DOI 10.12694/scpe.v20i2.1501
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 2, pp. 285–297. http://www.scpe.org 

# THE SLOW HTTP DISTRIBUTED DENIAL OF SERVICE ATTACK DETECTION IN CLOUD

A. DHANAPAL* AND P. NITHYANANDAM†

**Abstract.** Cloud computing became popular due to nature as it provides the flexibility to add or remove the resources on-demand basis. This also reduces the cost of investments for the enterprises significantly. The adoption of cloud computing is very high for enterprises running their online applications. The availability of online services is critical for businesses like financial services, e-commerce applications, etc. Though cloud provides availability, still these applications are having potential threats of going down due to the slow HTTP Distributed Denial of Service (DDoS) attack in the cloud. The slow HTTP attacks intention is to consume all the available server resources and make it unavailable to the real users. The slow HTTP DDoS attack comes with different formats such as slow HTTP headers attacks, slow HTTP body attacks and slow HTTP read attacks. Detecting the slow HTTP DDoS attacks in the cloud is very crucial to safeguard online cloud applications. This is a very interesting and challenging topic in DDoS as it mimics the slow network. This paper proposed a novel method to detect slow HTTP DDoS attacks in the cloud. The solution is implemented using the OpenStack cloud platform. The experiments conducted exhibits the accurate results on detecting the attacks at the early stages. The slowHTTPTest open source tool is used in this experiment to originate slow HTTP DDoS attacks.

**Key words:** DDoS, slowloris, RUDY attacks, slow HTTP attack, OpenStack, Cloud Security, Layer 7 Attack

**AMS subject classifications.** 68M14, 68M12

## 1. Introduction.

**1.1. Cloud computing and classifications.** Cloud computing helps enterprises to minimize the initial investments and operational cost [1] on the data centres. The enterprises running online applications are moving to the cloud. National Institute of Standards and Technology (NIST) defines the cloud computing [2] is a model for enabling convenient on-demand network access to a shared pool of configurable computing resources such as networks, servers, storage, applications and services that can be rapidly provisioned and released with minimal management effort or service provider interaction. The benefit of making the services available anywhere and anytime, adding or removing the resource based on the demands, pay per usage model are the most important factors for promoting cloud across the enterprises.

Cloud computing broadly classified as shown in Fig. 1.1.

The Service Delivery Model is based on the type of cloud service provided. This model sub-divided [3] as Software as a Service, Platform as a Service and Infrastructure as a Service. The Deployment Model is based on cloud deployment. This is further classified as [4] Public cloud, Private cloud and Hybrid cloud.

Cloud computing threats: Cloud computing has many potential threats and generally categorized as shown in Fig. 1.2.

**1.2. Distributed Denial of Service (DDoS) attacks .** The DDoS attacks are designed to disturb the service provided and make it unavailable to the real end-users. The DDoS attacks divided into [5] [6] following types:

- Volumetric-based attacks: The network bandwidth is targeted to bring down the services. Example: UDP flooding attacks.
- Protocol-based attacks: The resources of the server is consumed to bring the service down. Example: Ping of death.
- Application Layer attacks: The goal of this attack is to make application un-available to real users. Example: slow HTTP DDoS attack.

This paper focused on one of the application layer specific HTTP DDoS attack known as the slow HTTP DDoS attacks. This work proposed a solution to effectively detect such slow HTTP DDoS attacks in cloud computing.

*School of Computing Science and Engineering, VIT University, Chennai, India (Dhanapal.a2013@vit.ac.in, dhanapal.ang@gmail.com ).

†School of Computing Science and Engineering, VIT University, Chennai, India (Nithyanandam.P@vit.ac.in)

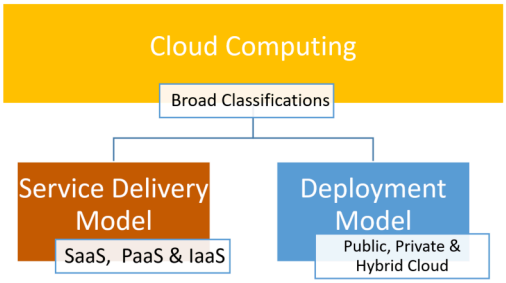


FIG. 1.1. *The cloud computing classifications*

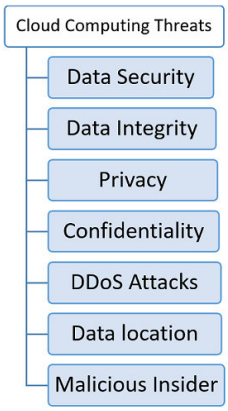


FIG. 1.2. *Cloud computing threats*

**1.3. The HTTP DDoS Attacks and Types.** The HTTP DDoS Attacks are application layer attacks in the system. These attacks aimed to make the online web services unavailable to the legitimate end users. The HTTP DDoS attacks type is visualized as in Fig. 1.3.

