Low False Alarm Ratio DDoS Detection for ms-scale Threat Mitigation

Peter Orosz, Balazs Nagy, Pal Varga
Department of Telecommunication
Budapest University of Technology and Economics
Budapest, Hungary
orosz, bnagy, pvarga {@tmit.bme.hu}

Mitch Gusat
Zurich Lab
IBM Research
Rüschlikon, Switzerland
mig@zurich.ibm.com

***Abstract*—The dynamically changing landscape of DDoS threats increases the demand for advanced security solutions. The rise of massive IoT botnets enables attackers to mount high-intensity short-duration *”volatile ephemeral”* attack waves in quick succession. Therefore the standard human-in-the-loop security center paradigm is becoming obsolete.**

To battle the new breed of volatile DDoS threats, the intrusion detection system (IDS) needs to improve markedly, at least in reaction times and in automated response (mitigation). Designing such an IDS is a daunting task as network operators are traditionally reluctant to act – at any speed – on potentially false alarms. The primary challenge of a low reaction time detection system is maintaining a consistently low false alarm rate. This paper aims to show how a practical FPGA-based DDoS detection and mitigation system can successfully address this.

Besides verifying the model and algorithms with real traffic ”in the wild”, we validate the low false alarm ratio. Accordingly, we describe a methodology for determining the false alarm ratio for each involved threat type, then we categorize the causes of false detection, and provide our measurement results. As shown here, our methods can effectively mitigate the volatile ephemeral DDoS attacks, and accordingly are usable both in human out-of-loop and on-the-loop next-generation security solutions.

***Index Terms*—FPGA, Intrusion detection and prevention, DDoS, Network security, Data Center Networks**

I. Introduction

Distributed Denial of Service (DDoS) attacks are plaguing the Internet with their uncanny harming ability. These attacks are cheap and simple to launch, fast to ramp up and exceedingly effective in results. There is such a rapid growth of these attacks, that even a phenomenon called DDoS-as-a-Service [1] exists. Unlike the earlier DoS attacks, modern DDoS attacks are mostly driven by automated botnets. The Reaper botnet alone has infected more than a million IoT devices, which actually serve as an easy-to-reach resource-pool to mount massive attacks [2]. On the contrary, a recent attack against GitHub did not necessarily resort to botnets – rather, it exploited the vulnerability to amplification attacks of the infrastructure's Memcached servers [3].

Corero identified two *key trends* during 2017 [4]: One is the emergence of volatile ephemerals: i.e., brief and low volume attacks, while the another trend is in their increased frequency. That is, 58% of all attacks last less than 5 minutes, down from 10s of minutes and hours in the past. The attacks' frequency meanwhile has increased by 35% quarterly, which is more than 200% p.a. Thus the detection of such volatile ephemeral attacks is challenging if the Intrusion Detection System (IDS) has reaction times in the seconds or minutes.

Aggravating the effect of machine-driven high-scale/frequency bursty ephemeral DDoS attacks, the human component is also becoming a weak spot and exposure. If each IDS decision requires human input, the (much) higher frequency of attacks can add a huge workload to the security operator, who is overwhelmed before the volatile ephemeral DDoS. The GISW2017 [5], which included 19000 security professionals worldwide, forecasts an information security workforce gap of 1.8 million by 2022. One answer to this problem is increasing the automation: offloading decisions to AI. To achieve fast mitigation we also have to aid or remove – partly or fully – the human component from the decisions related to the machine-driven attacks.

Such automation is currently prevented by a challenge: Any false positive actions may entail a great cost – e.g., in service denial and loss – to the customer. According to Cloud Security Alliance, 31.9% of security operators face alert fatigue: ignoring critical alerts, mostly because they are confronted already with too many false positives [6].

Fast, hardware accelerated, non-DPI IDS's – that are also accurate in their detection, – are becoming favorable in DCNs for two reasons. First, the GDPR update became EU law in May 2018 [7], thus data security is increasingly important for the both cloud providers and their tenants. Besides, end-to-end encryption is becoming more and more prevalent [8], making all DPI-based anomaly detection at least cumbersome, if not impossible [9].

The motivation for our work was to answer, whether a high-speed and -confidence IDS is practically feasible for mitigating DDoS within milliseconds for end-to-end encrypted traffic? As a proof of concept, we developed an FPGA-based IDS [10], which can detect over 96% of DDoS attacks – the 9 most frequent attacks reported by Akamai [11] – in ms timescales. This proves that indeed, ms-scale mitigation is possible.

The current paper seeks the answer to the second part of the burning question: confidence. In order to answer this, we describe the methodology for determining the false alarm ratio, then we categorize the causes of false detection, and provide our measurement results.

