informera en operatör via e-post om att en åtgärd kan behövas.

I de nationella projekten används den förmåga som kommer med SIEM-systemet Wazuh för att köra kod på maskiner som övervakas. Utveckling och testning har utförts för åtgärder som gör:

- Account locking: deaktiverar en angiven användare som anges i windowsmaskiner, windowsdomäner eller linuxmaskiner.
- Network Traffic Filtering: isolerar en angiven IP-adress med windows inbyggda brandvägg eller med iptables.

- Host Shutdown: stänger av maskinen.

Samtliga åtgärder har testats i de nationella projekten. Åtgärder som isolerar användare har inte fungerat särskilt bra då coreLang-modellerna inte representerat behörigheter med hög upplösning och det exempelvis blivit fel när en windowsmiljö instruerats att stänga ner systemkontot *LocalSystem* som behövs för basala funktioner. Experiment har därför främst stängt av maskiner och isolerat maskiner via brandväggar. Dessa åtgärder har ibland fungerat dåligt för att kommandon inte når fram till maskinerna som förväntat. Detta verkar bero på att maskiners logg-agenter blir överbelastade när det produceras mycket loggar, vilket dessvärre sammanfaller med att det sker angrepp.

De flesta åtgärder som använts och testats förutsätter behörighet i systemen. Exempelvis behövs att kommandon enkelt kan skickas till brandväggar, vilket inte är självklart i system omgärdade av restriktiva säkerhetsregler. Att åtgärder funkat i labbmiljö betyder alltså inte att de är enkla att använda till driftsatta system i en försvarskontext.

5 Slutsatser och rekommendationer

Det finns många mogna tekniska lösningar för logginsamling. Några av dessa har testats i projekten. En erfarenhet av att jobba med detta är att det i praktiken ofta är svårt att få hela kedjan att fungera som den ska. Exempelvis blir en detektionsregel verkningslös om den är beroende av data eller fält som aldrig loggas i systemen som övervakas. Insamling av rätt data var ett återkommande problem under projekten. Forskare undviker ofta detta praktiska arbete med logginsamling och har istället idealiserade testfall där all data är fint ordnad från början eller enkel att samla in (som nätverkstrafik). I dessa projekt gjordes omfattande logginsamling med verktyg som är utbredda bland praktiker. Det togs dock andra genvägar. Framförallt antog projekten att det fanns behörighet att utföra säkerhetsrelaterade förändringar i system från en central plats.

- ⇨ Forskare inom detta område har sällan löst praktiska hinder som kan finnas för observationer eller utförande av åtgärder i driftsatta system.

EU-projektet genomförde en demonstration som inte hanterade den simulerade incidenten på ett meningsfullt sätt, trots omfattande kalibrering mot hotscenariot. Många komponenter lämnades också utanför demonstrationen. Vissa komponenter fungerade nämligen inte i realtid eftersom mängden händelser blir för stor eller andra komponenter gav analysresultat som inte var korrekta. De nationella projekten har visat en lösning som försvarar mot angrepp på ett effektivt sätt, men under förutsättning att detektionerna är väl kalibrerade på förhand. Dessutom verkar lösningen med grafbaserade neurala nätverk som skapats inte vara mycket bättre än enkla regelbaserade lösningar som stänger ner de maskiner som orsakar säkerhetslarmen.

- ⇨ Lita inte på forskare inom detta område som säger att de löst problemet med att automatisera incidenthantering.

Orsakerna till att automatiserad incidenthantering fungerar dåligt tycks främst handla om att olika fel (t.ex. falsklarm) introduceras i stegen för observationer och orientering. Det har lagts ett betydande arbete med att korrigera de delarna, och det gäller framförallt i EU-projektet. Inga av de försök som vi har varit involverade i har lagt fokus på att aktivt söka ytterligare information för att bekräfta eller avfärda misstankar om pågående angrepp, på det sätt som mänskliga operatörer gör när misstänkta händelser sker. Exempelvis har verktygen inte utfört åtgärder som verifierar larm eller slår på ytterligare loggning för att samla in mer information.

- ⇨ Framtida forskning bör i högre grad inriktas på diagnos av säkerhetstillstånd, i linje med hur mänskliga logganalytiker arbetar.

Det kan konstateras att plattformen Crate varit mycket användbar för både skapande av dataset att träna på och för test av prototyper. Plattformen har också utvecklats betydligt under projektet och är numera mer kapabel än när projekten startade. En extern bekräftelse på detta är att FOI höll slutdemonstrationerna i EU-projektet trots att andra aktörer med liknande anläggningar fanns med i projektet.

- ⇨ Prototyper på automatisk incidenthantering kan med fördel testas i Crate eller liknande plattformar innan försök görs i driftsatta system.