- HTTP flooding DDoS Attacks: The HTTP flooding attacks is application layer attacks and focused on flooding with an excessive request to the web server so that to overload and make the web server unable to process incoming requests. The service will be brought down eventually.
- Slow HTTP DDoS Attacks: This is another form of HTTP DDoS attacks that exploit the HTTP protocol behaviour and implementation of the application in a legitimate way. This attack consumes all the available resources in a slow manner. This slow attack aimed to make the web service unavailable to the real end-users.

**2. The slow HTTP DDoS attacks .** The slow HTTP attack to the web server can take places in three different formats. The types of slow HTTP attacks is given in Fig. 2.1.

The detailed explanation of each type is given below. following characteristics create the special needs and complexities associated with the HTTP flooding detection in the cloud environment.

- Slowloris Attacks (or) Slow HTTP Header Attacks [7]: The HTTP Get requests are usually send with valid HTTP Header. The web server processes the HTTP Get request upon receiving the complete header. The attackers make use of this HTTP protocol behaviour to carry out the DDoS attack on the web server. In this method, the attacker sends incomplete HTTP header to the web server and make the server to wait for the complete message. The attacker creates many such requests to the server until the server is unable to process any requests. The valid HTTP Header always ends with consecutive CR LF CR LF (ASCII Value 0d 0a 0d 0a) as shown in Fig. 2.2. During the attack, the HTTP header has only one CR LF (ASCII value 0d 0a) to indicate incomplete header to the web server as depicted in

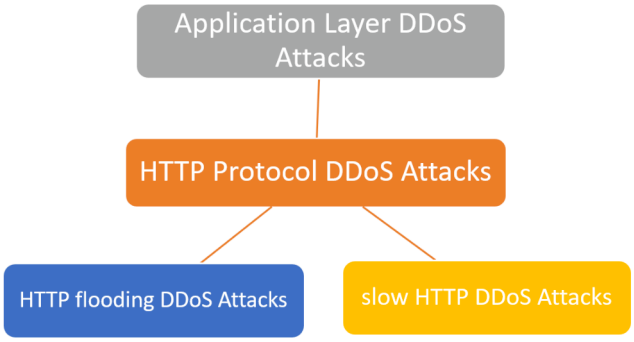


FIG. 1.3. *The HTTP DDoS attacks and types*

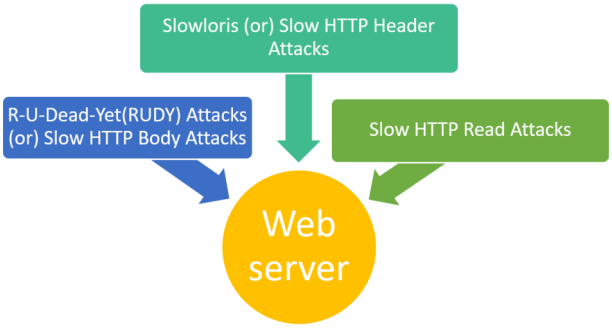


FIG. 2.1. *The slow HTTP DDoS attack classifications*

Fig. 2.3.

- RUDY Attacks (or) Slow HTTP Body Attacks [8]: RUDY or slow HTTP body attack is carried out using the HTTP Post requests. Unlike the slowloris attacks, this attack comes with a valid complete header in place. The length of the message has a bigger value. The attacker sends HTTP Post request with very small content size varies from one to few bytes with slow intervals of time. This makes the web server to wait for a long time to get the complete Post request. The attacker opens many connections with the server to consume all its resource so that the server will not be available to legitimate users. During the attack, the HTTP Post request contains an abnormal content-length field when compared to the normal request. The normal HTTP Post request and abnormal request are covered in Fig 2.4 and Fig. 2.5 respectively.
- Slow HTTP Read Attacks [9]: This is one another method of DDoS attack to the web server. In this model, the attacker sends the valid header and Get request to the web server, but it delays the read response from the server. The web client/browser at the attacker side frequently sends a message to the web server saying that it does not have enough space to receive the message so that to delay the server response. The TCP window size field is used during the 3-way handshaking method to exchange the window size between web client and server for agreeing upon the number of messages that they can handle. After establishing the connection web browser asks for more data, but it says not enough space to receive it. This makes the server to reserve the allocated resources for that transaction. The multiple transactions of this type being opened from attacker end to the victim web server to make web server unable to process any incoming requests. During the attack scenario, the web browser sends multiple TCP ZeroWindow messages to the web server as depicted in Fig. 2.6.

