Enabling Visual Analytics via Alert-driven Attack Graphs

Nadeem, A.; Verwer, S.E.; Moskal, Stephen; Yang, Shanchieh Jay

DOI
10.1145/3460120.3485361

Publication date
2021

Document Version
Final published version

Published in
CCS 2021 - Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security

Enabling Visual Analytics via Alert-driven Attack Graphs

Azqa Nadeem*
Delft University of Technology
Delft, The Netherlands
azqa.nadeem@tudelft.nl

Sicco Verwer
Delft University of Technology
Delft, The Netherlands
s.e.verwer@tudelft.nl

Stephen Moskal
Rochester Institute of Technology
Rochester, United States
sfm5015@rit.edu

Shanchieh Jay Yang
Rochester Institute of Technology
Rochester, United States
jay.yang@rit.edu

ABSTRACT

Attack graphs (AG) are a popular area of research that display all the paths an attacker can exploit to penetrate a network. Existing techniques for AG generation rely heavily on expert input regarding vulnerabilities and network topology. In this work, we advocate the use of AGs that are built directly using the actions observed through intrusion alerts, without prior expert input. We have developed an unsupervised visual analytics system, called SAGE, to learn *alert-driven attack graphs*. We show how these AGs (i) enable forensic analysis of prior attacks, and (ii) enable proactive defense by providing relevant threat intelligence regarding attacker strategies. We believe that alert-driven AGs can play a key role in AI-enabled cyber threat intelligence as they open up new avenues for attacker strategy analysis whilst reducing analyst workload.

CCS CONCEPTS

• **Human-centered computing** → **Visualization**; • **Security and privacy** → **Intrusion detection systems**; • **Computing methodologies** → *Unsupervised learning*.

KEYWORDS

Attack graphs; Intrusion alerts; Finite state automaton;

ACM Reference Format:
Azqa Nadeem, Sicco Verwer, Stephen Moskal, and Shanchieh Jay Yang. 2021. Enabling Visual Analytics via Alert-driven Attack Graphs. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3460120.3485361

1 MOTIVATION & RELATED WORKS

Expert input is expensive. Attack graphs (AG) are visual models of attacker strategies. Existing approaches for AG generation rely mostly on vulnerability scanning and expert knowledge [2, 13, 14], which is costly and ineffective for many real-world operations. Meanwhile, Security Operations Centers (SOC) often possess an abundance of rapidly evolving cybersecurity observables, such as intrusion alerts, from prior security incidents, which contain valuable insights regarding attacker strategies [7].

Alert overload. SOC analysts receive millions of intrusion alerts on a daily basis, leading to alert fatigue [4]. While alert correlation techniques help reduce the volume of alerts [1, 15, 16], they do not show how the attack transpired. Thus, attacker strategy identification is still largely a manually process.

To the best of our knowledge, utilizing intrusion alerts for attack graph construction remains an open problem [5]. This is an important question because such AGs can discover and visualize hidden patterns in large amounts of alerts to augment human intelligence.

ML application is challenging. In recent years, machine learning (ML) has emerged as a promising solution for obtaining insights into attacker behavior [6, 9, 11, 12]. ML application requires that the following three challenges be addressed:

(1) Class imbalance between severe (*e.g.,* exploitation) and non-severe (*e.g.,* scanning) alerts presents a huge difficulty. Severe alerts are infrequent, while non-severe alerts reflect an important aspect of an attacker's strategy. A solution that keeps both type of alerts, while highlighting infrequent alerts is required. This is a tricky problem because most ML solutions discard infrequent events.
(2) The future and past of a given alert captures important contextual cues about the intent of an attacker. Thus, the proposed solution must model this context to distinguish between similar alerts that lead to different attacks.
(3) Black-box solutions that security analysts cannot understand are undesirable, thus calling for an explainable approach.

2 CONTRIBUTIONS

We have developed an unsupervised visual analytics system, called SAGE (Intru**S**ion alert-driven **A**ttack **G**raph **E**xtractor)[1]. Details about SAGE's workflow are given in [10]. Essentially, SAGE processes raw intrusion alerts into episode (*hyper-alert*) sequences. The temporal and probabilistic dependence between alerts is leveraged using a suffix-based probabilistic deterministic finite automaton (S-PDFA). We propose a suffix-based PDFA to accentuate infrequent severe alerts. Further, the model distinguishes between episodes with different contexts but identical signatures: if the future and past of two episodes are statistically different, then the S-PDFA

*Corresponding author.