978-3-903176-14-0 © 2018 IFIP

II. Related Work

There are currently a wide range of DDoS detection solutions available on the market. The more comprehensive systems include Incapsula by Imperva [12], DefensePro by Radware [13], FortiDDoS by Fortinet [14], Arbor Networks [15], and CloudFlare [16], among others. While these solutions cover a large variety of attacks, their advertised detection times remain in the range of seconds to minutes, which is arguably insufficient against the new breed of volatile ephemeral massive DDoS.

The reduction of false positives is a crucial issue. System operators and researchers keep developing various new methods to reduce the false alarm rate. Still, the vendors of industrial-grade detection mechanisms generally do not publish their capabilities and validation methods. On the other hand, the industrial players do publish methods of DDoS attack generation, and some general practices of DDoS detection [17].

There are both classical and modern methods in signal detection theory [18] for determining the probability of false decisions. Classical methods include the Gaussian and the Markovian approaches, while modern methods are more concerned with specific problems – such as stochastic problems with apriori uncertainty, non-Gaussian signals, or the use of sequential analysis.

An advanced methodology for decreasing false alarms through fusion of DDoS detection algorithms has been demonstrated in [19]. This work is based on the Dempster-Shafer Rule of Combination [20]. The method aims for reasoning with uncertainty, and involves collection of evidence, as well as marking the certainty of that evidence.

Tran Ngoc Thinh, et. al. proposed a novel FPGA-based DDoS filtering architecture [21] in 2015. They used the Management Information Base (MIB) to identify DDoS attacks and the FPGA to filter the attacks with egress and ingress filtering. They used a 10 Gbps NetFPGA board to implement their design, and achieved a throughput of 9.869 Gbps. They published a further paper [22] two years later, with an extended architecture. This included soft-core processors for DDoS detection, while keeping their novel FPGA-based filtering architecture. They achieved 0.74% false positive detection and 0% false negative detection ratios, though they mostly used synthesized traffic. Calculation of false alarm ratio can be precise this way, but synthetic traffic has limited complexity.

The lack of corpus curse: In order to properly evaluate the capabilities of an IDS, large datasets are needed in a corpus for machine learning ingestion, in which the *evidence* of attacks are labeled as ground truth. Such attack corpus datasets are unfortunately not available yet. The publicly available datasets are either too old (e.g., by DARPA or KDD Cup from 1999), or dedicated to specific attacks (e.g., by CAIDA [23], ANT [24], DDoSDB [25]) – and even the latest are long obsolete. It remains impossible to find *borderline* traffic traces that contain clearly annotated false positive or false negative patterns.

Without such a corpus, we have resorted to a more tedious manual verification: We checked all the detected cases (over 100) in the 3-months measurement period when our system was under validation at NIIFI, the National IT Infrastructure Development Institute in Hungary. Our aim was to determine the capability of our algorithms in detecting attacks with minimum false positive (or negative) hits (or misses).

III. Architectural overview

We briefly describe here the design and the key features of our FPGA-accelerated DDoS detection and mitigation system able to cope with end-to-end encrypted traffic. A more detailed description of the ms-scale detection algorithms is presented in [26]. In order to accelerate the ephemeral DDoS detection and mitigation, we identified seven key features for the FPGA-based IDS. These include (1) high confidence detection of the Akamai Top-9 DDoS attacks, (2) automated mitigation, (3) millisecond range attack detection, (4) detection of ephemeral/volatile/transient/stealth attacks, (5) minimal false positive detection, (6) lossless packet processing up to 100 Gbps line-rate, and (7) fast re-development cycle for mitigation of new attacks.

A. Overview of the monitoring-detection-mitigation pipeline

In the monitoring-detection-mitigation pipeline shown by Figure 1, the pre-processing step is done by an FPGA-accelerated node [27]. Although it would be possible and beneficial in many applications, for the privacy of our network traffic we opt not to perform deep-packet inspection (DPI), but rather to rely solely on the information included in the Layer 2 to Layer 4 headers.

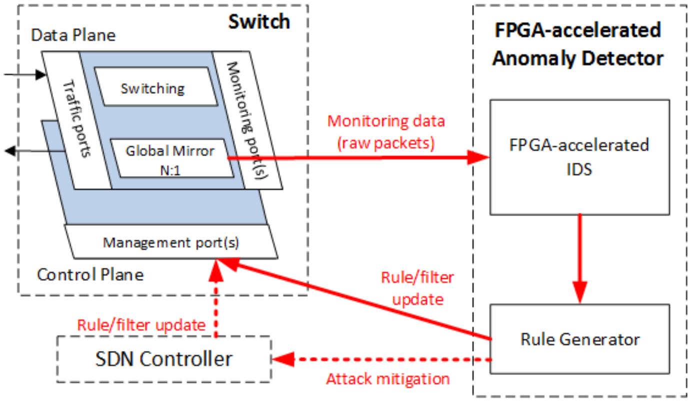

Fig. 1. The monitoring loop for hardware-accelerated anomaly detection

The summary of the proposed architecture can be found in [10]. The preprocessing stage in Figure 1 converts the mirrored data stream into a suitable format for anomaly detection. Depending on the desired application or service, this analysis may include the estimation of flow-level statistics (e.g., per-flow throughput) or traffic matrices (e.g., subnet level).

