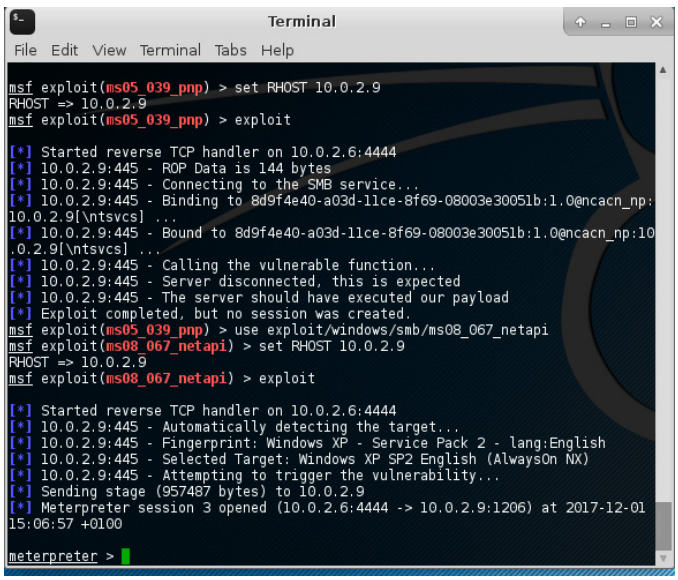
Publika angreppskoder och intrångssignaturer

Kvantitativa tester av träffsäkerhet

TEODOR SOMMESTAD OCH HANNES HOLM



[**] [1:2102465:9] GPL NETBIOS SMB-DS IPC$ share access [**][Classification: Generic Protocol Command Decode] [Priority: 3]

[**] [1:14782:17] OS-WINDOWS DCERPC NCACN-IP-TCP srvsvc NetrpPathCanonicalize path canonicalization stack overflow attempt [**][Classification: Attempted Administrator Privilege Gain] [Priority: 1]

[**] [1:7209:16] OS-WINDOWS DCERPC NCACN-IP-TCP srvsvc NetrPathCanonicalize overflow attempt [**][Classification: Attempted Administrator Privilege Gain] [Priority: 1]

FOI-R--4499--SE
ISSN 1650-1942 December 2017

FOI-R--4499--SE

Teodor Sommestad och Hannes Holm

Publika angreppskoder och intrångssignaturer

Kvantitativa tester av träffsäkerhet

Titel	Publika angreppskoder och intrångssignaturer: Kvantitativa tester av träffsäkerhet
Title	Public exploits and intrusion detection signatures: Quantitative tests of accuracy
Rapportnr/Report no	FOI-R--4499--SE
Månad/Month	December
Utgivningsår/Year	2017
Antal sidor/Pages	28
ISSN	1650-1942
Kund/Customer	Försvarsmakten
Forskningsområde	4. Informationssäkerhet och kommunikation
FoT-område	Ledning och MSI
Projektnr/Project no	E72679
Godkänd av/Approved by	Christian Jönsson
Ansvarig avdelning	Ledningssystem
Exportkontroll	Innehållet är granskat och omfattar ingen information som är underställd exportkontrollagstiftningen.

Sammanfattning

Denna rapport beskriver två tester kopplade till logganalysarbete inom cybersäkerhet. Det första testet undersökte hur ofta publika angreppskoder fungerar och ger information om hur sådana påverkar hotbilden som en logganalytiker behöver förhålla sig till i sina beslut. Totalt gjordes 1545 angreppsförsök med 211 olika angreppskoder för att tillskansa sig privilegier på maskiner en sårbarhetsskanner pekat ut som sårbar för angreppskoden. Endast 70 av angreppsförsöken (4,5 %) lyckades och endast 18 angreppskoder (8,5 %) fungerade mot någon maskinkonfiguration.

Det andra testet undersökte hur väl olika publika signaturdatabaser upptäcker angrepp. Information om detta kan hjälpa logganalytiker att välja signaturdatabas(er) och förstå vanliga intrångsdetektionssystems begränsningar. Signaturdatabaser utgivna mellan 2011 och 2016 från tre olika utgivare testades på nätverkstrafik från 246 angrepp med 125 olika angreppskoder. Den bästa signaturdatabasen producerade larm av rätt prioritet för 61 angrepp och den sämsta producerade larm av rätt prioritet för 20. Nyare signaturdatabaser var något bättre; publikt kända sårbarheter upptäcktes oftare; angrepp mot Windowsmaskiner upptäcktes oftare än angrepp mot Linuxmaskiner och många fler angreppsförsök upptäcktes när inaktiverade signaturer aktiverades.

Nyckelord: intrångsdetektion, signaturdatabaser, logganalys, angreppkoders tillförlitlighet.

Summary

This report describes two tests associated to log analysis within cyber security. The first test investigated how often publicly known exploits work, and gives information about the threat environment that a log analyst can use in their decision-making. In total, 1545 exploitation attempts were made with 211 exploits to obtain privileges on the machines identified as vulnerable for the exploit by a vulnerability scanner. Only 70 attempts (4.5%) were successful and only 18 exploits (8.5%) worked against some machine configuration.

The second test investigated how often different public signature databases discovers public exploits. Information about this can help log analysts select signature database(s) and understand the limitations of common intrusion detection systems. Signature databases released between 2011 and 2016 from three sources were tested with traffic traces from 246 exploitation attempts with 125 exploit codes. The best signature database produced alerts of the right priority for 61 exploitation attempts; the worst produced alerts of the right priority for 20 exploitation attempts. Newer signature databases performed better; publicly known vulnerabilities were detected more often; attacks on Windows machines were detected more often than attacks on Linux machines; many more exploitation attempts were detected when inactivated signatures were activated.

Keywords: intrusion detection systems, signature, log analysis, exploit reliability.

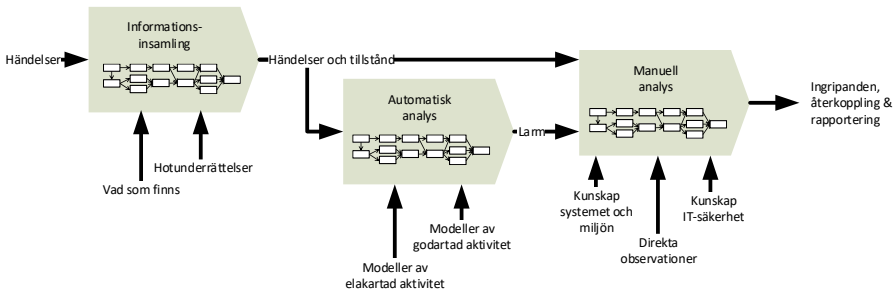
Innehållsförteckning

1 Inledning

Denna rapport har producerats inom projektet *Övning och experiment för operativ förmåga i cybermiljön* (ÖvExCy), som finansieras via Försvarsmaktens samlingsbeställning inom forskning och teknikutveckling. 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 upptäcka och förstå angrepp för att kunna fatta beslut kopplade till cybersäkerhet. Detta första kapitel ger bakgrund till rapporten och beskriver dess syfte.

1.1 Logganalysarbete

Det amerikanska ramverket rörande cybersäkerhetsarbete kallat National Initiative for Cybersecurity Education (NICE)[1] anger 928 uppgifter, 359 kompetenser och 614 kunskapsaspekter som är kopplade till yrkesroller inom cybersäkerhet. Denna rapport fokuserar på det som i NICE betecknas som *Computer Network Defence Analysis* och den yrkesroll som kallas logganalytiker. Dennas huvudsakliga uppgifter är att övervaka, analysera och fatta beslut om lämpliga motåtgärder när en incident upptäckts [2–4]. I projektet ÖvExCy har logganalys brutits ner i de tre processer som visas i Figur 1 [5].



