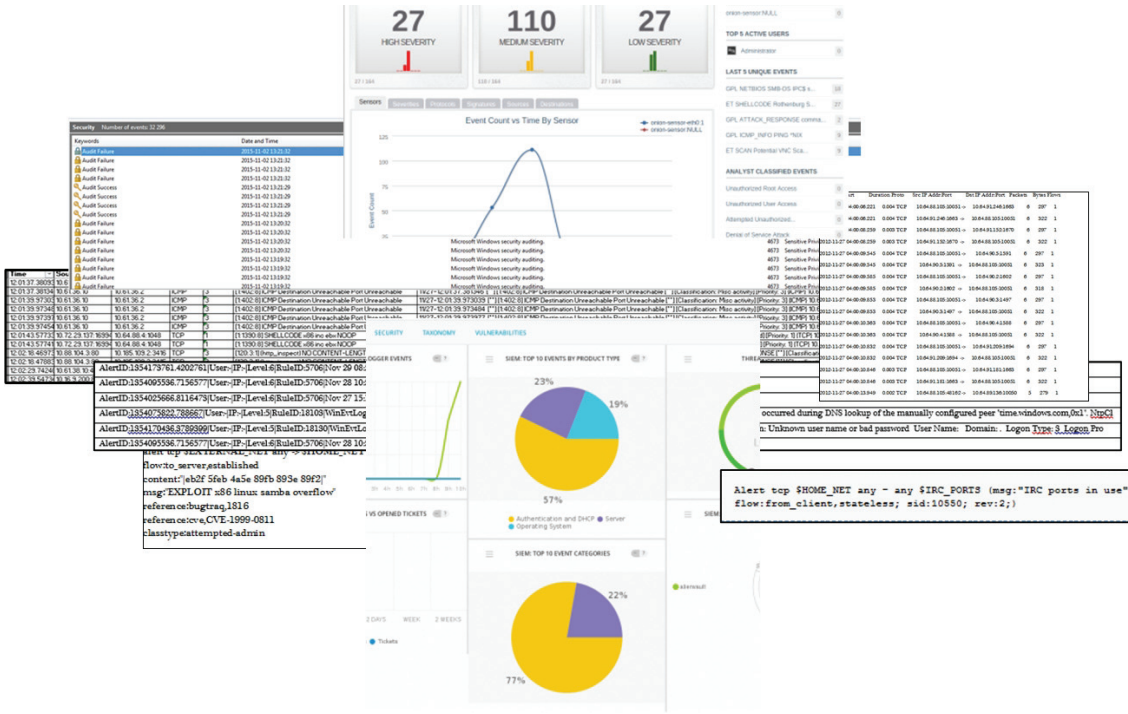
Variabler av vikt för förmågan att analysera cybersäkerhetsloggar

TEODOR SOMMESTAD OCH HANNES HOLM



FOI-R--4126--SE
ISSN 1650-1942 November 2015

Teodor Sommestad och Hannes Holm

Variabler av vikt för förmågan att analysera cybersäkerhetsloggar

Bild/Cover: Teodor Sommestad

Titel	Variabler av vikt för förmågan att analysera cybersäkerhetsloggar
Title	Variables of importance for the capability of analyzing cyber security logs
Rapportnr/Report no	FOI-R--4126--SE
Månad/Month	November
Utgivningsår/Year	2015
Antal sidor/Pages	54
ISSN	1650-1942
Kund/Customer	Försvarsmakten
Forskningsområde	4. Informationssäkerhet och kommunikation
FoT-område	Ledning och MSI
Projektnr/Project no	E36083
Godkänd av/Approved by	Christian Jönsson
Ansvarig avdelning	Informations- & aerosystem

Sammanfattning

Inom projektet *Övning och experiment för operativ förmåga i cybermiljön* kommer förmågan att analysera cybersäkerhetsloggar studeras. Denna rapport ger en allmän översikt över logganalys inom cybersäkerhet och identifierar variabler som förmodas ha stor betydelse för att göra detta framgångsrikt. På den översta nivån identifieras tre delar som är viktiga: insamling av information, automatisk analys och manuell analys. Variabler som bedöms som viktiga inom insamling av information är val och placering av sensorer, systemkartläggning och hotunderrättelser; inom automatisk analys bedöms de viktigaste variablerna vara analystypen, dess träffsäkerhet och dess prestanda (snabbhet); de mest betydelsefulla variablerna inom den manuella analysen bedöms vara arbetsplatsens utformning samt logganalytikerns kunskap, informationsresurser och behörigheter. Baserat på översikten identifieras ett antal forskningsfrågor som projektet kan försöka besvara.

Nyckelord: cybersäkerhet, IT-säkerhet, logganalys, intrångsdetektion, lägesbild, litteraturöversikt, mänskliga faktorer.

Summary

Within the project *Exercises and experiments for operational capability in the cyber environment*, studies will be performed of cyber security log analysis capability. This report provides a general overview of log analysis in cyber security and identifies variables presumed to be of great importance for its success. On the highest level of abstraction, three parts are identified as important: collection of information, automatic analysis and manual analysis. Variables believed to important for collection of information are the placement of sensors, information collected about the monitored system and threat intelligence; variables judged as most important for automatic analysis are the type of analysis performed, its accuracy and its computational performance; variables judged as important for the manual analysis are the workspace's design as well as the log analyst's cognitive ability, knowledge, system resources and information resources. Based on the overview, a number of research questions which the project may attempt to answer.

Keywords: cyber security, IT-security, log analysis, intrusion detection, situational awareness, literature review, human factors.

Innehållsförteckning

1 Inledning

Denna rapport har producerats inom projektet *Övning och experiment för operativ förmåga i cybermiljön* (ÖvExCy). Operativ förmåga i cybermiljön kan innebära många saker och kräva många delförmågor. ÖvExCy fokuserar på en av dessa delförmågor: förmågan att analysera loggar för att kunna fatta beslut kopplade till cybersäkerhet. I rapporten används ibland ord som loggar, loggposter, logganalys, lägesförståelse eller logganalytiker. I alla fall då detta görs är det i kontexten cybersäkerhet. Till exempel syftar logganalytiker i denna rapport på en logganalytiker som analyserar loggar från IT-system för att fatta beslut kopplat till cybersäkerhet, och inget annat. Vidare används begreppet cybersäkerhet som synonymt med IT-säkerhet och cybermiljön används som synonymt med IT-miljön.

Det första kapitlet presenterar bakgrunden till rapporten, ger en översikt över det område som studeras i projektet (logganalys inom cybersäkerhet) och ger en översikt av rapportens innehåll och struktur.

1.1 Bakgrund

Projektet ÖvExCy startade 2015 inom ramen för Försvarsmaktens anslag för forskning och teknik (FoT). Projektet syftar till att utveckla kunskap om metod, organisation och teknik kopplat till operativ förmåga i cybermiljön. Både kunskap om hur operativ förmåga bör tränas och värderas ska byggas upp. Dessutom ska projektet, när så är lämpligt och möjligt, föreslå hur den operativa förmågan inom cybermiljön kan förbättras.

Operativ förmåga i cybermiljön kräver ett stort antal delförmågor. Alla dessa kan inte studeras i ett forskningsprojekt av ÖvExCy:s storlek. ÖvExCy kommer därför fokusera på utvalda delar av den förmåga som Försvarsmakten har behov av. Efter diskussioner med berörda parter inom Försvarsmakten identifierades lägesbild inom cybermiljön som en viktig förmåga där forskning utförd av FOI skulle kunna resultera i värdefulla bidrag. Fokus lades på lägesbild i cybermiljön av två skäl:

1. Försvarsmakten initierar nu en central enhet som med hjälp av systemloggar ska skapa sig en lägesbild av cybermiljön inom Försvarsmakten. Ökad kunskap om lösningar och olika träningsmetoder rörande lägesbild i cybermiljön är därför lämpligt att bygga upp vid denna tidpunkt.

2. FOI har sedan tidigare en övnings- och testanläggning vid namn CRATE[1] som är lämplig att använda för tester och träning kopplat till lägesbild i cybermiljön.

Under de tre år som ÖvExCy pågår kommer FOI tillsammans med den centrala logganalysenheten i Försvarsmakten använda CRATE för att studera frågor kopplade till logganalys i cybermiljön och ta fram lösningar för träning och övning inom området. I detta ingår bland annat studier av vilka faktorer som är av betydelse för att på ett effektivt sätt utföra logganalys och skapa en korrekt lägesbild. Denna rapport syftar till att ge en översiktlig beskrivning av processen kring analys av cybersäkerhetsloggar samt variabler som av författarna bedöms vara relevanta för framgång, och därmed bör studeras av framtida forskning.

1.2 Lägesbild och logganalys

En korrekt lägesbild är centralt för att effektivt kunna utföra en uppgift. Det gäller oavsett om uppgiften i fråga är att rädda en fotbollsstraff, desarmera en bomb eller detektera en cyberattack. Lägesbild, lägesvy, lägesförståelse och lägesuppfattning är dock begrepp som saknar en allmän vedertagen definition i cyberdomänen. FOI har tidigare studerat litteratur och intervjuat intressenter i Försvarsmakten för att utreda dessa begrepp. Lundholm et al. [1] gjorde ett försök att bena upp begreppet lägesbild i:

- lägesinformation (den information som är tillgänglig),
- lägesvy (presentation av den tillgängliga informationen) och
- lägesuppfattning/lägesförståelse (vad individer har i medvetandet).

De konstaterar vidare att vilken lägesvy som är relevant beror på den roll mottagaren har och att det finns så många rimliga tolkningar av begreppet lägesbild att det inte bör användas utan vidare förklaring.

Franke och Brynielsson [2] gjorde en genomgång av den forskningslitteratur som beskriver forskning kopplat till lägesbild i cybermiljön och har använt den engelska frasen ”cyber situational awareness”. Över 100 artiklar identifierades och kategoriserades. Slutsatserna från denna genomgång var att vissa områden studerats mer än andra. En ansenlig del av de publicerade artiklarna handlar till exempel om algoritmer som ska bidra till god lägesbild genom att processa lägesinformation medan frågor som informationsutbyte studerats i begränsad omfattning. Överlag finns också få empiriska studier. Mindre än hälften av artiklarna hade validerat en lösning på något sätt. Ofta utgjordes valideringen av en enkel simulering, matematisk övning, utförlig diskussion om lösningens komplexitet eller liknande. Ordentliga tester och empiriskt grundande slutsatser är

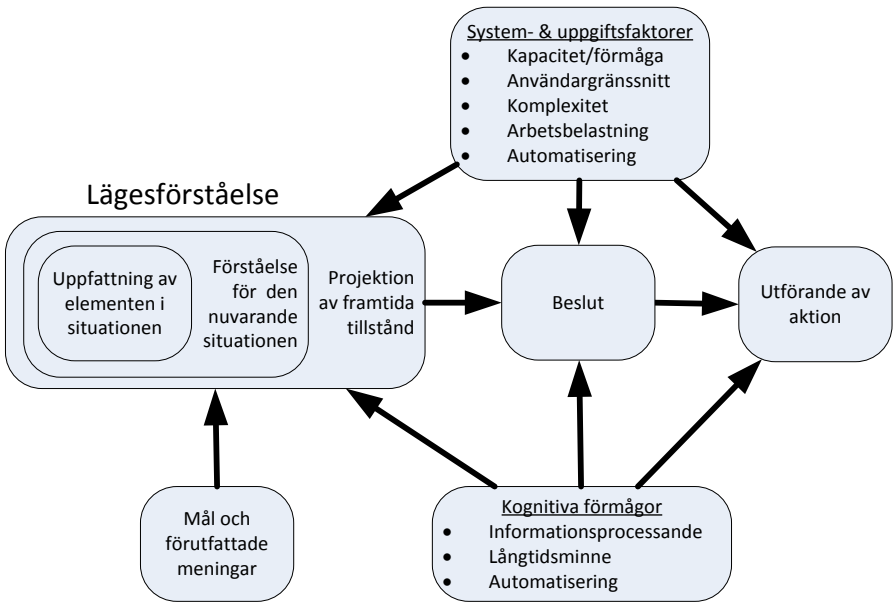
[1] En förkortning av Cyber Range And Training Environment, se www.foi.se/crate

sällsynta och det mesta stannar på en konceptuell nivå eller i idéstadiet. Detta trots att den sammantagna bedömningen av Franke och Brynielsson är att det inom detta område finns god potential för empirisk forskning, särskilt i samband med övningar och träning kopplat till cybersäkerhet.

Forskningen om lägesbild inom cyberdomänen är omogen, men tar i regel avstamp i den forskning som görs i andra domäner. Den mest etablerade teorin kopplat till lägesbild är den som skapats av Endsley [3]. Modellen kopplat till Endsleys teori beskriver tre nivåer av lägesbild (se Figur 1):

- uppfattning av elementen i den nuvarande situationen
- förståelse för den nuvarande situationen
- projektion av framtida tillstånd.

Tillsammans med kognitiva förmågor (inklusive den kunskap som finns i minnet), mål och förutfattade meningar (t.ex. om hur angrepp sker), samt faktorer kopplade till arbetsmiljön (t.ex. arbetsbelastning och användargränssnitt) påverkar lägesförståelsen vilket beslut som tas och i slutändan hur väl en person presterar i uppgifter som kräver lägesförståelse.



Figur 1. Lägesförståelse, anpassad från Endsley [3].

Denna rapport fokuserar på förmågan att analysera cybersäkerhetsloggar. Förmågan att analysera cybersäkerhetsloggar kan syfta till att upptäcka och förstå såväl pågående som historiska incidenter kopplade till cybersäkerhet genom att

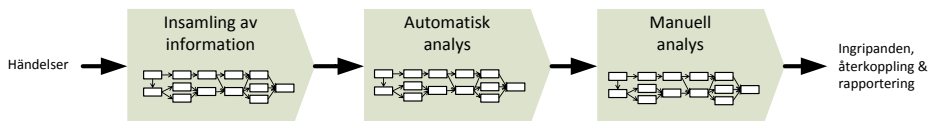
studera de händelser som registreras i IT-system. Det slutgiltiga målet är i båda dessa fall att kunna fatta bra beslut kopplat till cybersäkerhet. För att kunna fatta bra beslut behöver såväl mänskliga logganalytiker som automatiska algoritmer ha en uppfattning av element kopplade till situationen, förstå vad dessa innebär och prediktera hur händelser kan komma att utvecklas. Endsleys modell över lägesförståelse passar alltså väl för logganalys (och förmågan att analysera cybersäkerhetsloggar handlar således om att skapa lägesbild med hjälp av loggar från system i cybermiljön).