B. Architecture of the FPGA-accelerated IDS Node

The FPGA-accelerated IDS system is organized in three functional units; Packet Decoder and Parser, End-point Packet Rate Profiler, Behavioral and Heuristic DDoS Detector. Based on the results of the inspection, ACL (Access Control List) rules are generated and directly fed back into SDN-enabled switches, allowing for ms timescale attack mitigations (with further potential for μs in optimized implementations). Alternatively – as a more generic, although slower reaction – the rules can trigger the SDN controller, as shown in Figure 1.

The *packet decoder and parser module* extracts the useful fields from the packet headers and creates an 11-tuple for each packet. The *end-point packet rate module* measures the packet rate of each Layer-3 end-point according to Equation (1), where *f(t)* is the corresponding traffic function and *k* is the packet rate limit. Every end-point is online monitored in real-time. The network operator sets a rate limit, above which the end-point will be considered suspicious and will be forwarded to the *DDoS Detector unit*. After the signal, a *DDoS Detector unit* will start its inspection window, in which its going to sample every incoming packet for signs of DDoS attack.

A similar solution is employed by the Palo Alto DDoS protector [28].

$$\frac{1}{t_{n+1} - t_n} \cdot \int_{t_n}^{t_{n+1}} f(t)dt \geq k \tag{1}$$

The purpose of the *behavioral and heuristic DDoS Detector unit* is to check for traces of DDoS traffic. It runs behavioral and heuristic algorithms to detect attacks.

C. Calibration of the Inspection Window

The DDoS inspection starts when the end-point packet rate profiler found an end-point suspicious. The correct calibration of the inspection window is a key issue in every measurement application. We observed that constant inspection frequency in a delicate real-life security application entails multiple potentially harmful side-effects. First, the high intensity attacks are oversampled, hence (many) more than necessary packets are used for a high-confidence decision, i.e., higher effort and longer detection lags. Second, the low intensity attacks are undersampled, since at the end of a inspection period the IDS has insufficient data to reach a high-confidence decision that may increase the false alarm rate. Our proposed solution is to employ sequential analysis methods with adaptive inspection window, based on the ingress packet rate from the suspected attack. Each inspection window is based on the partial results of the corresponding measurement. In Section VI, we show the improved effectiveness of this approach.

D. Threat Mitigation

According to the best practice common in security systems, the decision confidence level can be mapped to their attack status and the corresponding action: *certain, probable, possible (attack), and non-attack*. Decisions with high confidence level, i.e., certain (positive) attack and non-attack status, can trigger direct actions, such as ACL filtering (e.g., blackholing) or leaving the traffic unfiltered (pass-thru). In contrast, a traffic with possible or probable attack status can be specifically handled. In this latter scenario, the suspect traffic can be diverted to a specific scrubbing network within the cloud, to be handled by a dedicated server cluster in the scrubbing datacenter.

IV. Achieving Low False Detection Rate

A. Boundary Conditions of this Assessment

The results of our assessments are valid within the following circumstances.

- Merely Layer 3-4 volumetric attacks are executed;
- Attack types that are included in the Akamai big-9 group (roughly 95% of all DDoS attacks) [11];
- The rate of the attack is higher than the attack volume limit set by the operator.

The Attack volume limit is a hard threshold set within the end-point packet rate profiler module. Every end-point, which receives traffic at a lesser rate than the limit is considered attack-free. This limit is reasonable to be set between 400Mbps and 10Gbps. A similar range is employed by Forti [14] and Palo Alto [28]. This limit is derived from the size of the protected network, while the resource availability within the FPGA hardware determines its lower bound. Equation (2) describes this limit, where $n_{maxhost,limit}$ is the maximal number of concurrent over-the-limit end-points of the protected network, n_{adv} is the number of parallel behavioral and heuristic DDoS detector modules instantiated in the FPGA, t_{resamp} is the time after an end-point is re-tested for attacks, and t_{dec} is the time required to make decision. The n_{adv} is dependent on the FPGA resources, the bottleneck in this case is the number of DSP slices. Our current hardware [29] can contain $70 - 100$ of these modules, while a similarly priced modern Kintex UltraScale+ can contain $600 - 800$. By increasing t_{resamp} the lower bound can be further extended, but the detection times may increase. The t_{resamp} is currently set to 0.1s. The t_{dec} is between 0.5 ms and 15 ms based on the incoming traffic rate of the end-point.

