Detection of Stealthy TCP-based DoS Attacks

Azeem Aqil*, Ahmed O. F. Atya*, Trent Jaeger§, Srikanth V. Krishnamurthy*,
Karl Levitt‡, Patrick D. McDaniel §, Jeff Rowe‡, Ananthram Swami†
*UC Riverside, †U.S. Army Research Laboratory, ‡UC Davis § Penn State University
{aaqil001, afath001, krish}@cs.ucr.edu, ananthram.swami.civ@mail.mil
{rowe, levitt}@cs.ucdavis.edu {tjaeger, mcdaniel}@cse.psu.edu

***Abstract*—Denial of service (DoS) attacks are among the most crippling of network attacks because they are easy to orchestrate and usually cause an immediate shutdown of whatever resource is targeted. Today's intrusion detection systems check if specific single scalar features exceed a threshold to determine if a specific TCP-based DoS attack is underway. To defeat such systems we demonstrate that an attacker can simply launch a combination of attack threads, each of which on its own does not break a system down but together can be very potent. We demonstrate that such attacks cannot be detected by simple threshold based statistical anomaly detection techniques that are used in today's intrusion detection systems. We argue that an effective way to detect such attacks is by jointly considering multiple features that are affected by such attacks. Based on this, we identify a possible set of such features and design a new detection approach that jointly examines these features with regards to whether each exceeds a high threshold or is below a low threshold. We demonstrate that this approach is extremely effective in detecting stealthy DoS attacks; the true positive rate is close to 100 % and the false positive rate is decreased by about 66 % as compared to traditional detectors.**

I. Introduction

Denial of service (DoS) attacks are among the most common of all network attacks [1]. Despite being well studied and several detection/preventive measures in place, DoS attacks continue to be prevalent today. The inherent ease with which DoS attacks can be initiated makes them attractive. Most attacks require minimal resources but can induce potentially crippling effects on the target.

Traditional detection engines for such attacks are of two types. Anomaly detection systems try to flag statistically significant deviations from normal behavior ([2] [3]) but such systems are limited by their choice of features and by the definition of normal behavior. Signature based systems essentially look at single scalar features for the purposes of detecting anomalies. For example, a high number of open but relatively unused ports would suggest that a TCP SYN flood attack is possibly occurring. Signature based schemes see much greater practical deployment[4]

In this paper, we argue that an attacker can simply undermine the efficiency of such practically deployed systems by combining a plurality of TCP-based DoS attacks. Essentially, he would use multiple different attack threads, each of which by itself would not overwhelm the system; however, jointly the threads would have the potency of a powerful DoS attack. More importantly, such an attack strategy defeats traditional intrusion detection systems that look for a single scalar threshold to be exceeded to issue an alert with regards to a TCP-based DoS attack; since the aggressiveness of each thread is moderate (controlled), these scalar thresholds are never exceeded causing the detection to fail. Thus, it is essential for a detection system to identify the effects caused by each attack individually and carefully assess the joint occurrence of such effects. If the evidence suggests that this to be the case, the detection engine can issue an alert.

We argue in this paper that an effective way to detect such attacks is by jointly considering a plurality of features (as opposed to a single scalar feature). We identify a basis set of such features and show that these are all affected in different ways by different TCP-based DoS attacks. Our experiments also help us design and implement a detection approach that jointly considers whether each of these features is (a) above a high threshold or (b) below a low threshold. This examination facilitates the identification of stealthy combinations of TCP-based DoS attacks while maintaining a relatively low false positive rate. In brief, we make the following contributions in this paper.

- Via extensive experiments we demonstrate the potency of a stealthy DoS attack and its ability to defeat traditional threshold based intrusion detection systems.
- Using an experimental approach, we identify features that are affected by each type of considered DoS attack at multiple layers. A combined examination of these features can yield a better assessment of whether or not the system is under attack.
- We propose an approach to combine the considered features in an effective way. Via experiments, we show that our approach is extremely effective (True positive $\approx 98\%$ and false positive $\approx 20\%$) in detecting both the stealthy DoS attack and traditional DoS attacks.