The rest of the paper is organized as follows: Section 3 discussed the related works. Section 4 explained

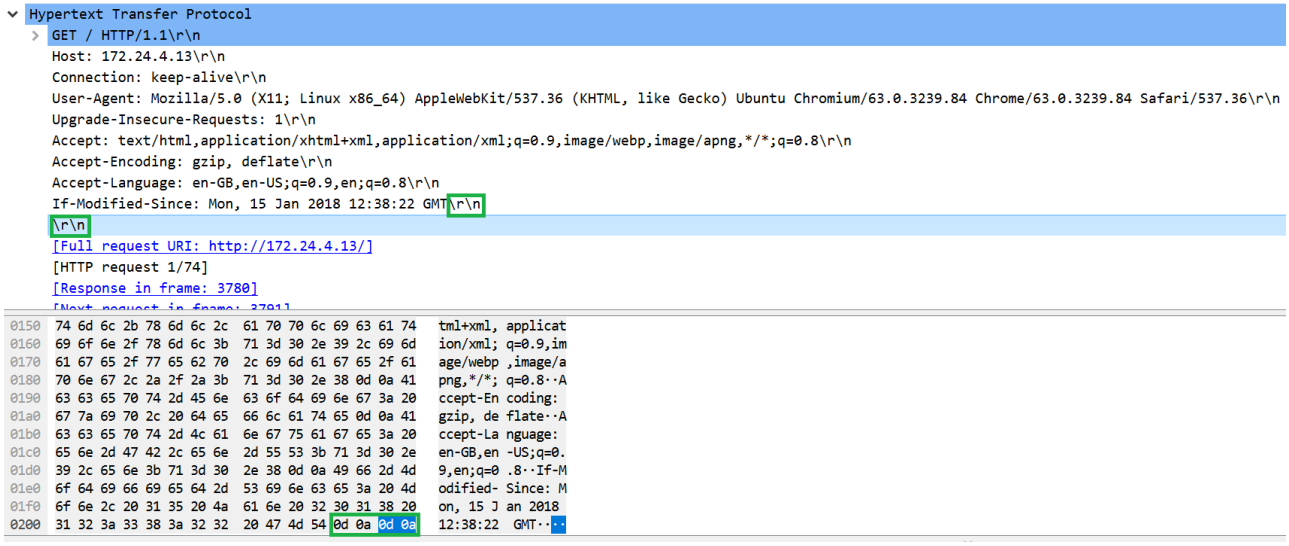


FIG. 2.2. *Normal HTTP Get Request*

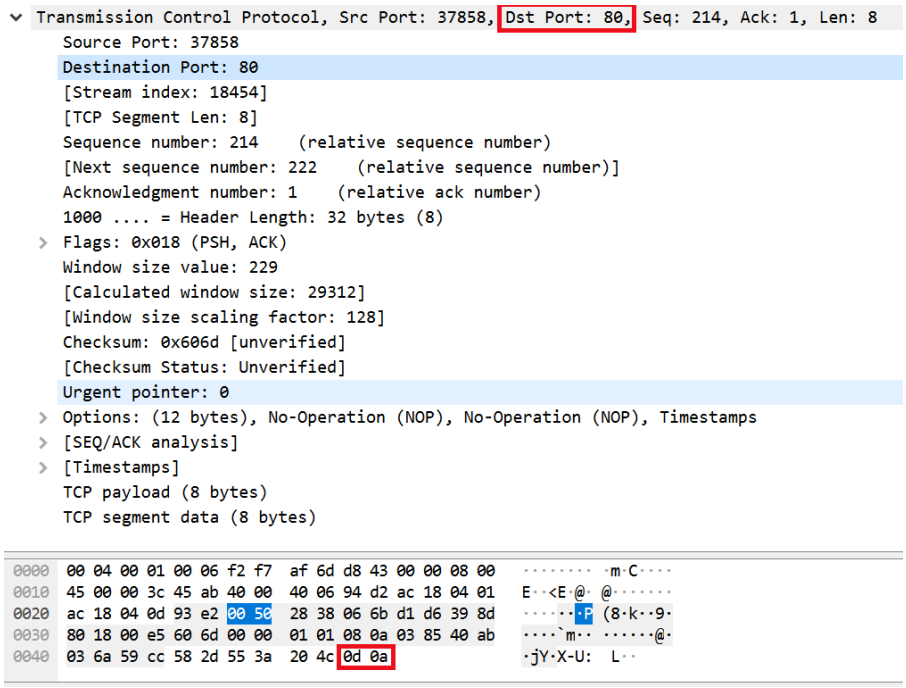


FIG. 2.3. *The HTTP header during slowloris attack*

the details of the proposed solution architecture. Section 5 captured the details on experimentation details and the results. The conclusion and future direction of research are covered in Sec. 6.

**3. Literature Review of the related works.** The authors carried out a literature review of a large number of works related to DDoS detection and this section limits the number of papers very relevant to the

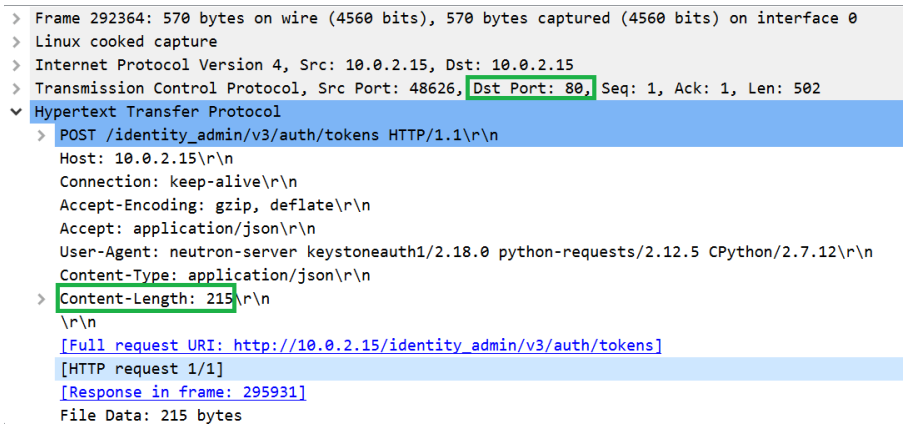


FIG. 2.4. *Normal HTTP POST Request*

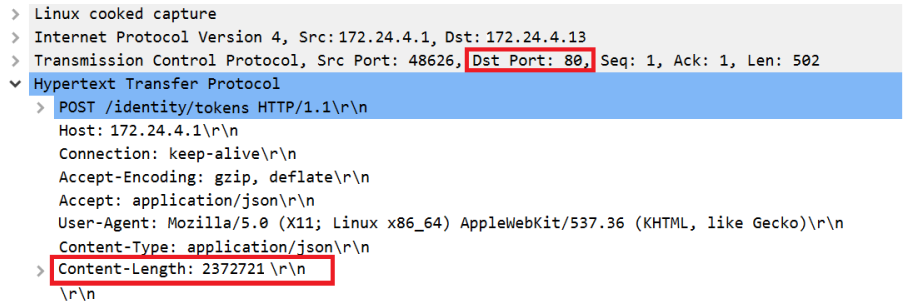


FIG. 2.5. *The HTTP Post request during RUDY Attack*

topic of interest slow HTTP DDoS attack detection and mitigation.

Tanishka Shorey et al. [10], This work studied and compared the DDoS attack tools such as slowloris, GoldenEye and Xerxes. The performance comparison is done based on the three parameters time to launch attacks successfully, Rate of Traffic and Size of packet. This gives insight into the potential tools used for HTTP DDoS attacks.

Clifford Kemp et al. [11], proposed a solution to detect slow HTTP Read DDoS attack detection using the machine learning techniques. The slowHTTPTest tool is used for generating the attack. This proposed technique works for the standalone environment. It has a scope for improvement to the cloud environment. Also, the solution has to be extended to address other slow HTTP DDoS attack types such as RUDY and slow HTTP Read attacks.