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8454-4/21/11.
https://doi.org/10.1145/3460120.3485361

[1] SAGE is open-source: https://github.com/tudelft-cda-lab/SAGE.

considers them to be different states even if they have the same signature. Thus, an S-PDFA is an explainable and deterministic graphical model of all attack paths present in an alert dataset. We extract objective-oriented AGs from the S-PDFA on a per-victim, per-objective basis.

An *alert-driven* AG can be considered as an aggregated representation of relevant alerts, where each attack path originates from one of the starting (*i.e.,* yellow) vertices and leads to the root (*i.e.,* objective) vertex. Each attacker that obtains the objective is shown using a different edge color (along with the attacker IP next to the starting vertex) and multiple attack attempts are broken into individual attack paths. The context of an episode is denoted using the state identifier from the S-PDFA. Since low-severity episodes are too frequent, we remove their state identifiers to reduce the number of resulting vertices. This post-processing step further highlights the infrequent high-severity episodes and their varying contexts.

SAGE can directly augment existing IDSs and SIEMs for alert triaging and visual analytics. The alert-driven AGs are powerful because they not only enable forensic analysis of prior attacks (*i.e.,* displaying and comparing attack paths), but they also provide relevant threat intelligence about attacker strategies (*i.e.,* insights into behavioral dynamics, fingerprinting paths for attacker re-identification, and ranking attackers based on the uniqueness and severity of their actions). We believe that alert-driven AGs can play a key role in AI-enabled cyber threat intelligence as they open up new avenues for attacker strategy analysis whilst reducing analyst workload.

2.1 Extensions

We show several different use-cases for alert-driven AGs using intrusion alerts collected through security testing competitions. In addition to the experiments conducted on the Collegiate Penetration Testing Competition (*i.e.,* CPTC-2018 [8]) by Nadeem *et al.* [10], we apply SAGE on two additional alert datasets: one collected through a penetration testing competition (*i.e.,* CPTC-2017[2]) and the other through a blue team exercise (*i.e.,* CCDC-2018[3]). Table 1 shows the summary of the experimental datasets. For CCDC-2018, no additional information is known, which reinforces the claim that SAGE does not need any expert input to produce insightful AGs.

We also compare the quality of the S-PDFA against two alternative modeling approaches, *i.e.,* a *suffix tree* and *Markov chains*. Using Perplexity [3] to measure the predictive power of each model, the results suggest that the suffix tree is best at modeling the training data, which is to be expected since it is a tree representation of the input data. The S-PDFA is best at modeling unseen test data (and second best at modeling the training data). The Markov chains struggle to achieve optimal values. In addition, the AGs generated from each of the models show a different perspective: The Markov chain-AGs do not model the context and make vast over-generalizations, thus producing no added-benefit of the modeling step. The suffix tree-AGs and the S-PDFA-AGs are highly similar, except the S-PDFA-AGs are smaller due to the state merging algorithm. The real benefit of the S-PDFA becomes apparent in larger graphs: Similar paths are merged in an S-PDFA. Thus, repeated (sub-)strategies are displayed using already-existing vertices, whereas the suffix tree adds duplicate vertices, making the graph bigger. This analysis raises the question: when is learning (*i.e.,* making generalizations) a good idea, and when does simply showing raw data suffice?

Lastly, we validate the completeness of the AGs by matching them against the teams' self-reported claims. We found that most of the AGs supported at least one of the claims. In fact, the AGs

[2]https://globalcptc.org/
[3]https://www.nationalccdc.org/

Table 1: Summary of experimental datasets.

	CPTC-2018	CPTC-2017	CCDC-2018
# alerts	330,270	43,611	1,052,281
# teams	6	9	Unknown
Duration (hrs)	9	11	25
Victim hosts known?	Yes	No	No
Competition type	Pen. testing	Pen. testing	Blue teaming