Scope: While our approach is generic and can account for stealthy attacks that combine a large number of different TCP-based DoS attacks, for clarity and tractability, we only consider a stealthy attack that combines two popular TCP-based DoS attacks viz., the TCP SYN flood and the Slowloris attacks. To account for other possible attacks, an offline study to understand the effects of such attacks, and features that may be effective in identifying them is needed. However, we believe that the generic approach

978-1-5090-0073-9/15/$31.00 ©2015 IEEE

that we use to jointly consider the TCP SYN flood and Slowloris attacks can be applied in such cases.

Roadmap: The rest of the paper is organized as follows. In Section II, we present brief overviews of the TCP SYN flood and Slowloris attacks. In Section III we provide a summary of related work. In Section IV, we showcase the stealthy DoS attack, and identify features that can be used as a basis set for detecting such attacks. In Section V, we discuss the key insights drawn from our experiments ; these lead to guidelines for design of detection approaches; experimental evaluation of such a design is then presented. Our conclusions are in Section VI.

II. Background

We provide a brief background on the DoS attacks we consider in our study. Specifically, to demonstrate the effectiveness of a stealthy DoS attack, we combine a SYN Flood attack[5] and a Slowloris attack [6].

A. The TCP SYN Flood Attack

The TCP SYN flood attack is a TCP-based DoS attack and has been known to the community for a long time. The attack takes advantage of the traditional TCP three way handshake mechanism. Most implementations of TCP establish some system state when a TCP connection is initiated through a SYN packet. Since there are practical limits on how much state can be maintained, attackers send a high volume of TCP SYN packets until the combined effect of all the half open connections saturates some system resource. This attack can be launched with minimal resources by an attacker since he is not required to maintain any state. In fact, most SYN flood attacks are launched using spoofed IP addresses. There are numerous approaches to defending against SYN floods but the most widely used approach is Syncookies[5].

B. The Slowloris Attack

Slowloris is a TCP-based DoS attack that exploits HTTP. The attack establishes multiple connections to a HTTP server and keeps them alive by regularly sending incomplete HTTP headers. The attack is maintained by sending partial, incomplete HTTP requests to the server and this continues to hog the connection as the headers are received regularly by the server. The idea is to cause a single server machine to maintain multiple connections until it runs out of all allocatable sockets and is thus, subject to DoS.

Slowloris is a different DoS attack as compared to the TCP SYN flood attack; it has different attacker and victim semantics. The attack requires much more attacker resources because it requires a single (powerful) machine that maintains multiple connections. However, the resource requirements are not too limiting because typically one could instrument the machine such that the attack program only wakes up periodically to send HTTP headers and then goes back to sleep. Application layer attacks like Slowloris are considered stealthy attacks because it's very hard to discriminate between legitimate traffic and attacks like Slowloris.

The attack semantics on the victim are also different. Instead of a traditional DoS attack like SYN floods where the victim machine is flooded with traffic, Slowloris attacks are characterized by bursts of traffic at regular intervals. This, potentially, makes this attack much more stealthy than high volume DoS attacks. The Slowloris attack can be further optimized if the server's connection timeout is known. This allows the Slowloris to minimize the number of times it has to send incomplete HTTP requests. There are various techniques, like load balancing and reverse proxies, that can be used to diminish the effect of Slowloris, but some versions of Apache, one of the most widely deployed servers, is still susceptible.

III. Related Work

In this section, we discuss existing detection schemes for DoS attacks. We also describe work relevant to mixed DoS and stealthy DOS attacks.

There are several efforts that target the detection of TCP SYN flood attacks. In [7], the authors use sequential change point detection to flag SYN floods. They analyze TCP behavior by looking at the number SYN and RST packets. However their approach is applicable only at leaf routers. The most prevalent detection and defense mechanism against SYN flood is the use of SYN cookies [5]. This approach detects attacks if the SYN buffer fills up. This is essentially a single threshold based technique and fails when a low intensity SYN flood is being employed with other attacks. Approaches such as the above, or popular signature based detectors such as those employed in Snort [8] and Bro [9] do not work against mixed DoS attacks because they do not collect and use adequate evidence information. In [10] the authors present a statistical approach to compute the correlation between requests and acknowledgments to detect anomalous behaviors; such an approach is reliant on some definition of normal traffic which can be tricky to characterize.There are other anomaly based detectors (such as [2] [3]) but they all face the problem of trying to accurately model or define normal behavior.

