Robust Self-Protection Against Application-Layer (D)DoS Attacks in SDN Environment

Chafika Benzaïd*, Mohammed Boukhalfa* and Tarik Taleb*†
* Aalto University, Espoo, Finland
†University of Oulu, Oulu, Finland
Email: firstname.lastname@aalto.fi

***Abstract*—The expected high bandwidth of 5G and the envisioned massive number of connected devices will open the door to increased and sophisticated attacks, such as application-layer DDoS attacks. Application-layer DDoS attacks are complex to detect and mitigate due to their stealthy nature and their ability to mimic genuine behavior. In this work, we propose a robust application-layer DDoS self-protection framework that empowers a fully autonomous detection and mitigation of the application-layer DDoS attacks leveraging on Deep Learning (DL) and SDN enablers. The DL models have been proven vulnerable to adversarial attacks, which aim to fool the DL model into taking wrong decisions. To overcome this issue, we build a DL-based application-layer DDoS detection model that is robust to adversarial examples. The performance results show the effectiveness of the proposed framework in protecting against application-layer DDoS attacks even in the presence of adversarial attacks.**

I. Introduction

The disruptive capabilities of 5G and beyond networks are envisioned to enable an extensive range of new applications and services, such as tactile Internet, virtual/augmented reality, autonomous driving and industrial automation. Meanwhile, they are expected to open the door to increased and sophisticated cybersecurity threats. One major security concern that may hamper the potential of the anticipated applications/services is the compromise of their availability. According to Gartner [1], Distributed Denial of Service (DDoS) attacks continue to escalate in frequency, volume and complexity, leading to availability breach. In fact, Kaspersky's DDoS Q2 2019 report states that the number of attacks in Q2 has risen by 18% compared with the same period of 2018. Considering the massive number of connected devices and the high bandwidth that will feature 5G and beyond networks, their number will even continue to soar.

DDoS attacks can be broadly classified into two types, namely [2]: (i) *network-layer DDoS* attacks, which aim at saturating the network bandwidth by generating volumetric traffic or high-rated packets , and (ii) *application-layer DDoS* attacks, which focus on exhausting the server's computational and memory resources. Application-layer DDoS attacks are usually stealthy in nature trying to mimic genuine behavior with low-bandwidth usage, making their detection and mitigation harder. The complexity of handling application-layer attacks is a key driver of the significant growth in their number these last years. Indeed, Kaspersky's DDoS Q2 2019 report reveals that their amount has increased by 32% compared with Q2 2018 and their share in Q2 2019 escalated by 46%. Although extensive work has been engaged and several solutions have been proposed (e.g. [2]–[4]), addressing the application-layer DDoS issue is far from being completely resolved.

The recent years have seen a trend toward the development of Machine Learning (ML) driven Software-Defined Security solutions [5]–[7] that are able to empower key security functions (e.g., prediction, detection, mitigation, etc.). Indeed, ML fosters security self-managing functionalities, resulting in improved robustness and lower operational costs [8]. While the recent contributions (e.g., [2], [9]–[11]) have focused on the opportunities of ML techniques to empower intelligent detection of DDoS attacks, they have overlooked their security concerns. In fact, the use of ML techniques is a source for new attack vectors. It has been proven that ML techniques are vulnerable to several attacks [12] targeting both training phase (i.e., poisoning attacks) and test phase (i.e., evasion attacks). The attacks against ML techniques aim to fool the ML model into taking wrong decisions (e.g., misclassifying a malicious traffic as a legitimate traffic) by introducing carefully crafted perturbations to training and/or test samples. Such perturbations are called adversarial examples. Thus, the adoption of ML techniques for security management operations in next-generation networks could be waned if their security issues are not addressed [13].

To fill the aforementioned gap, this work proposes a robust application-layer DDoS self-protection framework. The framework empowers a fully autonomous detection and mitigation of the application-layer DDoS attacks, leveraging Deep Learning (DL) and Software Defined Networking (SDN) enablers. A key contribution of this work is to build a DL-based application-layer DDoS detection model that is robust to adversarial examples.