Figur 1. Översikt över logganalysarbete och den information som vanligtvis hanteras.

Som figuren visar behöver såväl den manuella som den automatiska analysen information om händelser och tillstånd. I den manuella analysen behöver logganalytiker också ha bra förståelse för det egna systemet och kunskap om IT-säkerhet. Ibland görs även direkta observationer genom att logga in på övervakade maskiner eller liknande. Det är i regel samma typ av information som används för att konstruera verktyg för automatisk analys som ger larm till logganalytiker då något av särskilt intresse har skett. De automatiska analyserna kan grovt delas in i de som utgår från en modell av elakartad aktivitet (till exempel signaturer för paket som är vanliga i angrepp) och de som utgår från en

modell av godartad aktivitet (till exempel vanliga trafiklaster i nätverket). Tillsammans bestämmer dessa informationsinsamling, automatisk analys och manuell analys vilken logganalysförmåga som uppnås.

1.2 Behovet av tester

Cybersäkerhetsdomänen (eller IT-säkerhetsdomänen) är ett förhållandevis ung och växer fort. Forskningen som utförs inom området fokuserar i huvudsak på framtagandet av nya tekniska lösningar som ska lösa aktuella behov; inte på tester eller utvärderingar av vedertagna lösningar eller förslag på alternativa lösningar [6]. Det är också en domän där flera olika discipliner med helt olika fokus arbetar. När amerikanska JASON[1] fick i uppdrag att analysera cybersäkerhetsdomänen som ett vetenskapligt område listar de till exempel följande områden som exempel på sådana där vetenskaplig aktivitet pågår: pålitlighet, kryptografi, spelteori, kontrollering mot modeller, obfuskering[2], maskininlärning och sammansättning av systemkomponenter [7]. De konstaterar även att tillgången till data för tester, som är kritisk för vetenskapliga framsteg, är ett problem i cybersäkerhetsdomänen. De sätter hopp till möjligheten att använda testanläggningar (på engelska *cyber ranges*) för att skapa relevant data och möjliggöra tester.

Bristen på bra data för experiment är tydlig inom logganalysområdet. Mycket av forskningen fokuserar på att ta fram nya tekniska lösningar för automatisk analys och det finns ett stort antal förslag kopplat till intrångsdetektion i akademisk forskning som bör undersökas [5]. Få av dessa är dock implementerade i lösningar som är färdigställda och allmänt tillgängliga. Under 2015 gjorde FOI en systematisk genomgång av vetenskapliga artiklar för att identifiera färdiga lösningar som skulle kunna testas i ÖvExCy [8]. Av cirka 8 000 vetenskapliga artiklar bedömdes 671 som tillräckligt relevanta och aktuella för att läsas ytligt och 159 som tillräckligt intressanta för att läsas i detalj. Femtio av dessa beskrev egna tester av en lösning, vilket togs som en indikation på att en lösning fanns implementerad. Men när FOI kontaktade författarna till dessa femtio artiklar och erbjöd sig att testa dem var det endast tre som var beredda att göra sin implementation tillgänglig. Överlag tycks de tekniska lösningar som presenteras i vetenskaplig litteratur inte komma längre än till konceptstadiet och de konceptuella lösningar som tas fram testas också sällan på ett meningsfullt sätt. De tester som görs, exempelvis i samband med att en forskargrupp presenterar en ny lösning, görs ofta mot inaktuell och tveksam data. Till exempel utförs en stor del av testerna som faktiskt genomförs mot den trafikinspelning av angrepp som kallas KDD-cup eller DARPA-datasetet, vilket producerades för nästan 20 år

[1] JASON är en oberoende grupp av framstående forskare som ger råd åt USA:s regering i frågor kopplade till vetenskap och teknik. För mer information, se https://fas.org/irp/agency/dod/jason/.
[2] Avsiktlig tillkrångling av data i syfte att göra den svår att förstå.

sedan och sedan länge har kända validitetsproblem [9]. Som avsnitt 3.1 i denna rapport visar har också akademiker ett begränsat intresse av att kvantitativt testa de intrångsdetektionslösningar som används av praktiker, trots att det är allmänt känt att intrångsdetektionslösningarna har påtagliga brister i träffsäkerhet och producerar en stor mängd falsklarm.

Avsaknaden av ordentliga tester är problematisk. Dels finns en risk att praktiker och forskare idag arbetar under felaktiga antaganden, exempelvis om vilka detektionsmetoder som fungerar till vad eller under vilka förhållanden en detektionslösning fungerar väl. Dessutom sätts ingen press på tillverkare av detektionslösningar eller forskare på intrångsdetektionslösningar att ta fram nya förslag med substantiella förbättringar. Det avgörs ju nämligen inte idag om en ny lösning innebär en förbättring.

Logganalysområdet är alltså ett bra exempel på hur avsaknad av tester och testdata är problematiskt. Men det ska påpekas att detta problem också finns i andra delar av cybersäkerhetsdomänen. Ordentliga tekniktester utförda under realistiska förhållanden är överlag ovanliga i publikt tillgänglig litteratur och tester av en extern part (till exempel annan forskargrupp eller oberoende institut) är ytterst sällsynta.

1.3 Rapportens syfte

Ett av målen med projektet ÖvExCY är att svara på hur olika faktorer påverkar logganalysförmåga. Det har tidigare i projektet gjorts en genomgång av litteraturen på området och konstaterats att merparten av forskningen handlar om automatiska analysmetoder [5]. Som nämndes i avsnitt 1.2 har de försök som gjorts för att testa dessa lösningar hindrats av svårigheten att få tillgång till dem. De flesta har aldrig utvecklats till mer än en konceptuell idé; de som har utvecklats till en mjukvara tycks antingen förlagts, förkastats av utvecklaren eller vara omgärdade av ett kommersiellt intresse som begränsar möjligheten att testa effektiviteten.

En teknisk lösning som har varit möjlig att testa i projektet är SnIPS [10]. Denna lösning väger samman flera larm från det vedertagna, nätverksbaserade, intrångsdetektionssystemet Snort för att skatta sannolikheten att angripare har skaffat sig privilegier på datorerna i datornätverket. Även om SnIPS tycks överskatta sannolikheterna för att angripare fått privilegier finns mycket som tyder på att den underliggande idén är rimlig och kan användas av praktiker [11,12]. Tre tänkbara förklaringar till SnIPS överskattning är att

1. sammanvägningen överskattar hur ofta försök till nätverksangrepp lyckas under realistiska förhållanden
2. Snort producerar en stor mängd falsklarm som antyder att angreppsförsök gjorts trots att så inte är fallet

3. Snort missar många faktiska angrepp och sammanvägningen därför är konstruerad så att den övertolkar de få larm som indikerar angrepp.

Det finns ytterligare tänkbara förklaringar, men dessa tre räcker för att illustrera hur kunskap om de grundläggande förutsättningar som gäller i cybersäkerhetsdomänen behövs för att konstruera effektiva lösningar och fatta kloka beslut. Denna rapport dokumenterar två tester som syftar till att ge sådan kunskap. Kapitel 2 beskriver tester av attackkoders tillförlitlighet. Kapitel 3 beskriver tester av hur träffsäkra olika signaturdatabaser i Snort är. Båda testerna fokuserar på vedertagna publika verktyg och lösningar som används av praktiker. Kapitel 4 diskuterar resultaten från testerna och presenterar slutsatser.

