Projektnummer/Project no
E38530
Uppdragsgivare/Client
Försvarsmakten
FoT-område
Operationer i cyberdomänen

Författare/Author
Hannes Holm, Lars Helgeson

Datum/Date
2024-11-29
Memo nummer/Number
FOI Memo 8662

Utvärdering av verktyg som emulerar hotaktörer

1 Inledning

Detta memo presenterar årets omvärldsbevakning inom projektet *Verktyg och Experiment för Computer Network Operations* (VECNO). Memot beskriver förstudie till en större studie vars resultat förväntas presenteras på vetenskaplig konferens.

Årets arbete är en fortsättning på en serie tester av olika automatiska angreppsverktyg. Tidigare studier inom projektet är:

- FOI memo 6941 (2019): tester av nio olika angreppsverktyg
- FOI memo 7365 (2020): test av angreppsverktyget Caldera[1]
- FOI memo 8354 (2023): test av angreppsverktygen Caldera och Infection Monkey[2].

Av verktyg (mestadels) skrivna i öppen källkod kan Caldera ses som den största konkurrenten till Lore. Av denna anledning har flera tidigare studier fokuserat på just Caldera. De största skillnaderna gentemot tidigare studier är:

- det gjordes ett mer omfattande arbete för att identifiera verktyg
- verktygens egenskaper och utfall kartlades mot MITRE ATT&CK[3]
- verktygen har genomgått anpassningar för att fungera i Crate
- den empiriska studien av hur väl verktygen presterar var mer omfattande.

Målet med studien var att identifiera hur väl olika automatiska angreppsverktyg realiserar en framgångsrik hotemulering. Med framgångsrik hotemulering menas hur väl utfallet från tillämpning av ett verktyg matchar mot förväntade utfall för olika taktiker och tekniker i MITRE ATT&CK.

2 Metod

Det utfördes en sökning[4] mot GitHub för att identifiera relevanta verktyg. Som stöd för att säkerställa att relevanta träffar erhölls skapades en lista med verktyg som bedömdes viktiga att inkludera. Denna lista grundades på våra tidigare studier. För att begränsa studien till förhållandevis uppdaterade och relativt populära verktyg tillämpades två krav:

- de behövde vara uppdaterade senast under 2022

[1] https://github.com/mitre/Caldera

[2] https://github.com/guardicore/monkey

[3] https://attack.mitre.org

[4] Söktermerna var: OR(["network", "security"]) AND OR(["vulnerability", 'metasploit', "nmap", "attack", "pentest", "penetration", "pen-test", "adversary", "red-team", "red-teams", "'red team"']) AND OR(["'reinforcement learning"', "self-learning", 'auto', "automation", "automated", "automatic", "emulation", "emulated", "intelligence", "reasoning", "autopwn"])

- de behövde ha 50 GitHub-stjärnor.

Sökningen identifierade 23 257 GitHub-projekt. Denna lista bantades sedan ner till 354 träffar genom tillämpning av ej önskvärda nyckelord, vilka matchades mot verktygens readme-filer och beskrivningar. Utöver vårt egenskapade verktyg Lore (som ej är öppen källkod) identifierades fem verktyg:

- AutoPentest-DRL[5]
- Deep Exploit[6]
- DeathStar[7]
- Caldera
- Infection Monkey.

Av dessa verktyg exkluderades AutoPentest-DRL och DeathStar efter kodgranskning och tekniska tester då de inte fungerade. AutoPentest-DRL fungerade bara för ett hårdkodat exempel medan DeathStar och Powershell Empire (det verktyg som Deathstar automatiserar) hade kritiska funktionsfel i samtliga testade versioner.

2.1 Testmiljö

Crate nyttjades som plattform för att testa verktygen. En ny miljö kallad ”TESTCO” skapades för ändamålet. Denna miljö var en klon av den miljö som tillämpades för två cybersäkerhetsövningar under 2024, fast med ett antal extra maskiner som tillhandahöll exploaterbara mjukvaruservrar. En översikt av nätverkstopologin ges i appendix A. De viktigaste säkerhetsrelaterade egenskaperna presenteras nedan:

- två nätverkssegment hade särskilda brandväggsregler: SCADA kunde bara nås från HMI, och HMI kunde bara nås från HQCLIENT
- det fanns två Windows-domäner: en för maskiner på HQCLIENT, DMZ, SRV och STOCKHOLM, och en för maskiner på HMI och SCADA
- flera av Linux-maskinerna på HQCLIENT hade servrar med mjukvarusårbarheter som kunde medföra fjärrkodsexekvering
- alla maskiner hade fjärråtkomststjänster aktiverade (SSH, WinRM, WMIC eller PSEXEC)
- vanliga härdningsfunktioner, såsom Applocker och Windows Credential Guard[8], tillämpades inte på någon av Windows-maskinerna
- det fanns inga aktiva skydd för att förhindra skadliga kommandon på maskiner (såsom anti-virus)
- alla maskiner hade ett gemensamt och lättknäckt[9] lösenord för systemanvändarkonton (root, administrator, admin eller localadministrator, beroende på maskin)
- övervakningssystemet Wazuh tillämpades för att detektera hot i miljön
- Windows-användare som bland annat surfade på webbsidor simulerades i kontorsmiljön genom det FOI-utvecklade verktyget SVED[10].

---

[5] https://github.com/crond-jaist/AutoPentest-DRL

[6] https://github.com/TheDreamPort/deep_exploit och https://github.com/13o-bbr-bbq/machine_learning_security/tree/master/DeepExploit

[7] https://github.com/byt3bl33d3r/DeathStar

[8] https://learn.microsoft.com/en-us/windows/security/identity-protection/credential-guard/

[9] Lösenordet var med i vanliga lösenordslistor för lösenordsknäckning, såsom https://github.com/rapid7/metasploit-framework/blob/master/data/wordlists/common_roots.txt

[10] Holm, Hannes och Teodor Sommestad. "SVED: Scanning, vulnerabilities, exploits and detection." *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016.

Miljön genererades i fyra exemplar för att möjliggöra parallella tester av verktygen.

2.2 Testade scenarier och anpassning av verktyg

Två av verktygen (Caldera och Infection Monkey) ställde krav på initial åtkomst till ett målsystem för att kunna fungera. Med grund i detta utvärderades fyra olika scenarier:

1. initial åtkomst till en Windows-arbetsdator i kontexten för en klientadministratör i Windows-domänen
2. initial åtkomst till en Windows-arbetsdator i kontexten för en vanlig användare (ej heller administratör för den egna maskinen)
3. initial åtkomst till en Linux-arbetsdator i kontexten för root
4. initial åtkomst till en Linux-arbetsdator i kontexten för en vanlig användare (ej med sudo-behörighet på maskinen).