Slowloris is already considered a stealthy DoS attack. This is because it is very hard to differentiate between Slowloris traffic and normal traffic. It is also a relatively new attack.

There is little work on emerging application layer, TCP-based DoS attacks. In [11] the authors explore several application layer attacks but fail to provide any substantial defense mechanisms. In [12], an approach to detect application layer flooding attacks is presented. The approach however, only focuses on the application layer and will miss any mixed attacks that target the network layer. In [13], the authors model HTTP traffic with the aim of flagging anomalous behavior but it too suffers from being limited to the application layer.

There have been various Internet reports of attackers using multiple kinds of DoS attacks to achieve their goals ([14], [15]) but to the best of our knowledge, our work is the first to analyze the affects of a joint network and application layer TCP-based DoS attack.

IV. Stealthy DoS attacks: Impact and Key Characteristics

In this section, we conduct an experimental study to demonstrate the potency of a mixed or stealthy DoS attack and how such attacks can easily slip under the radar with respect to today's DoS detectors. We also identify key features that should be jointly considered for detecting such attacks.

Experimental Setup: All our experiments are conducted on the DeterLab test bed [16], a state of the art scientific computing facility for cyber security research. Our network topology consists of one victim machine, and two attacker machines that are connected to the victim machine. The links between the machines can be tuned to incorporate varying delays and different degrees of reliability (packet success). We also have a legitimate machine that issues requests and measures response times to the victim. The victim machine is running Ubuntu server version 14.04 and apache web server[17] version 2.2. The attacker machines are both running Kali Linux[18], a Linux based penetration testing distribution.

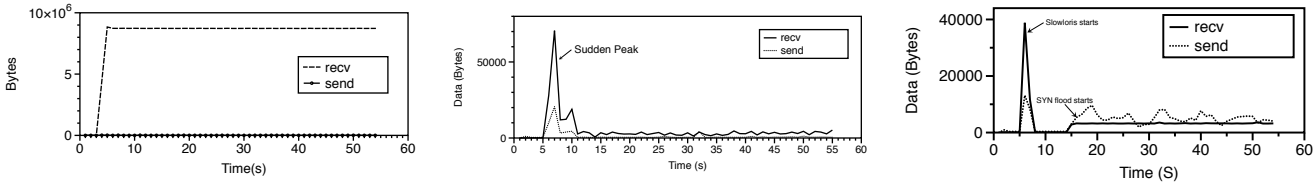
Designing a stealthy attack: As discussed in Sec II, traditional DoS attacks can be detected by simple statistical scalar measures. SYN floods are typically detected by examining if the rate at which SYN packets are received exceeds a certain threshold [5]. A Slowloris alert is issued if the number of incomplete HTTP headers is higher than a threshold. Our goal here is to experimentally determine the optimum detection threshold for our server by launching full scale Slowloris and SYN flood attacks and measuring the change in response as measured by the observer machine. The stealthy attack would then combine a mix of the two attacks, but each individually below the threshold determined as above. Full scale SYN flood is the maximum number of packets our SYN Flood program is able to generate (100 SYN Packets/Second). Full Scale Slowloris, is the max number of malicious connections our program can maintain (500 simultaneous connections).

To determine detection thresholds we launched each individual attack, starting from the highest possible frequency, and reducing this gradually, while measuring response time (with respect to a benign scenario) from the observer machine. We set the detection threshold to the point where the attacks were completely imperceptible from the point of view of the observer machine. We set our stealthy attack rates to be below this threshold; this results in our mixed attack evading the single threshold detection mechanisms. The detection thresholds are presented in table I. **Determining Key Features:** What are the features that facilitate the effective detection of both full scale and stealthy mixed attacks? This is the question we seek to answer here. Given how these attacks function, we chose a set of 6 performance metrics as features which can potentially characterize these attacks. The first two features are (i) the volume of data sent and (ii) the volume of data received. These two metrics are applicable for the following reasons. Attacks like SYN floods are characterized by a disproportionate amount of traffic received versus traffic sent. For high volume SYN flood DoS attacks, this single feature is often enough to identify attacks. We do not expect stealthy DoS attacks like Slowloris to display the same disproportionate traffic levels, they do however cause the volume of traffic sent to almost zero out. This feature by itself is not enough to identify an attack but does warrant suspicion.