2 Publika angreppkoders tillförlitlighet

Detta kapitel beskriver ett test av publika attackkoders tillförlitlighet. Först beskrivs bakgrunden till testet och tidigare forskning. Därefter beskrivs det material och de metoder som används i testet. Sist beskrivs resultatet av testet.

2.1 Bakgrund

Inom cybersäkerhetsdomänen finns en utbredd tro att angrepp blir lättillgängligare för varje år och att det numera, eller inom kort, kommer gå att utföra sofistikerade angrepp utan någon egentlig kunskap om cybersäkerhet. Till exempel skrev Liu och Chen att ”attackverktyg och attacktekniker har blivit allt mer automatiserade och sofistikerade under det senaste decenniet, medan den tekniska kunskapen och förmågan som krävs för att använda dessa verktyg har minskat dramatiskt” [13] (vår översättning). Liknande trendbeskrivningar har lagts fram i bland annat NATO Review [14]. Hur enkelt det är att utföra angrepp har en påverkan på många prioriteringar och beslut inom cybersäkerhet. Till exempel skriver ENISA att en ”scriptkiddie” kan realisera runt 30 olika hot kopplade till cybersäkerhet [15]. Som det nämndes i inledningen kan sådana värderingar också påverka hur träffsäkra intrångsdetektionslösningar blir.

Det är utan tvekan så att det idag finns gott om verktyg för att utföra angrepp. Vid skrivandet av denna rapport finns exempelvis 38 218 angreppskoder i den publika databasen Exploit Database[3]. Det finns dock få tester av hur tillgängligheten påverkar användandet av angreppskoderna eller hur ofta de kan användas till lyckade angrepp.

Allodi med flera [16,17] står för ett av undantagen, och har studerat hur tillgänglighet påverkar hur ofta angreppskoder används genom att samköra data från flera olika datakällor. De fann bland annat att det var av liten betydelse ifall någon variant på angreppskod för sårbarheter var publikt tillgängliga i Exploit Database eftersom det inte innebar att angreppet oftare hamnade i Symantecs databas över angrepp mot driftsatta system. Andra har undersökt huruvida angrepp mot sårbarheter blir mindre vanliga om sårbarheterna kommuniceras via betaltjänsterna Zero Day Initiative eller iDefense istället för att göras publika [18]. De fann att de slutna betaltjänsterna gjorde att det tog längre tid innan angrepp mot sårbarheterna skedde och att de även skedde i mindre omfattning. Sannolikheten för att angreppen skulle ske minskade dock inte.

När det kommer till hur ofta angreppen lyckas är antalet studier ännu färre. Bara en tidigare studie av detta har identifierats. Denna studie testade tillförlitlighet genom att återkommande utföra tolv olika angreppskoder mot åtta olika

[3] https://www.exploit-db.com/

Windows-installationer [19]. De konstaterade att med omstarter mellan försöken fungerade två tredjedelar av angreppskoderna varje gång medan en tredjedel inte fungerade alls. De konstaterade också att det inte spelade någon roll om uppkopplingen gjordes med så kallad ”Bind TCP” eller så kallad ”Reverse TCP”. Att generalisera utifrån ett sådant begränsat test är naturligtvis vanskligt och frågan ”hur ofta fungerar publika angreppskoder?” kvarstår. Test med fler angreppskoder och fler sårbara maskiner behövs för att kunna skatta hur ofta de fungerar.

2.2 Material och metod

Detta test använde verktyget SVED, som beskrivs utförligt i andra dokument [8,20,21], för att på ett kontrollerat sätt utföra och logga angrepp. SVED är integrerat med test- och övningsanläggningen CRATE[4] och gör det möjligt att konstruera händelsekedjor som schemaläggs på förhand. De typer av aktioner som är av särskild vikt för detta test är

- sårbarhetsgenomsökningar med sårbarhetsskannern OpenVAS
- hantering av ögonblicksbilder på virtuella maskiner
- flyttning av angreppsmaskiner till godtyckligt nätverk
- kopiering av filer till maskiner
- exekvering och öppnande av filer i maskinerna
- utförande av musrörelser och knapptryckningar i maskinerna
- exekvering av angreppkoder som finns i ramverket Metasploit.

Testet utgick från de 1 179 virtuella maskiner som fanns installerade i CRATE hösten 2017. Dessa genomsöktes med sårbarhetsskannern OpenVAS så att detaljerad information om maskinernas sårbarheter fanns tillgänglig. I samband med det togs en ögonblicksbild av maskinen så att det sårbara tillståndet bevarades. De 719 132 sårbarheter som fanns i maskinerna (i snitt 199 per maskin) matchades mot de cirka 1 600 attackkoder som kommer med attackramverket Metasploit. Denna matchning användes för att schemalägga tester för alla angrepp som borde fungera mot dessa maskiner. Varje angrepp föranleddes av en bakrullning till en känd ögonblicksbild som motsvarade den tidpunkt när OpenVAS utfört skanningen, samt en kontroll att målet svarade på nätverkskommunikation. Varje angrepp slutfördes med att spara en skärmbild av målmaskinens grafiska gränssnitt. De allra flesta ”hjälp-händelser” utfördes via

[4] CRATE (Cyber Range And Training Environment) är en simuleringsmiljö som finns på FOI. För mer information, se: www.foi.se/CRATE.

virtualiseringsmjukvaran VirtualBox API. Olika typer av angrepp ställde olika krav på utförandet av testerna:

- **Fjärrangrepp mot servertjänster** som webbservrar utfördes om angreppsmaskinen kunde nå dem.
- **Fjärrangrepp mot klientmjukvaror** som webbläsare realiserades med hjälp av cirka 100 specialskrivna moduler som först slog av utvalda accesskontrollinställningar i målen (till exempel godkännande för exekvering av Java-applets). Sedan exekverades klientmjukvaran med rätt parametrar (till exempel Internet explorer med en skadlig webbserver på antagonistmaskinen som mål) innan en sekvens musklickningar och tangentbordstryckningar utfördes med syfte att godkänna popups och liknande.
- **Lokala angrepp** realiserades genom att antagonistmaskinen först genererade en applicerbar bakdörr kompatibel med Metasploit för att sedan kopiera över denna till målet och aktivera den. Slutligen exekverades det lokala angreppet med den ursprungliga bakdörren som bas (med målet att eskalera privilegier).
- **Filangrepp** realiserades genom cirka 30 specialskrivna moduler. Dessa genererade filer med angrepp inbakade på angreppsmaskinen för att sedan kopiera och exekvera dessa filer i den sårbara maskinen. Slutligen genomfördes det olika sekvenser med musklickningar och tangentbordstryckningar med syfte att godkänna popups och liknande.

I alla dessa typer av test kontrollerades att angreppsdatorn erhållit en session till den sårbara maskinen och detta användes för att avgöra om angreppet var lyckat eller ej. Eftersom attackkoderna skiljer sig åt går det inte att injicera samma kod i alla test. Ett urval robusta koder som injiceras genom angreppet (så kallade ”payloads”) användes

2.3 Resultat

Under testet utfördes angrepp mot 725 maskiner med olika konfigurationer eller tillstånd. Totalt utfördes 165 463 aktioner av SVED, producerades 565 307 loggposter kopplade till dessa och det gjordes 1 545 angreppsförsök med 211 olika angreppskoder. Som Tabell 1 visar lyckades endast 70 angrepp (4,5 %) ge de privilegier som de borde ge och etablera en session med angriparmaskinen. Av serverangreppen lyckades 4,1 %; av klientangreppen lyckades 9,2 %; av de lokala angreppen lyckades 3,2 %; av filangreppen lyckades endast 0,6 %. Det är alltså tydligt att en sårbarhet identifierad med en sårbarhetsskanner och en tillgänglig attackkod inte räcker för att kunna utföra ett angrepp.