6 Referenser

Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). *Computer security incident handling guide* (NIST Special Publication 800-61 Rev. 2). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-61r2

Cyberförsvarsledningen. (2024). Doktrinansats Cyberförsvar. FM2024-21569. Försvarsmakten.

Daniel, P. (2025). Triggering False Alarms in Computer Networks: Evaluation of an Intrusion Detection System's Performance in Accurately Distinguishing Attacks from Regular Computer Activity [Master thesis]. KTH Royal Institute of technology.

Eckhoff, M. W., Flydal, P. M., Peters, S., Eian, M., Halvorsen, J., Mavroeidis, V., & Grov, G. (2026). A graph-based approach to alert contextualisation in security operations centres. In S. K. Cha & J. Park (Eds.), Information Security. ISC 2025. Lecture Notes in Computer Science (Vol. 16186, pp. 411–430). Springer, Cham. https://doi.org/10.1007/978-3-032-08124-7_24

Eckhoff, M. W., Halvorsen, J., Hansen, B. J., Eian, M., Mavroeidis, V., Chetwyn, R. A., Skjøtskift, G., & Grov, G. (2025). Experimenting with Neurosymbolic Artificial Intelligence for Defending Against Cyber Attacks. Neurosymbolic Artificial Intelligence, 1. https://doi.org/10.1177/29498732251377352

Ekstorm, B., & Sethi, R. (2024). DCO-konceptet. FM2024-9629. Försvarsmakten.

Flood, R., Engelen, G., Aspinall, D., & Desmet, L. (2024). Bad Design Smells in Benchmark NIDS Datasets. Proceedings - 9th IEEE European Symposium on Security and Privacy, Euro S and P 2024, 658–675. https://doi.org/10.1109/EuroSP60621.2024.00042

Fuchsberger, A. (2005). Intrusion detection systems and intrusion prevention systems. Information Security Technical Report, 10(3), 134–139. https://doi.org/10.1016/j.istr.2005.08.001

Granstedt, E. (2024). Evaluating the Realism and Effectiveness of Automated APT Emulation in Cybersecurity Training : A Lore Case Study [Master thesis]. KTH Royal Institute of technology.

Holm, H. (2023). Lore a Red Team Emulation Tool. IEEE Transactions on Dependable and Secure Computing, 20(2), 1596–1608. https://doi.org/10.1109/TDSC.2022.3160792

Holm, H., & Sommestad, T. (2025). Realistic and balanced automated threat emulation. Computers & Security, 151, 104351. https://doi.org/10.1016/j.cose.2025.104351

Husák, M., & Cermak, M. (2022). SoK: Applications and Challenges of using Recommender Systems in Cybersecurity Incident Handling and Response. I ACM International Conference Proceeding Series (Vol. 1, Nummer 1). Association for Computing Machinery. https://doi.org/10.1145/3538969.3538981

Husák, M., Sadlek, L., Špaček, S., Laštovička, M., Javorník, M., & Komárková, J. (2022). CRUSOE: A toolset for cyber situational awareness and decision support in incident handling. Computers and Security, 115(January). https://doi.org/10.1016/j.cose.2022.102609

Jaber, A. (2024). Transforming Cybersecurity Dynamics: Enhanced Self-Play Reinforcement Learning in Intrusion Detection and Prevention System. 2024 IEEE International Systems Conference (SysCon), 1–8. https://doi.org/10.1109/SysCon61195.2024.10553626

Jaber, A., Endregard, M., Mancini, F., & Grov, G. (2025). Mission-Aware Cyber Incident Response Generation Using Reinforcement Learning. International Conference on Military Communication and Information Systems, ICMCIS, 2025. https://doi.org/10.1109/ICMCIS64378.2025.11047892

Jalalvand, F., Baruwal Chhetri, M., Nepal, S., & Paris, C. (2024). Alert Prioritisation in Security Operations Centres: A Systematic Survey on Criteria and Methods. ACM Computing Surveys, 57(2). https://doi.org/10.1145/3695462

Kaloroumakis, P. E., & Smith, M. J. (2020). Toward a Knowledge Graph of Cybersecurity Countermeasures. Mitre, https://apps.dtic.mil/sti/citations/trecms/AD1156977

Kampourakis, V., Gkioulos, V., & Katsikas, S. (2025). A step-by-step definition of a reference architecture for cyber ranges. Journal of Information Security and Applications, 88. https://doi.org/10.1016/j.jisa.2024.103917

Karlzén, H. (2021). Honungsfällor: Att vilseleda och studera cyberangripare (FOI-R--5217--SE). Totalförsvarets forskningsinstitut.