The rest of this paper is organized as follows. Section II summarizes related work in the literature. In Section III, the proposed application-layer DDoS defense framework is described. Section IV presents the performance evaluation results. Finally, Section V concludes the paper.

II. Related work

Many research efforts have been devoted to tackle DDoS attacks leveraging ML and/or SDN [2]–[4], [14]. In what follows, we will review the main defense mechanisms proposed in the literature to handle DDoS attacks exploiting the intelligence of ML techniques and/or the flexibility provided by network programmability (i.e., SDN).

Braga *et al.* [15] proposed an intelligent method for detecting network-layer DDoS attacks in an SDN environment. The proposed method uses a Self Organizing Maps (SOM) [16] model, an unsupervised artificial neural network, trained on traffic flow features. The contribution in [9] rely on Deep Neural Network (DNN) models to detect intrusion in an SDN network. The authors in [10] devised a ML-based collaborative DDoS mitigation strategy in a multi-SDN controller environment. The detection is performed using Naive Bayes classifier based on flow features extracted by the SDN controller. Upon detection of malicious behavior, the SDN controller in the attacker's network is automatically notified to create a deny IP based flow. Similar to [15], the work in [9], [10] consider only network-layer attacks. Moreover, the proposed models are trained on NSL-KDD, a relatively old dataset that cannot reflect the current trend in network attacks.

Hong *et al.* [17] devised a SDN-assisted defense method to detect and mitigate slow HTTP DDoS attacks. The defense solution is deployed as a SDN application and triggered by the web server when the number of open connections that sent incomplete HTTP requests exceeds a given threshold. The major weakness of threshold-based schemes is their lack of accuracy. In fact, threshold-based schemes are unsuitable for detecting application-layer DDoS attacks due to the resemblance between the traffic patterns generated by those attacks and benign activities. The authors in [11] demonstrated the potential of ML techniques in detecting low-rate application-layer DDoS using the characteristics of malicious TCP flows. A detection accuracy of over 97% has been achieved using K Nearest Neighbor, Decision Trees and DNN techniques.

While the aforementioned contributions have focused on the opportunities of ML techniques to empower intelligent detection of DDoS attacks, they have overlooked their security concerns. In fact, it has been demonstrated that ML techniques, even the emerging ones (e.g., Reinforcement Learning (RL)), are prone to several attacks targeting both training phase (i.e., poisoning attacks) and test phase (i.e., evasion attacks) [12], [13], [18]. In poisoning attacks, an attacker aims at tampering the training data, by injecting carefully crafted malicious samples, to impact the learning outcome. Meanwhile, an evasion attack focuses on bypassing the learned model by introducing small perturbations to the test samples. Such perturbations are called adversarial examples. Adversarial Machine Learning (AML) [19] is an emerging research discipline that focuses on making ML techniques resilient to adversarial attacks by assessing their vulnerability to attacks and devising appropriate countermeasures. Unlike computer vision field, very few contributions (e.g., [20]) have been targeted at ML security in the context of networking field. The work in [20] investigates the resilience of RL to different forms of poisoning attacks in the context of autonomous cyber-defense in SDNs. While the authors have briefly discussed the potential countermeasures, they did not implement any defense strategy.

To deal with application-layer DDoS attacks in presence of adversarial attacks and in a fully autonomous way, we propose a robust application-layer DDoS self-protection framework leveraging both ML and SDN enablers.