1.3 Rapportens innehåll och avgränsningar

Denna rapport innehåller en genomgång av de variabler som, av författarna och andra forskare inom området, tros vara av vikt för att framgångsrikt kunna analysera cybersäkerhetsloggar. Rapporten kommer att användas som underlag när alternativ för experiment övervägs inom ÖvExCy och för att sätta framtida resultat i en kontext.

Genomgången är avsiktligt översiktlig, främst för att den forskning som gjorts i denna domän är ojämnt fördelad över de variabler som författarna tror är relevanta och en djupdykning i några få variabler inte tillför mycket till helheten. Ytterligare ett skäl till att genomgången hålls översiktlig är att det i stort saknas empiriska studier på variablers betydelse för helheten och det även inom de mer studerade variablerna saknas gedigen kunskap om vad som är viktigt.

I denna rapport beskrivs logganalysarbete utifrån de tre delar som illustreras i Figur 2: insamling av information (avsnitt 2), automatisk analys (avsnitt 3) och manuell analys (avsnitt 4).



Figur 2. Översikt över logganalysarbete och den input och output som hanteras.

Denna nedbrytning återspeglar det informationsflöde som finns i en typisk logganalysverksamhet. Som läsaren kommer märka är det dock svårt att sätta skarpa gränser mellan de tre delarna. Insamling av information kräver ofta viss behandling av informationen, och denna behandling kan i vissa avseenden ses som automatisk analys. I avsnitt 3 av denna rapport fokuseras det på automatisk analys som har ett tydligt cybersäkerhetsrelaterat syfte medan mer generell behandling av rådata läggs i avsnitt 2. Gränsen mellan manuell analys och automatisk analys är också otydlig i vissa avseenden. Till exempel använder eller skapar ofta logganalytiker verktyg i sitt manuella analysarbete. I denna rapport fokuseras

avsnitt 3 på sådana automatiska analyser som är standardiserade och generella medan verktyg och lösningar som behöver ”köras” av logganalytiker hamnar i avsnitt 4.

Inom vart och ett av de tre områdena (insamling av information, automatisk analys och manuell analys) ges en översikt av den kunskap som dokumenterats i tillgänglig litteratur. För att göra denna sammanställning så konkret som möjligt ges ibland tekniska exempel. Dessutom listas i slutet av varje avsnitt variabler som tros vara av stor betydelse för logganalysförmågan i stort. Dessa ska inte tas som en absolut sanning utan är, på samma sätt som mycket av den existerande litteraturen, baserade på kvalificerade gissningar. Det är också värt att notera att andra nedbrytningar än den som används i rapporten är tänkbara och rimliga. Till exempel skulle logganalysförmåga kunna brytas ner enligt vilken information som behandlas (olika loggar etcetera), vad analysen syftar till (detektion eller orsaksanalys), de fyra nivåerna i JDL:s modell för datafusion [4] eller de tre nivåerna av lägesbild som finns i Endsleys modell [3].

Sist men inte minst bör det poängteras att logganalys bara är ett medel att uppnå en korrekt lägesförståelse. Andra sätt för att skapa lägesförståelse, så som goda underrättelser, kan både ersätta och komplettera logganalys.

2 Insamling av information

En förutsättning för ett framgångsrikt logganalysarbete är att relevant information samlas in. Såväl insamling av vad som sker, insamling av vad som finns och insamling av hotunderrättelser är av relevans för logganalysarbetet. I detta kapitel ges en kort genomgång av dessa tre typer av information och hur de kan samlas in. Sist i kapitlet ges en sammanfattning av de mer betydande variablerna kopplade till insamling av information.

2.1 Insamling av vad som sker

Ett IT-system har många olika platser där det lagrar information om vad som händer i systemet. Vissa lagringsmedium lagrar bara information under en kort tid eller precis när en händelse sker. Exempelvis lagras information som en applikation behöver för sin exekvering i en dators Random Access Memory (RAM), ett minnesutrymme som nollställs när strömmen bryts; en särskild del i RAM-minnet kallat ”stacken” lagrar små egenskaper som en applikation bara behöver komma ihåg för den närmaste exekveringstiden (såsom en variabels värde); en router eller switch behöver temporärt lagra nätverkstrafik för att kunna skicka den vidare; en brandvägg måste temporärt lagra nätverkspaket för att kunna matcha dem mot kriterier som bestämmer om trafiken skall blockeras eller inte. Andra minnesutrymmen långtidslagrar information som enbart är tänkt att försvinna om användaren själv raderar den. Exempelvis lagrar en Solid State Drive (SSD) eller ett Universal Serial Bus (USB) minne information som inte nollställs när strömmen försvinner. Informationen finns kvar tills att den skrivs över av någon annan information.

Information om händelser som är relevanta för cybersäkerhet kan finnas i alla dessa lagringsmedium. Exempelvis kan information från RAM-minnet berätta om skadliga exekveringsinstruktioner som injicerats av någon typ av elakartad kod; information från en disk kan berätta om något olämpligt skrivs till den eller läses från den; information från en switch kan berätta om något olämpligt skickas från en applikation till en annan över ett IP-baserat nätverk.

Av denna anledning har det byggts diverse verktyg som samlar in och studerar information från olika lagringsmedium. Ett exempel på verktyg är libpcap, som loggar information om nätverkspaket som passerat. Ett annat exempel är Windows event log, som rapporterar om händelser i Windows-baserade system, såsom om en användare läggs till eller om en särskild mapp läses av en viss användare.

Olika verktyg bidrar med olika typer av behandling av insamlad data, och ofta bygger ett verktyg vidare där ett annat slutar. Exempelvis bygger verktyget tcpdump vidare på libpcap genom att lägga på filter på sparad nätverkstrafik som tolkar dess egenskaper och till exempel avgör om det handlar om fildelning eller

e-posttrafik. Vissa verktyg, såsom antivirus, utför även automatisk analys av insamlad data för att förebygga intrång eller underlätta för en logganalytiker med cybersäkerhetsfokus. Dessa automatiska analyser beskrivs vidare i avsnitt 3.

Olika sensorer har olika tekniska och administrativa styrkor och svagheter när det kommer till rapportering av cybersäkerhetsrelevant information. En sensor som rapporterar om nätverkstrafik är oftast resurseffektivare än en sensor på en maskin då den kan studera trafiken till/från många system samtidigt. En nätverkssensor har dock sällan samma precision som en maskinsensor då det är svårt för den att förstå kontexten och betydelsen av ett informationsflöde. Exempelvis är det är svårt för en nätverkssensor att pussla ihop enskilda paket till ett komplett begripligt informationsutbyte, och om ett sådant informationsutbyte kan identifieras är det fortfarande mycket svårt för sensorn att förstå vad olika instruktioner innebär utan se hur de observerade instruktionerna påverkar de deltagande applikationernas interna tillstånd. Det är också av naturliga skäl svårt för en nätverkssensor att övervaka krypterad kommunikation såsom HTTPS. Det finns dock en fördel som bör poängteras med att nätverkssensorer typiskt inte exekverar koden den analyserar – den löper då heller ingen risk att infekteras av den potentiella skadliga kod som den rapporterar om, till skillnad från maskinsensorer. En maskinsensor är därför inte alltid att lita på. Till exempel är det vanligt att datorvirus stänger av antivirus utan att berätta detta för användaren.

I nätverk med mycket kommunikation och mycket aktivitet är fördelning av trafik och maskinhändelser till flera sensorer, till exempel med lastbalansering, ett tänkbart tillvägagångssätt [5]. För såväl sensorer på maskiner som sensorer på nätverk finns det i regel anledning att filtrera och förprocessa loggar så nära insamlingspunkten som möjligt. Sådant måste då vägas mot risken att man vid dessa insamlingspunkter saknar den information eller helhetsbild som behövs för att kunna avgöra vad som är relevant att sända vidare. Sist men inte minst är säkerheten i själva insamlingssystemet av vikt. Det är vanligt att insamlingen av systemhändelser separeras från ”vanlig” trafik för att förebygga informationen förvanskas eller läcker till obehöriga [5].

2.2 Insamling av vad som finns

Händelser är centrala för att logganalys, men för att kunna tolka innebörden av dessa händelser är det ofta också nödvändigt att ha en god förståelse för det tillstånd systemet är i och vilka egenskaper det vanligtvis har. Till exempel kan det vara viktigt veta vilken roll en viss maskin eller applikation spelar i ett nätverk för att förstå vad olika händelser betyder. Det är till exempel tänkbart att en händelse har större prioritet om den kan kopplas samman med en känd sårbarhet i ett system. Exempel på sådant som kan vara relevant för att tolka systemhändelser är:

- namn på maskiner och användare i systemet

- behörigheter i nätverket, dess maskiner och mjukvaror
- tjänster, applikationer och andra som finns i systemet
- kända mjukvarusårbarheter och konfigurationssårbarheter i systemet
- det vanliga beteendet som användare och mjukvaror har.

I avsnitt 3 av denna rapport framgår det hur fusion av händelser och information som denna kan vara av vikt när loggar ska analyseras för att detektera och förstå elakartat beteende. Förutom att fusionera informationen med händelser för att detektera och förstå elakartat beteende finns förslag på hur information som denna användas för att med formella metoder optimera placeringen av sensorer i systemet [6].

Information om systemet kan samlas in på flera olika sätt och olika metoder ger olika datakvalitet. Dataimport från ett noga kontrollerat konfigurationssystem i en statisk miljö ger information om vilka mjukvaror som finns; en aktiv sårbarhetsskanner ger information om vilka mjukvaror som finns i nätverket och detekterar ungefär hälften av de kända sårbarheterna [7]; en passiv sårbarhetsskanner har endast möjlighet att detektera de mjukvaror som kommunicerar över nätverket under skanningen (och ibland sårbarheter i dessa). Vad som typiskt anses vara cybersäkerhetsrelevanta tillstånd kan också utläsas ur modeller som ska stödja cybersäkerhetsarbete. Ett exempel på en sådan modell är Open Vulnerability Assessment Language (OVAL), en Mitre-utvecklad standard för vad amerikanska myndigheter vill kunna samla in strukturerat. OVAL är ett paraply för många andra standarder för att klassa säkerhetsinformation (till exempel Common Vulnerability Enumeration och Common Platform Enumeration) och ska beskriva systems statiska egenskaper så att sårbarhetsbedömningar och sårbarhetshantering är möjliga. Mer abstrakta modeller finns också för att beskriva den säkerhetsinformation som kan vara relevant vid sårbarhetsbedömningar och sårbarhetshantering, till exempel CySeMoL [8].

Förutom information om vad som finns i systemet vid en viss tidpunkt kan information om vad som borde finnas i systemet vara av intresse. Till exempel kan en lista på tillåtna protokoll och en specifikation av hur dessa fungerar vara av värde när händelser ska tolkas. Mer eller mindre etablerade standarder för hur detta ska anges finns inom vissa områden. Till exempel anger Substation Configuration Language, en del av standarden IEC 61850, hur en IT-utrustning i en kraftstation ska fungera.

Farwick et al. [9] har resonerat om vas som är viktigt för att för att automatiskt skapa bra informationsmodeller av system i syfte att hantera deras säkerhet. Följande anses vara av vikt:

- Processer för att hantera och uppdatera sin informationsmodell. I detta ingår att besluta vad den ska innehålla och hur ofta det behöver samlas in.

- Möjlighet att samla in data som systemägarna har, till exempel sådant som är lagrat i lokala systemadministrationsverktyg.
- Ett system för att upptäcka förändringar i systemet och koppla dessa mot entiteter i datamodellen på korrekt sätt.
- Stöd för att upprätthålla datakvalitet, aktualitet, propagering av förändringar och konflikter. Till exempel genom datumstämplar och beroendekopplingar mellan entiteter.

I en enkät av riktad till en systemmodelleringsverktygsleverantörs klienter fann Farwick et al. [10] att den tredje punkten ovan (upptäcka förändringar) var viktigast att få automatisera för att få god informationskvalitet. Nya systemgränssnitt, nya informationssystem och förändringar av befintliga system ansågs särskilt viktigt att upptäcka.

2.3 Insamling av hotunderrättelser

Som det kommer att förklaras i avsnitt 3 är ett huvudspår inom automatisk analys av cybersäkerhetsloggar att jämföra pågående eller historiska händelser mot en modell av elakartade händelser. Att ha en bra idé om vad hotaktörer är kapabla till, vad de är intresserade av och hur de arbetar är såklart av vikt för att kunna göra en bra analys av detta slag. Tyvärr finns än så länge få etablerade modeller eller metoder för att hantera hotunderrättelser i cybersäkerhetsdomänen.

I FM UndR [11] diskuteras tre perspektiv som är av vikt i underrättelsetjänst i allmänhet: tematiskt, informationsteoretiskt och temporalt. Det finns få etablerade modeller som matchar det *tematiska* perspektivet inom cybersäkerhetsvärlden. Det vill säga, det finns få modeller av vad hotunderrättelser kan handla om. I relation till den allmänna indelning av underrättelser i som ges i FM UndR (*eget*, *neutralt*, *fientligt* och *övrigt*) handlar hotunderrättelser om det fientliga. I den text om underrättelsehot som finns i FM H SÄK Hot [12] diskuteras skyddsvärden, angriparen och dennes metoder/modus. Denna typ av information är rimligtvis relevant även för cybersäkerhetshot.

Inom det *informationsteoretiska* perspektivet är bearbetningsgraden central enligt FM H SÄK Hot. Informationen som används i praktiskt cybersäkerhetsarbete i dag kan vara atomärt utformad och ostrukturerad som poster i internetforum; atomärt utformad men strukturerad, som en svartlista på IP-adresser; komplex och ostrukturerad, som en muntlig beskrivning av en tidigare incident; komplex och strukturerad, som ett meddelande uttryckt med språket Structured Threat Information eXpression (STIX). När European Network and Information Security Agency (ENISA) bryter ner variabler kopplat till information för incidenthantering används fyra nivåer: lågnivådata, detektionsindikatorer, råd (eng. advisories) och strategiska rapporter [13].