Tabell 1. Antalet angreppsförsök av olika typ och antalet framgångsrika angrepp.

Operativsystem	Maskin-varianter	Antal angreppsförsök / Antal framgångsrika				
		Server	Klient	Lokal	Filformat	Totalt
Debian	18	18/0	0/0	0/0	0/0	18/0
Debian (x64)	2	1/0	0/0	0/0	0/0	1/0
Fedora (x64)	4	2/0	0/0	0/0	0/0	2/0
Gentoo	50	204/0	0/0	3/0	0/0	207/0
Gentoo (x64)	33	134/0	0/0	2/0	0/0	136/0
RedHat	4	21/0	0/0	0/0	0/0	21/0
RedHat (x64)	16	52/0	0/0	0/0	0/0	52/0
Ubuntu	40	136/0	0/0	16/0	0/0	152/0
Ubuntu (x64)	249	176/0	0/0	54/0	0/0	230/0
Win. 2000	35	69/21	37/1	12/0	45/1	163/23
Win. Server 2003	37	9/8	38/13	2/0	3/0	62/21
Win. Server 2008 (x64)	40	6/0	54/8	64/0	55/0	179/8
Win. 2012 (x64)	6	0/0	0/0	0/0	0/0	0/0
Win. 7	92	1/0	139/2	30/6	53/0	223/8
Win. 7 (x64)	5	0/0	1/0	0/0	9/0	10/0
Win. XP	94	66/8	14/2	7/0	2/0	89/10
Total	**725**	**905/37**	**283/26**	**190/6**	**167/1**	**1545/70**
Unika angreppskoder	**-**	**44**	**107**	**29**	**31**	**211**

Testet matchade totalt 211 olika attackkoder tillgängliga i Metasploit mot sårbarheter i maskinerna. Av dessa lyckades 18 ge privilegier mot någon konfiguration eller tillstånd. De angreppskoder som finns i Metasploits databas har en gradering som anger hur tillförlitlig angreppskoden är. De 18 som fungerade någon gång under försöket listas i Tabell 2. För dessa märks en tydlig koppling mellan den tillförlitlighet som är angiven för angreppskoden i Metasploit och hur ofta de lyckas. De som har getts tillförlitlighetsbedömningen ”excellent” lyckas i snitt 93 % av gångerna; de som getts tillförlitlighetsbedömningen ”great” eller ”good lyckas i snitt 67 % av gångerna; de som getts tillförlitlighetsbedömningen ”normal” lyckas i snitt 46 % av gångerna; de som getts tillförlitlighetsbedömningen ”average” lyckas i snitt 14 % av gångerna; och ingen av de som getts tillförlighetsbedömningen ”low” eller ”manual” är med bland de attackkoder som lyckats någon gång. De tillförlitlighetsbedömningar som gjorts tycks alltså fånga sannolikheten att angreppskoden gör det den ska. Det ska dock påminnas om att många av de andra 193 attackkoder som prövades, ofta mot flera maskiner bedömda som sårbara, misslyckades. Flera av dessa hade tillförlitlighetsbedömningar som var höga, och kopplingen är överlag svag mellan bedömd tillförlitlighet och sannolikhet att lyckas.

Tabell 2. Antalet angreppsförsök med olika angreppskoder och deras angivna tillförlitlighet.

Angreppskod	Försök	Lyckade	Typ	Tillförlitlighet
multi/browser/java_atomicreferencearray	1	100 %	Klient	Excellent
windows/local/ms10_092_schelevator	1	100 %	Lokal	Excellent
multi/browser/firefox_webidl_injection	10	80 %	Klient	Excellent
windows/mmsp/ms10_025_wmss_connect_funnel	3	100 %	Server	Great
windows/smb/ms08_067_netapi	33	94 %	Server	Great
windows/browser/ms07_017_ani_loadimage_chunksize	11	55 %	Klient	Great
windows/dcerpc/ms03_026_dcom	14	21 %	Server	Great
windows/browser/ms11_003_ie_css_import	1	100 %	Klient	Good
windows/fileformat/adobe_collectemailinfo	3	33 %	Filformat	Good
windows/browser/ms13_022_silverlight_script_object	1	100 %	Klient	Normal
windows/browser/ms10_002_aurora	4	50 %	Klient	Normal
windows/browser/msvidctl_mpeg2	9	33 %	Klient	Normal
windows/browser/ms09_072_style_object	6	33 %	Klient	Normal
windows/browser/ms13_037_svg_dashstyle	3	33 %	Klient	Normal
windows/browser/ms12_037_ie_colspan	4	25 %	Klient	Normal
windows/local/ms13_053_schlamperei	12	17 %	Lokal	Average
windows/local/ppr_flatten_rec	12	17 %	Lokal	Average
windows/local/ms13_081_track_popup_menu	11	9 %	Lokal	Average
Total	**139**	**70**	**-**	

3 Signaturdatabasers träffsäkerhet

Detta kapitel beskriver ett kontrollerat test av olika intrångssignaturers träffsäkerhet. Först ges en bakgrund till testet och därefter beskrivs materialet och metoden som användes för att utföra testet. Sist beskrivs resultatet.

3.1 Bakgrund

Intrångdetektionssystem är en viktig komponent i logganalysförmåga. Det är också något som getts mycket uppmärksamhet av forskare. I driftsatta system är den allra vanligaste lösningen så kallade signaturbaserade intrångsdetektionssystem, som jämför händelser och tillstånd mot en databas över kända hot med hjälp av signaturer för dessa. Det finns flera kommersiella lösningar, men den vanligaste och mest omskrivna tekniska lösningen är Snort [22], som skapades redan 1998, som undersöker nätverkstrafik, har öppen källkod och är gratis att använda. Eftersom systemet bygger på databaser med signaturer för kända hot är dess värde avhängigt träffsäkerheten hos dessa signaturer. Det finns flera olika signaturuppsättningar, med olika licensvillkor, och tiotusentals signaturer att utgå ifrån. Naturligtvis går det också att skapa egna signaturer och signaturdatabaser.

Vid intrångsdetektion önskas att en stor andel av faktiska angrepp genererar informativa larm samtidigt som få larm produceras för sådant som inte är angrepp. Det är trivialt att uppnå en av dessa eftersom en lösning kan ställas in så att allt betraktas som angrepp eller att inget betraktas som angrepp. Att uppnå båda samtidigt är svårare och varje system av denna typ måste göra en avvägning mellan hur många falsklarm som ska tolereras. Antalet falsklarm som produceras anges vanligtvis som det största problemet med Snort. Enligt vissa tester är det så mycket som 96 % falsklarm [23]; enligt andra är det så mycket som 98 % [12]. Det är också allmänt känt att mycket arbete och expertis krävs för att trimma in systemet mot den egna miljön så att antalet falsklarm begränsas medan angrepp fortfarande kan upptäckas [24][25][26]. Men det är också ett problem att systemet inte upptäcker tillräckligt många angrepp. Ofta lyfts svårigheten att upptäcka okända varianter av angrepp fram som ett problem för signaturbaserade lösningar som Snort [27][28]. Ett jämförande test [29] mellan signaturdatabaser från olika tidpunkter och ett stort antal angrepp mot servrar ger också visst stöd för det. Angrepp som mot sårbarheter som var publikt kända då signaturerna skapades gav ett larm av prioritet ett eller två i 54 % av fallen; angrepp mot sårbarheter som inte var publikt kända gav ett larm av prioritet ett eller två i 17 % av fallen. Angrepp som var kända när signaturerna skrevs upptäcktes alltså betydligt oftare.