$$n_{maxhost,limit} \leq n_{adv} \cdot \frac{t_{resamp}}{t_{dec}} \tag{2}$$

B. Methodology of this Assessment

The first step of the design was the background research. We collected traffic traces, both malicious and non-malicious, in the NIIF network. With those samples and others available online we created the specification for the IDS. We improved the specification based on researches done by security firms, especially Akamai. We designed methods and implemented them into our FPGA based system. Then we verified our system with the collected traces. We demonstrated in previous works [10], [26] that our system works properly according to the specification. In this paper, we propose novel methods to validate our detection algorithms, i.e., to verify the false detection rate of the IDS. First, we identified the possible causes of false detection. We synthesized various traffic pattern based on real DDoS traces, which are capable of inducing false detection. This way the dataset, which guided us in the creation of the detection methods is completely separated from the dataset used to assess the false rate of the system.

V. Causes of False Detection

This section discusses the algorithms of the advanced detector units by categories. We analyze how false positive and false negative alarms can be generated and what measures were used to eliminate the false positives and minimize the false negatives. The Akamai big-9 can summarized into 2 groups by detection method: i) attacks that can be detected based on protocol behavior deviations (UDP frag, DNS, NTP, CLDAP, SSDP, SNMP, ACK), ii) attacks that can only be identified by heuristics (UDP). The first group can be handled with 1 or 2 algorithms per attack, on the other hand UDP is detected by 5 algorithms in our implementation. The tighter the definition of an attack, the easier to find a rule for its detection. For the more vaguely-defined attacks, such as UDP floods, heuristic methods are more appropriated. **Protocol behavioral algorithms:** False negatives can be induced by network anomalies – i.e., packets not reaching the detector – and attack masking methods. False positives can be generated by network anomalies. **Heuristic algorithms:** False negatives can be triggered by network anomalies and elaborated Day-0 attacks that apply methods unknown to our heuristics. False positives can be raised due to network anomalies and particularly unusual, albeit legitimate traffic. The causes of the false detection are summarized in Table I.

TABLE I
POSSIBLE CAUSES OF FALSE DETECTION

	False +	False -
Protocol behavior	Packet loss, Sampling	Attack masking Packet loss, Sampling
Heuristic	Anomalous traffic Packet loss, Sampling	Masking, Day-0 attacks Packet loss, Sampling

A. Handling the Adverse Effect of the Inspection Window

The detector unit inspects the data flow for a few *ms* only, so we cannot assume that the current number of incoming packets are the expected value of the packets.

For example, consider a normal data transfer between two hosts that includes some fragmented packets, with a (possibly high) amount of last-fragment packets not arriving within the inspection period, while all of the first fragments were successfully received.

We applied statistical parametrization to calculate such occurrences. To model this we used a Poisson distribution, because in modeling networks with large population of independent users contributing to the aggregate, user sessions can be assumed to follow a Poisson arrival process [30].

B. Coping with Packet Loss (from network anomalies or the nature of inspection)

Some scenarios for asymmetric packet loss can cause both false negative and positive alarms. While our DDoS detector is capable of lossless packet processing, we cannot assume that the rest of the network path is lossless as well. Each detection algorithm was calibrated to effectively handle packet loss. Let us make two assumptions for the sake of simplicity.

First, the probability of packet types within the inspection window is the theoretical probability $p = q = 0.5$, – i.e., the first and last fragment have equal probability. In subsection V.A, we analyzed the effect and probability of unexpected number of packets.

Second, the packet loss can be modeled with binomial distribution – see Equation (3). This is a good model, if the loss is caused by the inspection window.

$$\sum_{k=0}^{N/4} \binom{N}{k} \cdot p^k \cdot q^{N-k} \tag{3}$$

In Section VI, we show that the first false positives are generated when the natural 1:1 ratio (of p:q) becomes 3:1 so we take that as a base for our calculation. Each measured attack can be modeled with 1:1 natural ratio, e.g., there is a last fragment for every first fragment in IPv4 fragmentation and there is a DNS response first fragment for each DNS request. The 3:1 ratio is based on a heuristic approach. If we set the rate to as high as 7:1 then a regular traffic can mask the attack. If we set it too low (e.g., 2:1) then the probability of false alarms corresponding to packet loss can be quite significant. Figure 2 depicts the false positive probability ratio. For packet loss rates typical for regular operation, the false positive probability is extremely low. But for undersampling cases, the false positive probability can be quite significant. Our solution for this problem is a sequential analysis-based detection method, in which the sampling window is automatically extended if the number of incoming packets is low.