Oavsett hur komplex och strukturerad informationen är finns typiskt behov av att bearbeta den och sätta den i rätt kontext med automatiska eller manuella processer. FM H SÄK Hot framhåller för det *temporala perspektivet* att underrättelser ska syfta till att prediktera framtida händelseutveckling och att det är information som värderats och lagrats för framtida bruk, till skillnad från ”combat information”. Hotunderrättelserna är alltså inte de loggposter som sensorerna avger just nu eller vad som kommer att ske de närmsta minuterna, utan sådant som möjliggör prediktioner om vad som kan hända om dagar, månader eller år.

ENISA föreslår att informationens användbarhet bestäms av dess relevans, aktualitet, träffsäkerhet, fullständighet och importbarhet/interoperabilitet [13]. FM UndR har en enklare modell, som säger att ska värdet av underrättelser ska bedömas utifrån informationens sakriktighet och källans tillförlitlighet. Dessa två används för att ange en konfidensgrad och en giltighetstid. En bedömning av sakriktighet kräver att informationen jämförs mot annan relevant information. Inom cybersäkerhet behöver alltså analytikern ha goda teorier över cybersäkerhetsdomänen för att kunna bedöma vad som är relevant och vilka förhållanden som är kompatibla. För att bedöma en källas tillförlitlighet behöver tidigare information från källan jämföras med faktiska händelser. Hotunderrättelser inom cybersäkerhetsdomänen kan bland annat hämtas från källor som

- betaltjänster som eNorse[2], Internet Identity[3], Cyveillance[4], Malcovery[5], Recorded Future[6]
- öppna källor som media eller publika databaser som Exploit Database[7] och VERIS[8]
- andra organisationer som delar med sig av information.

Tyvärr finns inga kända tester av hur träffsäkra aktuella, träffsäkra, fullständiga eller interoperable dessa källor är. Inte heller finns tester på deras tillförlitlighet eller rapporternas sakriktighet.

[2] http://www.norse-corp.com/products/norse-intelligence-service/
[3] http://internetidentity.com/threat-intelligence/
[4] https://www.cyveillance.com/home/what-we-do-threat-intelligence/
[5] http://www.malcovery.com/protect-your-network
[6] https://www.recordedfuture.com/cyber-threat-intelligence/
[7] https://www.exploit-db.com/
[8] https://github.com/vz-risk/VCDB

2.4 Variabler av betydelse för insamling av information

Syftet med insamlingen är att möjliggöra automatisk och manuell analys. Med andra ord bör insamlingen av information möta de krav som analysen ställer. Till exempel med avseende på vad som samlas in, dess kvalitet och dess format. Eftersom olika analyser ställer olika krav är det svårt att identifiera ett konkret och generellt mått på hur väl insamlingen av information fungerar. Det är dock i enlighet med texten ovan rimligt att förvänta sig att följande är av stor betydelse för hur väl insamlingen möter de krav som ställs i analysen:

- **Val och placering av sensorer**: Hur väl sensorerna kan samla in olika typer av händelser och tillstånd.
- **Systemkartläggning**: Hur väl information om det övervakade systemet samlas in.
- **Hotunderrättelser**: Hur väl logganalysen bereds med hotunderrättelser

Hur väl sensorerna kan samla in olika typer av händelser och tillstånd beror på hur väl de täcker in olika systemkomponenter, hur stor andel av alla händelser de registrerar och hur granulärt de registrerar händelserna. Det är till exempel fördelaktigt om nätverksbaserade sensorer kan registrera all kommunikation i systemet, om de klarar av att processa all sorts trafik och om de kan registrera innehållet i trafiken (och inte bara metadata).

Vilken information om det övervakade systemet som en kartläggning ska samla in beror på vilken typ av analys som ska utföras. Att veta vad som behövs i den automatiska och manuella analysen är därför av betydelse. Dessutom spelar det roll om det finns gott stöd för att faktiskt samla in information från det övervakade systemet och om det finns en god förändringshantering. Om analysen baseras på attackgrafer är det till exempel bra att identifiera nätverkstoplogier, behörigheter och maskiners sårbarheter. Dessutom är det bra om denna information hålls uppdaterad och kan samlas in direkt från systemets inventariesystem, nätverksutrustning och behörighetslistor istället för med en skanner (som har en sämre träffsäkerhet).

När det kommer till hotunderrättelser är naturligtvis kalibreringen av bedömningarna av stor vikt. De hot som kommer att påverka organisationen har i bästa fall föregåtts av underrättelser med hög konfidens och de hot som *inte* kommer att påverka organisationen har i bästa fall föregåtts av underrättelser med låg konfidens eller helt förbisetts. Dessutom är det i regel önskvärt att underrättelserna är konkreta och lättbehandlade. Till exempel om det kan förutsägas vilka maskiner som kommer angripas och att angreppets signatur är känd på förhand.

3 Automatisk analys

I de flesta IT-system processas en ansenlig mängd data. Till exempel kräver loggning av en enda veckas nätverksdata i ett nät som skapar 0,5 Gb/s hela 37.8 TB lagringsutrymme. I de allra flesta fall är det därför orimligt att manuellt inspektera alla händelser för att identifiera och förstå elakartad aktivitet. I många fall är det inte heller görbart att manuellt gå igenom de händelser som av sensorer klassats som säkerhetsrelaterade, som alla autentiseringsförsök som görs. Automatisk analys, i någon utsträckning, är därför en nödvändighet om elakartad aktivitet ska identifieras ur systemloggar på ett effektivt sätt.

Det finns många sammanställningar av algoritmer och modeller som används för automatisk analys i forskning inom logganalys. Exempel på sådana sammanställningar är [14]–[33]. Detta avsnitt kommer inte gå in på alla klassificeringssystem som föreslagits eller de egenskaper som de tillskrivits. Det huvudsakliga skälet är att den kunskap som dokumenterats om vad som fungerar bra respektive dåligt är alltför otillförlitlig. Inför skrivandet av denna rapport gjordes en systematisk litteraturgenomgång av de akademiska forskningsartiklar som rör olika metoder för att detektera missbruk och skadlig kod. Av cirka 8000 sökträffar bedömdes 671 artiklar vara relevanta och tillräckligt nya för att läsas. Av dessa bedömdes 159 som tillräckligt intressanta för att läsas i detalj. De allra flesta av dessa artiklar beskrev lösningar som i bästa fall testats mot det femton år gamla datasetet KDD Cup [34], som sedan länge varit kritiserat för betydande brister [35], [36]. Det verkar dessutom svårt att som oberoende part testa de lösningar som föreslagits. FOI kontaktade författarna till de 50 artiklar som beskrev egna tester för att erhålla deras lösningar och därmed kunna studera deras effektivitet i CRATE. Endast en tre positiva svar erhölls.

Istället för att presentera en detaljerad genomgång över olika lösningar presenteras i detta avsnitt en översiktlig beskrivning av de två huvudsakliga ansatser som återfinns i litteraturen:

- Modeller av elakartad aktivitet (dvs. angrepp och missbruk) som jämförs med händelser i systemet (avsnitt 3.1).
- Modeller av godartad aktivitet och tillstånd som jämförs med händelser i systemet (avsnitt 3.2).

De flesta sammanställningar av lösningar som finns i litteraturen kategoriserar metoder utifrån de algoritmer som används. Indelningen som görs här utgår utifrån vilken information de olika alternativen använder i analysen. Detta innebär bland annat att maskininlärning, som har getts stort fokus i forskningen kring intrångsdetektion, inte här tas upp som ett eget avsnitt.

Den automatiska analysen syftar till möjliggöra bra och snabba beslut kopplat till cybersäkerhet. Den automatiska analysen kan därför avslutas med en automatisk

åtgärd eller ett förslag på sådan, till exempel blockering av trafik eller spridning av information. Detta tas upp i avsnitt 3.3.

3.1 Modeller av elakartad aktivitet

Ett intuitivt sätt att elakartad aktivitet är att leta efter direkta tecken på att de sker genom att jämföra händelser med en modell av hur elakartad aktivitet ser ut. Detta kan göras på många olika sätt. Den utan tvekan vanligaste metoden är att matcha systemhändelser mot en databas med hårdkodade signaturer som beskriver olika elakartade eller riskabla beteenden. Om en överenstämmelse hittas förmodas händelsen vara elakartad på det sätt som specificerats i signaturen. Till exempel kan en signatur larma för att en NOP-släde[9] syns i nätverkstrafik, när upprepade misslyckade autentiseringsförsök har gjorts mot en maskin, eller när en viss sekvens av systemanrop utförts av en applikation. Figur 3 ger ett konkret exempel på en signatur från intrångsdetektionssystemet Snort i. Signaturen i Figur 3 är en av över 100 signaturer som detekterar angrepp mot MS08-067, en sårbarhet i Microsoft-operativsystem som användes av datormaskarna Conficker och Stuxnet. För att signaturen skall larma krävs det dels trafik över UDP mot en övervakad IP-adress på port 139. Det krävs också att det finns två särskilda hexadecimala avsnitt på särskilda platser i paketets innehållsdel.

```
alert udp any any -> $HOME_NET 139 (msg:"ET DELETED Microsoft Windows
NETAPI Stack Overflow Inbound - MS08-067 (6)"; content:"|0B|";
offset:2; depth:1; content:"|C8 4F 32 4B 70 16 D3 01 12 78 5A 47 BF
6E E1 88|"; classtype:attempted-admin; sid:2008695; rev:5;)
```

Figur 3. Exempel på signatur i Snort.

Signaturbaserade verktyg så som Snort är sedan länge mogna och att uppfinna nya sådana verktyg är inte något som den akademiska forskningen ägnar mycket tid åt. Det forskas dock kring hur dessa system bäst bör operationaliseras och tolkas. Till exempel genom testa prestandaförbättringar av signaturmatchningsmetoder så att de kan analyser trafik i realtid (ett exempel finns i [37]), automatisering av signaturskapande (som är dyrt att göra manuellt [38]) och lösningar för att binda ihop signaturer med varandra. I de flesta fall görs det sista i syfte att minska mängden falsklarm, men det görs även för att kunna ge lägesförståelse på högre nivå genom att kunna presentera vilka sekvenser av steg angripare har försökt genomföra. Exempel på forskning i detta spår är:

- Modeller av hur larm och olika klasser av angrepp hör samman, som ska göra larmen mer begripliga. [39], [40]

[9] En serie av processorinstruktioner som ökar programräknaren, vilket ofta görs som en del av en attack som försöker utnyttja en bufferöverskridningssårbarhet

- Enkla modeller av vilka larm som borde komma samtidigt för att öka träffsäkerheten i bedömningar [41] eller enkla modeller av händelser som borde följa av ett lyckat angrepp kopplat till ett larm [42]
- Modeller av nödvändiga/rimliga förutsättningar för attacker, som ska hjälpa att filtrera bort irrelevanta larm. Till exempel genom att titta om det finns en känd sårbarhet på maskinen och den tjänst som angrips finns på maskinen. [40], [43]–[49].
- Komplexa modeller av vilka relationer larm bör ha till varandra i form av attackgraf för att ge bättre träffsäkerhet, ge mer begripliga händelsebeskrivningar eller indikera angriparens planer. [50]–[56]

I de flesta fall föreslås regelbaserade system för hur detta ska göras och en expert använder kunskap eller information för att bygga modellen. Kunskap om egna sårbarheter, hotagenters resurser och liknande är såklart viktigt för att dessa ansatser skall fungera. Vissa av förslagen (till exempel det i [41]) inkluderar moment av maskininlärning och tränas istället upp med data. När så är fallet är det viktigt att det träningsdata som lär upp modellen stämmer överens med den typ av data som ska analyseras [57].

3.2 Modeller av godartad aktivitet

Angrepp kan även detekteras genom att jämföra händelser och tillstånd mot en modell över vad som är godtagbart för att sedan reagera på det som faller utanför denna ram. Vilka egenskaper som skall inkluderas i en modell av godartad aktivitet är inte lika uppenbart som det är för modeller av elakartad aktivitet och tillstånd. I en inflytelserik artikel skriven på 80-talet föreslog Denning [58] att man skulle modellera sessioner, program som körs och filer som används på en maskin. Dagens akademiska forskning nyttjar en mängd olika informationstyper för att skapa signaturer över godartad aktivitet. Det är till exempel vanligt att använda trafik eller trafikmönster (exempelvis [59], [60]), systemanrop och egenskaper associerade med systemanrop (exempelvis [61]–[65]), samt belastning på resurser (exempelvis [66]). Med hjälp av sådan information kan modellerna över godartad aktivitet skapas genom att mäta följande:

- Avvikelse från ett förväntat tillstånd eller händelse. Till exempel om checksumman för en fil är fel eller om kommunikation avviker från ett protokolls specifikation [67].
- Avvikelse från den vanliga frekvensen av händelser/tillstånd. Till exempel avvikelse från frekvens av olika anrop [63] eller datautbyten [66].

- Skillnad från sekvenser eller kedjor av händelser/tillstånd. Till exempel en sekvens av systemanrop [62] eller den normala interaktionen med ett protokoll [68].
- Kombinationer av de tre modelltyperna ovan, eller korrelation mot andra variabler (som till exempel värdet av en tillgång), är också ett alternativ.

En slags informell tumregel är att ju mer komplicerad algoritmen som ska härleda fram godartad aktivitet är, desto färre variabler och tillstånd kan övervakas (på grund av prestandaskäl). Precis som ovan kan modellerna skapas med hjälp av existerande kunskap och information om vad som är godartat. Till exempel föreslår Sekar et al. [67] att man skapar modeller av godkänt beteende med hjälp av systemspecifikationer. I akademisk litteratur är det dock vanligare att det föreslås en algoritm för att lära sig känna igen godartat beteende med hjälp av statistiska metoder. Markovmodeller (som beskriver sannolikheten för tillståndsövergångar) är till exempel vanligt att använda i kombination med sekvenser av systemanrop och protokollanrop. I den akademiska litteraturen finns också förslag på användning av maskininlärningsansatser (som neurala nätverk) som har skapar okända modeller enligt ett förutbestämt mönster. De maskininlärningsmetoder som använt okända modeller har fått kritik för att de inte ger tillräckligt stöd till logganalytikerers beslutsfattande eftersom de inte pekar ut orsaken till larm är på ett tydligt sätt [57].