III. Robust Application-Layer DDoS Self-Protection Framework

A. *Framework's High-Level Architecture*

Fig. 1 depicts the basic architectural components of the proposed solution to mitigate the application-layer DDoS attacks in a fully autonomous way. The "App-Layer DDoS Protection" component is in charge of detecting the malicious activity and issuing the security policy in case the attack is detected. It consists of four main modules: the "Network Flow Collector", the "Features Extractor" and the "Detector". The Network Flow Collector permanently collects network flows via port mirroring. To limit the impact of mirroring on the network performance, only traffic flowing from/to the monitored asset (e.g., Web server) is mirrored. The collected traffic is periodically exported to Features Extractor to retrieve flow's features relevant to application-layer DDoS attack detection. Once extracted, the flow features are passed to the Detector for uncovering suspicious behavior. If a malicious traffic pattern is identified, the Detector issues a security policy (e.g., flow dropping or steering) to the Security Policy Manager. Upon receiving the security policy, the Security Policy Manager converts the policy into a flow command and sends it to the SDN controller. Based on the received flow command, a flow rule is pushed by the SDN controller to the corresponding virtual Switch (vSwitch) to fulfill the defined security policy.

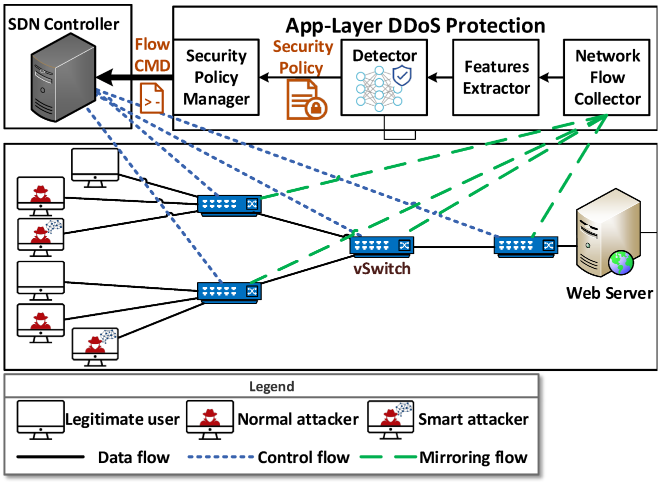


Fig. 1. The Robust App-Layer DDOS Self-Protection Framework's High-Level Architecture.

B. *Attacker Model*

The attacker's goal is to exhaust the server's resources (e.g., CPU, memory, I/O), preventing the server from providing services to legitimate users. To this end, we assume that the attacker has the capability to launch application-layer DDoS attacks, particularly HTTP-based flooding attack that aims to overwhelm the server by a voluminous number of legitimate HTTP requests. The HTTP-based flooding attack can be performed either in high-rate or slow-rate mode. In high-rate mode, the attacker mimics a flash-crowd event by flooding the web server with a large number of legitimate HTTP requests in a short period of time. The low-rate mode, however, consists in establishing multiple HTTP connections with the web server by sending partial HTTP requests at a very slow rate.

We consider that an attacker may be *smart*; that is, he/she has the capability to launch an application-layer DDoS attack while evading detection by a ML-based detector. A smart attacker is supposed to be able to craft application-layer (D)DoS flow that will be misclassified as a legitimate flow by the ML-based model. In this work, we assume that a smart attacker has a full-knowledge (i.e., white-box attack) on the targeted ML model, including its architecture and parameters. The white-box attack Fast Gradient Sign Method (FGSM) [21] is considered for the purpose of this work. The FGSM attack generates adversarial examples by performing a one step gradient update in the direction of the gradient's sign of the loss function relative to the input. The input is then altered by adding a perturbation that can be expressed as:

$$\eta = \epsilon.sign(\bigtriangledown_x J(\theta, x, y)) \quad (1)$$

where x is a sample (i.e., network flow), y is the label of x (i.e., Benign or DDoS flow), $J(\theta, x, y)$ is the loss function used to generate the adversarial example and ϵ denotes the perturbation magnitude.

C. Detector Module

To identify the application-layer DDoS attacks, Deep Learning (DL) is leveraged in this study. DL approaches have recently gained momentum for solving several problems in networking ranging from resource allocation to network security [22]. The adoption of a DL model is motivated by its capacity of uncovering complex non-linear relationships between inputs and outputs, yielding higher accuracy in distinguishing application-layer DDoS flow patterns from legitimate flow patterns.