Testproceduren för dessa fyra scenarier var som följande:

1. återställa den virtuella miljön genom snapshots
2. säkerställa att miljön var i ett fungerande tillstånd
3. flytta fjärrstyrningsservrar (eng: Command and Control, C2) till nätverket HQCLIENT
4. ladda upp och exekvera en agent på äskad dator genom VirtualBox
5. utföra hotemuleringen
6. ladda ner loggar från Wazuh
7. producera statistik.

Var och en av de fyra scenarierna beskrivna ovan exekverades under sex timmar. Denna tidsrymd valdes för att den kan tänkas representera en vanlig arbetsdag (8 timmar givet 2 timmar för förberedelse och analys). Verktygen konfigurerades översiktligt som följande:

- det delgavs ingen information om målsystemen
- det definierades inga mål
- den mest omfattande angreppsprofilen valdes
- en ordlista som enbart fick användas för lösenordsknäckning[11] av erhållna lösenordshashar tillhandahölls
- alla verktyg exekverades helt i Crate – dvs., de nyttjade samma typ av hårdvara med liknande prestandabegränsningar
- kommunikationsservrar (eng: Command and Control [C2]) placerades på kontorsnätverket där maskinerna som nyttjades för initial åtkomst befann sig för att minska behovet av pivotering.[12]

Mer verktygsspecifika konfigurationer ges i följande avsnitt.

2.2.1 Deep Exploit

Då den förgrening av Deep Exploit som kvalificerade sig för tester inte var funktionell (exempelvis innehöll koden referenser till olika med varandra inkompatibla versioner av maskininlärningsbibliotek) togs beslutet att utgå från den ursprungliga versionen från 2019. Även efter installation enligt den tillhandahållna guiden visade sig koden innehålla fel som gjorde att programmet kraschade. Koden patchades för att inte krascha, samt rätta till en bugg som gjorde att det inte gick träna upp en redan upptränad modell mot ytterligare datorer. Deep Exploit har inte

[11] https://github.com/rapid7/metasploit-framework/blob/master/data/wordlists/common_roots.txt. Denna ordlista fick ej nyttjas för andra ändamål än lösenordsknäckning, såsom att försöka logga in i en maskin via SSH.

[12] I skrivande stund klarar bara Lore av att automatiskt utföra pivotering.

förmågan att först identifiera datorer på ett näverk och sedan testa dessa, utan verktyget behöver startas med en (och endast en) IP-adress som mål, för såväl träning som testning.

Eftersom Deep Exploit använder Metasploit för att utföra angrepp, och endast ett fåtal (5) av maskinerna i testmiljön hade sårbarheter i servermjukvaror som var möjliga att utnyttja, kördes verktyget först i ”träningsläge” mot dessa maskiner. Här visade det sig att trots att Metasploit innehöll moduler för sårbarheterna och att nmap identifierade de körande tjänsterna, valde verktyget inte ut de relevanta modulerna för utvärdering, vilket resulterade i att inga sessioner lyckades etableras.

Med den tränade modellen från föregående steg, kördes sedan verktyget i ”testläge” mot samtliga maskiner i miljön.

2.2.2 Caldera

Det finns två huvudsakliga val att ta ställning till gällande Caldera: val av planerare och val av vilka moduler (kallade ”abilities” i Caldera) som är tillåtna att exekveras. Planerare är den huvudsakliga metoden för hur ett test skall utföras. I skrivande stund är planeraren ”batch” den som utvecklarna av Caldera rekommenderar för scenarier som innefattar repeterbara moduler, där utförandet av en modul kan möjliggöra utförandet av en annan modul:

> ”The batch planner should be used for profiles containing repeatable abilities.”[13]

Av denna anledning nyttjades batch-planeraren. Den ursprungliga planen var att automatisera alla moduler i Caldera (2157 stycken). Denna plan fick dock revideras då det visade sig att:

- flera moduler startar om maskinen som kör agenten utan att först skapa persistens (vilket därmed avslutar testet)
- exekvering av alla moduler utom förstörande moduler blir för tungt för verktyget trots att den exekverande maskinen har delgivits så mycket resurser som är möjligt i Crate.

Utöver dessa två anledningar är inte alla moduler lämpade för automatiserad exekvering då de saknar regler som specificerar när de är giltiga att exekvera. Av dessa anledningar valde vi att enbart tillämpa de 162 modulerna i kategorin ”stockpile”. Denna kategori är den som utvecklarna av Caldera tänkt skall automatiseras med batch-planeraren. Till skillnad från moduler i exempelvis ”atomic” finns det definierade regler som säkerställer att de körs i en korrekt ordning.

Alla moduler analyserades i syfte att identifiera verktyg som laddades ner från Internet. Då Internet-åtkomst saknas i miljöerna, laddades verktyg (skript, komprimerade arkiv, installationspaket etc.) ner för att kunna tillhandahållas lokalt, och sökvägarna i modulerna anpassades för detta. Runt hundra unika adresser anpassades på detta sätt. Det är dock värt att påpeka att majoriteten av dessa rörde andra moduler än de som tillhandahölls av ”stockpile”.

Noterbart är att vi också var tvungna att reparera en bugg i Caldera som medförde att batch-planeraren slutade att fungera efter cirka en timme.

2.2.3 Infection Monkey

Infection Monkey tillhandahåller funktionalitet som kan exekvera på agenten i form av plugins. Samtliga tillgängliga plugins förutom två (ransomware och cryptojacker) installerades och aktiverades med sina default-konfigurationer. Den yttre brandväggens externa IP-adress lades in i en svartlista för att undvika att bli angripen.

Det visade sig att docker och dess default-konfiguration var ett problem då docker fanns installerat både på servern (injektorn) och på de Linux-baserade maskiner verktyget fick initiala sessioner från.

[13] https://caldera.readthedocs.io/en/latest/Basic-Usage.html#planners

Detta skapade IP-konflikter som medförde att servern inte kunde ta emot några klientsessioner. Docker på injektorn ändrades därför till en annan IP-konfiguration, varefter sessioner etablerades.

2.2.4 Lore

Lore instruerades att nyttja sin default-profil. Inga tröskelvärden eller liknande tillämpades, vilket innebar att Lore tilläts utföra en handling även om dess utfall bedömdes vara oanvändbart, eller i vissa fall till och med skadligt (t.ex. medföra upptäckt eller en kraschad agent).

2.3 Metriker

Metrikerna rör utfall som vi ansåg var relevanta givet målet att uppnå en framgångsrik hotemulering med avseende på utfall för taktikerna i MITRE ATT&CK. Vi skapade metriker för 12 av de 14 taktikerna i ATT&CK:

- **Reconnaissance and Discovery (5 metriker)**: Antal identifierade maskiner, nätverkssegment, Windows-domäner, användarkonton och respektive användargrupper.
- **Initial Access, Execution, Privilege Escalation, and Lateral Movement (4 metriker)**: Övertagna maskiner och nätverkssegment med behörighet som användare eller system/root.
- **Persistence, Credential Access, and Collection (2 metriker)**: Erhållna lösenord och lösenordshashar.
- **Defense Evasion, Command and Control, and Exfiltration (1 metrik)**: Antalet larm med en prioritet på 6 eller högre från intrångsdetektionssystemet Wazuh.