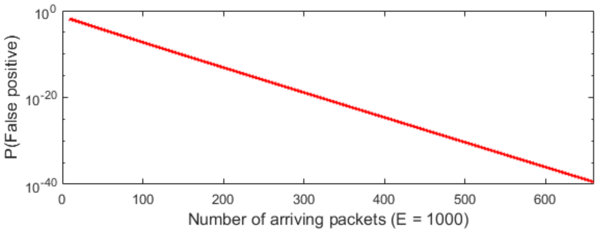

Fig. 2. The probability of false positive versus the number of non-lost packets (lin-log scale)

C. Non-Malicious Anomalous Application Traffic

Some highly unlikely anomalous traffic can also trigger the heuristics algorithms. Our heuristics include the comparison of the payload hashes of the packets arriving to the target. This can be done since the current IDS implementation receives 96B truncated packets, which contain enough data to create hashes. If the majority of the packets' payload hashes are extremely similar to each other, we assume that it is a DDoS attack. Most direct DDoS attacks use cloned packets to achieve maximal efficiency on the given hardware. These anomalies can happen in case of ”Heartbeat”, ”Sync” and ”Keep-alive” applications. Such applications use 1-1000 packets per sec., which is too small to trigger the IDS. It is quite possible nonetheless for such applications to persistently trigger the IDS; hence they should be manually added to the whitelist.

D. Handling Attack Masking

According to our measurements in the NIIF network, merely a small number of attackers are using such methods to mask their tracks. We captured an UDP fragmentation attack, which was masked with ”last fragment” IP packets. The last fragments were on different IPv4 identifications, making the packet reassembly impossible. Primitive or naive detection methods can be easily fooled in this case. NTP attacks can be masked with spoofed NTP requests, whereby the $src.IP = Target$, $dst.IP = Address\ of another\ datacentertenant$. As this masking always leaves a trace, the designer can refine the detection algorithms to the point where such masking can be detected. The UDP fragmentation masking can be identified with a stateful pseudo-reassembler.

VI. Results

Here we validate our IDS with traffic that was specifically created to generate false alarms. Our goal was to create plausible scenarios for each false alarm category. Unlike in our previous ”in-the-wild” papers, here we had to synthetically generate the majority of the traffic used for validation.

A. Handling the Adverse Effect of the Sampling Window

In this case, we performed a worst case analysis using one-sided variance, i.e., the response is the expected value and the number of requests varies. The test file contains 1000 pieces of 0.5s bursts (e.g., NTP requests), with $0\ - 100\% \cdot \lambda$ packet number on (e.g., NTP-) request packets. Interarrival times of request packets within the bursts are generated by an exponential distribution, hence the timewise spacing of packets follow a Poisson distribution within the burst. The detector was modified to generate ”not attack” messages, if the burst was not considered as attack. The IDS messages were logged to a text file and manually analyzed. For each step a relative probability is calculated based on the Poisson distribution. Figure 3 shows our results versus the calculated probability.

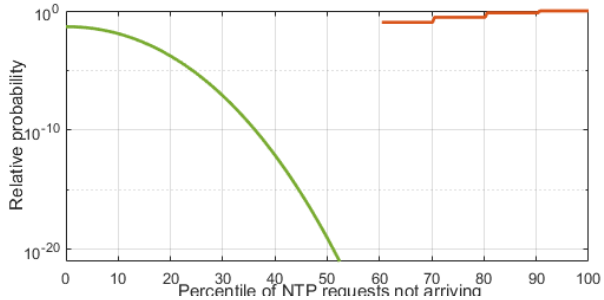

Fig. 3. The measured probability (red, ’step-like’) of false positive detection and its theoretical probability (green, ’falling’). The red line is the product the number of false detection aggregated in 10% steps, e.g., number of false positive : number of tests with 10%-20% packet loss

B. Handling Packet Loss (Network Anomalies or Sampling)

Here we simulate our adaptive sampling time method versus the constant sampling frequency. The two methods are virtually similar at higher packet rates, but different at lower packet rates. Figure 4 shows that *our method reduces the loss probability* by orders of magnitudes (10^{-2} vs. 10^{-4} around the lower bounds), for only a small relative-increase in the detection time (see Figure 4). This trade-off can be beneficial in applications, where the cost of false detection is high.

We trade a single digit of detection time for two orders of magnitude in false positive probability, by adaptively extending the inspection window. The false detection probability here spans from (10^{-40} to 10^{-4}).