När det kommer till signaturerna finns begränsad dokumenterad kunskap som anger vad som utmärker bra respektive dåliga signaturer. Såvitt författarna till

denna rapport känner till finns endast ett test av som karaktäriserar bra intrångsdetektionssignaturer. I det testet [30] utgick man från data insamlat på ett universitet och gjorde med manuell analys baserat på ytterligare informationskällor (bland annat internetsökningar och ytterligare loggar) en bedömning av huruvida larm var falska eller inte. De fann att bra signaturer, som i den studien var sådana som upptäckt något angrepp och gav få falsklarm, var mer specifika och precisa. Det innebar till exempel att de använde reguljära utryck, preciserade portnummer eller angav paketstorlek.

Även om det finns begränsad kunskap om vad bra respektive dåliga signaturer har för egenskaper finns det gott om tekniska förslag på lösningar för att skapa signaturer, typiskt med ett förslag på sätt att göra detta automatiskt. Till exempel förslag

- på hur kommunikation med ”*command and control*”-servrar ska kunna identifieras baserat på trafik som genererats av ett stort antal insamlade skadliga koder [31]
- som genererar signaturer utifrån minnet eller loggfiler i så kallade ”honeypots” som syftar till att attrahera angripare [32] [33] [34]
- som skapar signaturer baserat på en beskrivning av mjukvarusårbarheter av särskild typ [35] [36]
- på hur protokollspecifikationer ska kunna användas för att skapa signaturer som indikerar hotfull aktivitet [37][38].

Dessa förslag ger en indikation på under vilka förutsättningar det är enkelt att skriva signaturer. Det kan till exempel antas att angrepp som observerats i skadliga koder eller mot honeypots är enklare att skriva signaturer för än de som ännu inte kunnat observeras på detta sätt. Likaså är det rimligt att tänka sig att angrepp som utnyttjar kända sårbarheter som är lätta att analysera också är jämförelsevis enkla att skriva signaturer för.

Det enda experiment som undersöker signaturdatabasers träffsäkerhet och går att finna i litteraturen rapporterade att 54 % av de serverangrepp som var kända genererade larm av prioritet ett (”*dangerous and harmful attacks*”) eller två (”*suspicious signatures potentially preparing attacks*”) när signaturdatabasen från Sourcefire Vulnerability Research Team (VRT) användes [29]. Observationer i samband med cybersäkerhetövningar har visat snarlika siffror. Exempelvis gav VRT-signaturerna larm med prioritet ett i 45 % av angreppen som gjordes med publika verktyg i en övning hos FOI 2012 [39]. Tester som inkluderar angrepp mot klientmjukvaror över datornätverk saknas i princip i dessa tester. Inte heller finns jämförelser mellan olika signaturdatabaser.

Testet som beskrivs nedan försöker ge svar på frågan ”hur väl upptäcker publika signaturdatabaser publika angreppskoder?” och dessutom ge ytterligare indikationer på vad som påverkar ifall ett angrepp upptäcks eller ej.

3.2 Material och metod

Också denna studie använde verktyget SVED för att på ett kontrollerat sätt utföra och logga angrepp. För att få ett representativt urval av angreppskoder slumpades 200 fram från Metasploits databas av nätverksangrepp. Dessa utfördes under kontrollerade former mot två maskiner preparerade för att vara mottagliga för angreppsförsök medan inspelning av all nätverkstrafik gjordes. Inspelningen gavs därefter som indata till Snort tillsammans med olika signaturdatabaser.

Att finna äldre signaturdatabaser var svårt eftersom de officiella källorna endast sparar och offentliggör den allra senaste versionen. Signaturdatabaserna för detta test inhämtades från äldre versioner av säkerhetsverktyg tillgängliga på internet och var daterade från 2011 till 2016. Mer specifikt testades databaser från Sourcefire Vulnerability Research Team (VRT) daterade 2015-04-06 och 2014-09-22; databaser från Emerging Threats (ET) daterade 2016-01-01, 2014-02-18, 2012-12-18 och 2011-04-02; databaser med de så kallade community-signaturerna daterade 2016-03-31 och 2013-10-14. Signaturerna från VRT och ET är förmodligen de vanligaste i driftsatta system; community-signaturerna representerar en enklare uppsättning signaturer utan begränsningar på hur de får användas eller spridas vidare. De åtta signaturdatabaserna testades i version 2.9.9.0 av Snort så som de laddas ner från internet. Signaturer som kommenterats bort och signaturer som markerats som borttagna inkluderades inte.

Snort producerar larm som är graderade i tre nivåer. Ett larm med prioritet ett betyder ”*dangerous and harmful attacks*”; ett larm med prioritet två betyder ”*suspicious signatures potentially preparing attacks*”; ett larm med prioritet tre betyder ”*unusual traffic not identified as dangerous*” [40]. Huruvida Snort producerar sådana larm när angrepp sker, och vilken betydelse signaturdatabaserna har, var det testet sökte svar på. Eftersom alla angreppen hade potential att ge behörigheter på den angripna maskinen skulle de rimligen generera larm av prioritet ett för att kunna anses korrekt upptäckta, men larm av alla tre typer noterades och togs med i analysen.

Då vissa angrepp kräver flerstegsinteraktion med en sårbar för mjukvara, och alla sårbara mjukvaror (exempelvis gamla versioner av affärssystem) var svåra att tillgå, lades mycket arbete på att skapa en labbmiljö där angreppen kunde utföras utan den faktiska mjukvaran som var sårbar. Två maskiner sattes upp för att agera målmaskiner. Den ena av dessa hade operativsystemet Windows och utsattes för angrepp mot windowstjänster som SMB samt de angrepp som inriktade sig mot webbläsare. Den andra använde Linux, körde ett antal vanliga tjänster (till exempel SSH, Telnet, HTTP och FTP) samt ett specialbyggt skript som möjliggjorde svar på frågor mot tjänster som använde portar som inte var standard men vissa av attackkoderna riktade sig mot. I praktiken innebar detta antingen *port forwarding* till någon av de faktiska servrarna på maskinen eller ett trivialt svar på lager fyra i TCP/IP-stacken. Det sistnämnda nyttjades enbart när

det inte bedömdes påverka de korresponderande angreppskodernas funktion, till exempel för att de endast vill starta en session på den porten.

Även om rimliga svar gavs var det inte alltid de svar en sårbar maskin skulle ge för att vara sårbar. Angreppskoderna innehåller ofta kontroller för detta och attacken utförs endast om maskinen är av rätt typ. Med anledning av detta gjordes en del justeringar av attackkoderna. Vissa av angreppskoderna skrevs om så att de inte avslutades ifall en initial utvärdering (till exempel av webbläsarversion) visade att den angripna maskinen inte var sårbar. Andra skrevs om så att de själva genererade acceptabel data ifall den angripna maskinen inte gav något svar som användes för att utföra angreppet, exempelvis i form av sessionsnummer i HTTP-trafik. Några få innefattade så komplex interaktion att de exkluderades och ersattes av andra attackkoder. Exempel på sådana med komplex interaktion var angrepp som krävde interaktion i en proprietär utökning av SMB-protokollet. Det ska noteras att inga av de ändringar som gjordes bedömdes påverka de delar som en signaturdatabas rimligen undersöker. Till exempel antogs att sessionsnummer i HTTP-trafik inte används av signaturdatabasen eller att signaturdatabasen kräver att angreppet ska lyckas för att skapa ett larm.