1) Model structure: The detection model is built using Multi-Layer Perceptron (MLP) algorithm. The proposed model consists of 1 input layer, 2 hidden layers with 64 neurons each, and a two-class softmax output layer. The model's input is the flow features received from the Extractor. The model's output is the traffic class; that is, DDoS traffic or legitimate traffic.

2) Adversarial Training: The detector module is resilient to adversarial attacks performed by the smart attackers. To counteract the white-box attacks, we adopt the adversarial training defense. In adversarial training, the DL model is explicitly trained on adversarial examples in order to learn how to resist them. Considering the FGSM attack, the adversarial training is performed based on adversarial examples generated using FGSM.

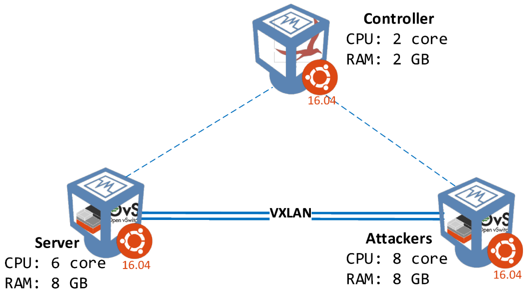


Fig. 2. Testbed Architecture.

IV. Performance Evaluation

Fig 2 illustrates the setup environment used to evaluate the performance of the proposed framework. We have used three (03) VirtualBox VMs hosted on the same physical server and running Ubuntu 16.04 operating system. The first VM (Controller), with a minimal configuration of 2 CPU cores and 2GB RAM, acts as an SDN controller, where an instance of ONOS [1] (Open Network Operating System) is deployed. The second VM (Attackers) is used to simulate the attackers' network. Eight (8) attackers, deployed on 8 LXD containers, perform simultaneous App-layer DDoS attack against the web server. The high-rate HTTP-based flooding attack is launched using Hulk [2] tool. The hulk attack generates a high volume of unique and obfuscated HTTP GET requests. The low-rate HTTP-based flooding attack is launched using Slowloris [3] tool. The slowloris attack allows to make the web server inaccessible by holding multiple connections open for a long time. The attackers can generate either ordinary malicious traffic or adversarial malicious traffic. The Cleverhans [4] library is used to craft adversarial application-layer DDoS flows based on FGSM attack using a perturbation magnitude $\epsilon = 0.5$. In fact, the smart attacker's aim is to generate application-layer DDoS traffic on which slight perturbations are introduced by the FGSM attack in order to be recognized as legitimate traffic, resulting in detection evasion. To make the DL-based model robust against adversarial examples, the model has been adversarially trained on adversarial flows generated by FGSM attack using the same perturbation magnitude (i.e., $\epsilon = 0.5$). The third VM (Server) contains the Apache web server with its default configuration and the App-Layer DDoS Protection component, each of them deployed on an LXD container. This VM is connected to the Attackers VM through Open vSwitch (OVS) using a Virtual Extensible LAN (VXLAN) tunnel as illustrated in Fig. 3. The OVSs are controlled by the SDN controller.

The MLP-based model integrated in the Detector module is trained on the recent intrusion detection dataset, CI-

[1] https://onosproject.org
[2] https://github.com/grafov/hulk
[3] https://github.com/gkbrk/slowloris
[4] https://github.com/tensorflow/cleverhans