Inga metriker skapades för taktikerna *Resource Development* och *Impact*. Vi bedömde taktiken *Resource Development* som irrelevant för testerna då den innefattar hur hotaktören förbereder sig *inför* ett angrepp mot ett utvalt offer, såsom anskaffning av scenario-relevanta behörigheter och infrastruktur innan angreppet utförs. Taktiken *Impact* bedömdes som irrelevant för testerna då den innefattar scenario-specifika handlingar som hotaktören utför när den lyckas med angrepp, exempelvis att kryptera filer eller stänga ned system. Det är därmed avsevärt svårare att generalisera utfallet än för de utvalda metrikerna.

Facit för metrikerna, såsom hur många maskiner som kunde kompromettеras och hur många lösenord som kunde erhållas, extraherades från Crate. Detta medförde vissa begränsningar då Crates datamodell för miljöerna inte innefattade metrik-relevant information som verktygen samlade in. Exempelvis var Windows–kontona ”DefaultAccount” och ”Guest” inte med i Crates datamodell för miljöerna. Dessa begränsningar bedömdes dock som acceptabla och bör inte ge för- eller nackdelar för något av de studerade verktygen.

3 Kartläggning mot MITRE ATT&CK

En översikt av hur de studerade verktygen i teorin uppfyller olika taktiker och tekniker i MITRE ATT&CK presenteras i tabell 1. Tabellen presenterar den procentuella andelen av teknikerna som har åtminstone en delteknik som stöds av de olika verktygen. Siffrorna inom parenteserna anger den procentuella andelen deltekniker som stöds inom varje taktik.

Kartläggningen utfördes av författarna genom tester och studier av verktygens olika förmågor. Ett verktyg måste tillhandahålla stöd för ett generellt testfall för att en delteknik skall anses stödjas. Exempelvis kan Lore enbart skicka automatiska nätfiske-epost om det används i datorklustret Crate. Nätfiske-tekniker i ATT&CK ansågs därmed inte som uppfyllda för Lore. Utöver detta krav ansågs det som acceptabelt om ett verktyg bara nyttjade en förmåga för detektionssyfte. Exempelvis finns det ett stort antal förmågor i Caldera som enbart utförs för att testa intrångsdetektionssystem, såsom att starta kalkylatorn med en eskalerad behörighet.

Som framgår i Tabell 1 har Caldera störst stöd, där åtminstone en delteknik stöds för 54% av alla tekniker i ATT&CK. Lore hade näst högsta snittstöd på 51%. Den största skillnaden mellan verktygen är att Caldera stödjer många fler Impact-typ-tekniker, såsom moduler som ger störningar eller avbrott. I Lore är det i allmänhet[14] tänkt att det är upp till operatören att planera sådana handlingar. Deep Exploit och Infection Monkey har lågt stöd för tekniker i ATT&CK.

Tabell 1. Verktygen och MITRE ATT&CK. Antal tekniker inom respektive taktik samt den procentuella andelen av teknikerna inom respektive taktik som har åtminstone en delteknik som stöds av de olika verktygen. Siffrorna inom parenteserna anger den procentuella andelen deltekniker som stöds inom varje taktik.

Taktik	**Antal tekniker**	**Caldera**	**Deep Exploit**	**Infection Monkey**	**Lore**
Reconnaissance	10	20 (11)	10 (6)	20 (6)	50 (40)
Resource Development	8	0 (0)	13 (3)	0 (0)	38 (18)
Initial Access	10	30 (28)	10 (6)	30 (28)	60 (50)
Execution	14	57 (50)	0 (0)	14 (6)	64 (72)
Persistence	20	70 (28)	0 (0)	0 (0)	50 (13)
Privilege Escalation	14	93 (33)	0 (0)	0 (0)	71 (26)
Defense Evasion	43	64 (49)	5 (2)	2 (2)	42 (27)
Credential Access	17	59 (52)	0 (0)	12 (5)	71 (52)
Discovery	32	81 (75)	3 (3)	0 (0)	69 (63)
Lateral Movement	9	56 (50)	11 (5)	44 (30)	67 (55)
Collection	17	59 (40)	0 (0)	0 (0)	59 (37)
Command and Control	18	44 (27)	33 (27)	6 (3)	50 (39)
Exfiltration	9	44 (43)	0 (0)	0 (0)	22 (14)
Impact	14	79 (50)	0 (0)	7 (5)	7 (5)
Medelvärde	*17*	*54 (38)*	*6 (4)*	*10 (6)*	*51 (36)*

[14] Det finns undantag, t.ex. profilerna ”APT29” och ”RYUK”, som bland annat krypterar filer på vissa övertagna maskiner.

4 Resultat från tester

Medelvärden och standardavvikelser från alla tester (12 per verktyg) för de olika metrikerna efter testernas utförande presenteras i Tabell 2. I tabellen (och resultatet i övrigt) exkluderas maskinen som verktygen ursprungligen fick en agent på samt dess nätverkssegment från övertagna maskiner och nätverk. Motsvarande tabeller för de fyra olika scenarierna ges i appendix B. Utfallet över tid för de studerade metrikerna presenteras av figurerna i appendix C. I dessa figurer representerar de tjocka linjerna medelvärden och de färgade volymerna 95% konfidensintervall för de tester som genomfördes för varje verktyg.

Att utfallet skiljer sig åt mellan tester beror på att miljön i Crate inte beter sig på precis samma sätt varje gång. Exempelvis kan en agent krascha, eller en back-off funktion för misslyckade SSH-inloggningar blockera anslutningar, i ett test men inte ett annat.

Tabell 2. Översikt av resultat (medelvärden och standardavvikelser från alla tester). För Wazuh presenteras antalet larm, för de övriga metrikerna presenteras andelen i procent baserat på facit från Crate.

