Enhancing Network Intrusion Detection Robustness via Dataset Augmentation: A CIC-IDS-2017 Case Study

Jozef Jankaj
Faculty of Sciences
Ghent University
Ghent, Belgium
jozef.jankaj@ugent.be

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

I. Introduction

With the advent of the internet, physical threats to our security have moved to the virtual world. Our privacy is continually threatened as we give more and more of it up for the convenience of the interconnected world. Plenty of adversaries are more than happy to attack, steal and destroy our systems. To address this Sword of Damocles, network security defenses has been a hot topic of research. While significant improvements in network security have been made thanks to the improvements in encryption, implementation of firewalls and general security practices, attacks against the networks are still prevalent. Timely detection and interruption of these attacks is paramount for maintaining the confidentiality, integrity and availability of the data. A Network Intrusion Detection System (NIDS) fulfills this purpose by monitoring the traffic and raising an alarm upon detection of malicious traffic. Many such existing systems [**TODO: cite here**] are implemented using preprogrammed rules or features of traffic.

However, the fast-paced nature of cybersecurity renders these systems quickly ineffective, as new attacks and variants of existing attacks get discovered and deployed. For this and other reasons, researchers turned their attention to Machine Learning based Network Intrusion Detection Systems (ML-NIDS). Both classical, shallow-learning methods [**TODO: cite here**] and deep learning methods [cite here] have been explored in the literature. These methods rely heavily on high-quality data, and naturally, many datasets [**TODO: cite here**] have been constructed and proposed as the benchmark for the ML-NIDS research.

Unfortunately, these datasets have been shown to contain mistakes [**TODO: cite here**], reducing their usefulness. Issues such as poor documentation, misimplementations of attacks and faulty labeling have been raised and as a result, the NIDS practitioners are reserved about implementing these models.

To address these and other issues, Verkerken et al. have developed the ConCap framework [**TODO: cite here**], leveragin the power of Kubernetes clusters to build out networks in-silico, conduct attacks and capture traffic for dataset creation. In this article, we have looked at using this technology for enhancing robustness of existing ML-NIDS datasets through dataset augmentation and take the CIC-IDS-2017 [**TODO: cite here**] as a case study.

We conduct this as follows: In Section II, we describe our process of reconstructing the CIC-IDS-2017 dataset into ConCap scenarios and choices we make along the way. In Section III, we experiment with the dataset to show that our reconstruction is sound and useful. In Section IV, we study [**TODO: find a better word here**] the robustness improvements by augmenting the CIC-IDS-2017 dataset with extra traffic from ConCap and testing it on the CSE-CIC-2018 [**TODO: cite here**].

II. Dataset reconstruction

The CIC-IDS-2017 dataset consists of traffic captures over the course of a workweek in July 2017, during which a test network is hit with numerous attack classes: Bruteforce, Denial-of-Service, Heartbleed, Web Attacks, Infiltration, Botnet, Distributed Denial-of-Service and Portscan.

We reconstruct the attacks in the dataset by collecting and writing Docker images for the various attackers and targets. By including these images as part of various ConCap scenarios, we can construct the dataset of PCAP files representing the attack.

A. Monday

No attacks take place on Monday, only background, benign traffic generated by the B-Profile [**TODO: cite here**]. This benign traffic is present on all days and serves as a baseline. We do not specifically reconstruct this traffic, but rather reuse the traffic from this day in our experiments

B. Tuesday

FTP and SSH bruteforce attacks take place on Tuesday. These attacks are implemented using Patator [**TODO: cite**

here]. It is unclear what wordlists the authors use to conduct these attacks, we therefore opt to use lists from the SecLists [**TODO: cite here**] repository.

C. Wednesday

Denial-of-Service and Heartbleed attacks take place on Wednesday. Authors perform attacks with Slowhttptest [**TODO: cite here**], Slowloris [**TODO: cite here**], GoldenEye [**TODO: cite here**] and Hulk [**TODO: cite here**] tools, each attacking a different part of the HTTP protocol. We reconstruct these attacks with the respective tools, reusing Slowhttptest to implement both Slowhttptest and SlowLoris attacks, as this tool is capable of both.

Heartbleed bug is also exploited in addition to the Denial-of-Service attacks. We follow the authors in using the Heartleech [**TODO: cite here**] tool to conduct the attack against a vulnerable OpenSSH server version 1.0.1f.

D. Thursday

Web Attacks take place on Thursday, consisting of three subclasses: Bruteforce, SQL-Injection and Cross-Site Scripting. Each of these is conducted against a part of the Damn Vulnerable Web App [**TODO: cite here**] using Selenium framework. As we do not have access to the original code, we write own Python scripts to conduct these attacks.

Infiltration attack also takes place on this day, by having the victim run a malicious executable and create a reverse Meterpreter shell to the attacker, after which a Portscan is executed. We opt not to reconstruct this attack for two reasons: First, a lack of support for multi-stage attack execution by ConCap limits our ability to reproduce this attack faithfully. Second, there is little interesting traffic happening on the network: it is the specific file that makes a file transfer malicious, not the act of transferring a file.

E. Friday

On Friday, the remaining attack classes are executed. A Botnet attack is launched using the Ares [**TODO: cite here**] botnet. We do not include this attack in our reconstruction due to technical limitations. Ares does not allow for controlling the botnet from Linux hosts, limiting our ability to conduct an attack from inside Docker containers.