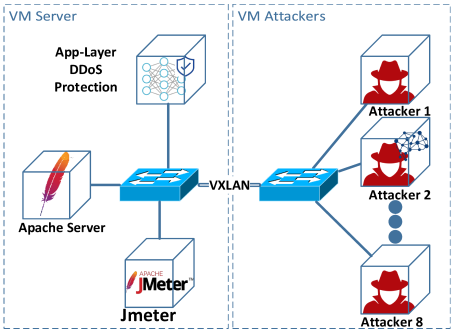


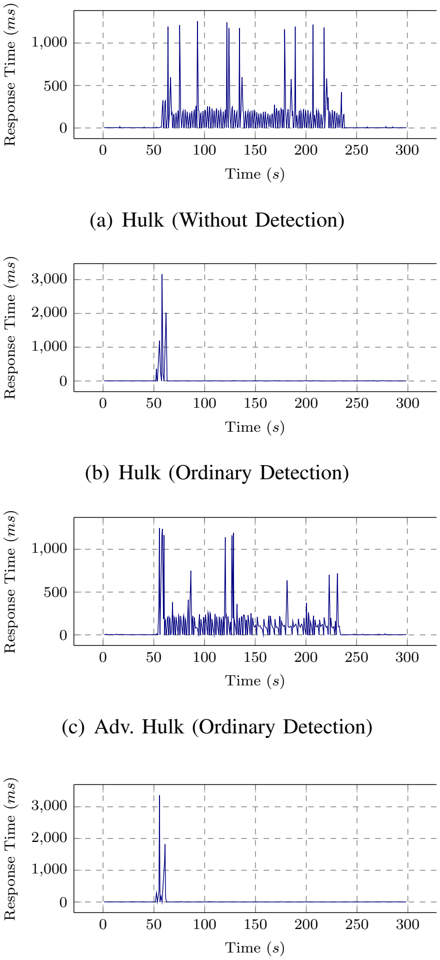
Fig. 3. Testbed's Network Topology.

CIDS2017 [23] (where only network flows corresponding to legitimate traffic and DoS/DDoS attacks are used) augmented by application-layer DDoS flows generated in our testbed. The dataset used in this study is available on [24]. 70% of the datasets flows are used to train the model and the remaining 30% flows are used as a test set to assess the models performance on unseen data. The model is trained for 10 epochs with a batch size of 128, Adam as an optimizer, and a learning rate of 0.001. The model is implemented using the Python's DL library Keras running on a TensorFlow backend. It achieved an accuracy of 99.65% on the test set.

The framework performance is assessed in terms of web server's response time and system load. The response time is defined as the time elapsed between when the request is sent and when the corresponding response is fully received. The system load is measured in terms of CPU and RAM usage. The performances are measured before ($1min$), during ($3min$) and after ($1min$) the App-layer DDoS attack is launched. To this end, Apache JMeter is used to simulate a legitimate client sending an HTTP request every one second. Four scenarios are considered, namely: (1) Normal attack without the proposed defense system - this scenario is used as a baseline to evaluate the effectiveness of the DL-based detection module; (2) Normal attack while using the original DL-based model (i.e., without adversarial training) for detection; (3) Adversarial attack while using the original DL-based detection model; and (4) Adversarial attack while using the adversarially-trained DL-based detection model.

A. Web Server's Response Time

Fig. 4 and Fig. 5 show the web server's response time when Hulk and Slowloris attacks are launched, respectively. The server's average response time is $3.5ms$ without attack. The depicted results show a high increase in the response time during the attack phase. A key observation is that Slowloris attack exhibits a significant impact on the response time ($279.4s$ in average) compared to Hulk attack ($2.6s$ in average). The proposed defense framework using the original DL-based model succeeds in detecting and blocking normal attacks in a very short time, allowing the web server's response time to go to normal. Indeed, the attack is counteracted in roughly $10s$ in the case of Hulk (See Fig. 4(b)) and $25s$ in the case of Slowloris (See Fig. 5(b)). Nevertheless, the original DL-based model fails in detecting the adversarially-generated attacks (See Fig. 4(c) for Hulk and Fig. 5(c) for Slowloris). In fact, the original DL-based model's accuracy dropped considerably in presence of adversarially-crafted App-layer DDoS flows, which are misclassified as legitimate flows. Unlike the original DL-based model, the use of its adversarially-trained variant allowed to prevent both ordinary and adversarially-generated attacks (See Fig. 4(d) for Hulk and Fig. 5(d) for Slowloris). Thus, adversarial training has significantly improved the model's robustness against adversarial flows. It is worth mentioning that while the adversarial Hulk attack is stopped in $10s$, the adversarial Slowloris is thwarted in $45s$, which is about 2 times longer than the mitigation of the ordinary Slowloris attack by the original DL-based model. This can be explained by the fact that the patterns of adversarial Slowloris flows will become much closer to legitimate patterns, making their detection more complex.