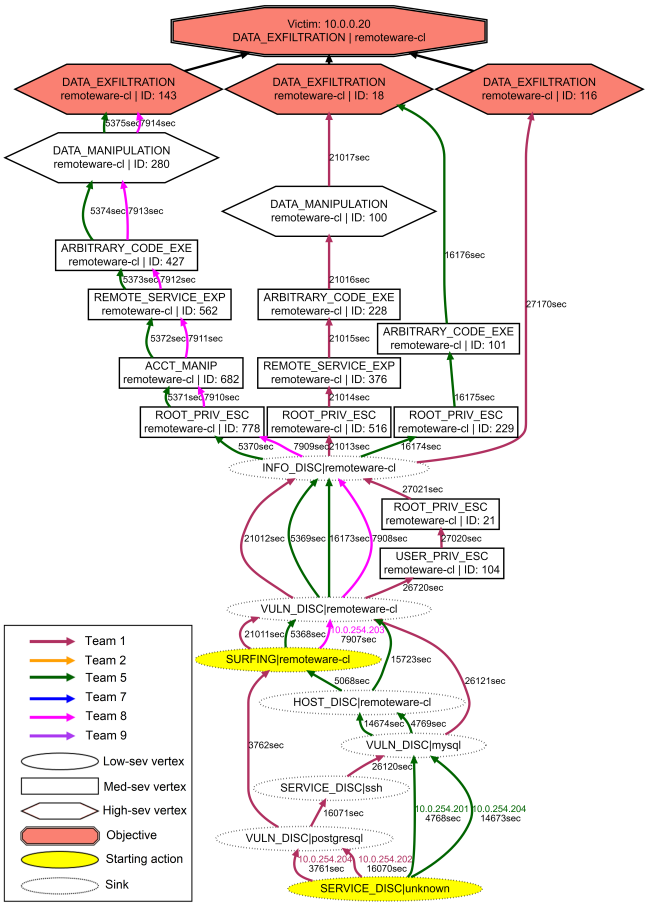

Figure 1: An alert-driven attack graph of data exfiltration over remoteware-cl (IDs are state identifiers, capturing context). Three attacker teams exploit it: Teams 1 and 5 exploit it twice, where subsequent attempts are shorter than the previous ones. There are three *ways of exploiting* the objective, based on the actions that lead up to it, as determined by the S-PDFA. Sinks are states that are too infrequent for the S-PDFA. Edge labels show time progression in seconds.

provide significantly more detail into attacker strategies than the steps described by the teams. Some claims did not have corresponding attack paths, which could indicate that those actions did not trigger any alerts. Further investigation is required to understand what causes such missing paths. We also conducted an informal user-study with two senior security researchers regarding the correctness and usability of the AGs, whose responses suggest SAGE is a promising technique for getting insights from intrusion data.

3 PRELIMINARY RESULTS

Alert triaging. SAGE compresses thousands of intrusion alerts into a handful of *alert-driven attack graphs*. For CPTC-2018, SAGE compresses over 330k alerts into 93 AGs; for CPTC-2017, SAGE compresses ~43k into 169 AGs; and for CCDC-2018, SAGE compresses ~1052k alerts into 139 AGs. Instead of investigating thousands of alerts, analysts can triage alerts based on a few AGs of interest.

Behavior dynamics. The AGs capture the strategies used by the participating teams, producing directly relevant insights for SOC analysts. Figure 1 shows an AG with five attack paths, conducted by three attacker teams. T1 and T5 conduct two attempts, where each subsequent attempt is shorter than the previous one. In fact, the AGs reveal that attackers follow shorter paths to re-exploit an objective in 84.5% of the cases. The S-PDFA is critical in identifying various ways of obtaining the same objective. In this case, it discovers that there are three ways to reach the objective, based on significant differences in the paths that lead up to it.

Strategy comparison. The AGs provide an intuitive layout to compare attacker strategies for discovering strategic similarities, scripted (simultaneous) attacks, and fingerprintable paths. For example, T5 and T8 share a significant portion of a strategy, as seen in Figure 1. In case of a scripted attack on multiple victims, their AGs appear identical, including the time progression information. A simple graph edit distance is enough to automate the detection of such identical AGs. Lastly, because some of the objectives are only exploited by a single attacker, these paths can uniquely fingerprint attackers. SAGE finds 29 such fingerprintable paths in CPTC-2018 that can be used for attacker re-identification.

4 ROADMAP OF THE NEXT STEPS

Learning from infrequent data is a difficult problem, which is further exacerbated by the unavailability of labeled data. SAGE leverages explainable sequence learning to compress thousands of alerts into a few objective-oriented attack graphs (AG). We lay the roadmap for what we believe are interesting next steps.

The completeness of the AGs cannot easily be determined due to the lack of ground truth. Further investigation is required to understand why a certain path could be missing, and whether this information can be used to improve faulty IDS signatures. Moreover, the current method for episode sequence construction does not show distributed attacks in the same AG. Although changing the granularity of the sequence construction is a simple fix, it produces considerably larger AGs. Thus, a trade-off is required between sequence granularity and AG size. Additionally, a more rigorous validation study is required to measure the correctness of the AGs and to understand which design decisions enable the analysts to reach correct conclusions. It is also currently unclear how to empirically measure interpretability and the usefulness of an AG. Further research is required into the design of a metric to measure AG quality. Finally, the impact of adversarial attacks on the resulting AGs is yet to be established.

