FOI MEMO

Projekt/Project
VECNO

Sidnr/Page no
1 (5)

Projektnummer/Project no
E716589

Uppdragsgivare/Client
FM

FoT-område
Operationer i cyberdomänen

Författare/Author
Hannes Holm

Datum/Date
2022-12-09

Memo nummer/Number
FOI Memo 8047

Omvärldsbevakning och statusmemo 2022

1 Arbete inom VECNO under 2022

Detta memo presenterar genomförd omvärldsbevakning av akademisk forskning kring maskininlärning, samt redogör kortfattat för det arbete som har genomförts inom projektet *Verktyg och Experiment för Computer Network Operations* (VECNO) under 2022. Resterande del av detta kapitel presenterar årets genomförda arbete. Kapitel 2 presenterar den genomförda omvärldsbevakningen.

Publicering av artikeln ”Lore A Red Team Emulation Tool” i tidskriften IEEE Transactions on Dependable and Secure Computing [1]. Denna tidskrift är en av de högst rankade akademiska tidskrifterna som finns för den typen av forskning som VECNO innefattar.

Omvärldsbevakning av akademisk forskning kring maskininlärning. Målet var att identifiera effektiva sätt att tillämpa förstärkt inlärning (eng: reinforcement learning, RL) på Lore för att öka korrektheten för verktygets bedömningar. En sammanfattande rapport är under arbete.

Omvärldsbevakning av verktyg som kan obfuskera bakdörrar. Målet var att identifiera verktyg som kan obfuskera de bakdörrar som används av Lore, och på det sätt möjliggöra kringgång av anti-virus och liknande detektorer som används under cybersäkerhetsövningar. Arbetet beskrivs vidare i en rapport[1].

Planering, utförande och uppföljande av cybersäkerhetsövningen Cybon. Cybersoldaternas slutövning under 2022 innefattade ett delmoment som handlade om att detektera hot skapade av Lore mot virtuella nätverk och datorer i CRATE. Under fem dagar i maj utsattes cirka 40 cybersoldater för angrepp automatiserade av Lore. VECNO stod för allt arbete kring Cybon, såsom planering och uppsättning av virtuella nätverk och maskiner, exekvering av angrepp och simulerad användartrafik, och uppföljning genom olika enkäter. Resultatet återges i mer detalj i ett kommande memo[2].

Tillämpning av Lore under cybersäkerhetsövningen SAFE Cyber 2022. Lore tillämpades under SAFE Cyber 2022 för automatisering av angrepp och jämfördes med utförande av angrepp manuellt definierade i SVED (dvs. ett ”vanligt” rött lag). Resultatet visade på att övningens deltagare inte kunde förstå om de utsattes för det röda laget eller för Lore och att kvaliteten för deras incidentrapporter inte berodde på vilken typ av hot de utsattes för. Resultatet ämnas skickas till den vetenskapliga konferensen IFIP SEC 2023.

Vidareutveckling av Lore. Det har, likt tidigare år av projektet, genomförts mycket ingenjörsarbete för att öka prestanda, korrekthet, modularitet och användarvänlighet för Lore.

Utveckling av automaträttningsfunktioner av incidentrapporter under cybersäkerhetsövningar. Det har arbetats fram en mjukvara som automatiskt rättar incident-rapporter skapade av säkerhetsanalytiker under cybersäkerhetsövningar. Denna mjukvara nyttjar loggarna producerade av verktyget SVED och systembeskrivningar i CRATE som underlag för bedömningar.

[1] FOI-R--5366--SE

[2] “Cyber Defence Exercise - Cybon 2022”, FOI Memo under arbete.