3.3 Format på analysresultat och val av automatisk åtgärd

Det är vanligt att resultat från automatiska analyser kommuniceras till en mänsklig logganalytiker som förväntas fatta rimliga beslut baserat på dess innehåll. Tripletter av subjekt, handling och objekt har föreslagits som format för att beskriva säkerhetshändelser som en logganalytiker bör undersöka [69]. Till exempel kan det då anges att objektet *Program X* gör handlingen *skriva* till objektet *C:\Mapp_A*. Mer invecklade och komplexa sätt att beskriva av säkerhetshändelser har också föreslagits. Intrusion Detection Message Exchange Format (IDMEF) säger till exempel att varningar ska innefatta: sensor/analyserare, datumstämplar för olika steg i insamlingsprocessen, källa om mål för händelsen, en eventuell utvärdering av händelsen och eventuell fritext. Common Intrusion Specification Language (CIDF) föreslår fält för händelser av olika slag (till exempel öppna en session och ta bort en fil), tiden det skedde och inblandade entiteter (till exempel användare, maskin och port). Mitre förslår med Cybox en än mer komplex modell som gör det möjligt att utrycka både specifika och översiktliga händelser, som att ett e-post med en bilaga observerats eller beskriva en kampanj av nätfiske med e-post. I Cybox breda vokabulär finns bland annat hundratals händelser, aktioner, subjekt, objekt och tillstånd fördefinierade. Till exempel: Created_By, Deleted_By, Read_From, Paused, Resumed, Moved,

Allocated, Initialized, Sent, Mapped_Into, Killed_By, Encrypted_By, osv. Det finns också en stor mängd objekt definierade, som: IRI, Whois, Win Event Log, Win Event, User session, Network Package, Memory, File, och så vidare.

Logganalyslösningar erbjuder ofta funktioner för att meddela administratörer, exempelvis via e-post, om ett förmodat angrepp sker. Det är också vanligt med funktioner för ange automatiska interventioner för att försvåra pågående angrepp [70]. Exempel på sådana interventioner åtgärder är:

- När skadlig kod detekteras av ett antivirusprogram och filen tas bort samtidigt som förändringar återställs.
- När en brandvägg ser kommunikation med en svartlistad adress och hindrar trafik att passera.
- När misstänkt aktivitet på en maskin upptäcks och ytterligare loggning eller ytterligare skydd aktiveras.

Vilka åtgärder som ska utföras vid olika analysresultat följer i vissa fall naturligt från den typ händelse som sker och vilken produkt som upptäcker det. Ett antivirusprogram tar till exempel i regel bort skadlig kod när så är möjligt och notifierar såväl användare och administratörer om att det skett. Men en mycket hög träffsäkerhet krävs i regel för att automatiska interventioner ska vara rimliga att utföra då ett falskt larm kan få dyra konsekvenser. Dessutom behövs det vid val av åtgärd vägas in hur allvarlig hotet är, vilket värde tillgångar som kan påverkas av en intervention har och vilka åtgärder som finns att välja mellan [71]. Det har skissats på generella lösningar för att fatta interventionsbeslut där konfidens, systemets tillstånd med mera vägs in i beslutet. Till exempel föreslår [72] att kostnader för olika åtgärder på nätverksnivå bedöms utifrån modeller av resurser, systemanvändare, nätverkstoplogi och brandväggsregler. Dessvärre har förslag som dessa inte lämnat konceptstadiet ännu.

3.4 Variabler av betydelse för automatisk analys

Intrångsdetektion är ett av de mer centrala problemen i automatisk analys. För system som ska upptäcka elakartad aktivitet finns därför förslag på hur kvaliteten i ett system bör mätas. Mell et al. [73] tar upp tio kvantitativa mått: 1) sannolikheten för falsklarm, 2) sannolikheten för upptäckt, 3) möjlighet att upptäcka nya typer/varianter av attacker, 4) möjlighet att upptäcka olika typer av attacker, 5) förmåga att ange attacktyp eller exploaterad sårbarhet, 6) förmåga att koppla ihop händelser, 7) möjlighet att bedöma attackens framgång, 8) förmåga att hantera attacker med stor bandbredd, 9) kapacitet att hantera datamängder och 10) motstånd mot attacker. Vissa av dessa handlar om vad för typ av analyser systemet gör (5-7), vissa om hur träffsäkra dessa analyser är (1-4) och andra om lösningens kapacitet eller prestanda (8-9). Automatisk analys i denna rapport avser även analyser utanför intrångsdetektion, men om man bortser från skydd mot

angrepp riktade mot själva analysverktyget (10) kan alltså rimligt att tänka sig att denna den automatiska analysens värde beror på:

- **Analystyp**. Om manuella beslut skall tas baserat på analysresultatet behöver dess klassningssystem och syfte passa de manuella analyser som görs av en logganalytiker och utgöra ett bra beslutsunderlag.
- **Träffsäkerhet**. Hur träffsäker analysen är. Till exempel hur många falska larm, korrekta larm och missade larm som en detektionslösning producerar.
- **Prestanda.** Hur snabbt analysresultaten finns tillgängliga och hur enkelt det är att komma åt dem.

Analystypen är av vikt eftersom den automatiska analysen ska ge ett underlag som logganalytikern kan använda för att besluta om åtgärder såsom ingripanden, återkoppling och rapportering. För ingripanden kan det till exempel vara lätt att omvandla information till beslut om den automatiska analysen indikerar vilka maskiner som är kompromitterade, vilken sårbarhet som exploaterats, vilka steg angriparen kan förväntas ta härnäst och vilka åtgärder som hindrar eller begränsar angriparen. Om logganalytikern bör ge återkoppling och ändra i den automatiska analysen efter det att ett falsklarm konstaterats kan det vara av värde att veta vilka signaturer som bör justeras. Om logganalytikern ska rapportera händelsen vidare kan det vara av värde att den automatiska analysen kan generera beskrivande rapporter, extrahera relevanta loggar och skicka dem till berörda parter. I praktiken ger automatiska analyser dock sällan information som enkelt kan omsättas till åtgärder, utan istället skapas indikatorer som en logganalytiker nyttjar tillsammans med andra informationskällor för att erhålla en grund för rimliga beslut. Exempelvis kan ett larm om en förmodad injektionsattack från Snort korreleras av en analytiker (eller verktyg) med information om kända sårbarheter i det förmodat angripna systemet som identifierats via nätverksskannern Nessus.

En *träffsäker* analys är såklart nödvändig för att åtgärderna ska vara de rätta. I de flesta fall syftar den automatiska analysen till att helt enkelt klassificera händelser eller avgöra om en händelse är godartad eller elakartad. I fall med dikotomier av typen {elakartad, godartad} mäts träffsäkerhet ofta som sannolikhet att korrekta respektive falska larm ges. För mer komplexa analyser, som prediktion av framtida angrepp, behövs dock mer komplexa mått på träffsäkerhet. Oavsett komplexitet bör man förvänta sig att en träffsäker analys kräver en analysmodell som är rättvisande såväl kvalitativt och kvantitativt korrekt. Kvalitativt bör modellen innehålla allt relevant elakartat och/eller godartat beteende; kvantitativt bör frekvenser, sannolikheter och prioriteringar vara korrekta. När det kommer till träffsäkerhet är det dels av vikt att verktyget får tillräcklig och tillförlitlig indata som det kan processa, dels av vikt att verktyget konfigurerats på ett tillförlitligt sätt och underhålls med omsorg: om verktyget nyttjar vit- eller svartlistor för att bedöma vad som är godartad respektive elakartad aktivitet bör dessa regellistor

uppdateras för att passa både den nuvarande omgivningen och de förmodade hoten; om verktyget nyttjar maskininlärning bör det tränas vid behov.

När det gäller *prestanda* är det av vikt att analysen är färdig snart efter det att angrepp utförts. Att upptäcka ett angrepp långt efter det slutförts omöjliggör nämligen interventioner som begränsar skadan. Eftersom många verktyg, såsom Security Information Event Management (SIEM) system, analyserar stora mängder data är det också av vikt hur data sparas och processas av analyser. Till exempel visar [74] på att en SQL-fråga som räknar fram medelvärden för en tabell med 7.9 miljarder poster mot en distribuerad SQL-databas kan ta upp emot 12 timmar. Denna tid skulle vara betydligt längre om det handlade om en icke-distribuerad databas eller om sökningen även krävde att tabeller kopplades ihop. Effektiva lösningar för att hantera sökningar i stora datamängder är alltså av betydelse för logganalysarbetet.

4 Manuell analys

I så gott som alla praktiska sammanhang innefattar logganalysförmåga manuellt arbete. Logganalysarbetet går till stor del ut på att filtrera bort falsklarm och oviktiga loggposter och prioritera bland de korrekta och viktiga loggposterna, men det går också ut på att skapa förståelse för vad som skett och identifiera lämpliga åtgärder [75]. Aningen förenklat kan det sägas att logganalytikerns jobb är att komplettera den automatiska analysen och täcka upp för de brister som finns i algoritmerna som används. Den process som logganalytiker utför består av tre faser [76]:

- **Övervakning**, där logganalytikern har den enformiga uppgiften att gå igenom larmlistor och loggar för att identifiera anomalier och annat som tyder på intrång och missbruk.
- **Orsaksanalys**, där logganalytikern gör en grundligare undersökning av möjliga intrång och missbruk för att bedöma vad som faktiskt har hänt. Om något bedöms som ett faktiskt hot görs en konsekvensanalys för att bedöma vilka effekter hotet redan har fått eller kan tänkas få givet olika omständigheter. I samband med konsekvensanalysen görs dessutom ofta en spårning av hotets aktivitet för att bedöma hur det uppstod och vad det har innefattat.
- **Åtgärd**, där logganalytikern tar fram lämpliga åtgärder. Till exempel att konfigurera om sensorer eller sökströngar vid frekventa falsklarm, att underrätta systemägare via e-post eller telefon, att isolera komprometterade resurser eller utföra systemåterställningar och mjukvaruuppdateringar. Själva åtgärdsfasen är en kritisk och svår aktivitet då det per definition inte går att lita på ett komprometterat system; all kod i det kan tänkas vara omskriven och all trafik genom det avlyssnad.

Det bör poängteras att övervakning, orsaksanalys och åtgärder i verkligheten inte är en ”vattenfallsmodell”, utan en invecklad process där resultatet från en fas matas in i en annan. Exempelvis kräver goda åtgärder ofta inkrementella förändringar där varje förändring föregås av en omfattande orsaksanalys [77].

Trots att logganalytikern onekligen är central och viktig för praktiskt intrångsdetektionsarbete så finns få studier som undersöker logganalytikerers arbete eller testar logganalytikers förmåga i dessa tre faser. Kvalitativa undersökningar har gjorts av Werlinger et al. ([75], [78], [79] och Goodall et al. ([76], [80]). I dessa görs observationer av logganalytiker eller intervjuer med logganalytiker för att förstå förutsättningarna för intrångsdetektionsarbete i praktiken och vad praktiker anser vara viktigt för god detektionsförmåga. I form av kvantitativa tester finns en handfull studier publicerade. Sommestad och Hunstad [81] testade hur väl en logganalytiker kan filtrera larm från

intrångsdetektionssystem; Sawyer et al. [82] gjorde ett experiment som fann att logganalytiker har svårare att korrelera information när den är synlig kortare tid och när attacker är sällsynta; Ben-Asher och Gonzalez [83] har gjort experiment som indikerat att kunskap om den angripna miljön och cybersäkerhet sammanfaller med förmåga att avgöra vilken typ av angrepp som skett; Ben-Asher [84] har i ett annat experiment funnit att detaljerad feedback till logganalytikern om hur väl den presterar i att klassa händelser ger bättre förmåga att detektera vanliga angrepp men det ger samtidigt sämre förmåga att detektera nya/ovanliga angrepp. Ett experiment där olika tekniska hjälpmedel jämförs med varandra finns också i litteraturen. Thompson et al. [85] testade prestation med visualiseringsstöd jämfört med prestation med kommandotolk. Detta enda experiment kan sättas i kontrast till det stora antal förslag på visualiseringsstöd som finns i litteraturen (bara i [86] beskrivs 38 lösningar) och det stora antalet visualiseringsstöd som finns att tillgå i praktiken – författarna kunde exempelvis identifiera över trettio SIEM-system på marknaden.

Sammanfattningsvis, eftersom logganalytiker har en viktig roll i logganalysarbetet finns goda skäl att reda ut vilka förmågor och förutsättningar som leder till god logganalysförmåga. På grund av det magra antalet evaluerande studier som finns tillgängliga inom cybersäkerhetsdomänen är tyvärr kunskapen om detta bristfällig. Fler studier behöver alltså genomföras. I nedanstående avsnitt diskuteras vilka förmågor och förutsättningar som kan antas vara viktiga för att logganalytiker ska lyckas med den manuella analysen.

4.1 Övervakning av händelser

Övervakning syftar till att identifiera avvikelser, oregelbundna mönster eller andra tecken på verkliga hot, ofta i stora mängder loggdata. Det ofta monotona arbetet innefattar att tillämpa sökfilter och visuellt kontrollera tabeller med loggposter, men även att ha koll på nya hot såsom skadlig kod och mjukvarusårbarheter.

Olika verktyg producerar logginformation som är av helt olika karaktär. Ett exempel kan ges för en lyckad attack mot sårbarheten CVE-2008-4250[10] i en Windows XP-maskin. För denna attack producerar Wireshark cirka 370 poster, medan Snort (med Emerging Threats regelverk) producerar två poster och Windows Event Log (med säkerhetsloggar igång) inte en enda. Om antagonisten sedan väljer att lägga till en ny användare till systemet producerar Wireshark ytterligare cirka 50 poster, Windows Event Log sex poster, medan Snort inte producerar några nya poster alls. Vidare innefattar alla dessa poster mycket varierad information, från paketdata som markerats upp (med Wireshark) till indikationer på attacker (”ET SHELLCODE Rothenburg Shellcode”, med Snort), till Windows-händelser (”Security Enabled Local Group Member Added”, med

[10] Se US National Vulnerability Database

Windows Event Log). För att en logganalytiker ska vara framgångsrik vid övervakning krävs det med andra ord att den har god förståelse för verktygen som producerar de loggar som studeras. Det krävs också att analytikern förstår den tekniska omgivningen från vilka loggarna produceras av. En Windows Event Log post om en tillagd användare kan vara helt naturligt för en domänansluten arbetsdator i ett stor kontorslandskap, men något som verkligen bör analyseras vidare om systemet i fråga är en central server i ett industriellt styrsystem.