Utöver dessa ändringar lades kodrader till i angreppskoderna som loggade all kommunikation som skedde och huruvida denna var elakartad eller ej. Många angreppskoder startar direkt med att försöka injicera skadlig kod i den sårbara maskinen; andra inleder angreppet med att ställa helt normala frågor till den sårbara maskinen eller logga in på den tjänst som ska angripas. Loggningen av kommunikationen i angreppskoderna gjorde det möjligt att både säkerställa att angreppet utfördes som det var tänkt och att isolera de delar som är faktiska angrepp. Genom att undersöka de larm som producerades vid de dokumenterade tidpunkterna för angrepp kunde träffsäkerheten bestämmas.

Sist men inte minst, angreppskodernas syfte är att injicera kod som utför något i den sårbara maskinen. Typiskt för att möjliggöra fjärrstyrning av den. Detta test fokuserade på angreppskoderna (det som på engelska kallas *exploits*) och huruvida de upptäcktes och inte om den kod som injiceras i den sårbara maskinen (det som på engelska kallas *payload*) upptäcktes. Den kod som injicerades hölls därför enkel. Till exempel injicerades den kod som heter en *generic/debug_trap* och bara lägger in en avlusningsflagga i den sårbara processen eller enkla kommandon för att lista filer. Dessa koder varierades mellan försöken för att kunna utvärdera om det var den injicerade koden som upptäcktes snarare än angreppskoden. Flera angreppskoder ser också något olika ut beroende på vilken typ av maskin som angrips. Till exempel finns skillnader mellan angreppskoderna mot olika språkpaket i *windows/smb/ms08_067_netapi*. Angreppsförsök gjordes mot alla olika mål angreppskoderna kunde hantera.

Ett enkelt manuellt test gjordes för att verifiera att alla ändrade angreppskoder fungerade i testmiljön och att uppmärkningen gjordes korrekt, men som resultatet

nedan visar utfördes inte alla angreppskoder under försöket. Orsaken till detta är vid skrivandet av denna rapport oklar. Eventuellt är skälet en krasch i angriparmaskinen till följd av ändringarna i angreppskoderna, som fått angriparmaskinen att låsa vissa funktioner under en del av testet. Alternativt har de delar som skulle aktivera tjänster eller få de sårbara maskinerna att besöka webbsidor misslyckats med detta efter en krasch. Det upptäcktes också att den trafikinspelning som gjorts inte sparats korrekt, utan skapade paket av en storlek som Snort inte kunde hantera. Programmet tcprewrite användes för att om-fragmentera trafiken i efterhand. Mer arbete kommer att behövas för att göra de mer stabila i syntetiska förhållanden och resultatet från detta test ska tolkas med försiktighet.

3.3 Resultat

Under testet utfördes kontrollerade angrepp med 125 av de 200 angreppskoderna. Flera försök gjordes med varje angreppskod; ett för varje mål (exempelvis mjukvaruversion) som angreppet kunde anpassas för. Totalt kördes 246 angreppsförsök som med säkerhet involverade skadlig kod som skickades över nätverket, och därmed skulle kunna upptäckas av Snort.

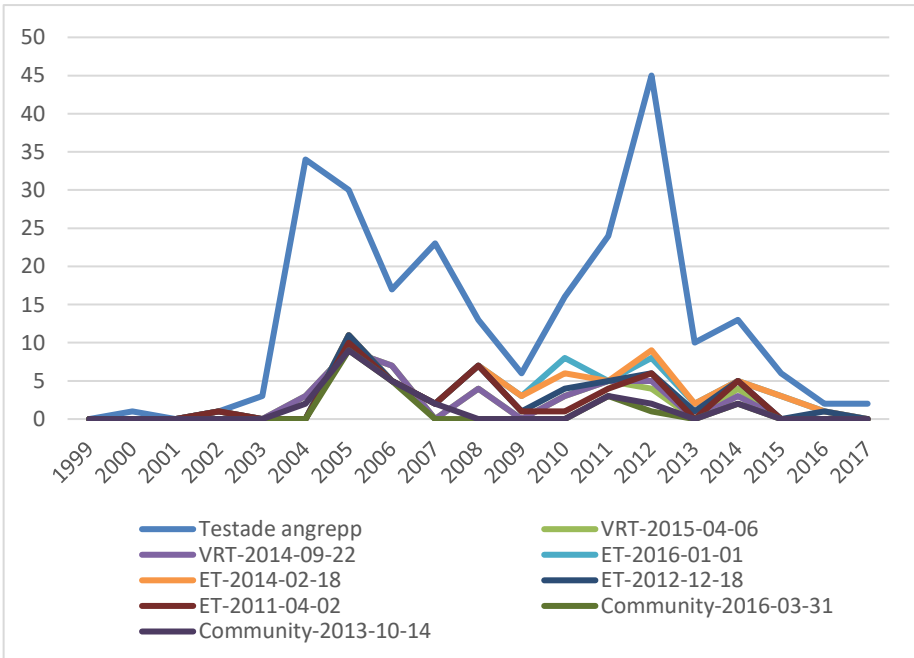
Tabell 3 visar en sammanställning av hur många angrepp som resulterade i larm av olika slag med olika signaturdatabaser och hur många larm de producerade totalt. De mest relevanta kolumnerna är de för larm av prioritet ett, som bäst motsvarar den hotnivå angreppen hade.

Överlag presterar ET bäst i detta test. De nyare signaturerna från ET upptäcker drygt 50 % fler angrepp än databaserna från VRT och cirka tre gånger så många som community-databaserna. Även den äldsta signaturdatabasen från ET ger oftare larm när det sker angrepp än den fyra år nyare databasen från VRT.

Tabell 3. Antal angrepp som resulterade i larm och antal larm producerade vid angreppen.

		Antal angrepp som resulterade i larm av prioritet					Totalt antal larm genererade av olika prioritet				
Utgivare	**Datum**	**1**	**2**	**3**	**1-2**	**1-3**	**1**	**2**	**3**	**1-2**	**1-3**
VRT	2015-04-06	40	36	43	69	101	63	56	51	119	170
VRT	2014-09-22	39	36	43	91	101	53	56	51	109	160
ET	2016-01-01	61	51	66	91	135	117	84	81	201	282
ET	2014-02-18	60	51	66	92	135	116	84	81	200	281
ET	2012-12-18	49	51	66	80	125	95	84	77	179	256
ET	2011-04-02	42	44	51	75	106	58	75	59	133	192
Community	2016-03-31	20	36	43	51	84	35	56	51	91	142
Community	2013-10-14	25	36	43	56	89	34	56	51	90	141

När det kommer till tidens inverkan på resultatet märks också en viss effekt. Figur 2 visar hur många angrepp som utfördes mot sårbarheter publicerade olika år och hur många av dessa som resulterade i larm av prioritet ett.



Figur 2. Fördelning över när sårbarheter som angreps har publicerats i tid och hur många av dem som resulterade i larm av prioritet ett med olika signaturdatabaser.

Angrepp mot sårbarheter publicerade senare år upptäcks i något mindre utsträckning. Detta är väntat då sårbarheter som blivit publikt kända efter att en signaturdatabas släppts rimligen kan betraktas som okända för de som skriver signaturerna. Detta är sådana angrepp som i cybersäkerhetsterminologi kallas ”zero days”. Att upptäcka sådana kräver framsyn eller generella signaturer. Tabell 4 visar hur stor andel av angreppen som upptäcktes givet att de riktade sig mot sårbarheter publicerade samma år eller tidigare än signaturdatabaserna skapades respektive efter det år signaturdatabaserna skapades. Även här producerar ET bäst och gör på så sätt skäl för namnet ”Emerging threats”, vilket på svenska kan översätts till ”framväxande hot”.