2 Kartläggning av forskningsfronten

Den metod som tillämpades för kartläggning av forskningsfronten under 2018 [2] och 2020 [3] nyttjades även för årets studie. I kort innefattar denna metod att utföra en systematisk sökning och analys av forskning i databasen Scopus, och komplettera resultaten med mer öppna sökningar i Google Scholar. Forskningsdatabasen Scopus genomsöktes den 26:e oktober 2022 med samma nyckelord som nyttjades under 2018 och 2020, fast för forskning publicerad från och med 2021.[3] Detta gav 937 träffar. Sammanfattningar och titlar för dessa artiklar granskades av en forskare för att bedöma vilka som är relevanta för VECNO. Tio av de 937 artiklarna bedömdes som relevanta och lästes i sin helhet. Sökningarna på Google Scholar identifierade ytterligare en artikel som bedömdes som relevant [4].

Dessa elva artiklar presenteras i resterande del av detta kapitel.

Filiol m.fl. [5] föreslår en metod för att automatisera identifiering och åtgärdande av sårbarheter i nätverksanslutna system. De testar metoden genom ett experiment med enklare skript som kedjar ihop resultatet från verktyg såsom Nmap och Metasploit för sex mjukvaruservrar. Ingen maskininlärning tillämpas.

Erdődi m.fl. [6] föreslår en metod för att automatiskt identifiera och utnyttja SQL-injektionssårbarheter. De modellerar processen som en Markov-beslutsprocess (eng: Markov Decision Process, MDP) och tränar upp modellen genom RL. De testar sin metod genom simulering av en fiktiv egenskapad webbsida som har ett indatafält vilket är sårbart för SQL-injektion. För att lära sig utnyttja denna sårbarhet delges RL-agenterna viss förkunskap om den simulerade webbsidans databasstruktur samt kunskap kring SQL-syntax i allmänhet.

Zang m.fl. [7] föreslår en metod som automatiskt konverterar data från utförda penetrationstester av nätverk till Bayesianska nätverk. Syftet är att extrahera semantiska relationer, såsom att koppla ihop sårbarheter och specifika mjukvaror. Författarna beskriver ej hur de rent konkret ämnar samla in data från penetrationstester, eller exakt hur semantiska relationer skall extraheras ur data.

Lyu m.fl. [8] beskriver en metod för att bedöma huruvida mjukvarusårbarheter kan utnyttjas. Författarna tillämpar djup inlärning (eng: deep learning) baserat på beskrivningsfälten för sårbarheter i kombination med metrikerna delgivna av CVSS (Common Vulnerability Scoring System). Författarna anser att en sårbarhet går att nyttja om det finns någon angreppskod i publika databaser såsom Exploit-DB eller SeeBug. De testar sin metod genom att prediktera om det finns träffar i angreppskodsdatabaserna för sårbarheter i databasen National Vulnerability Database (NVD). Deras resultat visar på f1-score's mellan 0.91 och 0.93, vilket kan tolkas som hög precision.

Duan m.fl. [9] presenterar ett ramverk för automatiska säkerhetsgranskningar av internetanslutna cyber-fysiska system (eng: Internet of Things, IoT). Ramverket kombinerar naturlig språkbehandling (eng: Natural Language Processing, NLP) och maskininlärning (genom Light Gradient Boosting Machine, LGBM) av beskrivningsfält i mjukvarusårbarheter för att prediktera deras signifikans. Resultatet nyttjas i kombination med data för operativa sårbarheter identifierade via automatiska sårbarhetstester som indata till en attackgraf. Slutligen tolkas attackgrafen i ett grafiskt gränssnitt av en operatör.

---

[3] (TITLE-ABS-KEY (security) AND (TITLE-ABS-KEY ("attack graph*" OR "attack path*" OR "penetration testing" OR pentesting OR ("game theory" AND "network security")) AND TITLE-ABS-KEY (learning OR optim* OR "markov decision process" OR "decision making" OR enumeration OR assessment) AND NOT TITLE-ABS-KEY ("intrusion detect*" OR iot OR "internet of things"))) OR (TITLE-ABS-KEY (("attack graph*" OR hack* OR penetrat* OR "red team" OR "security test*") AND (optimal OR learning OR plan* OR automat* OR emulat* OR "script*") AND ("network security" OR "cyber defense" OR "computer security" OR "cyber security" OR "cyber defence"))) OR (TITLE-ABS-KEY (("automat*" OR "script*" OR "simulat*") AND ("hack*" OR "penetrat*") AND "attack*")) AND (LIMIT-TO (PUBYEAR , 2022) OR LIMIT-TO (PUBYEAR , 2021)) AND (LIMIT-TO (SUBJAREA , "COMP"))