I relation till den allmänna modell över informationsprocessande som Wickens [87] tagit fram, behöver logganalytiker under övervakningsfasen ta in information från datorskärmar. Bland den information som visuellt tas in kommer endast en liten del uppmärksammas och uppfattas av logganalytikern. Det som uppfattas kan antingen leda till

- direkta instinktiva handlingar, som att bortse från loggposter på grund av deras låga prioritet.
- behandling i närminnet för att därefter välja en mer genomtänkt åtgärder, till exempel efter att ha tittat på flera loggposter för att se om det finns något hotfullt mönster.
- inläsning i långtidsminnet eftersom informationen bedöms som intressant och tros kunna vara användbar i framtiden.

Intervjuer med logganalytiker pekar på att effektiv övervakning av ett system kräver kunskap om aktuella hot, IT-system, cybersäkerhet samt situations- och kontextrelaterad information om det egna systemet (såsom sårbarheter, vanligt systemanvändande och nätverksinställningar) [76], [81]. Sådan information borde därför vara bra att ha oavsett hur loggposter uppmärksammas av en logganalytiker.

Utöver förmågan att uppfatta inkommande loggar och matchningen mot relevant information är det rimligt att tro att uppmärksamhet är en viktig komponent för att lyckas i övervakningsfasen. Uppmärksamhet påverkar vilken sensorisk information som uppfattas, vilken information som finns i närminnet och vilken information som sparas i långtidsminnet. Om övervakning sker över lång tid på ett monotont sätt så är risken uppenbar att logganalytikerna inte klarar att fokusera på rätt saker. I enlighet med detta argumenterar såväl Thompson et al. [88] som Goodall et al. [76] för att informationen i larm och kalibrering av sensorerna är av mycket stor vikt för att inte logganalytikerna ska översvämmas av information, tröttna eller fästa uppmärksamheten fel.

Sammantaget är det rimligt att definiera logganalytikerns prestation i övervakningsfasen som förmågan att identifiera faktiska hot (loggposter som bör utredas vidare) utan att ägna för mycket tid åt felaktiga loggposter som inte utgör ett hot. Baserat på ovanstående text och erfarenheter från andra domäner är det rimligt att tro att följande variabler påverkar denna prestation: det visuella stöd logganalytikern får, den mängd loggposter som logganalytikern utsätts för,

logganalytikerns kognitiva förmågor (t.ex. uthållighet) och logganalytikerns kunskap om de områden som listas ovan.

Antalet kvantitativa studier som undersöker förmåga i denna fas är få. I ett experiment med simulerad data undersökte Sommestad och Hunstad [89] en IDS-logganalytikers förmåga att filtrera ut relevanta loggposter. Resultaten visar att logganalytikern kan reducera antalet falska loggposter utan att missa speciellt många riktiga attacker. Intrångsdetektionssystemet larmade om sex av tio angrepp och gav tio falsklarm för varje riktigt larm; logganalytikern larmade för fem av tio angrepp och gav drygt ett falsklarm för varje korrekt larm. Sawyer et al. [82] använde mätmetoden NASA-TLX för att undersöka mental arbetsbelastning när en logganalytiker gavs uppgiften att matcha IP-adresser i en tabell med två kolumner. Slutsatserna från studien visade att det var svårare att uppfatta korrelationer när tabellen med IP-adresser arrangerades om kontinuerligt och när attacker förekom sällan. Hur detta ska generaliseras till de uppgifter som en logganalytiker utför är oklart eftersom uppgiften som genomfördes under experimenten inte överensstämmer med de uppgifter som faktiskt genomförs av en logganalytiker i praktiken.

4.2 Orsaksanalys

I en orsaksanalys utgår logganalytikern från händelser som prioriterades i övervakningsfasen och som bedöms behövas analyseras ytterligare för att avgöra om det är nödvändigt att vidta någon form av åtgärd. Till skillnad från övervakningsfasen, som utgår från ett konstant flöde av loggposter och ständigt pågår, så varierar det hur ofta en logganalytiker påbörjar en orsaksanalys och hur länge en sådan varar. Informationen från övervakningsfasen, det vill säga loggdata från sensorer och annan information som var tillgänglig, fusioneras i orsaksanalysen med ytterligare information för att bättre kunna avgöra varför systemet larmat. Denna djupare analys handlar inte bara om att klassa larmen som korrekta eller felaktiga utan också ofta om att förstå orsakerna olika loggposter.

Det är rimligt att se analysfasen som ett antal iterationer där olika hypoteser kopplade till situationen testas. I förhållande till Wickens modell itereras då de fyra stegen inhämtning med sensorer, perception, val av åtgärd och utförande av åtgärd upprepade gånger för flera informationskällor som tros kunna ge viktiga ledtrådar. Fram till det att en slutsats nåtts består åtgärderna i denna process typiskt av att ta fram ny information av något slag. Den information som anses relevant i denna fas är i stor utsträckning densamma som den som anses relevant under övervakningsfasen:

- allmän kunskap om cybersäkerhet (t.ex. angrepp och skydd),
- information om det egna systemet (t.ex. sårbarheter, typiskt systemanvändande och nätverksinställningar),

- sensorernas förmåga och konfiguration och
- aktuella hot mot systemet.

Utöver detta kan logganalytikern under analysfasen ta sig tiden att hämta in mer information av detta slag, till exempel genom att detaljstudera loggar eller samtala med nätverksadministratörer och systemägare. Logganalytikern kan även hämta in annan information, till exempel genom att titta på loggar från andra system, logga in på maskiner för att undersöka dem, genomföra sökningar på internet eller genomföra analys av misstänkt binärkod som är inblandad i skapandet av loggar.

Logganalytikerns förmåga i denna fas kan ses som förmågan att under en begränsad tidsrymd identifiera faktiska attacker, intrång och missbruk samt erhålla en förståelse av hur dessa genomförts med tillgängliga informationskällor. Utöver analytikerns kunskap och tillgång till information kopplat till de fyra ovanstående punkterna är det rimligt att allmänna kognitiva förmågor påverkar framgång med orsaksanalys. I den mån ytterligare automatiserade analyser genomförs är det också rimligt att tänka sig att förtrogenhet med dessa analysverktyg är av betydelse.

Denna fas har studerats i två experiment som författarna av denna rapport känner till. I ett försök av Ben-Asher and Gonzalez [83] gavs deltagarna uppgiften att bedöma om ett systems status (till exempel i form av nätverkslast och aktiva tjänster) utgjorde en säkerhetsrisk. I denna förenklade logganalysuppgift upptäckte de att ökad kunskap inom cybersäkerhet gav bättre upptäckt av elakartade händelser, minskade felklassificeringen av godartade händelser och hjälpte logganalytikerna att identifiera vilken typ av elakartad aktivitet som skapade den aktuella situationen. Thompson et al. [85] testade förmågan att identifiera och klassificera angrepp med visualiseringsstöd och med kommandotolk. De fann att förbestämda uppgifter utfördes bättre med kommandotolk medan mer explorativa uppgifter som att se anomalier i nätverket utfördes bättre med visualiseringsstöd.

4.3 Utförande av åtgärd

I åtgärdsfasen vidgas lämpliga åtgärder utifrån arbetet som genomförts under övervakningen och orsaksanalysen. De vanligaste typerna av åtgärder är: *ingripanden*, *återkoppling* och *rapportering* [80]. Exempel på ingripanden är att dra ur nätverkssladden, ominstallera en dator eller uppdatera programvara. Återkoppling innebär vanligen att anpassa signaturernas regler i automationssystemet eller välja ny data att samla in. Rapportering innebär att sprida informationen om hotet till andra eller att fördjupa analysen av hotet (till exempel samla forensiska bevis och vidta juridiska åtgärder). Utöver detta är det också tänkbart att logganalytiker vill rekommendera åtgärder som vilseleder en angripare. Det kan till exempel finnas ett värde i att lura in en angripare i ett fiktivt

nät för att observera dess beteende eller plantera vilseledande information i system som angriparen kompromett erat.

Valet av åtgärd(er) beror på inte enbart på den aktuella händelsen utan även på logganalytikerns roll i organisationen och logganalytikerns mandat/ansvar/behörighet. Eftersom rätt åtgärd är beroende av incidenten och logganalytikerns mandat/ansvar/behörighet så är det svårt att fastslå en mall för exakt vilka åtgärder som bör genomföras vid uppmärksammande av olika attacker, intrång och missbruk. Cichonski et al. [77] menar att strategin för att begränsa skadeverkningarna bör ta hänsyn till: potentiell skada, krav på att bevara bevis, tillgänglighetskrav, kostnaden och effektiviteten (ur cybersäkerhetsperspektiv).

Trots att det är fullt möjligt att studera människor och deras agerande i åtgärdsfasen så verkar det inte finnas någon publicerad forskning som empirisk testar vad som påverkar prestation under åtgärdsfasen. Ben-Asher and Gonzalez [84] har dock testat hur detaljnivå i återkoppling till tidigare analyser (som ju skapas i åtgärdsfasen) påverkar framtida analysförmåga. Man fann som väntat att detaljerad återkoppling om vad som var korrekt ökade förmågan att identifiera vanliga angrepp, men också att det gav sämre förmåga att detektera nya typer angrepp jämfört med när återkopplingen var mer abstrakt.

4.4 Variabler av betydelse för manuell analys

Syftet med logganalytikerns arbete är i mångt och mycket att upprätthålla säkerheten genom att, med tillgänglig information, svara på de händelser som sker och de tillstånd som uppstår[11]. Detta kräver att övervakningen identifierar händelser och tillstånd som bör undersökas djupare, att orsaksanalysen identifierar den faktiska orsaken till händelser och tillstånd samt att rätt åtgärder utförs baserat på orsaksanalysen. Komplexiteten i detta arbete och den lösningsrymd som finns gör det svårt att peka ut de enskilda variabler som avgör hur väl detta fungerar. Författarna bedömer dock att variabler kopplade till följande bör ha stor betydelse i de flesta fall:

- **Arbetsplatsens utformning**. Till exempel den mängd loggposter en logganalytiker utsätts för i övervakningsfasen, hur grafiska gränssnitt är utformade och om logganalytikern har uppgifter som kräver multitasking.
- **Logganalytikerns kunskap**. Kunskap om cybersäkerhet, det egna systemet, sensorernas förmåga och konfiguration och aktuella hot är sådant som bör ha betydelse i den manuella analysen. Men det är också rimligt att tänka sig verksamhetens kostnader och mål har betydelse när

[11] Värt att notera är att ett annat tänkbart syfte med logganalys är att förstå angriparens kunskap, resurser och motiv. Eller att peka ut vem angriparen är med goda bevis. Detta lämnas dock utanför denna rapport och projektet ÖvExCy.

åtgärd ska tas fram samt att kunskap om fristående analysverktyg är av betydelse i orsaksanalysen.

- **Logganalytikerns informationsresurser och behörigheter.** Som alternativ och komplement till kunskap om cybersäkerhet, det egna systemet med mera så kan logganalytikerns ges informationsresurser och behörigheter som gör detta enkelt att ta reda på. I [90] listas 33 resurser som det anses av betydelse att det finns tillgängligt för en analytiker i ett cyber security operations center. Till exempel anses det att det är av hög betydelse att man har tillgång till internet och det rådata som skapat loggarna, ganska viktigt att man har översiktliga nätverkskartor och av mindre betydelse att man har tillgång till aktuella konfigurationer på nätverksutrustning.
- **Logganalytikerns allmänna kognitiva förmåga.** Såväl analytisk förmåga som korttidsminne och långtidsminne behövs i den manuella analysen. Dessutom kan god social förmåga vara av värde i de fall intern och extern kommunikation är nödvändig.

Då den manuella och automatiska analysen har samma syfte – att identifiera och utföra en lämplig åtgärd – finns vissa likheter mellan dem. Kanske framförallt när det kommer till vilken information som är användbar för analysen.

5 Sammanfattning och exempel på forskningsfrågor inom logganalys

Denna rapport har beskrivit logganalys utifrån tre delar:

- insamling av information
- automatisk analys
- manuell analys.

Alla dessa delar syftar till att en logganalytiker med stöd av olika verktyg ska utföra åtgärder som höjer cybersäkerheten i det övervakade systemet. Logganalys och åtgärder kan utföras i (nära) realtid för att verka mot pågående hot, men (som ofta är fallet) också undersöka historiska händelser för att kartlägga tidigare intrång och missbruk. Det kan till exempel handla om ingripanden där behörigheter i systemet ändrats, återkoppling som gör insamlingen av information eller automatisk analys effektivare, eller rapportering av trender till andra systemintressenter.

Det finns en stor mängd variabler som har en tämligen uppenbar påverkan på logganalysförmågan (se avsnitten 2.4, 3.4 och 4.4), varav några mer etablerade idéer beskrivs i Tabell 1. Det bör tilläggas att Tabell 1 inte är en komplett bild, utan ett urval av variabler som författarna bedömt vara av särskilt signifikanta för logganalysförmåga. Exempelvis är den utförandes kunskap och förståelse för hot naturligtvis inte bara viktiga vid manuell analys, utan även vid konfiguration av sensorer och skrivande av regler för automatisk analys.

Tabell 1. Översikt över variabler av vikt för olika delar av logganalysförmågan.

Logganalysförmåga	**Variabler av vikt**
Insamling information	Val och placering av sensorer Systemkartläggning Hotunderrättelser
Automatisk analys	Analystyp Träffsäkerhet Prestanda
Manuell analys	Arbetsplatsen utformning Logganalytikerns kunskap Logganalytikerns informationsresurser och behörigheter Logganalytikerns allmänna kognitiva förmåga

Som rapporten visat finns tyvärr begränsat med tillgänglig forskning som belyser hur viktiga dessa variabler är. Det finns till exempel inga tillgängliga tester av hur logganalytikers informationsresurser påverkar deras förmåga att skapa sig en lägesbild. Det finns inte heller forskning som reder ut vilka relationer som finns mellan variablerna. Till exempel är det rimligt att tänka sig att

- bättre systemkartläggningar (t.ex. sådana man får av nätverksskanners) möjliggör noggrannare analystyper (t.ex. med loggposter med mer kontextinformation)
- olika analystyper (t.ex. attackgrafer kontra enkla loggposter) är olika värdefulla vid olika tidpunkter (t.ex. forensiskt arbete kontra incidenthantering)
- vad som är en lämplig utformning av arbetsplatsen (t.ex. mängden loggposter) hänger ihop med vilken typ av analys som produceras (t.ex. enkla loggposter eller komplexa attackgrafer).