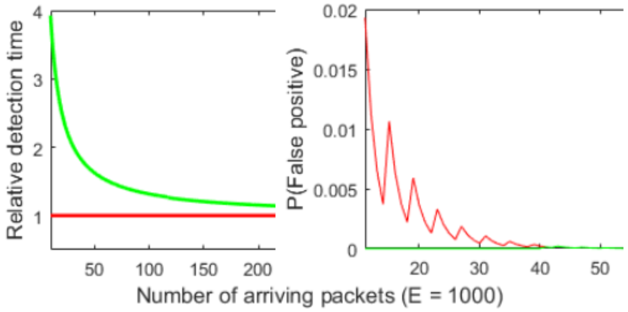

Fig. 4. Left: The false positive probability of the basic rule (red) versus our method (green, ’close-to-zero’). Right: The increase in detection lags, basic rule (red), our method (green)

C. Non-Malicious Anomalous Application Traffic

We captured our ”wild” DDoS traffic in the NIIF datacenter using sample rate based rules, as explained in [10], [26]. As this system generated many false positive alarms, we used our synthetic false positive traffic to test the IDS. We replayed five long samples, which could be verified manually as false positives. None of the replayed traffic was registered as attack by our IDS, which is a convincing result.

D. Handling Attack Masking

For this purpose we used real attack traces captured in NIIF and added masking packets to generate false negatives. The attacks were masked with the methods presented in Subsection V.E. We used four different attacks (i.e., NTP, DNS, UDP frag, RIP) and measured the detection times. The measurement methods are explained in [26]. As Figure 5 shows, detecting the masking over such attacks does not significantly impact the *millisecond-range* detection times experienced for attacks without masking (see Figure 6).

E. Results overview

In this section, we summarize the results, calculate the total false ratios, and we are going to estimate, how many decisions can be automated. The term over-the-limit host (OTH) will be used to describe a host, which has incoming traffic higher than the limit set in end-point packet rate profiler module (Subsection III.C). The average number of OTH is calculated based on (4), where k is the total number OTH and $t_{n,hight}$ is the time when the host receives traffic over-the-limit. The OTH is a very important metric for our system, because only OTH traffic can cause false positive detection. The value of

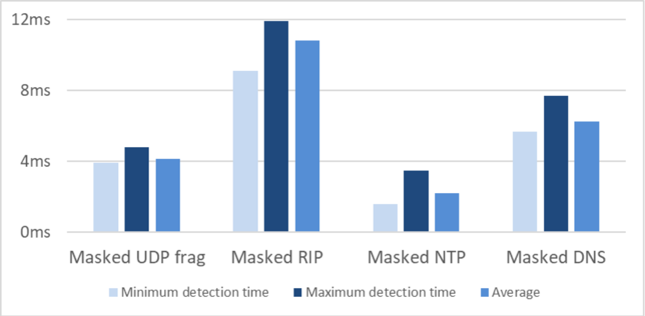

Fig. 5. The ms-range detection times of masked attacks

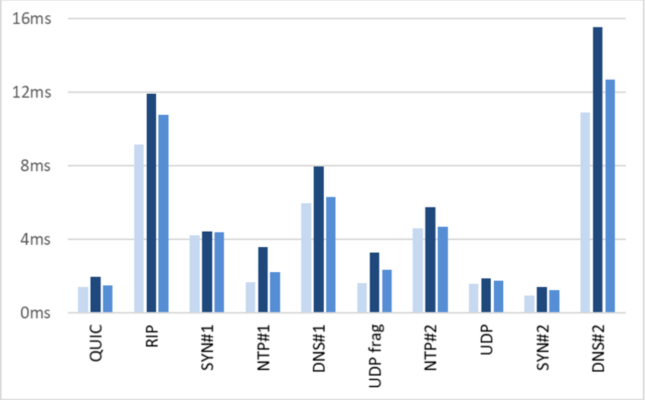

Fig. 6. The ms-range detection times of regular attacks captured in NIIF

the limit is irrelevant in this calculation (400Mbps - 1Gbps was used in NIIF). The total number of DCN hosts are much higher $(10^3, 10^4)$ than the average number of OTH.

$$A_{OTH} = \sum_{n=1}^{k} \frac{t_{n,hight}}{t_{n,total}} \tag{4}$$

The false detection ratios are calculated with total probability theorem. The $P_{totalfalse}$ has to be weighted with the following unique characteristics of our IDS: each host's traffic, which is labeled by the end-point packet rate profiler, is sampled over and over again. In an hour, this kind of traffic is sampled 36000 times, at 0.1 ms re-sampling rate. This is a necessary mechanism to detect attacks swiftly against high traffic hosts.