Metrik	Facit	Caldera	Deep Exploit	Infection Monkey	Lore
Hittade maskiner (%)	56	29.3 (29.3)	0.0 (0.0)	23.2 (0.0)	100.0 (0.0)
Hittade nätverkssegment (%)	8	48.2 (10.8)	0.0 (0.0)	12.5 (0.0)	100.0 (0.0)
Hittade domäner (%)	2	14.3 (23.4)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användargrupper (%)	8	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användarkonton (%)	134	1.1 (1.0)	0.0 (0.0)	0.0 (0.0)	92.4 (1.2)
Erhållna lösenord (%)	134	0.5 (1.0)	0.0 (0.0)	0.0 (0.0)	10.7 (1.3)
Erhållna lösenordshashar (%)	134	0.6 (1.3)	0.0 (0.0)	0.0 (0.0)	79.8 (19.0)
Övertagna nätverk (administratör) (%)	7	12.2 (20.1)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna nätverk (användare) (%)	7	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	54.8 (5.6)
Övertagna nätverk (%)	7	12.2 (20.1)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna maskiner (administratör) (%)	55	8.2 (13.7)	0.0 (0.0)	0.0 (0.0)	93.5 (2.1)
Övertagna maskiner (användare) (%)	55	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	25.3 (5.9)
Övertagna maskiner (%)	55	8.2 (13.7)	0.0 (0.0)	0.0 (0.0)	96.7 (2.0)
Wazuh larm (>= 6)	-	8923.8 (4667.2)	9760.7 (313.5)	8665.2 (367.0)	9151.6 (1452.9)

Deep Exploit gav inte utslag för någon metrik utöver Wazuh-larm. Verktyget utförde nmap mot alla nåbara adresser i miljön (hårdkodat av oss för att verktyget skulle kunna köra) samt cirka 2000 misslyckade serverangrepp och 4000 anslutningar till webbappar.

Infection Monkey utförde cirka 550 handlingar för Windows-scenarierna och cirka 130 000 handlingar för Linux-scenarierna. Anledningen till detta är att Linux-maskinerna som nyttjades för initial åtkomst hade ett nätverksinterface med en nätmask som medförde att det kunde finnas 65536 datorer på nätverket (172.17.0.1/16, vilket används av docker-installationen på maskinen). Infection Monkey körde nåbarhetstester (ping och TCP-anslutningar) mot alla möjliga adresser. Den lyckades identifiera datorer på agentens lokala nätverk. Den lyckades dock inte ta över några andra datorer eller erhålla några behörigheter, ens på datorn som exekverade agenten. En python-implementation av Mimikatz kallad pykatz[15] kördes, vilket i administratörsscenariot (scenario #1) bör ha extraherat lösenordshashen för administratörskontot. Då agentens loggfiler visade att pykatz till synes hade exekverats korrekt ligger problemet därmed möjligen i den parser som Infection Monkey använde för att tolka utfallet från pykatz.

[15] https://github.com/skelsec/pypykatz

Caldera utförde cirka 9000 handlingar för scenario #1, 1000 handlingar för scenario #2, och cirka 25 000 handlingar för scenario #3 och scenario #4. Den stora mängden handlingar för Linux-scenarierna var för att Caldera genererade cirka 200 agenter på den maskin som verktyget ursprungligen fjärrstyrde, och sedan exekverade ungefär samma skalkommandon på varje agent (t.ex. ~200 försök att installera python-paketet stormssh). Caldera lyckades ta över cirka 17 datorer givet scenario #1, men inga datorer givet de övriga scenarierna. Anledningen är att Caldera enbart spred sig genom wmic, och detta med antagandet att den agentens användarkontext hade behörighet att exekvera kod på andra datorer. I scenario #1, där agenten kördes i kontexten för en klientadministratör, fungerade detta. I scenario #2, där agenten fick en vanlig användare, lyckades den ej eskalera till en domänadministratör (eller administratör för den lokala maskinen för den delen).

Lore utförde runt 6000 handlingar per test och lyckades ta de allra flesta datorer oavsett scenario. Den översiktliga metoden var ungefär samma oavsett scenario:

- identifiering av domäner, nätverk och maskiner genom exempelvis nmap, BloodHound, omvända DNS-uppslag, ping, netstat och sniffning av nätverkstrafik
- exploatering av serversårbarheter på vissa av Linux-maskinerna
- erhållning av lösenord och lösenordshashar genom exempelvis Mimikatz, Lazagne och /etc/shadow
- knäckning av lösenordshashar med hashcat
- användning av lösenordshashar och lösenord i klartext för att logga in i fjärrsystem via SSH, WinRM, WMIC och psexec
- pivotering via övertagna maskiner på HMI till SCADA.

Som presenteras i Figur 9 och appendix B producerade verktygen liknande mängder Wazuh-larm för de flesta scenarier. En huvudsaklig anledning till detta var förmodligen falsklarm för aktiviteterna utförda av de simulerade konstorsanvändarna. Tidpunkten för larmen skiljer sig något mellan verktygen: För Deep Exploit och Infection Monkey genereras nästan alla larm tidigt under testerna, medan de är mer utspridda för Caldera och Lore. Val av scenario påverkar främst larmen skapade för Caldera, som ger upphov till långt fler larm (cirka 14 500 givet scenario #1 och mellan cirka 4000 och 9000 för de övriga scenarierna). Detta är förmodligen eftersom Caldera huvudsakligen producerar resultat givet scenario #1.

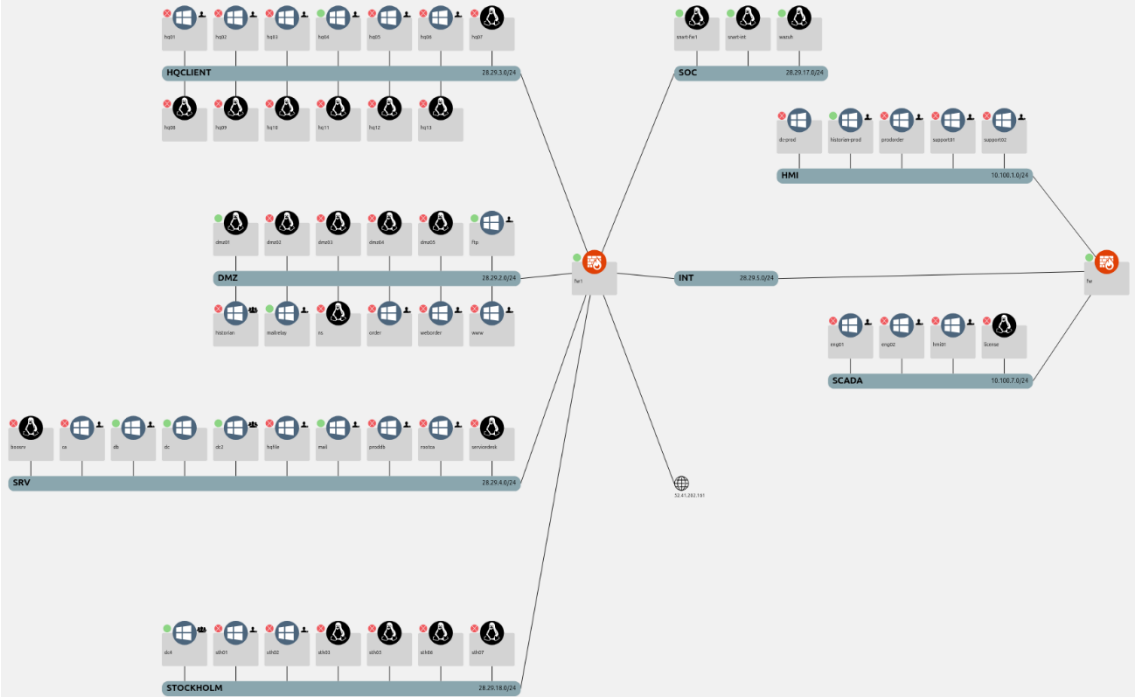
5 Slutsatser och fortsatt arbete

Caldera och Lore var de enda verktygen som lyckades ta över maskiner under testerna. Caldera täcker i teorin in många av teknikerna i MITRE ATT&CK, men i praktiken är verktyget inte särskilt duglig på att automatiserat utföra dessa tekniker. Medan Caldera helt misslyckas med att ta över datorer i tre av fyra scenarier, och i det fjärde scenariot tar över cirka 17 datorer, så tar Lore över i princip alla datorer oavsett scenario.