Det finns även skäl att tro att variabler hänger samma på komplexa sätt en abstraktionsnivå nedanför den som visas i Tabell 1. Till exempel är det rimligt att tänka sig att träffsäkerhet i den automatiska analysen kräver både att den modell som används för att klassificera angrepp är riktig och att den serveras med de händelser som är av relevanta. Det är också rimligt att tänka sig att logganalytikerns kunskap och kognitiva förmåga tillsammans med egenskaper

kopplade till hotet spelar en roll för hur väl modellen kan trimmas in mot den miljö som den ska övervaka.

Det finns en uppsjö frågor kopplat dessa variabler som skulle vara bra att besvara. Forskningsfrågor (med de mest relaterade variablerna inom parantes) som anses prioriterade att söka svar på inom ÖvExCy är beskrivna av följande uppräkning utan någon inbördes prioritering. I bilaga A beskrivs varje forskningsfråga tillsammans med en kort beskrivning av ett exempel på ett experiment eller test som skulle kunna svara på frågan.

1. Hur bra stämmer allmänt tillgänglig information om attackkoder med deras egenskaper? (Hotunderrättelser)
2. Vad predikterar träffsäkerheten för intrångsdetektionsregler i existerande system? (Hotunderrättelser & Träffsäkerhet)
3. Hur välanpassade behöver modeller av elakartad aktivitet vara för att de ska vara till nytta? (Träffsäkerhet)
4. Hur väl fungerar modeller av godartat beteende under ideala förhållanden? (Träffsäkerhet)
5. Vilken kontext- eller situationsbaserad information som behövs för att göra bra manuellt analysarbete? (Logganalytikerns kunskap, informationsresurser och behörigheter)
6. Hur viktig är förståelse för hur den automatiserade analysen fungerar? (Logganalytikerns kunskap)

Forskningsfrågorna och de förslagna testerna är på intet sätt uttömmande. De är enbart indikativa för vad som anses vara intressant, relevant och möjligt att besvara med de resurser som projektet ÖvExCy har. Det ses till exempel inte som prioriterat att undersöka hur man åstadkommer snabba (det vill säga med hög prestanda) automatiska analyser och det finns inga planer på att genomföra tester av hur allmänna kognitiva förmågor påverkar en logganalytikers prestation. Däremot finns flera prioriterade frågor kopplade till vilken kunskap och information en logganalytiker behöver och träffsäkerheten hos de lösningar som finns att välja bland för automatisk analys anses dock högst relevant.

6 Referenser

[1] K. Lundholm, J. Löfvenberg, A. Hunstad, and H. Karlzén, "Lägesbild på informationsarenan Översikt och diskussion (FOI-R--3240--SE)," Linköping, Sweden, 2011.

[2] U. Franke and J. Brynielsson, "Cyber situational awareness – A systematic review of the literature," *Comput. Secur.*, vol. 46, pp. 18–31, Oct. 2014.

[3] M. R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems," *Hum. Factors*, vol. 37, no. 1, pp. 32–64, 1995.

[4] J. Llinas and D. L. Hall, "An introduction to multi-sensor data fusion," *ISCAS '98. Proc. 1998 IEEE Int. Symp. Circuits Syst. (Cat. No.98CH36187)*, vol. 6, no. 1, 1998.

[5] K. Kent and M. Souppaya, "Guide to computer security log management," *NIST Spec. Publ.*, vol. 800, no. 92, 2006.

[6] S. Noel and S. Jajodia, "Optimal IDS sensor placement and alert prioritization using attack graphs," *J. Netw. Syst. Manag.*, vol. 16, no. 3, pp. 259–275, 2008.

[7] H. Holm, T. Sommestad, J. Almroth, and M. Persson, "A quantitative evaluation of vulnerability scanning," *Inf. Manag. Comput. Secur.*, vol. 19, no. 4, pp. 231–247, 2011.

[8] T. Sommestad, M. Ekstedt, and H. Holm, "The Cyber Security Modeling Language – A Tool for Vulnerability Assessments of Enterprise System Architectures," pp. 1–12, 2011.

[9] M. Farwick, B. Agreiter, R. Breu, S. Ryll, K. Voges, and I. Hanschke, "Automation processes for enterprise architecture management," *Proc. - IEEE Int. Enterp. Distrib. Object Comput. Work. EDOC*, no. OCTOBER, pp. 340–

349, 2011.

[10] M. Farwick, B. Agreiter, R. Breu, S. Ryll, K. Voges, and I. Hanschke, “A Requirements Analysis based on a Literature Review and an Exploratory Survey,” *13th Int. Conf. Enterp. Inf. Syst. (ICEIS), Beijing*, no. July 2015, pp. 325–337, 2011.

[11] Försvarsmakten, “Försvarsmaktens Underrättelsereglemente (FM UndR),” Stockholm, 2010.

[12] Försvarsmakten, “Handbok för Försvarsmaktens säkerhetstjänst, Hotbedömning (H Säk Hot),” Stockholm, 2006.

[13] ENISA, *Actionable Information for Security Incident Response*, no. November. 2014.

[14] S. Axelsson, “Intrusion detection systems: A survey and taxonomy,” Göteborg, Sweden, 2000.

[15] Y. Chen, Y. Li, X. Cheng, and L. Guo, “Survey and taxonomy of feature selection algorithms in intrusion detection system,” *Inf. Secur. Cryptol.*, pp. 153–167, 2006.

[16] I. Corona, G. Giacinto, C. Mazzariello, F. Roli, and C. Sansone, “Information fusion for computer security: State of the art and open issues,” *Inf. Fusion*, vol. 10, no. 4, pp. 274–284, Oct. 2009.

[17] H. Debar, M. Dacier, and A. Wespi, “Towards a taxonomy of intrusion-detection systems,” *Comput. Networks*, vol. 31, no. 8, pp. 805–822, Apr. 1999.

[18] P. Garciateodoro, J. Diazverdejo, G. Maciafernandez, and E. Vazquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Comput. Secur.*, vol. 28, no. 1–2, pp. 18–28, Feb. 2009.

[19] K. A. Jackson, “INTRUSION DETECTION SYSTEM (IDS) PRODUCT SURVEY.”

[20] K. Julisch, “Data mining for intrusion detection,” *Appl. data Min. Comput. Secur.*, pp. 1–14, 2002.

[21] P. Kabiri and A. Ghorbani, “Research on intrusion detection and response: A survey,” *Int. J. Netw. Secur.*, vol. 1, no. 2, pp. 84–102, 2005.

[22] R. A. Kemmerer and G. Vigna, “Intrusion detection: a brief history and overview,” *Computer (Long. Beach. Calif).*, vol. 35, no. 4, pp. supl27–supl30, Apr. 2002.

[23] J. Kim, P. J. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, “Immune system approaches to intrusion detection – a review,” *Nat. Comput.*, vol. 6, no. 4, pp. 413–466, Jan. 2007.

[24] A. Lazarevic, V. Kumar, and J. Srivastava, “Intrusion detection: A survey,” *Manag. Cyber Threat.*, 2005.

[25] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion detection system: A comprehensive review,” *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013.

[26] W. Lee, S. Stolfo, and K. Mok, “A data mining framework for building intrusion detection models,” *... 1999 IEEE Symp.*, 1999.

[27] E. Lundin and E. Jonsson, “Chalmers Publication Library Survey of Intrusion Detection Research,” 2006.

[28] T. Lunt, “A survey of intrusion detection techniques,” *Comput. Secur.*, 1993.

[29] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, “A

survey of intrusion detection techniques in Cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, Jan. 2013.

[30] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Networks*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007.

[31] A. Patel, Q. Qassim, and C. Wills, "A survey of intrusion detection and prevention systems," *Inf. Manag. Comput. Secur.*, vol. 18, no. 4, pp. 277–290, 2010.

[32] J. S. Sherif and T. G. Dearmond, "Intrusion detection: systems and models," *Proceedings. Elev. IEEE Int. Work. Enabling Technol. Infrastruct. Collab. Enterp.*, pp. 115–133.

[33] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, Dec. 2009.

[34] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA on-line intrusion detection evaluation," *Comput. Networks*, vol. 34, 2000.

[35] J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, Nov. 2000.

[36] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, no. Cisda, pp. 1–6.

[37] Y. Meng and L. F. Kwok, "Adaptive blacklist-based packet filter with a statistic-based approach in network intrusion detection," *J. Netw. Comput. Appl.*, vol. 39, no. 1, pp. 83–92, 2014.

[38] G. Suarez-Tangil, E. Palomar, A. Ribagorda, and I. Sanz, “Providing SIEM systems with self-adaptation,” *Inf. Fusion*, vol. 21, no. 1, pp. 145–158, 2015.

[39] H. Debar and A. Wespi, “Aggregation and correlation of intrusion-detection alerts,” *Recent Adv. Intrusion Detect.*, 2001.

[40] H. W. Njogu and L. Jiawei, “Using Alert Cluster to reduce IDS alerts,” in *2010 3rd International Conference on Computer Science and Information Technology*, 2010, pp. 467–471.

[41] B. Zhu and A. a. Ghorbani, “Alert correlation for extracting attack strategies,” *Int. J. Netw. Secur.*, vol. 3, no. 3, pp. 244–258, 2006.

[42] J. Carlson and M. Bishop, “Verify Results of Network Intrusion Alerts Using Lightweight Protocol Analysis,” *21st Annu. Comput. Secur. Appl. Conf.*, vol. 1, no. 4, pp. 117–126, 2005.

[43] R. Gula, “Correlating ids alerts with vulnerability information,” vol. 2011, no. Revision 4, 2002.

[44] S. Neelakantan and S. Rao, “A Threat-Aware Signature Based Intrusion-Detection Approach for Obtaining Network-Specific Useful Alarms,” in *2008 The Third International Conference on Internet Monitoring and Protection*, 2008, pp. 80–85.

[45] D. Gupta, P. S. Joshi, a. K. Bhattacharjee, and R. S. Mundada, “IDS alerts classification using knowledge-based evaluation,” *2012 Fourth Int. Conf. Commun. Syst. Networks (COMSNETS 2012)*, pp. 1–8, Jan. 2012.

[46] F. Gagnon, F. Massicotte, and B. Esfandiari, “Using Contextual Information for IDS Alarm Classifi□cation,” in *Proceedings of International Conference, DIMVA*, 2009, pp. 147–156.

[47] H. W. Njogu, L. Jiawei, J. N. Kiere, and D. Hanyurwimfura, “A comprehensive vulnerability based alert management approach for large networks,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 27–45, Jan. 2013.

[48] J. Haukeli, “False positive reduction through IDS network awareness,” Oslo University College, 2012.

[49] M. Xiao and D. Xiao, “Alert Verification Based on Attack Classification in Collaborative Intrusion Detection,” *Eighth ACIS Int. Conf. Softw. Eng. Artif. Intell. Networking, Parallel/Distributed Comput. (SNPD 2007)*, pp. 739–744, Jul. 2007.

[50] P. Ning, Y. Cui, and D. S. Reeves, “Constructing attack scenarios through correlation of intrusion alerts,” in *Proceedings of the 9th ACM Conference*, 2002.

[51] W. Wang and T. E. Daniels, “A Graph Based Approach Toward Network Forensics Analysis,” *ACM Trans. Inf. Syst. Secur.*, vol. 12, no. 1, pp. 1–33, Oct. 2008.

[52] E. Totel, B. Vivinis, and L. Mé, “A Language Driven Intrusion Detection System for Event and Alert Correlation,” *Secur. Prot. Inf. Process. Syst.*, vol. 147, pp. 208–224, 2004.

[53] S. Noel, E. Robertson, and S. Jajodia, “Correlating intrusion events and building attack scenarios through attack graph distances,” *Proc. - Annu. Comput. Secur. Appl. Conf. ACSAC*, pp. 350–359, 2004.

[54] X. Qin and W. Lee, “Attack plan recognition and prediction using causal networks,” *Proc. - Annu. Comput. Secur. Appl. Conf. ACSAC*, pp. 370–379, 2004.

[55] L. Wang, A. Liu, and S. Jajodia, “Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts,” *Comput. Commun.*, vol. 29, no. 15, pp. 2917–2933, 2006.

[56] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, S. Dubus, and A. Martin, "Success Likelihood of Ongoing Attacks for Intrusion Detection and Response Systems," *2009 Int. Conf. Comput. Sci. Eng.*, pp. 83–91, 2009.

[57] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," *Proc. IEEE Symp. Secur. Priv.*, pp. 305–316, 2010.

[58] D. E. Denning, "An Intrusion-Detection Model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[59] S. S. Kim and M. Vannucci, "Detecting Traffic Anomalies at the Source through aggregate analysis of packet header data," pp. 1–12, 1998.

[60] M. Mahoney and P. K. Chan, "PHAD: Packet header anomaly detection for identifying hostile network traffic," *Florida Inst. Technol. Tech. Rep. CS-2001-04*, no. 1998, pp. 1–17, 2001.

[61] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: alternative data models," *Proc. 1999 IEEE Symp. Secur. Priv. (Cat. No.99CB36344)*, 1999.

[62] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion Detection using Sequences of System Calls," *J. Comput. Secur.*, vol. 6, no. 3, pp. 151–180, 1998.

[63] D. Kang, D. Fuller, and V. Honavar, "Learning Classifiers for Misuse Detection Using," pp. 511–516, 2005.

[64] E. Eskin, W. L. W. Lee, and S. J. Stolfo, "Modeling system calls for intrusion detection with dynamic windowsizes," *Proc. DARPA Inf. Surviv. Conf. Expo. II. DISCEX'01*, vol. 1, 2001.

[65] H. H. Feng, O. M. Kolesnikov, P. Fogla, W. Lee, and W. G. W. Gong, “Anomaly detection using call stack information,” *2003 Symp. Secur. Privacy, 2003.*, 2003.

[66] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, p. 219, 2004.

[67] R. Sekar, a Gupta, J. Frullo, T. Shanbhag, a Tiwari, H. Yang, and S. Zhou, “Specification-based anomaly detection: a new approach for detecting network intrusions,” *Proc. 9th ACM Conf. Comput. Commun. Secur.*, vol. 26, no. 2, pp. 265–274, 2002.