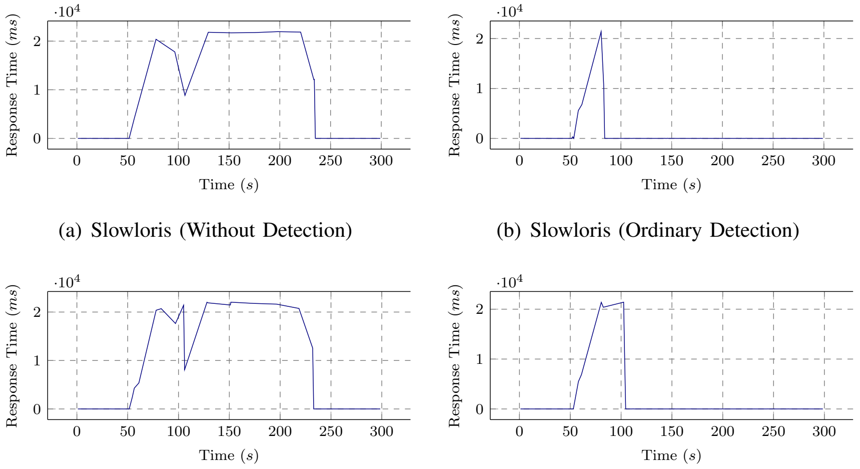
(a) Hulk (Without Detection)

(b) Hulk (Ordinary Detection)

(c) Adv. Hulk (Ordinary Detection)

(d) Adv. Hulk (Adv. Detection)

Fig. 4. The Web server's response time over the time in the case of Hulk attack



(a) Slowloris (Without Detection)

(b) Slowloris (Ordinary Detection)

(c) Adv. Slowloris (Ordinary Detection)

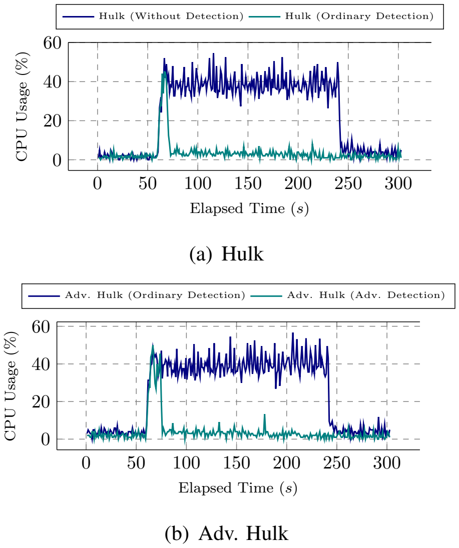
(d) Adv. Slowloris (Adv. Detection)

Fig. 5. The Web server's response time over the time in the case of Slowloris attack

B. Web Server's System Load

The results depicted in Fig. 6 demonstrate the significant impact of Hulk attack on the Web server's CPU usage, where the average CPU usage increased from 1.7% without attack to 36% during the attack. However, Fig. 7 shows that Slowloris attack has a negligible effect on the CPU usage. In the conducted tests, the average CPU usage increased to about 3.5%.

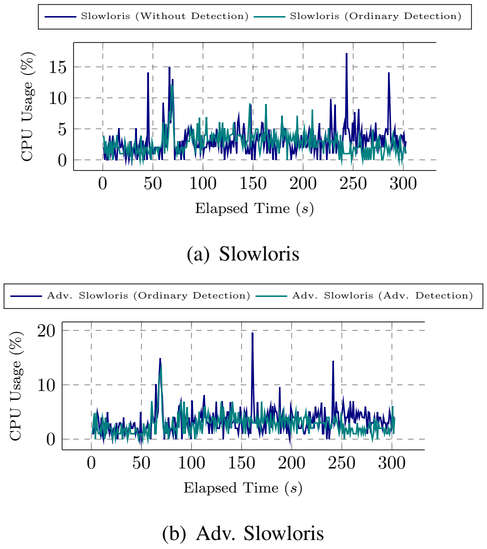
Fig. 8 and Fig. 9 display the Web server's RAM usage in the case of Hulk and Slowloris attacks, respectively. It is observed that both attacks exhibit almost the same behavior in impacting the RAM usage. In fact, the RAM usage is linearly increasing with the increase of attack duration, and starts gradually decreasing when the attack is stopped. It is worth observing from Fig. 8(b) that the original DL-based model was able to detect some adversarial Hulk samples during the attack period, which explains the gradual drop in RAM usage between $160s$ to $240s$.