Vi förmodar att anledningen till detta är att Caldera är tänkt att anpassas för varje scenario, till skillnad från Lore som är tänkt att kunna fungera väl oavsett scenario. När Lore använts i cybersäkerhetsövningar har det istället satts begränsningar på hur verktyget skall agera. Baserat på vår erfarenhet krävs det avsevärt mindre ansträngning och kunskap att begränsa ett beteende än att tillföra nya beteenden.

Detta memo beskriver en förstudie till ett mer omfattande arbete som förväntas publiceras på en vetenskaplig konferens. I synnerhet kommer detta framtida arbete innefatta långt mer empiri och mer ingående analyser.

Appendix A: Testad miljö i Crate



Figur 1. Nätverksskiss för den aktuella testmiljön.

Appendix B: Statistik

Tabell 3. Översikt av medelvärden och standardavvikelser för Scenario #1 (Windows och administratör). För Wazuh presenteras antalet larm, för de övriga metrikerna presenteras andelen i procent baserat på facit från Crate.

Metrik	Caldera	Deep Exploit	Infection Monkey	Lore
Hittade maskiner (%)	72.3 (12.5)	0.0 (0.0)	23.2 (0.0)	100.0 (0.0)
Hittade nätverkssegment (%)	62.5 (0.0)	0.0 (0.0)	12.5 (0.0)	100.0 (0.0)
Hittade domäner (%)	50.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användargrupper (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användarkonton (%)	1.7 (1.4)	0.0 (0.0)	0.0 (0.0)	92.5 (1.5)
Erhållna lösenord (%)	1.7 (1.4)	0.0 (0.0)	0.0 (0.0)	9.7 (1.5)
Erhållna lösenordshashar (%)	2.2 (1.6)	0.0 (0.0)	0.0 (0.0)	79.9 (23.3)
Övertagna nätverk (administratör) (%)	42.9 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna nätverk (användare) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	57.1 (0.0)
Övertagna nätverk (%)	42.9 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna maskiner (administratör) (%)	28.6 (6.0)	0.0 (0.0)	0.0 (0.0)	93.9 (1.0)
Övertagna maskiner (användare) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	28.5 (5.8)
Övertagna maskiner (%)	28.6 (6.0)	0.0 (0.0)	0.0 (0.0)	97.6 (1.0)
Wazuh larm (>= 6)	14428.5 (5234.9)	9492.3 (193.9)	8868.0 (232.9)	9030.0 (985.1)

Tabell 4. Översikt av medelvärden och standardavvikelser för Scenario #2 (Windows och användare). För Wazuh presenteras antalet larm, för de övriga metrikerna presenteras andelen i procent baserat på facit från Crate.

Metrik	Caldera	Deep Exploit	Infection Monkey	Lore
Hittade maskiner (%)	19.6 (0.0)	0.0 (0.0)	23.2 (0.0)	100.0 (0.0)
Hittade nätverkssegment (%)	50.0 (0.0)	0.0 (0.0)	12.5 (0.0)	100.0 (0.0)
Hittade domäner (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användargrupper (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användarkonton (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	93.8 (0.4)
Erhållna lösenord (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	10.7 (1.7)
Erhållna lösenordshashar (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	93.8 (0.4)
Övertagna nätverk (administratör) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna nätverk (användare) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	57.1 (0.0)
Övertagna nätverk (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna maskiner (administratör) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	95.8 (1.0)
Övertagna maskiner (användare) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	22.4 (6.9)
Övertagna maskiner (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	98.8 (1.0)
Wazuh larm (>= 6)	8546.0 (1068.6)	9796.7 (326.3)	8691.0 (215.6)	8805.3 (3039.6)

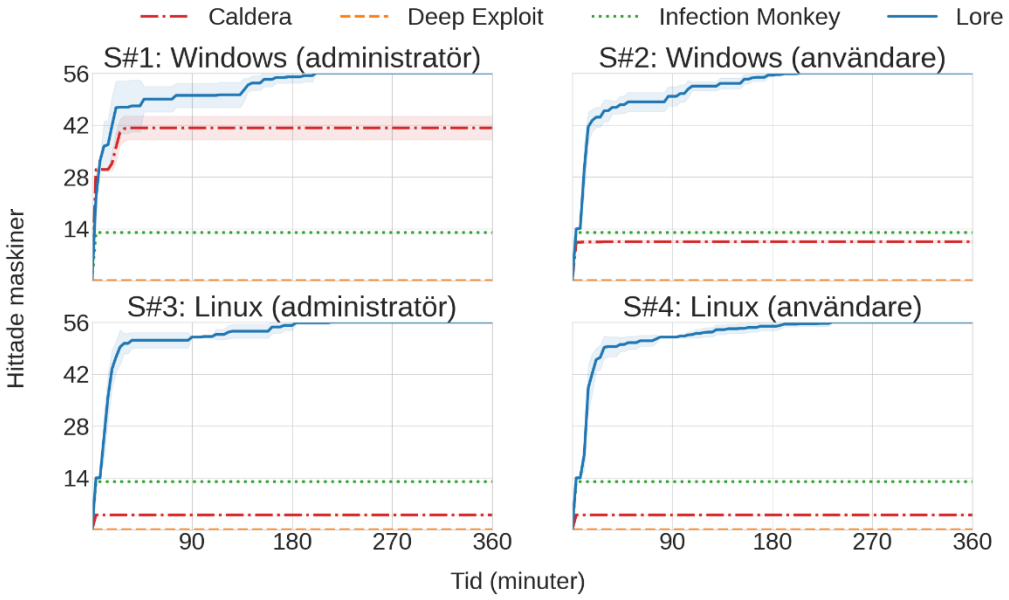
Tabell 5. Översikt av medelvärden och standardavvikelser för Scenario #3 (Linux och root). För Wazuh presenteras antalet larm, för de övriga metrikerna presenteras andelen i procent baserat på facit från Crate.