Tabell 4. Andel angrepp som resulterade i larm av prioritet för sårbarheter publicerade fram till det år signaturdatabasen skapades och de som publicerats senare år än signaturdatabasen.

			Sårbarhet publicerad tidigare eller samma år		**Sårbarhet publicerad senare år**	
Utgivare	**Datum**	**Signa-turer**	Angrepp	Detekterade	Angrepp	Detekterade
VRT	2015-04-06	6725	242	16 %	4	0 %
VRT	2014-09-22	5256	236	17 %	10	0 %
ET	2016-01-01	14799	244	25 %	2	0 %
ET	2014-02-18	16026	236	24 %	10	40 %
ET	2012-12-18	13077	213	20 %	33	21 %
ET	2011-04-02	7850	168	18 %	102	11 %
Community	2016-03-31	772	244	8 %	2	0 %
Community	2013-10-14	327	223	10 %	23	9 %
	Medelvärde		**226**	**17 %**	**23**	**10 %**

Signaturer kommenteras bort i signaturdatabaserna av olika skäl. Enligt den officiella hemsidan[5] kan de aktiveras enligt ett av fyra regelverk, som bland annat väger in antalet falska larm en signatur producerar och hur aktuell den angripna sårbarheten är. Grundutförandet är tänkt att representera en balanserad avvägning. Några enklare körningar gjordes också med de signaturer som kommenterats bort for att se om dessa skulle innebära en stor förändring[6]. Med alla signaturer aktiverade genereras betydligt fler larm av prioritet ett. Med ET från 2016-01-01 produceras larm av prioritet ett för 112 angreppsförsök istället för 61 och det produceras 320 larm av prioritet ett istället för 117; med VRT från 2015-04-06 produceras 124 larm av prioritet ett vid angrepp istället för 40 och 295 larm av prioritet ett produceras istället för 63; med Community från

[5] Se https://www.snort.org/faq/why-are-rules-commented-out-by-default

[6] En handfull signaturer som kommenterats bort i respektive signaturdatabas var inkompatibla med den version av Snort som användes eller var felaktigt konstruerade.

2016-03-31 produceras larm av prioritet ett vid 48 angrepp istället för 20 och 97 larm av prioritet ett istället för 36. Typiskt fördubblas alltså antalet angrepp som producerar larm av rätt prioritet och antalet larm av rätt prioritet tredubblas. Det gäller även angrepp som är nyare, och alltså inte utnyttjar äldre sårbarheter. Med alla signaturer aktiva pptäcker exempelvis ET från 2012-12-18 hela 29 angrepp mot sårbarheter som blev publika 2013 och framåt, jämfört med 7 utan alla signaturer aktiverade.

Angreppen riktar sig mot olika typer av mål. Tabell 5 visar hur stor andel av angreppen som genererar larm av prioritet ett med utgångspunkt från den indelning av angrepp som görs i Metaploits ramverk. Endast grupper med fler än tio angreppsförsök inkluderas här, och de är inte ömsesidigt uteslutande. Här märks tydliga skillnader. Exempelvis tycks ET vara bättre på att upptäcka angrepp mot webapplikationer och angrepp som görs via protokollet HTTP. Generellt är signaturdatabaserna även bättre på att ge larm när angrepp sker mot Windows-maskiner än när de sker mot Linux-maskiner.

Tabell 5. Andelen angrepp av olika typ som resulterade i larm av prioritet ett.

Utgivare	Datum	Windows	Multi	Linux	Browser	HTTP	Webapp
VRT	2015-04-06	25 %	13 %	13 %	5 %	9 %	7 %
VRT	2014-09-22	25 %	13 %	9 %	5 %	8 %	7 %
ET	2016-01-01	35 %	49 %	9 %	10 %	28 %	27 %
ET	2014-02-18	35 %	44 %	9 %	15 %	26 %	27 %
ET	2012-12-18	31 %	31 %	9 %	10 %	14 %	27 %
ET	2011-04-02	31 %	26 %	9 %	0 %	12 %	20 %
Community	2016-03-31	18 %	5 %	4 %	0 %	0 %	0 %
Community	2013-10-14	18 %	8 %	4 %	0 %	1 %	0 %
Medelvärde		**27 %**	**23 %**	**8 %**	**6 %**	**12 %**	**14 %**
Antal angrepp		**55**	**39**	**23**	**20**	**85**	**15**

4 Diskussion och slutsatser

Resultatet av de två testerna kan sammanfattas så här:

- Endast 4,5 % av angreppsförsöken med publika attackkoder lyckades och endast 8,5 % av de testade angreppskoderna fungerade mot någon maskinkonfiguration. Det finns en svag koppling mellan angiven tillförlitlighet och faktisk tillförlitlighet.
- Den bästa signaturdatabasen producerade larm av prioritet ett för 61 av de 246; den sämsta producerade endast larm av prioritet ett för 20 av angreppen. Nyare signaturdatabaser är något bättre; publikt kända sårbarheter upptäcks oftare; angrepp mot Windowsmaskiner upptäcks oftare än de som är generella; de som är generella upptäcks oftare än de mot Linuxmaskiner; många inaktiverade signaturer kan tillföra värde.

Implikationen av det första testet är både positiv och negativ ur ett cybersäkerhetsperspektiv. Det är positivt att angrepp sällan fungerar trots att det antyds att de ska fungera, vilket försvårar för angripare. Det är negativt då riskanalyser skulle förenklas om det gick att utgå från att den stora risken fanns då en sårbarhetsskanning anger att sårbarhet med publik angreppskod fanns i nätverket. Att träffsäkert prioritera vilka sårbarheter som ska åtgärdas eller gissa vilka angreppsförsök som lyckats tycks kräva mer information än så. Till viss del kan de låga siffrorna bero på förutsättningarna för testet. Till exempel behövs ibland att en användare startar en applikation och tar sig igenom en dialog för att den ska aktiveras och många av de maskiner som installerats i CRATE är sparsamt använda. Det är därför inte säkert att alla sårbara mjukvaror/funktioner är aktiverade eller fullt installerade på dem.

Det finns flera resultat i denna rapport som kan användas som utgångspunkt för den som ska besluta kring signaturuppsättning för Snort. Resultatet visar att valet av signaturdatabas spelar stor roll. Den allmänt tillgängliga ET-databasen ger till exempel larm betydligt oftare än den allmänt tillgängliga community-databasen. Resultatet visar också att det finns ett värde i att uppdatera signaturdatabasen, även om äldre signaturer också kan upptäcka nya angrepp. Därtill kan det finnas skäl att gå igenom de signaturer som lämnats inaktiva (dvs. kommenterats bort). Att aktivera sådana kan leda till betydande ökning i andelen upptäckta angrepp, men också att ett stort antal falsklarm produceras. Som nämnts tidigare är falsklarm ett stort problem med Snort. Falsklarm, liksom mycket annat, bör studeras i framtiden.

Även om de resultat som presenteras i denna rapport är förhållandevis tydliga kommer FOI fortsätta utföra ytterligare tester av både angreppskoders tillförlitlighet och signaturers träffsäkerhet. Till exempel för att undersöka hur ofta signaturdatabaserna producerar falsklarm. Dessutom kommer arbete med testverktygen att utföras. Det är bland annat önskvärt att utreda varför

trafikinspelningen blev korrupt och varför 75 av de 200 modifierade angreppskoderna inte exekverades som de skulle under försöket. Inte minst för att kunna göra bättre tester av liknande slag i framtiden. Testerna i denna rapport ska alltså ses som ett delresultat från pågående arbete – mer utförliga tester av liknande slag kommer att utföras i framtiden.