[68] J. M. Estevez-Tapiador, P. Garcia-Teodoro, and J. E. Diaz-Verdejo, “Stochastic protocol modeling for anomaly based network intrusion detection,” *First IEEE Int. Work. Inf. Assur. 2003. IWIAS 2003. Proceedings.*, no. August 2015, 2003.

[69] F. Chuan, P. Jianfeng, Q. Haiyan, and J. W. Rozenblit, “Alert fusion for a computer host based intrusion detection system,” *Proc. Int. Symp. Work. Eng. Comput. Based Syst.*, pp. 433–440, 2007.

[70] N. B. Anuar, M. Papadaki, S. Furnell, and N. Clarke, “An investigation and survey of response options for Intrusion Response Systems (IRSs),” *Proc. 2010 Inf. Secur. South Africa Conf. ISSA 2010*, 2010.

[71] A. Shameli-Sendi, M. Cheriet, and A. Hamou-Lhadj, “Taxonomy of intrusion risk assessment and response system,” *Comput. Secur.*, vol. 45, no. SEPTEMBER 2014, pp. 1–16, 2014.

[72] T. Toth and C. Kruegel, “Evaluating the impact of automated intrusion response mechanisms,” *18th Annu. Comput. Secur. Appl. Conf. 2002. Proceedings.*, 2002.

[73] P. Mell, V. Hu, R. Lippmann, J. W. Haines, and M. Zissman, “An overview of

issues in testing intrusion detection systems, (NIST IR 7007)," Citeseer, Gaithersburg, MD, USA, 2003.

[74] S. Van Wouw, J. Viña, A. Iosup, and D. Epema, "An Empirical Performance Evaluation of Distributed SQL Query Engines Categories and Subject Descriptors."

[75] R. Werlinger, K. Muldner, K. Hawkey, and K. Beznosov, "Preparation, detection, and analysis: the diagnostic work of IT security incident response," *Inf. Manag. Comput. Secur.*, vol. 18, no. 1, pp. 26–42, 2010.

[76] J. R. Goodall, W. G. Lutters, and A. Komlodi, "I know my network: collaboration and expertise in intrusion detection," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004, pp. 342–345.

[77] P. Cichonski and K. Scarfone, "Computer Security Incident Handling Guide Recommendations of the National Institute of Standards and Technology (SP 800-61)," Gaithersburg, MD, USA, 2012.

[78] R. Werlinger, K. Hawkey, K. Muldner, P. Jaferian, and K. Beznosov, "The challenges of using an intrusion detection system: is it worth the effort?," *SOUPS '08 Proc. 4th Symp. Usable Priv. Secur.*, no. 1, pp. 107–118, 2008.

[79] R. Werlinger, K. Muldner, K. Hawkey, and K. Beznosov, "Towards Understanding Diagnostic Work During the Detection and Investigation of Security Incidents," in *Proceedings of the Third International Symposium on Human Aspects of Information Security & Assurance (HAISA 2009)*, 2009, no. Haisa, p. 119.

[80] J. R. Goodall, W. G. Lutters, and A. Komlodi, "Developing expertise for network intrusion detection," *Inf. Technol. People*, vol. 22, no. 2, pp. 92–108, 2009.

[81] T. Sommestad and A. Hunstad, "Intrusion detection and the role of the system

administrator," *Inf. Manag. Comput. Secur.*, no. HAISA 2012 special issue, 2013.

[82] B. D. Sawyer, V. S. Finomore, G. J. Funke, V. F. Mancuso, M. E. Funke, G. Matthews, and J. S. Warm, "Cyber Vigilance: Effects of Signal Probability and Event Rate," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 58, no. 1, pp. 1771–1775, Sep. 2014.

[83] N. Ben-Asher and C. Gonzalez, "Effects of cyber security knowledge on attack detection," *Comput. Human Behav.*, vol. 48, pp. 51–61, Jul. 2015.

[84] N. Ben-Asher and C. Gonzalez, "Training for the unknown : The role of feedback and similarity in detecting zero- day attacks," in *6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences*, 2015, no. August.

[85] R. S. Thompson, E. M. Rantanen, W. Yurcik, and B. P. Bailey, "Command line or pretty lines?: comparing textual and visual interfaces for intrusion detection," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2007, p. 1205.

[86] H. Shiravi, A. Shiravi, and A. a. Ghorbani, "A survey of visualization systems for network security," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 8, pp. 1313–1329, 2012.

[87] C. Wickens, J. Hollands, S. Banburry, and R. Parasuraman, *Engineering Psychology and Human Performance*. New York, New York, USA: Pearson Education, 2013.

[88] R. S. Thompson, E. M. Rantanen, and W. Yurcik, "Network intrusion detection cognitive task analysis: Textual and visual tool usage and recommendations," in *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 2006, vol. 50, no. 5, pp. 669–673.

[89] T. Sommestad and A. Hunstad, "Intrusion detection and the role of the system

administrator," *Inf. Manag. Comput. Secur.*, vol. 21, no. 1, pp. 30 – 40, 2013.

[90] C. Zimmerman, *Ten Strategies of a World-Class Cybersecurity Operations Center*. Bedford, MA: The MITRE Corporation, 2014.

[91] M. Hafiz and M. Fang, "Game of detections: how are security vulnerabilities discovered in the wild?," *Empir. Softw. Eng.*, Sep. 2015.

[92] T. Sommestad and F. Sandström, "An empirical test of the accuracy of an attack graph analysis tool," *Inf. Comput. Secur.*

[93] J. D. Lee, K. A. See, and I. City, "Trust in Automation: Designing for Appropriate Reliance," in *Human Factors*, 2004, vol. 46, no. 1, pp. 50–80.

[94] K. Goddard, A. Roudsari, and J. C. Wyatt, "Automation bias: a systematic review of frequency, effect mediators, and mitigators," *J. Am. Med. Informatics Assoc.*, vol. September, pp. 121–127, 2012.

[95] J.-Y. Jian, A. M. Bisantz, and C. G. Drury, "Foundations for an Empirically Determined Scale of Trust in Automated System.," *Int. J. Cogn. Ergon.*, vol. 4, no. 1, p. 53, 2000.

Bilaga A

Denna bilaga beskriver ett antal forskningsfrågor som skulle kunna studeras av ÖvExCy tillsammans med metoder som kan nyttjas för att svara på dessa forskningsfrågor. Metodvalen begränsas i viss mån av de experimentella resurser som finns att tillgå i övningsanläggningen CRATE. Vissa av dessa resurser fanns tillgängliga innan projektets start, såsom ett webbgränssnitt för att konfigurera virtuella nätverk och maskiner. Andra resurser utvecklas inom ramen för ÖvExCy för att möjliggöra mer omfattande och kontrollerade experiment. Ett sådant exempel är SVED (Software Vulnerabilities, Exploits and Detection), ett verktyg som används för att på ett enkelt sätt skapa attackgrafer som automatiskt kan exekveras mot systemkonfigurationer i CRATE. SVED kan därmed användas för att skapa och automatiskt utföra intrång och missbruk riktat mot system i CRATE på ett kontrollerat och reproducerbart sätt. Ett annat exempel på resurser som tillförs till CRATE i och med ÖvExCy är verktyg för att simulera godartat systemanvändande – det logganalysen ska kunna särskilja från intrången och missbruket.

I exemplen på forskningsfrågor som ges nedan beskrivs ofta en övergripande forskningsfråga först, för att sedan brytas ned i delfrågor. Det ges en bakgrund till frågan och delfrågorna, beskrivs en metod som skulle kunna användas för ge svar på frågorna och det spekuleras kring hur testerna skulle ge nytta i form av ökade möjligheter att skapa god logganalysförmåga. Det ska noteras att exemplen nedan är långt ifrån färdiga planer på experiment – många detaljer utelämnas och variabler operationaliseras/definieras inte fullständigt. Det ska också noteras att en aktivitet i CRATE kan ge svar kopplade till flera av dessa forskningsfrågor samtidigt. Till exempel kan den loggdata som produceras under en övning användas för att besvara många frågor kopplade träffsäkerhet.

6.1 Hur bra stämmer allmänt tillgänglig information om attackkoder med deras egenskaper?

6.1.1 Bakgrund

Förståelse för de hot som kan tänkas uppstå mot ett system är av central vikt för förmågan att detektera och begränsa deras verkan. Till exempel krävs en teknisk förståelse av cyberattacker för att kunna skriva effektiva regler för intrångsdetekteringssystem. Dessutom är en generell förståelse för vilka hot som är rimliga något som bör öka en logganalytikers förmåga att identifiera faktiska intrång och missbruk från stora mängder larm. Allmän kunskap om detta är därför önskvärt att bygga upp.

Ett par frågor rörande hotmodeller som skulle kunna vara relevanta att studera inom ramen för ÖvExCy är:

1) **Hur har attackkoder för olika sårbarheter förändrats över tiden**? Till exempel, hur har olika skyddsmekanismer såsom Data Execution Prevention (DEP) och Address Space Layout Randomization (ASLR), safeSEH och stack cookies påverkat utvecklandet av attackkoder?

2) **Hur ofta är en skyddsmekanism anledningen till att en attackkod misslyckas?** Till exempel, hur ofta är ASLR orsaken till att de misslyckas och hur ofta är orsaken en felaktig konstruktion av attackkoden eller bara en saknad ”offset” för applikationskonfigurationen?

3) **Hur väl överensstämmer en attackkods bedömda tillförlitlighet med dess faktiska tillförlitlighet**? Till exempel betygssätts tillförlitlighet för attackkoder i ramverket Metasploit efter en skala från Manual (”*The exploit is unstable or difficult to exploit and is basically a DoS. This ranking is also used when the module has no use unless specifically configured by the user (e.g.: php_eval).*”) till Excellent (“*The exploit will never crash the service. This is the case for SQL Injection, CMD execution, RFI, LFI, etc. No typical memory corruption exploits should be given this ranking unless there are extraordinary circumstances (WMF Escape())*”)[12].

6.1.2 Metod

Den första av de tre frågorna ovan besvaras lämpligen genom en blandning av manuella och automatiska analyser av attackkoder, där varje attackkod bedöms enligt väl valda kriterier. Attackkoder kan hämtas från databasen Exploit DB, som är det mest populära publiceringsforumet för attackkoder [91]. Den 23e oktober 2015 fanns det 34790 färdiga attackkoder i Exploit DB.

Fråga två och tre studeras lämpligen genom att utföra attacker mot existerande konfigurationer i CRATE. Maskiner i CRATE kan kartläggas med sårbarhetskannrar och det konfigurationsverktyg som används i CRATE. Resultatet av kartläggningen kan användas av SVED för att automatiskt generera attacker som täcker in alla relevanta attackkoder i Exploit DB. Exekvering av attackgraferna avslöjar vilka attacker som lyckas och misslyckas; en blandning av manuell och automatisk analys avslöjar orsakerna till detta.

6.1.3 Förväntad nytta

En större förståelse av cyberattacker ökar förmågan att förstå vilken information som bör samlas in, hur den bör analyseras samt presenteras för en logganalytiker.

[12] https://github.com/rapid7/metasploit-framework/wiki/Exploit-Ranking

Exempelvis kan det identifieras trender och hypoteser för hur framtida okända angrepp (så kallade 0-days) kan tänkas ske.

6.2 Vad predikterar träffsäkerheten för intrångsdetektionsregler i existerande system?

6.2.1 Bakgrund

Att skriva regler till intrångsdetektionssystem är vanligen ett svårt och tidskrävande manuellt arbete. Detta gör att kvaliteten för en regel, och i förlängningen möjligheten att upptäcka cyberattacker, varierar från fall till fall. Det är därmed av intresse att försöka kartlägga vad som påverkar effektiviteten för olika typer av existerande regler och vilka attacker som är lättare att upptäcka än andra med dessa. Ett par preliminära frågor rörande detta ämne är:

1) **Är attacker som har fler unika regler enklare eller svårare att detektera**? Detektorer har ofta flera regler som rör en och samma attack. Hur triggas dessa vid exekvering av den eftersökta attacken, eller annan aktivitet som inte är tänkt att trigga dem? Påverkar antalet unika regler detektionsförmågan av en attack? Är anledningen bakom flera regler generellt sett att attacken är mer attraktiv att fånga eller att den helt enkelt är svårare att fånga (t.ex. att attacken kommer i många olika varianter)? Exempelvis har CVE-2008-4250 (en server-sårbarhet i Microsoft Windows som nyttjas av Stuxnet och Conficker) fler än 100 signaturer medan CVE-2007-0071 (en klient-sårbarhet i Adobe Flash) totalt 5 signaturer. Hur påverkar detta deras detektion?

2) **Hur mycket påverkar medieexponeringen av en sårbarhet eller attack effektiviteten för de motsvarande intrångsdetektionsreglerna**? Med andra ord, är sannolikheten större att detektera en cyberattack som är ”het” i media? Exempelvis, är det mer sannolikt att upptäcka attacker mot CVE-2008-4250 än CVE-2007-0071 (se föregående punkt)?

3) **Hur mycket påverkar egenskaperna för en attackerad sårbarhet effektiviteten för de motsvarande intrångsdetektionsreglerna?** Vanligen beskrivs sårbarheter enligt Common Vulnerability Scoring System (CVSS), som även kvantifierar hur allvarlig varje sårbarhet är, och med Common Weakness Enumeration (CWE), som beskriver vilken klass sårbarheten tillhör. Exempelvis har CVE-2008-4250 ett CVSS-värde på 10.0 (det högsta möjliga) och är av klassen CWE-94 (kodinjektion). En hypotes skulle vara att en sårbarhet som enligt CVSS kräver högre interaktion ger större möjlighet för detektion; en annan tänkbar hypotes är att en högre konsekvens för

sårbarheten (t.ex. att den ger administratörsprivilegier) gör den viktigare att detektera och att därmed mer prioriterad av regelskapare.

4) **Hur bra är olika regler på att upptäcka olika typer av obfuskerade attacker**? Det finns idag mängder av sätt att på enkla sätt obfuskera attacker. Exempelvis erbjuder Metasploit i grundutförande ett flertal olika kodare som med några knapptryckningar kan obfuskera injicerade payloads, och på samma sätt kan en buffert för en buffertöverskridningsattack skrivas om till i stort sett vad som helst.