Metrik	Caldera	Deep Exploit	Infection Monkey	Lore
Hittade maskiner (%)	7.1 (0.0)	0.0 (0.0)	23.2 (0.0)	100.0 (0.0)
Hittade nätverkssegment (%)	37.5 (0.0)	0.0 (0.0)	12.5 (0.0)	100.0 (0.0)
Hittade domäner (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användargrupper (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användarkonton (%)	1.5 (0.0)	0.0 (0.0)	0.0 (0.0)	91.8 (0.7)
Erhållna lösenord (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	11.2 (1.3)
Erhållna lösenordshashar (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	79.6 (19.8)
Övertagna nätverk (administratör) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna nätverk (användare) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	52.4 (8.2)
Övertagna nätverk (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna maskiner (administratör) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	90.9 (1.8)
Övertagna maskiner (användare) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	23.0 (6.9)
Övertagna maskiner (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	94.5 (1.8)
Wazuh larm (>= 6)	4993.0 (1015.1)	9934.3 (316.0)	8398.2 (184.1)	9582.0 (391.3)

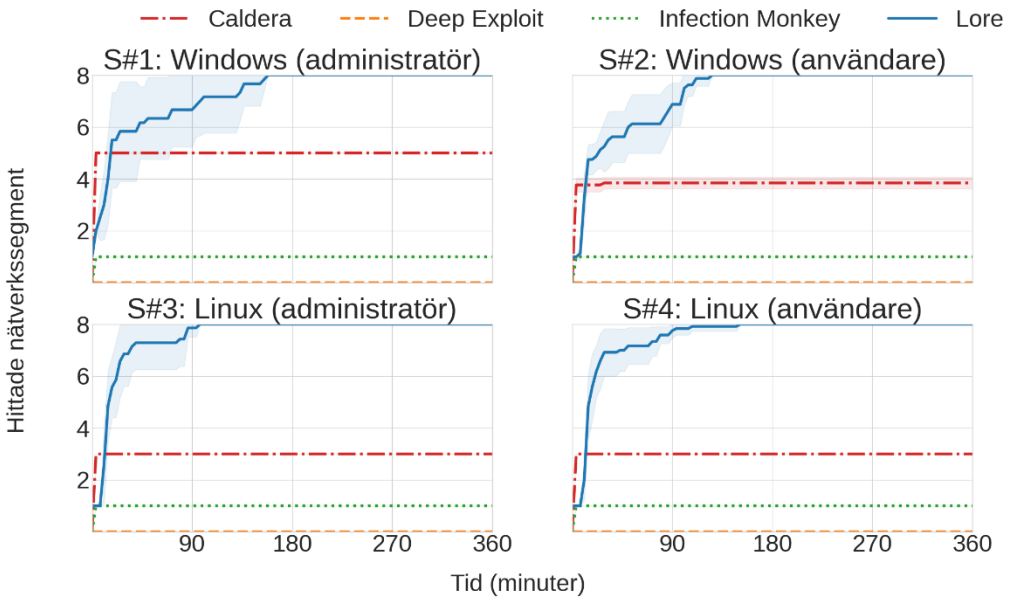
Tabell 6. Översikt av medelvärden och standardavvikelser för Scenario #4 (Linux och användare). För Wazuh presenteras antalet larm, för de övriga metrikerna presenteras andelen i procent baserat på facit från Crate.

Metrik	Caldera	Deep Exploit	Infection Monkey	Lore
Hittade maskiner (%)	7.1 (0.0)	0.0 (0.0)	23.2 (0.0)	100.0 (0.0)
Hittade nätverkssegment (%)	37.5 (0.0)	0.0 (0.0)	12.5 (0.0)	100.0 (0.0)
Hittade domäner (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användargrupper (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Hittade användarkonton (%)	1.5 (0.0)	0.0 (0.0)	0.0 (0.0)	91.3 (0.4)
Erhållna lösenord (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	11.2 (0.0)
Erhållna lösenordshashar (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	65.9 (21.8)
Övertagna nätverk (administratör) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna nätverk (användare) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	52.4 (8.2)
Övertagna nätverk (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)
Övertagna maskiner (administratör) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	93.3 (1.0)
Övertagna maskiner (användare) (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	27.3 (4.8)
Övertagna maskiner (%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	95.8 (1.0)
Wazuh larm (>= 6)	6018.7 (1007.3)	9844.4 (273.1)	8652.8 (633.2)	9189.0 (874.8)