The next two features in our set are system interrupts and context switches. Interrupts and context switches exhibit similar behaviors but are semantically different. A context switch occurs when the OS switches from the currently running execution thread. The kernel saves the state of the running execution thread and loads in a new one. An interrupt, however, occurs when the executing process receives an asynchronous signal requiring attention. Interrupt routines do not change context; they return to the same execution thread after the interrupt is handled. Examining context switches indicate how much, or how little, work is being done by the server. Interrupts are important indicators of unexpected events. Every new TCP connection request (SYN packet) will trigger an interrupt. Too many interrupts, coupled with too few context switches (a normal functional server has to serve multiple requests which results in an inevitably larger number of context switches) can be a possible indication that the server is not performing useful work. In other words, a high number of interrupts, coupled with few context switches and a high data reception rate can effectively identify a SYN flood. Slow loris is also likely to exhibit few context switches. The number is likely to be higher than that with SYN floods because actual connections are established here; however, it will still be low because these connections just wait rather than performing useful work.

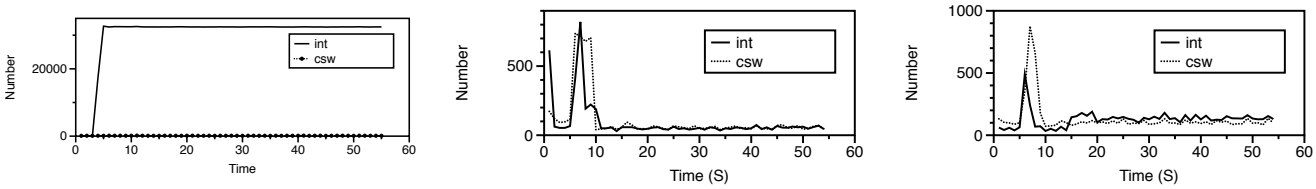
The final two features that we consider are the number of TCP sockets that are in the SYN state and in the established state (ACT sockets), respectively. Keeping track of sockets in the SYN state is useful because SYN flood attacks try and exhaust the SYN buffer. The number of sockets in the established state will include those connections that are induced by a Slowloris attacker. A slowloris attack will result in many sockets in being in the established state. High socket occupancy coupled with low data volumes and low system interrupt rates could allow us to identify a Slowloris attack.

From everything we have learned, we expect mixed attacks to exhibit the following characteristics (i) A large combined number of SYN and ACT sockets (ii) a relatively high interrupt rate (depending on the intensity of SYN floods), (iii) low volumes of sent data sent and (iv) few context switches.



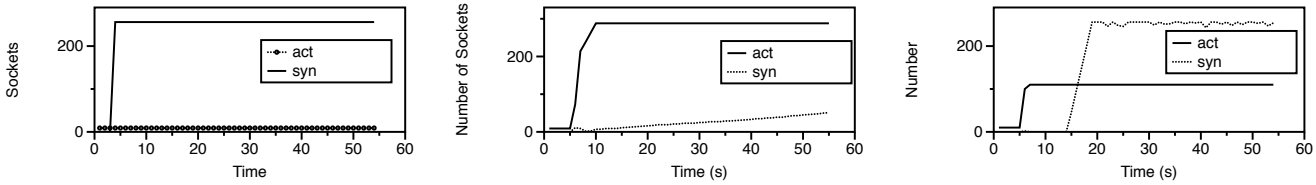
(a) SYN flood **(b)** Slowloris **(c)** Mixed attack

Fig. 1: Data received and sent data during (a) a full-scale SYN flood; (b) a full scale Slowloris attack (c) a mixed attack



(a) SYN flood **(b)** Slowloris **(c)** Mixed attack

Fig. 2: Interrupts and context switches during (a) a full-scale SYN flood (b) a full scale Slowloris attack (c) a mixed attack



(a) SYN flood **(b)** Slowloris **(c)** Mixed attack

Fig. 3: TCP Sockets during (a) a full-scale SYN flood; (b) a full scale Slowloris attack (c) a mixed attack

Discussion: The above features are readily obtained by reading certain hardware registers. However, we point out that these features are by no means exhaustive. By including additional features, the accuracy of detection of stealthy attacks can possibly increase significantly. Our objective here is to showcase the potential of the approach, rather than find an exhaustive set of such features.