5 Referenser

[1] B. Newhouse, S. Keith, B. Scribner och G. Witte, *Draft NIST SP 800-181, NICE Cybersecurity Workforce Framework (NCWF): National Initiative for Cybersecurity Eduction (NICE)*, 2017.

[2] P. Cichonski, T. Millar, T. Grance och K. Scarfone, *Computer Security Incident Hochling Guide Recommendations of the National Institute of Stochards och Technology*, 2012.

[3] ENISA, *Good practice guide for incident handling*, 2010.

[4] T. Sommestad och A. Hunstad, *Intrusion detection and the role of the system administrator*, Inf. Manag. Comput. Secur. (2013), .

[5] T. Sommestad och H. Holm, *Variabler av vikt för förmågan att analysera cybersäkerhetsloggar (FOI-R--4126--SE)*, Linköping, Sweden, 2015.

[6] T. Sommestad, Experimentation on operational cyber security in CRATE, in NATO STO-MP-IST-133 Specialist Meeting, 2015, pp. 7.1-7.12.

[7] JASON, *Science of Cyber-Security*, McLean, Virginia, 2010.

[8] T. Sommestad och H. Holm, *Utveckling av CRATE inom ÖvExCy (FOI Memo 5502)*, Linköping, Sweden, 2015.

[9] 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. 3 (2000), pp. 262–294.

[10] S.C. Sundaramurthy, L. Zomlot och X. Ou, Practical IDS alert correlation in the face of dynamic threats, i The 2011 International Conference on Security and Management, 2011.

[11] T. Sommestad och H. Holm, *Test av logganalysverktyget SnIPS*, Linköping, Sweden, 2016.

[12] T. Sommestad och H. Holm, *Alert verification through alert correlation—An empirical test of SnIPS*, Inf. Secur. J. A Glob. Perspect. 26 (2017), pp. 39–48.

[13] S. Liu och B. Cheng, *Cyberattacks: Why, what, who, and how*, IT Prof. Mag. 11 (2009), pp. 14.

[14] *NATO: changing gear on cyber defence*.

[15] C. Lévy-Bencheton, L. Marinos, T. King, C. Dietzel, J. Stumpf och R. Mattioli, *Threat Landscape and Good Practice Guide for Internet Infrastructure*, ENISA, 2015.

[16] L. Allodi, W. Shim och F. Massacci, Quantitative assessment of risk reduction with cybercrime black market monitoring, i Security and Privacy

Workshops (SPW), 2013 IEEE, 2013, pp. 165–172.

[17] L. Allodi och F. Massacci, A preliminary analysis of vulnerability scores for attacks in wild: the ekits and sym datasets, i Proceedings of the 2012 ACM Workshop on Building analysis datasets and gathering experience returns for security, 2012, pp. 17–24.

[18] S. Ransbotham, S. Mitra och J. Ramsey, *Are Markets for Vulnerabilities Effective?*, MIS Q. 36 (2012), pp. 43–64.

[19] M. Dondo, J. Risto och R. Sawilla, *Reliability of exploits and consequences for decision support*, NATO STO-MP-SAS-106 (2015), .

[20] H. Holm och T. Sommestad, SVED: Scanning, Vulnerabilities, Exploits and Detection, i MILCOM 2016, 2016.

[21] H. Holm, *Teknikutveckling under 2016 inom Övning och Experiment i Cybermiljön (ÖvExCy) (FOI Memo 5856)*, (2016), .

[22] M. Roesch, *Snort: Lightweight Intrusion Detection for Networks.*, LISA '99 13th Syst. Adm. Conf. (1999), pp. 229–238.

[23] G. Tjhai, M. Papadaki, S.M. Furnell och N.L. Clarke, Investigating the problem of IDS false alarms: An experimental study using Snort, i Proceedings of The Ifip Tc 11 23rd International Information Security Conference, 2008, pp. 253–267.

[24] R. Werlinger, K. Hawkey, K. Muldner, P. Jaferian och K. Beznosov, The challenges of using an intrusion detection system: is it worth the effort?, i SOUPS '08 Proceedings of the 4th symposium on Usable privacy and security, 2008, pp. 107–118.

[25] J.R. Goodall, W.G. Lutters och A. Komlodi, *Developing expertise for network intrusion detection*, Inf. Technol. People 22 (2009), pp. 92–108.

[26] T. Sommestad och A. Hunstad, *Intrusion detection and the role of the system administrator*, Inf. Manag. Comput. Secur. 21 (2013), pp. 30–40.

[27] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin och K.-Y. Tung, *Intrusion detection system: A comprehensive review*, J. Netw. Comput. Appl. 36 (2013), pp. 16–24.

[28] A. Patcha och J.M. Park, *An overview of anomaly detection techniques: Existing solutions and latest technological trends*, Comput. Networks 51 (2007), pp. 3448–3470.

[29] H. Holm, Signature Based Intrusion Detection for Zero-Day Attacks: (Not) A Closed Chapter?, i 2014 47th Hawaii International Conference on System Sciences, 2014, pp. 4895–4904.

[30] E. Raftopoulos och X. Dimitropoulos, *A quality metric for IDS signatures: in*

the wild the size matters, EURASIP J. Inf. Secur. 2013 (2013), pp. 7.

[31] A. Zand, G. Vigna, X. Yan och C. Kruegel, Extracting probable command and control signatures for detecting botnets, i Proceedings of the ACM Symposium on Applied Computing, 2014, pp. 1657–1662.

[32] Z. Liang och R. Sekar, *Automatic generation of buffer overflow attack signatures: An approach based on program behavior models. In Proceedings of the 21st*, Annu. Comput. Secur. Appl. Conf. December (2005), pp. 215–224.

[33] H. Altwaijry och K. Shahbar, *(WHASG) automatic SNORT signatures generation by using honeypot*, J. Comput. 8 (2013), pp. 3280–3286.

[34] A.N. Singh, S. Kumar och R.C. Joshi, *Intrusion detection system based on real time rule accession and honeypot*, Commun. Comput. Inf. Sci. 196 CCIS (2011), pp. 292–301.

[35] H.J. Wang, C. Guo, D.R. Simon och A. Zugenmaier, *Shield: Vulnerability-driven Network Filters for Preventing Known Vulnerability Exploits*, Proc. 2004 Conf. Appl. Technol. Archit. Protoc. Comput. Commun. (2004), pp. 193–204.

[36] M. Chandrasekaran, M. Baig och S. Upadhyaya, AVARE: Aggregated Vulnerability Assessment and Response against Zero-day Exploits, i 2006 IEEE International Performance Computing and Communications Conference, 2006, pp. 603–610.

[37] H. Li, G. Liu, W. Jiang och Y. Dai, Designing snort rules to detect abnormal DNP3 network data, i ICCAIS 2015 - 4th International Conference on Control, Automation and Information Sciences, 2015, pp. 343–348.

[38] T. Tran, I. Aib, E. Al-Shaer och R. Boutaba, An evasive attack on SNORT flowbits, i Proceedings of the 2012 IEEE Network Operations and Management Symposium, NOMS 2012, 2012, pp. 351–358.

[39] T. Sommestad och U. Franke, *A test of intrusion alert filtering based on network information*, Secur. Commun. Networks 8 (2015), pp. 2291–2301.

[40] S. Riebach, E.P. Rathgeb och B. Toedtmann, *Efficient Deployment of Honeynets for Statistical and Forensic Analysis of Attacks from the Internet*, i *Pr*oceedings of the 4th IFIP-TC6 International Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication Systems, Waterloo, Canada, 2005, pp. 756–767.

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