Saraswathi m.fl. [10] beskriver ett verktyg som kan nyttjas för att automatiskt utföra datainsamlingsfasen för ett penetrationstest av en webbsida. Verktyget kombinerar ett antal öppna-källkodsverktyg som återfinns i Linux-distributionen Kali Linux för att utföra denna uppgift. Verktyget bygger helt på enklare booleska uttryck och innefattar ingen inlärning eller dylikt.

Nguyen m.fl. [11] beskriver en RL-metod för att automatisera nätverkspenetrationstester. Denna metod optimeras genom RL-algoritmen A2C med fiktiva egenskaper om nätverksangrepp som indata. Författarna testar sin metod genom en mycket enkel simuleringsmodell för nätverksangrepp. Testerna visar på att den tränade modellen kan läras att välja rätt alternativ producerad av simuleringen.

Stan m.fl. [12] utvidgar attackgrafsverktyget MULVAL med ny funktionalitet för att ta hänsyn till cyber-fysiska system, fler kommunikationsprotokoll, designsårbarheter i nätverksprotokoll, samt för att modellera specifika styrsystemsarkitekturer. Författarna testar sitt verktyg genom att simulera fem typer av angrepp mot ett simulerat värmekraftverk. Resultat visar på olika sätt som dessa angreppstyper kan utföras.

Jeon och Kim [13] beskriver ett verktyg kallat AutoVAS som identifierar mjukvarusårbarheter genom statisk kodgranskning av C/C++ källkod. Författarna kombinerar metoder inom djup inlärning (i synnerhet metoden Word2Vec) och NLP för att konvertera källkod till word embeddings. Dessa word embeddings nyttjas tillsammans med information om sårbarheter beskrivet i databaserna Common Weakness Enumeration (CWS), Common Vulnerabilities and Exposures (CVE) och Common Attack Pattern Enumeration and Classification (CAPEC) för att identifiera sårbarheter i källkod. AutoVAS testades mot nio mjukvaror. Dessa tester identifierade sju kända sårbarheter och fyra nya sårbarheter.

Gavaudan m.fl. [14] förordar automatisering av cybersäkerhetsövningar som en metod för att möjliggöra cybersäkerhetsforskning. Författarna diskuterar tre olika typer av automation som behövs: automatisering av systeminstallation, systemkonfiguration och angrepp. Författarnas lösning tillämpar verktyget Terraform för systeminstallation, Ansible för systemkonfiguration, och CALDERA [15] för angrepp. Författarna jämför sin lösning mot att göra de tre stegen manuellt och finner att automatisering sparar mellan 5.5 och 26 timmars arbete.

Li m.fl. [4] implementerar RL ovanpå verktyget CALDERA. Författarna tränar upp två exempelmodeller, den första med en händelserymd på 10 möjliga val, och den andra med 16 möjliga val. Deep Q-Network (DQN) och klassisk Cross Entropy (CE) tillämpades för ändamålet. Båda exemplen involverade att välja rätt sekvens med val, där ett val kostade -1 poäng givet att det inte var målet för scenariot, och gav 99 poäng i annat fall. För det första exemplet krävdes att välja en sekvens på fyra rätta val; för det andra exemplet krävdes en sekvens på sex rätta val. Resultaten visade att det första träningsexemplet krävde 4600 aktioner fördelade över 45 träningsepisoder för att uppnå en hög belöning (96 poäng), medan det andra träningsexemplet krävde 25 000 aktioner för att uppnå en hög belöning (92 poäng). Det framgår inte hur många träningsepisoder som krävdes för det andra exemplet, men om det följer samma fördelning som det första exemplet torde det vara runt 250. Varje träningsepisod krävde mellan 20 och 30 minuter att utföra.