Experimental validation of our feature set: Next, we seek to demonstrate experimentally, that our chosen feature set can lead to an effective detection approach.

We launched full scale SYN Flood and Slowloris attacks (characterized in table I) against our server to analyze the effectiveness and behaviors of our chosen features. Panels a,b of Figs 1 to 2 depict the results of those experiments. By combining the previously described DoS attacks in stealthy modes (note that each component by itself in such mixed attacks do not cause a performance degradation), we launched our mixed (stealthy) attack. The effects of the stealthy attack on the considered features, are shown in Figures 1c, 2c and 3c.

We analyze the behavior of these features, and this is the lynchpin of our detection approach.

Figs. 1a and 1b depict the volume of traffic during full fledged (or full scale) attacks while the same feature is presented in Fig. 1c for the mixed attack. As expected, SYN floods are characterized by a very high volume of traffic received and disproportionately low volume that is sent. Slowloris (fig 1b) is characterized by a spike in traffic volume (when the connections are established) but the volume of data tapers off. The key observation is that in all three cases (SYN Flood, Slowloris and the mixed attack) the data sent is always below 800 bytes. For full scale SYN flood, our experiments show that the data received is always above 70,000 bytes. However, for the mixed attack the volume of data is much lower.

Figs. 2a and 2b demonstrate the behaviors in terms of interrupts and context switches with full scale attacks. For SYN floods we see that interrupts consistently dominate context switches (every new connection request triggers an interrupt) while for Slowloris we only see a spike that correlates with network data and then, few context switches and interrupts. For the mixed attack (Fig 2c), we observe relatively high values for Interrupts (corresponding to SYN Flood) and few context switches. For both full scale SYN flood and mixed attacks, the number of Interrupts (in a short span of time) is greater than 200 and the number of context switches is less than 100.

Finally, Figs. 3a and 3b capture the behavior of TCP Sockets for full scale attacks while Fig. 3c does the same for mixed attacks. As expected, we see that a full scale SYN flood results in a large number of SYN sockets and Slowloris results in many established sockets. Mixed attacks on the other hand, yield a high number when the two are combined.

The experiments demonstrate that the behaviors of the key features that we choose are as expected. We leverage

	SYN Flood (SYN Packets/S)	Slowloris (Connections)
Full Scale Attack	100	500
Stealthy Attack	7	100
Detection Threshold	10	120

TABLE I: Experimental Parameters

Feature	High Threshold	Low Threshold
Context switches	600	100
Interrupts	200	N/A
Data Received	70,000	450
Data Sent	N/A	800
TCP SYN sockets	200	9
TCP ACT sockets	95	9

TABLE II: Algorithm Variable Description

Algorithm 1: Attack Classification

Data: Set of features S aggregated over 3 seconds
Result: Attack Classification
if *Any feature in S above its high threshold* **then**
  θ = All feautures above their high thresholds;
  $\Theta = S - \theta$;
  if *All features in* Θ *below their low thresholds* **then**
    OUTPUT "attack";
  end
  else
    OUTPUT "no attack" ;
  end
end

these behaviors to design an effective detection approach as will be described in the following section.

V. A framework for detecting stealthy DoS attacks: design and validation

In this section we leverage the take aways from the previous section to design our detection framework and experimentally demonstrate its effectiveness.

A. Design of our approach

Key Insights: As discussed in section IV, in order to detect both full scale attacks and mixed (stealthy) attacks we propose to use a combination of features. When under attack, some of these features exhibit high values while others tend have low values. In essence, since DoS targets the consumption of specific resources, those will be over utilized while certain other resources will be left under utilized. Legitimate web requests on the other hand, usually utilize (and thereby affect) a majority of the system, each having a part to play in servicing requests. This observation is the cornerstone of our detection framework.