TABLE II
THE PROBABILITIES OF FALSE DETECTION. THE NUMBER IS CALCULATED WITH THE DATA DESCRIBED IN TABLE III [1] FALSE + PROBABILITY OF 1 HOUR OVER THE RATE LIMIT TRAFFIC [2] IF THE ATTACK IS NOT DETECTED IN THE FIRST SECOND: CLASSIFIED AS FALSE NEGATIVE

Probability	False + [1]	False - [2]
Packet loss	$7.32 \cdot 10^{-43}$	$3.39 \cdot 10^{-50}$
Inspection	$1.1 \cdot 10^{-25}$	$5 \cdot 10^{-33}$
Attack masking	NA	0
Anomalous traffic	$< 10^{-5}$	NA

Table II shows the false detection probabilities. The question arises, how these probabilities can be translated into real-life data-center operation. Unfortunately, no uniform answer can be given to this question, because the characteristics of each data center is different. Table IV provides a yearly estimation on what can be expected in DCNs with different sizes. The average number of OTHs used for this calculation are set this way to put our results into different context. On average, the KIFU-NIIF network has 5-10 OTHs, at 1Gbps rate limit.

TABLE III
THE PROBABILITY DATA SOURCES OF EACH CATEGORY. * THE PROBABILITY OF PACKET LOSS EVENTS WERE CALCULATED WITH THE MARKOVIAN MODEL, THE LOAD FACTOR WAS CHOSEN TO BE 0.9, WHICH IS LARGER THAN MOST REAL NETWORKS' LOAD

	P_{event}	$P_{relative}$
Packet loss	Mathematical model *	Simulation Fig. 4
Inspection	Mathematical model Fig. 3	Expected Result Fig. 3
Attack masking	NIIF Measurement Section II	Exp. Result Fig. 6
Anomalous traffic	NIIF Measurement Section II	NIIF Measurement Section II

TABLE IV
NUMBER OF EXPECTED YEARLY FALSE DETECTIONS FOR DIFFERENT DCNS.

	Average number of OTHs					
	12		120		1200	
	F+	F-	F+	F-	F+	F-
Packet loss	10^{-38}	10^{-45}	10^{-37}	10^{-44}	10^{-36}	10^{-43}
Inspection	10^{-20}	10^{-27}	10^{-19}	10^{-26}	10^{-18}	10^{-25}
Anom. traffic	1	NA	10	NA	100	NA

Our results show that we can automate many functions that was previously done by security experts, but the human component cannot be completely removed yet, mostly due to Day-0 attacks.

VII. CONCLUSION

The new breed of machine-driven high-scale/frequency bursty ephemeral DDoS attacks launched via multi-million node botnets mandate a new generation of IDS tools, vastly improved in reaction times and in the confidence of mitigation. Designing such an IDS is a risky endeavour as the security operators are adamant to act – at any speed – on what may later prove as false alarms with costly consequences. A paramount challenge of a fast (ms) reaction system was in maintaining a consistently low false alarm rate. We have shown that our detection and mitigation system can address this challenge at 100Gbps.

Besides verifying the model and algorithms with real traffic ”in the wild”, we have validated theoretically and practically the low false alarm ratio. Our further key contributions were the 'false'-proof and the (ms) detection methods amenable to FPGA acceleration. Such methods can timely mitigate the new ephemeral attacks and are effective both in human out-of-loop and on-the-loop security solutions.

Takeaway: Is a *high-speed* (ms mitigation lag) and *high-confidence* IDS practically feasible for e2e encrypted cloud traffic? Our answer is yes, despite the challenges to design and implement it in a real 100Gbps system.

References

[1] A. Zand, G. Modelo-Howard, A. Tongaonkar, S.-J. Lee, C. Kruegel, and G. Vigna, "Demystifying DDoS as a Service," *IEEE Communications Magazine*, vol. 55, no. 7, 2017, iEEE.

[2] Z. Sun, B. Feng, L. Lu, and S. Jha, "OEI: Operation Execution Integrity for Embedded Devices," Feb 2018, arXiv:1802.03462.

[3] GitHub Engineering. (2018) February 28th ddos incident report. [Online]. Available: https://githubengineering.com/ddos-incident-report/

[4] Corero. (2017 Q3) Ddos trends report q2-q3 2017. [Online]. Available: http://info.corero.com/

[5] Forst and Sullivan and Center for Cyber Safety and Education. (2017) 2017 global information security workforce study. [Online]. Available: https://iamcybersafe.org/wp-content/uploads/2017/06/Europe-GISWS-Report.pdf

[6] Cloud Security Alliance. (2016) Mitigating risk. [Online]. Available: https://cloudsecurityalliance.org/download/mitigatingrisk/

[7] EP and EU, "General Data Protection Regulation (GDPR)," *European Parliament and Council of the European Union (EU) 2016/679, in L119*, pp. 1–88, 2016.