Karlzén, H., & Sommestad, T. (2023). Automatic incident response solutions: a review of proposed solutions' input and output. Proceedings of the 18th International Conference on Availability, Reliability and Security, 1–9. https://doi.org/10.1145/3600160.3605066

Katsikeas, S., Buhaiu, A., Ekstedt, M., Afzal, Z., Hacks, S., & Mukherjee, P. (2024). Development and validation of coreLang: A threat modeling

language for the ICT domain. Computers and Security, 146. https://doi.org/10.1016/j.cose.2024.104057

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1). https://doi.org/10.1186/s42400-019-0038-7

Kotenko, I., Gaifulina, D., & Zelichenok, I. (2022). Systematic Literature Review of Security Event Correlation Methods. IEEE Access, 10, 43387–43420. https://doi.org/10.1109/ACCESS.2022.3168976

Landauer, M., Skopik, F., & Wurzenberger, M. (2024). A Critical Review of Common Log Data Sets Used for Evaluation of Sequence-Based Anomaly Detection Techniques. Proceedings of the ACM on Software Engineering, 1(FSE), 1354–1375. https://doi.org/10.1145/3660768

Landauer, M., Skopik, F., Wurzenberger, M., Sommestad, T., & Karlzén, H. (2025). Benign User Activities that Trigger False Positives in Intrusion Detection Systems: An Expert Survey. Lecture Notes in Computer Science, 15995 LNCS, 25–43. https://doi.org/10.1007/978-3-032-00633-2_2

Landauer, M., Wurzenberger, M., Skopik, F., Hotwagner, W., & Höld, G. (2023). AMiner: A Modular Log Data Analysis Pipeline for Anomaly-based Intrusion Detection. Digital Threats: Research and Practice, 4(1). https://doi.org/10.1145/3567675

Liu, J., Inam, M. A., Goyal, A., Riddle, A., Westfall, K., & Bates, A. (2025). What We Talk About When We Talk About Logs: Understanding the Effects of Dataset Quality on Endpoint Threat Detection Research. 2025 IEEE Symposium on Security and Privacy (SP), 112–129. https://doi.org/10.1109/SP61157.2025.00112

Lundholm, K., Sommestad, T., Persson, M., Gustafsson, T., & Hunstad, A. (2011). Detektion av IT-attacker Övningsuppställning och insamlad data (FOI-R--3342--SE). Totalförsvarets forskningsinstitut.

Maldonado, I., Meeuwissen, E., de Haan, P., & van der Mei, R. (2025). Telosian: Reducing False Positives in Real-Time Cyber Anomaly Detection by Fast Adaptation to Concept Drift. Proceedings of the 11th International Conference on Information Systems Security and Privacy, 84–97. https://doi.org/10.5220/0013320500003899

Mavroeidis, V., & Brule, J. (2020). A nonproprietary language for the command and control of cyber defenses – OpenC2. Computers and Security, 97, 101999. https://doi.org/10.1016/j.cose.2020.101999

Nunez, J., & Livingstone, D. (2025). Hype Cycle for Security Operations, 2025. Gartner.

Papadaki, M., & Furnell, S. M. (2006). Achieving automated intrusion response: A prototype implementation. Information Management and Computer Security, 14(3), 235–251. https://doi.org/10.1108/09685220610670396

Sawilla, R. E., & Wiemer, D. J. (2011). Automated computer network defence technology demonstration project (ARMOUR TDP): Concept of operations, architecture, and integration framework. 2011 IEEE International Conference on Technologies for Homeland Security (HST), 167–172. https://doi.org/10.1109/THS.2011.6107865

Shaked, A., Cherdantseva, Y., Burnap, P., & Maynard, P. (2023). Operations-informed incident response playbooks. Computers & Security, 134, 103454. https://doi.org/10.1016/j.cose.2023.103454

Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., & Zhang, J. (2023). Cyber Threat Intelligence Mining for Proactive Cybersecurity Defense: A Survey and New Perspectives. IEEE Communications Surveys and Tutorials, 25(3), 1748–1774. https://doi.org/10.1109/COMST.2023.3273282

Wurzenberger, M., Höld, G., Landauer, M., & Skopik, F. (2024). Analysis of statistical properties of variables in log data for advanced anomaly detection in cyber security. Computers & Security, 137, 103631. https://doi.org/10.1016/j.cose.2023.103631

Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015). Intrusion detection and Big Heterogeneous Data: a Survey. Journal of Big Data, 2(1). https://doi.org/10.1186/s40537-015-0013-4

FOI
Totalförsvarets forskningsinstitut
164 90 Stockholm

Tel: 08-55 50 30 00
Fax: 08-55 50 31 00

www.foi.se