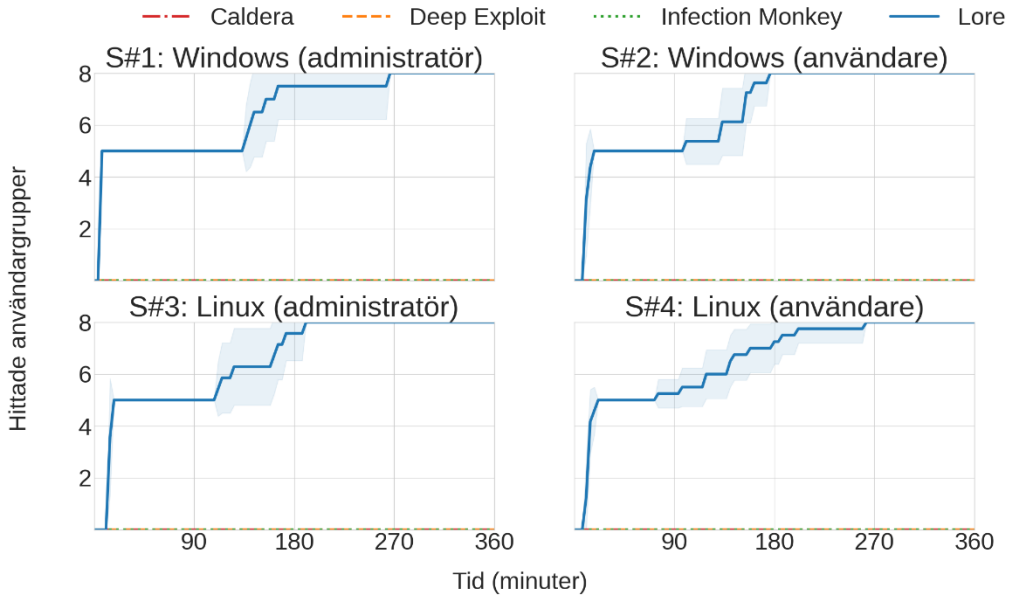
Appendix C: Figurer



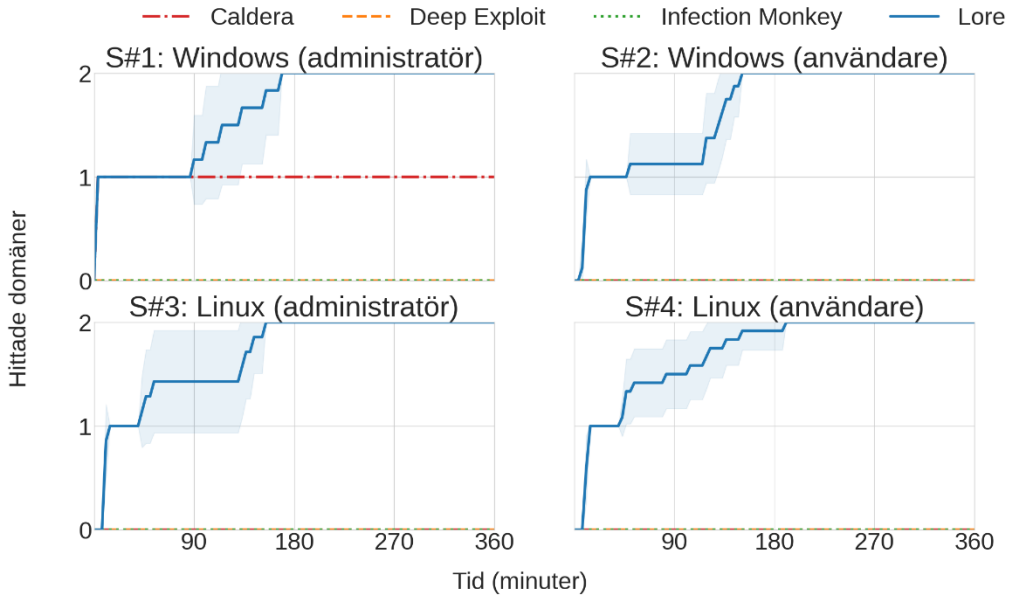
Figur 1. Kumulativa antalet hittade maskiner (av 56 möjliga) för de testade verktygen. De tjocka linjerna är medelvärden och de färgade volymerna är 95% konfidensintervall för de tester som genomfördes för varje verktyg.



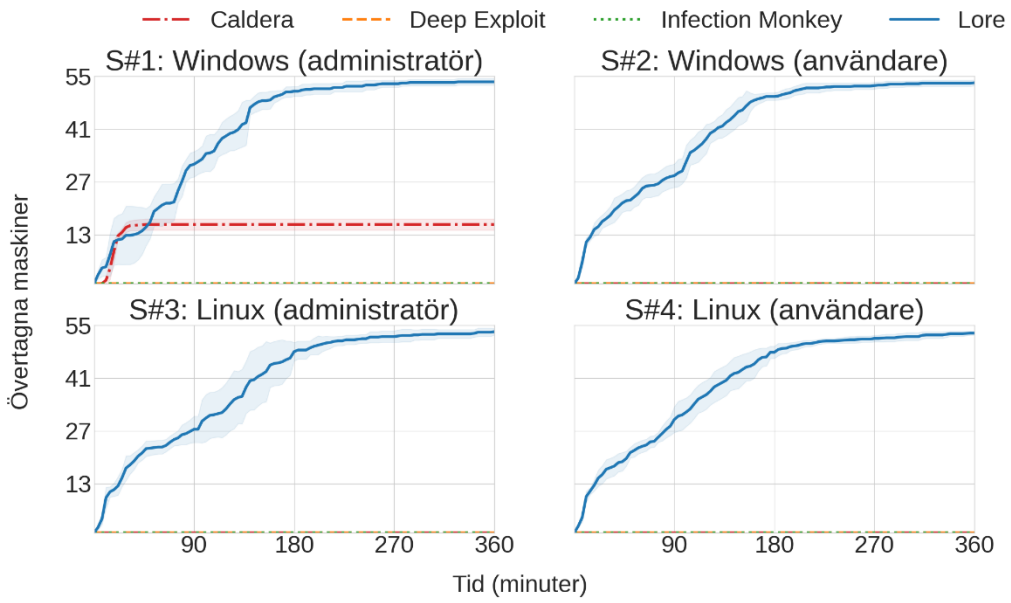
Figur 2. Kumulativa antalet hittade nätverkssegment (av 8 möjliga) för de testade verktygen. De tjocka linjerna är medelvärden och de färgade volymerna är 95% konfidensintervall för de tester som genomfördes för varje verktyg.



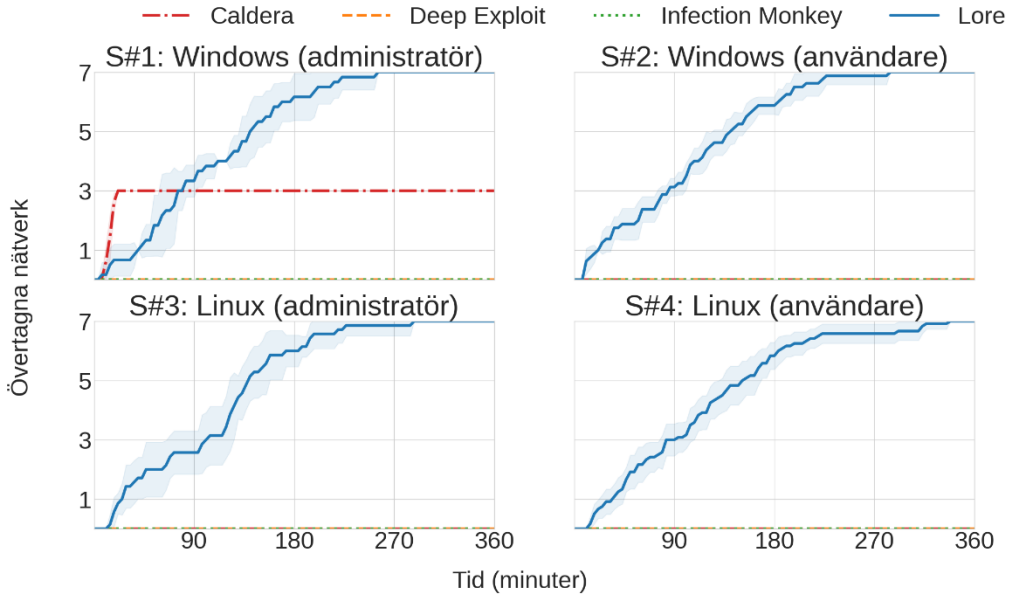
Figur 3. Kumulativa antalet hittade användargrupper (av 8 möjliga) för de testade verktygen. De tjocka linjerna är medelvärden och de färgade volymerna är 95% konfidensintervall för de tester som genomfördes för varje verktyg.