[8] D. Severson, "The Encryption Debate in Europe," in *Aegis Paper series no. 1702*. Hoover Institute, 2017.

[9] M. Kuehlewind, T. Buehler, B. Trammell, S. Neuhaus, R. Muentener, and G. Fairhurst, "A Path Layer for the Internet: Enabling Network Operations on Encrypted Protocols," in *Conference on Network and Service Management (CNSM)*. Tokyo, Japan: IFIP/IEEE, 2017.

[10] P. Varga, G. Kathareios, A. Mate, R. Clauberg, A. Anghel, P. Orosz, B. Nagy, T. Tóthfalusi, L. Kovács, and M. Gusat, "Real-Time Security Services for SDN-based Datacenters," in *Conference on Network and Service Management (CNSM)*. Tokyo, Japan: IFIP/IEEE, 2017.

[11] Akamai. (2017) State of the Internet security report Q4. [Online]. Available: https://www.akamai.com

[12] Imperva Inc. (2018) DDoS Mitigation. [Online]. Available: https://www.incapsula.com/ddos/ddos-mitigation-services.html

[13] Radware Inc. (2018) DefensePro DDoS Protection, DDoS Prevention and Attack Mitigation. [Online]. Available: https://www.radware.com/products/defensepro/

[14] Fortinet Inc., "FortiDDoS Handbook Version 4.3.0," 2017.

[15] Arbor Networks. (2018) DDoS Attack Solutions. [Online]. Available: https://www.arbornetworks.com/ddos-protection-products

[16] Cloudflare Inc. (2018) Protect Against DDoS Attack. [Online]. Available: https://www.cloudflare.com/ddos/

[17] FortiNET. Understanding FortiDDoS Prevention Mode, Reducing false positives. [Online]. Available: http://help.fortinet.com/fddos/4-3-0/FortiDDoS/Understanding$_{F}ortiDDoS_{P}revention_{M}ode.html$

[18] V. P. Tuzlukov, *Signal Detection Theory*. Springer Science and Business Media, 2001.

[19] F. Mahmood, "Minimization of DDoS False Alarm Rate in Network Security; Refining Fusion through Correlation," Ph.D. dissertation, University of Windsor, 2012.

[20] R. R. Yager, "On the Dempster-Shafer Framework and New Combination Rules," *Information Sciences*, vol. 41, no. 2, 1987, elsevier.

[21] T. N. Thinh, C. Pham-Quoc, B. Nguyen-Hoang, T.-C. Tran-Thi, C. Do-Minh, Q. Nguyen-Bao, and N. Q. Tuan, "FPGA-Based Multiple DDoS Countermeasure Mechanisms System Using Partial Dynamic Reconfiguration," in *Journal on Electronics and Communications*. REV Journal, 2015.

[22] C. Pham-Quoc, B. Nguyen, and T. N. Thinh, "FPGA-based Multicore Architecture for Integrating Multiple DDoS Defense Mechanisms," *SIGARCH Computer Architecture News*, vol. 44, pp. 14–19, 2016.

[23] Center for Applied Internet Data Analysis. CAIDA, Data Sharing. [Online]. Available: http://www.caida.org

[24] University of Southern California, ANT Lab. ANT Data sets. [Online]. Available: https://ant.isi.edu/

[25] SIDN fonds. DDoSDB. [Online]. Available: https://ddosdb.org/

[26] B. Nagy, P. Orosz, T. Tóthfalusi, L. Kovács, and P. Varga, "Detecting DDoS Attacks within Milliseconds by Using FPGA-based Hardware Acceleration," in *Network Operations and Management Symposium (NOMS)*. Taipei, Taiwan: IFIP/IEEE, 2018.

[27] P. Varga and L. Gulyas, "Traffic analysis methods to support decisions at the Knowledge Plane," *Infocommunications Journal*, vol. 65, no. 4, pp. 50–56, 2010.

[28] Palo Alto Networks. (2016) How to Set Up DoS Protection. [Online]. Available: https://live.paloaltonetworks.com/

[29] P. Varga, L. Kovács, T. Tóthfalusi, and P. Orosz, "C-GEP: 100 Gbit/s Capable, FPGA-based, Reconfigurable Networking Equipment," in *International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2015.

[30] G. W. Patrick, M. Roberts, and C. Wulff, "Stability of Poisson Equilibria and Hamiltonian Relative Equilibria by Energy Methods," *Archive for Rational Mechanics and Analysis*, vol. 174, no. 3, pp. 301–344, Dec 2004.