Next, Distributed Denial-of-Service attack is performed using LOIC [**TODO: cite here**]. Though LOIC is primarily controlled through a GUI, the author does provide a way to control it remotely using an IRC server. We use the multi-target functionality of ConCap to launch both the target and an IRC server. A bot connects to this server and waits for LOIC to make the connection, before directing it to perform the attack using the channel topic.

Finally, a Portscan is executed. After traffic analysis, we crafted a container with open ports 21, 22, 80, 139 and 445, pointing to respectively an FTP server, SSH server, HTTP server and an SMB service. The portscan attack is executed against this container. Due to difficulties with setting up Samba, we have decided to conduct SMB portscans separately, merging the traffic with the other portscans during preprocessing.

III. Experimental verification

We first want to verify that the traffic generated by ConCap has similar features to the physical traffic in CIC-IDS-2017. To this end, we preprocess the generated PCAP files into NetFlows using the CICFlowmeter [**TODO: cite here**]. CICFlowmeter will extract more than eighty features from the traffic, allowing us to train ML models on it. We employ two-way methodology in our verification: Train a model on ConCap traffic and measure its performance on CIC-IDS-2017 traffic, and vice-versa.

To do this, we use a Decision Tree with a single root node, single feature to make the decision on and a Gini criterion. After preprocessing the datasets, we add equal amount of benign NetFlows to the attacks NetFlows to achieve class balance. We take these benign NetFlows from CIC-IDS-2017 Monday. Afterwards, we train the model on each feature and measure its accuracy, precision and recall, as well as ROC AUC score.

We are looking for a set of features that show high predictive power in both ways. While preferable, we do not expect the set of best performing features to be the same in both ways, due to the underlying networks being fundamentally different: the ConCap network is a theoretically perfect network implemented fully in-silico. Though ConCap does provide a way to configure network artifacts (packet drops, corruption, delays, reordering...), we lack this information about the CIC-IDS-2017 network, making it impossible to reproduce.

These experiments show the presence of these common features with high predictive power as measured by the ROC AUC score. Naturally, different attacks require different features for detection. Below we report the top three most performant features, along with their average ROC AUC Scores.

A. Tuesday

For the Bruteforce attacks, we find Bwd RST Flags (0.977596), Average Packet Size (0.977235) and Packet Length Mean (0.977235) to be most predictive features for FTP bruteforcing. This is not surprising, as we expect a high amount of connections being opened and closed as different username-password combinations are tried. For SSH bruteforce attacks, we find Fwd Seg Size Min(0.921095), Fwd IAT Min (0.852054) and SYN Flag Count (0.810621) to be most predictive.

[**TODO: include graphs**]

B. Wednesday

For the SlowLoris attack, we find Total TCP Flow Time (0.910514), Bwd Packet Length Max (0.866311) and Total Length of Bwd Packet (0.866311) to be the most powerful predictors. For Slowhttptest, Total TCP Flow Time (0.901099), Fwd IAT Min (0.885101) and Fwd IAT Total (0.842183) to be most predictive. This result is not surprising, as these two

attacks focus on exhausting the server resources, increasing the time between packets coming from the server.

For Hulk attack, we find Bwd Packet Length Std (0.977901), Fwd RST Flags (0.955192) and Subflow Bwd Bytes (0.951457). Similarly, GoldenEye attacks are best predicted using Bwd Packet Length Std (0.939451), Packet Length Std (0.928317) and Packet Length Variance (0.928317). As GoldenEye exploits HTTP headers `Connection` and `Cache-Control` and HULK floods the server with UDP packets, servers memory is the resource that gets exhausted, rather than connection pool. This supports our findings of features that focus on size rather than timing.

For the Heartbleed attacks, we find Bwd Packet Length Std (1.0), Flow Bytes/s (0.8636365) and Packet Length Std (0.905702) to be the best predictors. We do note that due to the low amount of samples from CIC-IDS-2017, this model may suffer from heavy overfitting and therefore these features may not be the most representative.

C. Thursday

For the Bruteforce web attack, we find Bwd Init Win Bytes (0.969178), Fwd Seg Size Min (0.914384) and Fwd IAT Min (0.867808). Similarly, we find Fwd Seg Size Min (0.961538), Bwd IAT Min (0.912873) and Bwd Packet Length Std (0.843799) for SQL Injection attacks and Fwd Seg Size Min (0.930556), Flow IAT Std (0.847222) and Fwd Packets/s (0.819444) for Cross-Site Scripting attacks.

Overall, Fwd Seg Size Min seems to be the best predictor across these attacks, but once again we note the low amount of samples present in the dataset, potentially leading to classification issues.

D. Friday

For LOIC DDoS attack, we find Fwd Seg Size Min (0.934932) and FIN Flag Count (0.821918) to hold the most predictive power. Finally for portscan, we find Fwd Packet Length Max (0.946259), Fwd Packet Length Mean (0.944625) and Fwd Segment Size Avg (0.944625) in the top three predictive features.

E. Discussion

Overall, we find plenty of features that perform well for both ways of training, regularly scoring above .9 on the ROC AUC score and none going below .8. This shows that the ConCap framework can be used to replicate datasets and potentially extend them, as the models based on either can reliably predict the other.

IV. Augmentation

After verifying ConCap's usefulness as a stand-in for the CIC-IDS-2017 dataset, we

V. Results

- describe the results, show some nice graphs

VI. Conclusion

- conclude that it's all good, no worries, please give me my degree

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.