Based on the understanding we gained with respect to our features in Section IV we set two thresholds for each feature viz., a high threshold and a low threshold. Whenever any of the features crosses it's respective high thresholds (a sign of DoS), our detection approach examines the other features to check if they are below their low thresholds. It is important to realize that this is fundamentally different from single threshold based approaches; the use of multiple features and two thresholds (low and high) help's reduce both false positives and false negatives.

	True Positives	False Positives
SYN Flood-Full	100	N/A
Slowloris-Full	95	NA
Mix - 1	100	N/A
Mix - 2	100	N/A
Mix - 3	100	N/A
Normal Traffic	N/A	20

TABLE III: Results with our detection approach

	True Positives	False Positives
SYN Flood-Full	100	N/A
Slowloris-Full	100	NA
Mix - 1	0	N/A
Mix - 2	0	N/A
Mix - 3	60	N/A
Normal Traffic	N/A	60

TABLE IV: Results with scalar threshold detection

Determining Optimal Thresholds: To determine it's high and low thresholds, we recall results from Section IV. As determined earlier, Slowloris (both full scale and stealthy) attacks result in (a) a high number of sockets in the ACT state, (b) few interrupts, (c) few context switches, (d) few SYN sockets and (e) low volume of data transfered. SYN floods have (i) high interrupt rates, (ii) high volume of data received (SYNs) and (iii) high number of sockets in SYN sate. From the results of our mixed attack we observe that together, they result in high number of interrupts (true for both SYN flood and Slowloris), and a high total number of sockets that are in ACT and SYN states. Thus, we must look for the crossing of high thresholds for these features.

We get our high threshold value for interrupts from the mixed attack empirically from Fig 2c (a more sophisticated machine learning approach can be potentially used but we leave this for future work). We obtain our threshold for high received data volume from Fig. 1a for full scale SYN floods since, high data volume is only a feature that is manifested with full scale SYN floods. High thresholds for sockets together in SYN and ACT state are both extracted from the results in Fig. 3c.

Low thresholds for ACT and SYN sockets (important in identifying full scale Slowloris and SYN Flood) are empirically obtained from Figs. 3b and 3a. The low threshold for data sent (characteristic of mixed attacks, full scale SYN flood as well as a full scale Slowloris) is obtained from the results in Fig 1c. Table II summarizes the values of these thresholds.

Detection Algorithm: Our detection algorithm (Algo 1) is executed once every 3 seconds. It checks if any of the considered features are above their high thresholds; if there are such features, it checks to see if all other features are below their low thresholds. If this is true then, the algorithm flags an attack. It is easy to verify that when any of the DoS attacks is in progress (full scale or

stealthy), there are certain resources that are heavily used (e.g., high received data volume), but inevitably, there are other parameters that indicate low usage of certain other resources (e.g., number of context switches). In other words, there is a serious imbalance in the way in which resources are utilized in a system. This in essence, is the lynchpin of the algorithm. By using two thresholds, the algorithm is effective in all cases (again, because any attack cannot saturate all resources).

B. Evaluation of our approach

Next, we conduct experiments on the Deter testbed to showcase the effectiveness of our detection framework. In order to evaluate how well our approach works in a real world setting, we need to generate realistic "normal" web traffic. To do so we used the extensive set of traffic traces detailed in [2] and available at [19]. The trace set contains network traffic collected at the edge router at a major university. We translated each incoming request (TCP packets with destination port set to 80) to a HTTP request for our server. We used a set of 10 traces. We consider multiple attack scenarios (full scale TCP, full scale Slowloris and 3 different mixed attacks). Our metrics of interest are the false positive and false negative rates. The 3 different kinds of mixed attacks represent different intensities of the individual attack. Mix-1 is the initial mixed attack described in Section IV from Table I. Mix-2 incorporates a higher intensity Slowloris attack and a lower intensity SYN flood attack (Slowloris:110 connections, SYN:7 Packets/sec). Mix-3 includes a higher intensity SYN flood attack but a lower intensity Slowloris attack (Slowloris:100 connections, SYN:8.5 Packets/sec).