(a) Hulk

(b) Adv. Hulk

Fig. 6. Hulk attack: System Average Load (CPU)



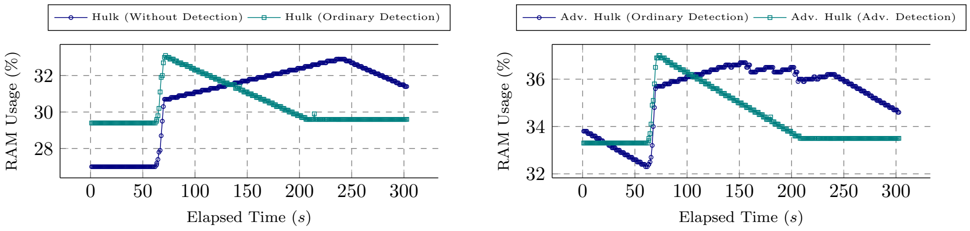
(a) Slowloris

(b) Adv. Slowloris

Fig. 7. Slowloris attack: System Average Load (CPU)

V. Conclusion

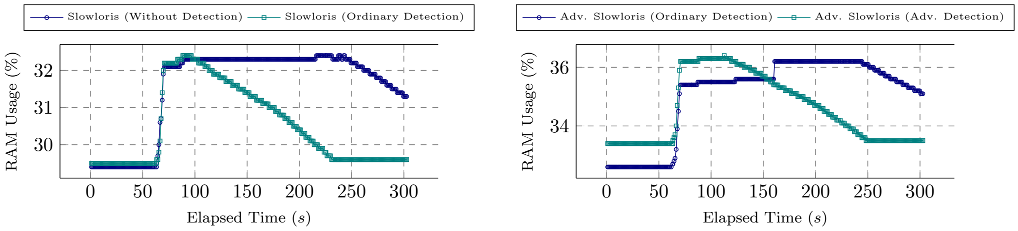
In this paper, we presented a Robust Application-Layer DDoS Self-Protection Framework that empowers the fully automated detection and mitigation of Application-Layer DDoS attacks leveraging both DL and SDN. The framework's robustness stems from its ability to mitigate adversarially-generated attack flows, thanks to the adversarial training defense used to train the model on adversarial DDoS flows. The devised framework was implemented and deployed on an experimental testbed. The obtained results demonstrate the effectiveness of the implemented solution in tackling the Application-Layer DDoS attacks even in the presence of adversarially-crafted



(a) Hulk

(b) Adv. Hulk

Fig. 8. Hulk attack: System Average Load (RAM)



(a) Slowloris

(b) Adv. Slowloris

Fig. 9. Slowloris attack: System Average Load (RAM)

malicious flows. In the near future, we will consider other ML techniques (e.g., RL) and defenses against adversarial attacks.

Acknowledgment

This work was supported in part by the European Union's Horizon 2020 research and innovation programme under the INSPIRE-5Gplus project (Grant No. 871808), the Academy of Finland Project CSN (Grant No. 311654), and the Academy of Finland Project 6Genesis Flagship (Grant No. 318927).

References

[1] T. Lintemuth and P. Hevesi, "DDoS: A Comparison of Defense Approaches," *Gartner*, Apr. 2019.

[2] A. Praseed and P. S. Thilagam, "DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Appl." *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 661 – 685, 2019.

[3] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues and Challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602 – 622, 2016.