5) **Hur väl motsvarar intrångsdetektionsreglers uppskattade effektivitet deras faktiska effektivitet**? Vissa analysverktyg, såsom Snort, har egenskattade värderingar av deras effektivitet i form av false positives och/eller false negatives. Hur korrelerar detta med verkligheten?

6.2.2 Metod

De fem frågorna kan besvara med ungefär samma experiment, och kan därmed med fördel sannolikt besvaras parallellt. Först kan dokumentationsstudier utföras av regelverk i intrångsdetektionssystem och medieexponeringen för attackerna och sårbarheterna som regelverken täcker in. Sedan implementeras relevanta konfigurationer i CRATE med lämplig simulerat användarbeteende (t.ex. användande av delade mappar, webbsurfande och epostinteraktion). Slutligen implementeras och exekveras motsvarande attackgrafer i SVED via dess existerande automatik. För att svara på fråga fyra krävs en utökning av SVED:s eftersom varje attack måste omkodas på olika sätt och utföras lika många gånger som antalet sätt de kodas. Detta är dock möjligt att automatisera via en tilläggsmodul till SVED.

6.2.3 Förväntad nytta

Bättre svar på denna fråga kommer ge kunskap om när man kan lita på en detektionsregel och allmän kunskap om hur väl tillgängliga regeluppsättningar fungerar.

6.3 Hur välanpassade behöver modeller av elakartad aktivitet vara för att de ska vara till nytta?

6.3.1 Bakgrund

Inom logganalys skapas ofta modeller av angrepp för att kunna upptäcka angreppen när de materialiserar sig. Signaturer som de i sensorn Snort utgör en

förhållandevis atomär modell över ett angrepp. En sekvens eller annan kombination av sådana signaturer skulle bilda en mer komplex modell över angrepp. Tanken är att en mer komplex modell ska hjälpa till att filtrera bort loggposter som inte orsakas av angrepp, utan som inträffar av rena tillfälligheter. De mer komplexa modellerna kan också hjälpa till att filtrera bort loggposter orsakas av misslyckade eller ofullständiga angrepp och som därför är av mindre intresse för logganalytikern. Dessutom kan en mer komplex modell över angrepp användas till att göra analysresultatet tydligare och mer precist. Till exempel genom att tydligare peka på vilka steg angriparen tagit. Men en felaktig modell kan också stjälpa logganalysen genom att filtrera bort angrepp som är relevanta.

Två grundläggande fråga kopplad till idén om attackmodeller är:

1) **Hur komplexa behöver modellerna vara för att vara till nytta?** Modeller kan vara enkla, och till exempel kräva att två händelser ska inträffa inom samma tidsram. De kan också vara komplexa och till exempel utrycka sannolikheter för att olika steg tas efter varandra under ett angrepp.

2) **Hur korrekta behöver modellerna vara för det övervakade systemet?** Modellerna kan också vara allmänna och baseras på generell kunskap om hur angrepp bör gå till eller vara väldigt specifika och baseras på information om det övervakade nätet och dess hotbild. Dessutom finns en risk att det baserade på felaktig information i någon utsträckning, till exempel för att den allmänna hotbilden är inkorrekt eller för informationen om det egna systemet är inaktuell.

6.3.2 Metod

Det finns idag ett stort antal förslag på hur modeller av elakartad aktivitet ska konstrueras. I punktlistan under avsnitt 3.1 ges exempel på sådana modeller. Den första av de två frågorna skulle kunna adresseras genom att existerande förslag på angreppsmodeller rankas som mer eller mindre komplexa och testas mot angrepp i CRATE. I punktlistan i avsnitt 3.1 finns bland annat förslag på modeller som bygger på attackgrafer. Eftersom varianter med olika komplexitet finns för denna typ av modell är den sannolikt lämplig att använda som utgångspunkt för tester. Attackgrafer kan skapa en modell över möjliga attackvägar baserat på information som:

- möjligheten att kommunicera mellan maskiner i nätverket
- sårbarheter i nätverket som angriparen tros kunna utnyttja
- behörigheter som användare har
- behörigheter som sårbarheter kan ge

- annan information kopplat till sårbarheterna (t.ex. enkelhet att utnyttja).

De två första av dessa är mer eller mindre nödvändiga för att skapa en attackgraf medan övriga kan användas för att skapa mer komplexa (och eventuellt mer träffsäkra) modeller.

Under ett test kan antalet informationstyper som används och relationerna mellan dessa som användas som ett mått komplexitet. Hur korrekt informationen är, det vill säga hur väl den stämmer med verkligheten, kan varieras i flera tydliga nivåer. Perfekt information kan ges genom modellen konstrueras med information som är tagen direkt från de konfigurationsverktyg som används i CRATE – det vill säga systemkonfiguration från konfigurationsverktyget och information kopplat till hotbild från SVED. Lägre nivåer av korrekthet kan skapas genom att fel injiceras i systemkonfigurationsinformationen och modellen av angriparens förmåga. Till exempel kan behörigheter för en viss andel av användarna sättas felaktigt eller en viss andel av brandväggsreglerna kan tas bort för att återspegla felaktig information om konfigurationen. För att återspegla inkorrekt information om hotet kan till exempel en viss andel sårbarheter som attacker saknas för kan lämnas kvar i modellen eller en viss andel fungerande angrepp kan tas bort ur modellen.

6.3.3 Förväntad nytta

Modeller över angrepp fungerar rimligtvis bra givet att de serveras med korrekt data. Men detta är inte självklart. Tidigare tester av attackgrafer har identifierat felaktigheter angående de antaganden som modeller bygger på [92]. Kunskap som denna, om vilka begränsningar som finns och vad som krävs för en träffsäker analys, kan hjälpa logganalytiker att välja rätt verktyg, förstå vad som krävs för att använda dem och vilka begränsningar de har.

6.4 Hur väl fungerar modeller av godartat beteende under ideala förhållanden?

6.4.1 Bakgrund

Som beskrivits ovan är modeller av godartad aktivitet populärt inom den akademiska forskningen. Ett stort antal förslag finns på hur de ska skapas, men det finns tyvärr lite dokumenterad kunskap om hur bra de fungerar i praktiken. Ett skäl till detta är att det är svårt att generera syntetiska händelser av samma komplexitet som de sker i verkliga miljöer. Det går inte heller att dra klara slutsatser om effektivitet i verkliga miljöer eftersom det i verkliga miljöer saknas kontroll över vad som är elakartad aktivitet och inte. Eftersom det syntetiska data som kan genereras under kontrollerade former är enklare och mindre komplex bör dock vara möjligt att slå fast en övre gräns för hur väl olika modeller av godartad

aktivitet kan prestera. Det vill säga hur väl de fungerar i enklare miljöer med begränsade varianter av godartat beteende. Det bör alltså vara möjligt att ge svar på frågan: **Hur väl fungerar modeller av godartat beteende under ideala förhållanden?**

6.4.2 Metod

Ett antal av de mer etablerade modellerna för godartat beteende kan välja ut baserat på litteraturstudier. De allra flesta förslag i litteraturen fokuserar på nätverkstrafik och/eller systemanrop, vilket kan återskapas i CRATE förhållandevis väl genom att återspegla vad riktiga användare gjort på sina maskiner. Genom att noggrant producera godartade systemhändelser enligt mönster som observerats i riktiga system och injicera angrepp från SVED kan data för tester produceras.

För att skapa ideala förutsättningar för modellerna ges de fullständigt korrekt information om det övervakade systemets normaltillstånd. Detta inkluderar till exempel en korrekt modell över nätverkstopologin och aktiva tjänster. I de fall metoder brukar maskininlärning för att bygga modellen så lärs den upp på samma godartade beteende om den sedan testas mot. Antingen genom att dela denna data i en halva för lärande (vecka ett) och en vecka för detektion (vecka två) där angrepp sker. Eller genom att modellen får normalt beteende för både vecka ett och två, för att sedan uppgiften att rapportera angrepp när dessa två veckor spelas upp med angrepp inbäddat i händelserna. Båda dessa är ideala i någon bemärkelse. Den första typen av test motsvarar situationen där man vid upplärningsfasen vet med all säkerhet att allt är godartat beteende, vilket sällan/aldrig är fallet i riktiga system. Den andra typen av test är än mer idealt och eftersom man då också vet vad som kommer vara godartat beteende just när angreppen inträffar och användarna är helt förutsägbara.

6.4.3 Förväntad nytta

Resultatet av dessa tester skulle ange en övre gräns för den kapacitet som existerande modeller har. Trots att de inte ger en skattning av hur väl de skulle fungera under normala förhållanden kommer denna kunskap sannolikt kunna användas för att avgöra om modeller av godartat beteende är värt att undersöka vidare och vilka verktyg/modeller för godartat beteende som är mest lovande.

6.5 Vilken kontext- eller situationsbaserad information som behövs för att göra bra manuellt analysarbete?

6.5.1 Bakgrund

Observationer av och intervjuer med logganalytiker har identifierat att de använder flera typer av kontext- och situationsbaserad information i sitt arbete. Exempel på kontext- eller situationsbaserad information som kan tänkas vara av vikt för olika delar av den manuella analysen är: nätverkstopologi, mjukvaror som används, sårbarheter i mjukvaror, olika delar syfte, behörigheter i system, vanliga trafikmönster, angriparnas förmågor och angriparnas intressen. Hur viktig kunskap om denna kontext- eller situationsbaserad information är av intresse. Det vore bra med svar på **hur viktig är kontext- eller situationsbaserad information för prestation i**

1) **övervakning?**
2) **orsaksanalys?**
3) **utförande av åtgärd?**

Det finns redan ett antal studier som pekar på att informationen har betydelse för prestationen, men det saknas kunskap om hur viktig olika typer av information är och hur den används.

6.5.2 Metod

Existerande litteratur kan användas som utgångpunkt för att identifiera lämpliga definitioner av de informationstyper som angetts ovan och andra typer av information som är av eventuellt värde i analysarbetet. Till exempel behövs ett sätt beskriva en nätverkstopologi som fångar det som tros vara av relevans för logganalysarbetet. Tester kan sedan utföras med denna information är tillgänglig för logganalytikern och när den inte är det. Tillgängligheten kan bestå av möjlighet att studera den innan testet och/eller möjlighet att inspektera den under försöket. Prestation kan mätas enligt följande:

- Hur god träffsäkerhet det är i prioriteringen av loggposter under övervakningen.
- Hur ofta attackstegen beskrevs korrekt under orsaksanalysen.
- Hur ofta den utförda åtgärden skulle skydda mot angreppet och andra liknande angrepp.

Dessa tre mått kan samlas in under en och samma experiment där logganalytiker får i uppgift att analysera loggar och utföra alla steg i följd. De kan också göras mer fokuserat, till exempel genom att enbart fokusera på prioritering av loggposter i en logg.

Värt att notera är att den oberoende variabeln (informationen som är tillgänglig) är monotont ökande under experiment som dessa ifall ett och samma datornätverk används. Det vill säga, kunskapen om ett nätverk får antas öka i takt med att tester genomförs och information görs tillgänglig. Detta är problematiskt i det fall tester ska göras där en person har olika förutsättningar vid olika tester. Det behövs därför en experimentplan som antingen hanterar detta genom ordningen av testerna eller genom att flera olika datornät används i testerna.

6.5.3 Förväntad nytta

Tidigare tester har indikerat att kunskap om hotbilden och vilka angrepp som är rimliga att förvänta sig är viktiga under övervakningsfasen. För orsaksanalysen antas mer detaljerad kunskap om topologi, behörigheter och sårbarheter vara av vikt. Detsamma antas vara viktig för utförande av åtgärd. Bekräftelse av detta och mer kunskap om vilken information som behövs kan användas för att identifiera vilka informationsresurser en logganalytiker behöver för att arbeta effektivt.

6.6 Hur viktig är förståelse för hur den automatiserade analysen fungerar?

6.6.1 Bakgrund

Inom forskning på mänskliga faktorer kopplade till system är automatisering ett område som studerats väl. Som avsnitt 3 av denna rapport beskrivit är möjligheterna till automatisering inom logganalys betydande. Två av de mer centrala frågorna kopplade till automation är [93]:

1) Vilken tillit och tilltro systemanvändare har för automationen och hur det påverkar arbetet? (En övertro har observerats i andra områden. [94])
2) Vad som påverkar systemanvändares förståelse för automationens begränsningar och hur detta påverkar prestation?

Dessa två frågor hänger ihop i stor utsträckning. En god förståelse för hur automationen fungerar innebär sannolikt att systemanvändaren (logganalytikern) har en rimlig bild av den tillit som bör fästas till automationen. Och en logganalytiker som själv skapa automationen har sannolikt god tillit till den.

6.6.2 Metod

Det går att manipulera en logganalytikers förståelse för den automatiska analysen på flera sätt. Logganalytikern kan exempelvis ges ett automationssystem som hen inte får någon information; logganalytikern kan ges ett automationssystem tillsammans en beskrivning av det; logganalytikern kan själv uppmanas skapa ett automationssystem. Manipulation av tillit kan till också göras. Till exempel genom att visa testresultat (korrekta och/eller missvisande) för automationslösningen som används. Etablerade mätinstrument för tillit till system finns att tillgå i litteraturen och dessa kan med enkelhet anpassas till logganalys för att se om manipulationen lyckats. Ett exempel på en sådan är de tolv frågorna i [95].

Då övervakning är den del av den manuella analysen som är mest automatiserad idag är automatiseringens påverkan på övervakning mest intressant att undersöka. Tester kan till exempel mäta hur väl operatörerna identifierar relevanta loggposter (prestation) med och utan system av olika kvalitet och tillit. Värt att notera är att det går att undersöka vilka variabler som påverkar hur korrekt logganalytikerns bedömning av pålitlighet som en del av dessa tester. Det kan till exempel undersökas hur allmän kunskap om cybersäkerhet och hur kunskapen om insidan av den automatiska analysen påverkar logganalytikerns förmåga att bedöma när man ska lita på den automatiska analysen.

6.6.3 Förväntad nytta

Dessa tester ger mer kunskap om när logganalytiker kan bedöma om en automationslösning fungerar och vilken kunskap som behövs för att göra korrekta bedömningar. Till exempel kan det avgöras om förmågan att bedöma tillit är korrelerad med den kunskap logganalytikerna har om den modell som används i den automatiska analysen och mot allmän kunskap inom cybersäkerhet.

FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

FOI
Totalförsvarets forskningsinstitut
164 90 Stockholm

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

www.foi.se