We compare our detection framework with traditional detection approaches that use a single scalar feature to determine if an attack is under way. Tables I and II describe the set up. Tables III and IV present our results. As evident from the results, our detection approach outperforms traditional approaches with respect to detecting mixed, stealthy attacks. The stealthy attacks Mix-1 and Mix-2 are completely undetected by the traditional schemes employing single thresholds. Mix-3 (which has a higher intensity of SYN Floods) was detected 60% of the time. This is because the combined SYN messages from the flood and those from the Slowloris component sometimes crossed the scalar detection threshold. Traditional approaches also do poorly in classifying normal traffic. In our experiments, traditional approaches erroneously issued alerts 60% of the time with normal traffic; these false alerts typically occurred during periods of high activity (i.e., sudden spikes of high volume but normal traffic). In contrast, the false alerts with our approach only occurred 20% of the time.

VI. Conclusions

In this paper, we demonstrate that by intelligently combining a plurality of low intensity TCP-based DoS attacks, an attacker can evade traditional single scalar threshold based intrusion detection systems. We argue that multiple features need to be considered for efficiently detecting such stealthy attacks. Via extensive experiments, we identify such a set of features, and jointly examine them in a new simple, yet effective detection framework. We demonstrate, via extensive experiments, that our approach can detect stealthy attacks effectively unlike traditional approaches.

Acknowledgment: The effort described in this article was partially sponsored by the U.S. Army Research Laboratory Cyber Security Collaborative Research Alliance under Cooperative Agreement W911NF-13-2-0045. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation hereon.

References

[1] "IBM reat Intelligence QuX-Force Tharterly, 1Q 2015," 2015. [Online]. Available: http://public.dhe.ibm.com/common/ssi/ecm/wg/en/wgl03073usen/WGL03073USEN.PDF
[2] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the source," in *In Proc. 10th IEEE International Conference on Network Protocols*, 2002.
[3] T. M. Gil and M. Poletto, "Multops: A data-structure for bandwidth attack detection," in *Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*, ser. SSYM'01, 2001.
[4] "Intrusion detection FAQ: Statistical based approach to intrusion detection," https://www.sans.org/security-resources/idfaq/statistic_ids.php.
[5] W. Eddy, "TCP SYN flooding attacks and common mitigations," in *RFC 4987*, Aug 2007.
[6] "Slowloris HTTP DoS," http://ha.ckers.org/slowloris/.
[7] H. Wang, D. Zhang, and K. Shin, "Detecting SYN flooding attacks," in *In Proc. IEEE INFOCOM 2002.*, 2002.
[8] "Snort," https://www.snort.org.
[9] "The bro network security monitor," https://www.bro.org/.
[10] H. Wang, D. Zhang, and K. Shin, "Change-point monitoring for the detection of DoS attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 4, pp. 193–208, Oct 2004.
[11] V. Durcekova, L. Schwartz, and N. Shahmehri, "Sophisticated denial of service attacks aimed at application layer," in *Proc. ELEKTRO, 2012*, 2012.
[12] T. Yatagai, T. Isohara, and I. Sasase, "Detection of http-get flood attack based on analysis of page access behavior," in *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. PacRim 2007*, 2007.
[13] L. C. Giralte, C. Conde, I. M. de Diego, and E. Cabello, "Detecting denial of service by modelling web-server behaviour," *Computers & Electrical Engineering*, vol. 39, no. 7, pp. 2252 – 2262, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790612001292
[14] "DoS attacks get more complex–are networks prepared?" http://defensesystems.com/articles/2013/12/19/dosattacks-complexity.aspx?admgarea=DS.
[15] "What we learned from anonymous: DDoS is now 3DoS," https://devcentral.f5.com/articles/whatwelearnedfrom-anonymousddosisnow3dos.
[16] "The DETER project," http://deter-project.org.
[17] "Apache, http server project," https://httpd.apache.org/.
[18] "Kali linux," https://www.kali.org/.
[19] "D-WARD: DDoS network attack recognition and defense," http://www.lasr.cs.ucla.edu/ddos/.