[4] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based Network Intrusion Detection System using Machine Learning Approaches," *Peer-to-Peer Networking and Appl.*, vol. 12, no. 2, pp. 493 – 501, March 2019.

[5] A. Boudi, I. Farris, M. Bagaa, and T. Taleb, "Assessing Lightweight Virtualization for Security-as-a-Service at the Network Edge," *IEICE Trans.*, vol. E102.B, no. 5, pp. 970 – 977, 2019.

[6] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 812 – 837, 2019.

[7] A. Boudi, I. Farris, M. Bagaa, and T. Taleb, "Lightweight Virtualization based Security Framework for Network Edge," in *Proc. of IEEE Conf. on Standards for Communications and Networking (CSCN)*, Oct. 2018.

[8] C. Benzaid and T. Taleb, "AI-driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions," *IEEE Network Magazine*, to appear.

[9] T. Tang, S. A. R. Zaidi, D. McLernon, L. Mhamdi, and M. Ghogho, "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," in *Proc. of the 4th IEEE Conf. on Network Softwarization and Workshops (NetSoft)*, June 2018, pp. 1 – 5.

[10] S. S. Mohammed, R. Hussain, O. Senko, B. Bimaganbetov, J. Y. Lee, F. Hussain, C. A. Kerrache, E. Barka, and M. Z. A. Bhuiyan, "A New Machine Learning-based Collaborative DDoS Mitigation Mechanism in Software-Defined Network," in *Proc. of WiMob*, Oct. 2018.

[11] M. Siracusano, S. Shiaeles, and B. V. Ghita, "Detection of lddos attacks based on tcp connection parameters," *CoRR*, vol. abs/1904.01508, 2019.

[12] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. Tygar, "Can Machine Learning Be Secure?" in *In Proc. of ASIACCS06*, 2006.

[13] C. Benzaid and T. Taleb, "ZSM Security: Threat Surface and Best Practices," *IEEE Network Magazine*, to appear.

[14] S. Bhatia, S. Behal, and I. Ahmed, "Distributed Denial of Service Attacks and Defense Mechanisms: Current Landscape and Future Directions," *Versatile Cybersecurity*, vol. 72, pp. 55 – 97, 2018.

[15] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS Flooding Attack Detection using NOX/OpenFlow," in *IEEE Local Computer Network Conference*, Oct. 2010, pp. 408 – 415.

[16] T. Kohonen, "The self-organizing map," *Proceesings of the IEEE*, vol. 78, no. 9, pp. 1464 – 1480, 1990.

[17] K. Hong, Y. Kim, H. Choi, and J. Park, "Sdn-assisted slow http ddos attack defense method," *IEEE Commun. Lett.*, vol. 22, no. 4, Apr. 2018.

[18] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust Deep Reinforcement Learning with Adversarial Attacks," in *In Proc. of the 17th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS'18)*, July 2018, pp. 2040 – 2042.

[19] L. Haung, A. D. Joseph, B. Nelson, B. I. Rubinstrein, and J. D. Tygar, "Adversarial Machine Leaning," in *In Proc. of 4th ACM Workshop on Artificial Intelligence and Security*, Oct. 2011, pp. 43 – 58.

[20] Y. Han, B. I. Rubinstein, T. Abraham, T. Alpcan, O. De Vel, S. Erfani, D. Hubczenko, C. Leckie, and P. Montague, "Reinforcement Learning for Autonomous Defence in Software-Defined Networking," in *In Proc. of the 9th Int. Conf. on Decision and Game Theory for Security (GameSec)*, Aug. 2018, pp. 145 – 165.

[21] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2014.

[22] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Commun. Surveys Tuts.*, 2019.

[23] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proc. of the 4th Int. Conf. on Inform. Syst. Security and Privacy (ICISSP)*, Jan. 2018.

[24] MOSA!C, "Application-Layer DDoS Dataset," https://drive.google.com/drive/folders/1h-b7px3RmgvzXS1notQl2dl9RobHko8x?usp=sharing, 2019.