Sammanfattningsvis kan det konstateras att de allra flesta identifierade arbeten inte är särskilt lika det arbete som görs med Lore inom VECNO [1]. Det mest relevanta är [4], och detta arbete presenterar en mycket förenklad bild av cyberangrepp som inte passar särskilt väl för faktiska cybersäkerhetsövningar. De 10-16 valen i exemplen i [4] kan ställas i kontrast mot de ~100 000 val som Lore hade att välja emellan under slutet av SAFE Cyber 2020 [1].

3 Referenser

[1] H. Holm, ”Lore A Red Team Emulation Tool”, *IEEE Trans Dependable Secure Computing*, vol. Pre-print, 2022.

[2] H. Holm, ”Arbete utfört inom ÖvExCND under 2018 (FOI Memo 6541)”, nov. 2018.

[3] H. Holm, ”Arbete utfört inom ÖvExCND under 2020 (FOI Memo 7365)”, nov. 2020.

[4] L. Li, R. Fayad, och A. Taylor, ”CyGIL: A cyber gym for training autonomous agents over emulated network systems”, i *IJCAI-21 1st International Workshop on Adaptive Cyber Defense*, 2021.

[5] E. Filiol, F. Mercaldo, A. S.-P. C. Science, och undefined 2021, ”A method for automatic penetration testing and mitigation: A red hat approach”, *Procedia Computer Science*, vol. 192, s. 2039–2046, 2021.

[6] L. Erdődi, Å. Å. Sommervoll, och F. M. Zennaro, ”Simulating SQL injection vulnerability exploitation using Q-learning reinforcement learning agents”, *Journal of Information Security and Applications*, vol. 61, sep. 2021, doi: 10.1016/j.jisa.2021.102903.

[7] Y. Zang, T. Hu, T. Zhou, och W. Deng, ”An automated penetration semantic knowledge mining algorithm based on bayesian inference”, *Computers, Materials & Continua*, vol. 66, nr 3, 2021.

[8] J. Lyu, Y. Bai, Z. Xing, och X. Li, ”A Character-Level Convolutional Neural Network for Predicting Exploitability of Vulnerability”, i *2021 International Symposium on Theoretical Aspects of Software Engineering (TASE)*, 2021, s. 119–126.

[9] X. Duan, M. Ge, T. Le, F. Ullah, och S Gao, ”Automated Security Assessment for the Internet of Things”, *ieeexplore.ieee.org*, Åtkomstdatum: okt. 27, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9667743/

[10] V. Saraswathi och I. Ahmed, ”Automation of Recon Process for Ethical Hackers”, i *2022 International Conference for Advancement in Technology (ICONAT)*, jan. 2022.

[11] H. V. Nguyen, S. Teerakanok, A. Inomata, och T. Uehara, ”The Proposal of Double Agent Architecture using Actor-critic Algorithm for Penetration Testing”, i *Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021)*, 2021, s. 440–449.

[12] O. Stan *m.fl.*, ”Extending attack graphs to represent cyber-attacks in communication protocols and modern it networks”, *IEEE Trans Dependable Secure Comput*, vol. 19, nr 3, s. 1936–1954, maj 2022.

[13] J. Sanghoon och K. H. Kim, ”AutoVAS: An automated vulnerability analysis system with a deep learning approach”, *Computers & Security*, vol. 106, s. 1–24, 2021.

[14] L. Gavaudan, S. Legras, och V. Ventors, ”Cyber range automation, a bedrock for AI applications”, i *Proceedings of the 28th C&ESAR*, 2021.

[15] D. Miller, R. Alford, A. Applebaum, H. Foster, C. Little, och B. Strom, ”Automated adversary emulation: A case for planning and acting with unknowns”, 2018.