Attack graph querying and prioritization is an important direction for future work since it will enable analysts to reach the most interesting attack paths quicker. Further, a big open question for SAGE is its handling of on-going attacks: If the AGs can be generated in real-time, evolving attacks can be monitored and highlighted. The AGs can potentially even be used to predict next attack steps, thus enabling proactive defense and dynamic risk assessment.

5 CONCLUSION

SAGE[4] is utilized to generate alert-driven attack graphs (AG) for two additional open-source alert datasets. We analyze the AGs produced using alternative modeling approaches to show the effectiveness of the S-PDFA. We demonstrate how these AGs can provide insights into past attacks and intelligence for future attacks. Finally, we lay the roadmap for further research into alert-driven AGs.

REFERENCES

[1] Faeiz M Alserhani. 2016. Alert correlation and aggregation techniques for reduction of security alerts and detection of multistage attack. *International Journal of Advanced Studies in Computers, Science and Engineering* (2016).

[2] Michael Lyle Artz. 2002. *Netspa: A network security planning architecture.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[3] Thomas Cover and Joy Thomas. 1991. *Elements of information theory.* John Wiley & Sons.

[4] Wajih Ul Hassan, Shengjian Guo, Ding Li, Zhengzhang Chen, Kangkook Jee, Zhichun Li, and Adam Bates. 2019. Nodoze: Combatting threat alert fatigue with automated provenance triage. In *NDSS*.

[5] Kerem Kaynar. 2016. A taxonomy for attack graph generation and usage in network security. *Journal of Information Security and Applications* (2016).

[6] Qin Lin, Sridha Adepu, Sicco Verwer, and Aditya Mathur. 2018. TABOR: A graphical model-based approach for anomaly detection in industrial control systems. In *Asia-CCS*.

[7] Stephen Moskal and Shanchieh Jay Yang. 2020. Framework to Describe Intentions of a Cyber Attack Action. *arXiv preprint arXiv:2002.07838* (2020).

[8] Nuthan Munaiah, Akond Rahman, Justin Pelletier, Laurie Williams, and Andrew Meneely. 2019. Characterizing Attacker Behavior in a Cybersecurity Penetration Testing Competition. In *ESEM*. IEEE.

[9] Azqa Nadeem, Christian Hammerschmidt, Carlos H Gañán, and Sicco Verwer. 2021. Beyond Labeling: Using Clustering to Build Network Behavioral Profiles of Malware Families. *Malware Analysis Using Artificial Intelligence and Deep Learning* (2021).

[10] Azqa Nadeem, Sicco Verwer, and Shanchieh Jay Yang. 2021. SAGE: Intrusion Alert-driven Attack Graph Extractor. In *2021 IEEE Symposium on Visualization for Cyber Security (VizSec)*. IEEE.

[11] Julio Navarro, Véronique Legrand, Aline Deruyver, and Pierre Parrend. 2018. OMMA: open architecture for Operator-guided Monitoring of Multi-step Attacks. *EURASIP Journal on Information Security* (2018).

[12] Julio Navarro, Véronique Legrand, Sofiane Lagraa, Jérôme François, Abdelkader Lahmadi, Giulia De Santis, Olivier Festor, Nadira Lammari, Fayçal Hamdi, Aline Deruyver, et al. 2017. Huma: A multi-layer framework for threat analysis in a heterogeneous log environment. In *FPS*. Springer.

[13] Steven Noel, Matthew Elder, Sushil Jajodia, Pramod Kalapa, Scott O'Hare, and Kenneth Prole. 2009. Advances in topological vulnerability analysis. In *CATCH*.

[14] Xinming Ou, Sudhakar Govindavajhala, and Andrew W Appel. 2005. MulVAL: A Logic-based Network Security Analyzer. In *USENIX security symposium*.

[15] Reza Sadoddin and Ali Ghorbani. 2006. Alert correlation survey: framework and techniques. In *International conference on privacy, security and trust: bridge the gap between PST technologies and business services*.

[16] Saeed Salah, Gabriel Maciá-Fernández, and JesúS E DíAz-Verdejo. 2013. A model-based survey of alert correlation techniques. *Computer Networks* (2013).

[4] https://github.com/tudelft-cda-lab/SAGE