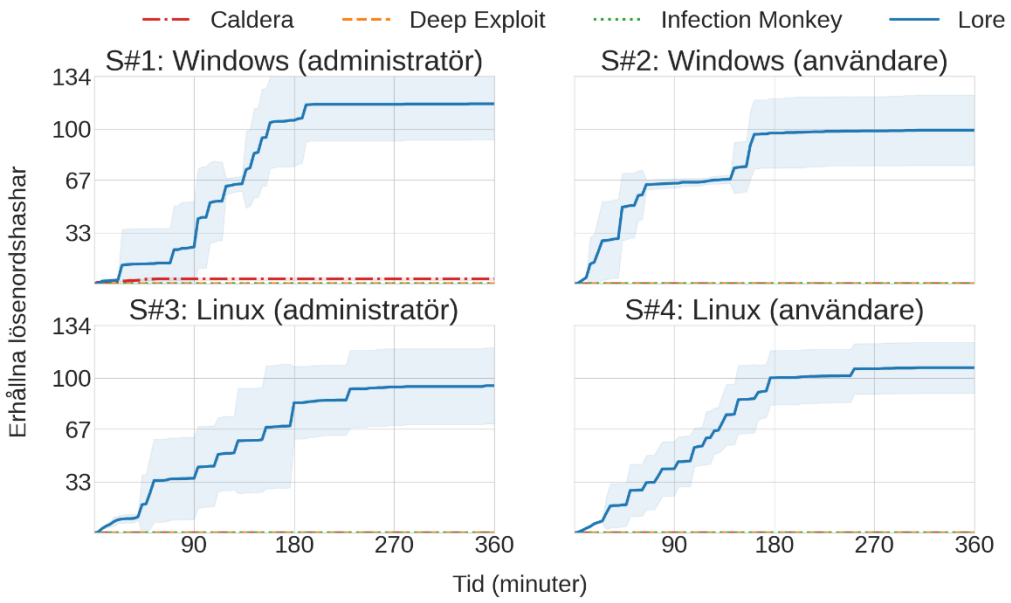
Figur 4. Kumulativa antalet hittade domäner (av 2 möjliga) för de testade verktygen. De tjocka linjerna är medelvärden och de färgade volymerna är 95% konfidensintervall för de tester som genomfördes för varje verktyg.



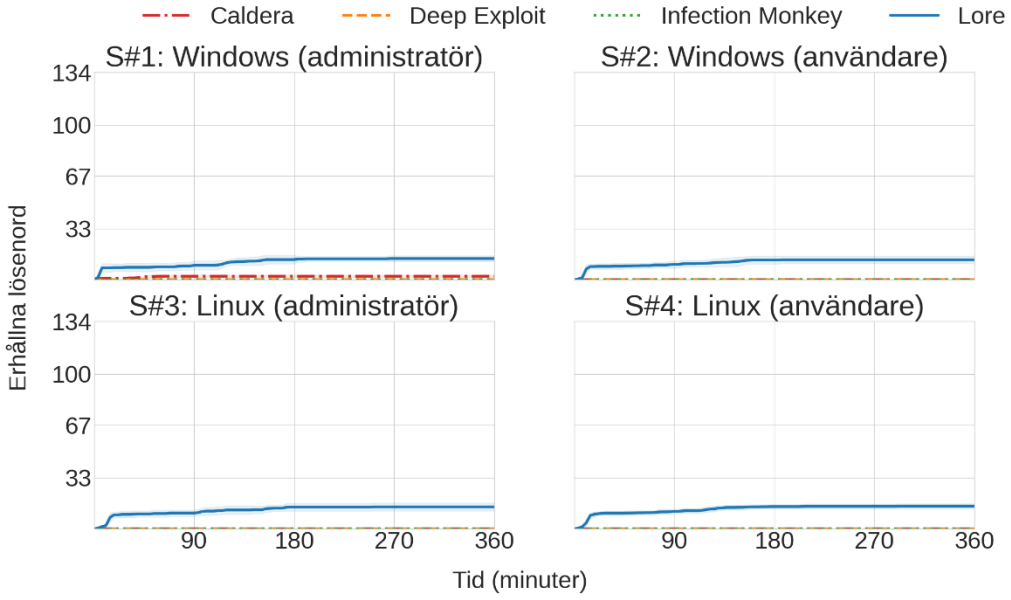
Figur 5. Kumulativa antalet övertagna maskiner (av 55 möjliga) för de testade verktygen. De tjocka linjerna är medelvärden och de färgade volymerna är 95% konfidensintervall för de tester som genomfördes för varje verktyg.



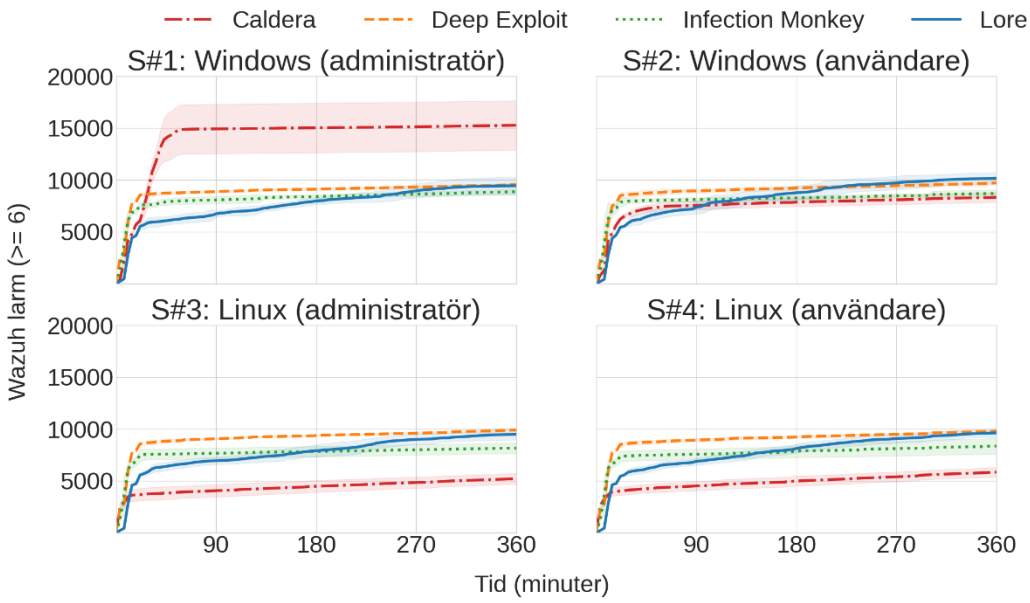
Figur 6. Kumulativa antalet övertagna nätverkssegment (av 7 möjliga) för de testade verktygen. De tjocka linjerna är medelvärden och de färgade volymerna är 95% konfidensintervall för de tester som genomfördes för varje verktyg.



Figur 7. Kumulativa antalet erhållna lösenordshashar för användarkonton (av 134 möjliga) för de testade verktygen. De tjocka linjerna är medelvärden och de färgade volymerna är 95% konfidensintervall för de tester som genomfördes för varje verktyg.



Figur 8. Kumulativa antalet erhållna lösenord i klartext för användarkonton (av 134 möjliga) för de testade verktygen. De tjocka linjerna är medelvärden och de färgade volymerna är 95% konfidensintervall för de tester som genomfördes för varje verktyg.



Figur 9. Kumulativa antalet Wazuh-larm för de testade verktygen. De tjocka linjerna är medelvärden och de färgade volymerna är 95% konfidensintervall för de tester som genomfördes för varje verktyg.