Kiwon Hong et al. [12], used the software-defined network method to implement their solution against the slow HTTP DDoS attacks. The traffic flow between the web server and the web browser via the switch is used by the defence application for analysis. This work classified the request as attacks, real slow connections and legitimate requests. This is a general solution proposed and need to be enhanced for cloud computing.

Shunsuke Tayama et al. [13], worked on the slow HTTP Read attack. The important parameter considered to detect the attack is the bandwidth rate. This work concluded that the browser having the connection limit equal to the processing capability of the web server and bandwidth greater than 500Kbps is good enough to carry out successful slow HTTP Read attacks. This work has gaps in analyzing the other types of slow HTTP DDoS attacks.

Aanshi Bhardwaj et al. [14], used the OpenStack cloud to analyze and evaluate the DDoS attacks. This work carried out to study DDoS attacks in the cloud using multiple DDoS attack tools. This paper requires an enhancement of providing the solution to detect the slow HTTP DDoS attacks in the cloud environment.

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 6493 | 0.737937893 | 172.24.4.13 | 172.24.4.1 | TCP | 68 | [TCP Keep-Alive] 80 → 57292 [ACK] Seq=1152 Ack=233 Win=29056 Le |
| 6494 | 0.737944873 | 172.24.4.1 | 172.24.4.13 | TCP | 68 | [TCP ZeroWindow] 57292 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6495 | 0.737949092 | 172.24.4.1 | 10.0.0.7 | TCP | 68 | [TCP ZeroWindow] 57292 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6496 | 0.737949439 | 172.24.4.1 | 10.0.0.7 | TCP | 68 | [TCP ZeroWindow] 57292 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6497 | 0.737952779 | 172.24.4.1 | 10.0.0.7 | TCP | 68 | [TCP ZeroWindow] 57292 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6498 | 0.738028853 | 10.0.0.7 | 172.24.4.1 | TCP | 68 | [TCP Keep-Alive] 80 → 57294 [ACK] Seq=1152 Ack=233 Win=29056 Le |
| 6499 | 0.738034426 | 10.0.0.7 | 172.24.4.1 | TCP | 68 | [TCP Keep-Alive] 80 → 57294 [ACK] Seq=1152 Ack=233 Win=29056 Le |
| 6500 | 0.738034794 | 10.0.0.7 | 172.24.4.1 | TCP | 68 | [TCP Keep-Alive] 80 → 57294 [ACK] Seq=1152 Ack=233 Win=29056 Le |
| 6501 | 0.738040589 | 172.24.4.13 | 172.24.4.1 | TCP | 68 | [TCP Keep-Alive] 80 → 57294 [ACK] Seq=1152 Ack=233 Win=29056 Le |
| 6502 | 0.738045309 | 172.24.4.1 | 172.24.4.13 | TCP | 68 | [TCP ZeroWindow] 57294 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6503 | 0.738048427 | 172.24.4.1 | 10.0.0.7 | TCP | 68 | [TCP ZeroWindow] 57294 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6504 | 0.738048687 | 172.24.4.1 | 10.0.0.7 | TCP | 68 | [TCP ZeroWindow] 57294 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6505 | 0.738051150 | 172.24.4.1 | 10.0.0.7 | TCP | 68 | [TCP ZeroWindow] 57294 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6506 | 0.738438999 | 172.24.4.1 | 172.24.4.13 | TCP | 68 | [TCP ZeroWindow] 57352 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6507 | 0.738452659 | 172.24.4.1 | 10.0.0.7 | TCP | 68 | [TCP ZeroWindow] 57352 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6508 | 0.738453488 | 172.24.4.1 | 10.0.0.7 | TCP | 68 | [TCP ZeroWindow] 57352 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |
| 6509 | 0.738458307 | 172.24.4.1 | 10.0.0.7 | TCP | 68 | [TCP ZeroWindow] 57352 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 |

FIG. 2.6. *The slow HTTP Read attack*

A. Dhanapal et al. [15], proposed a cloud testbed model using the OpenStack Infrastructure as a Service (IaaS) model. The different possible DDoS attack scenarios are explained in the work. This paper discussed only the framework details of the cloud environment.

Ghafar A. Jaafar et al. [16], reviewed the various work done to detect HTTP DDoS attacks in the time frame Jan 2014 to Dec 2018 and summarised the different parameters used in the solutions, attack detection levels, evalution methods, dataset used and performance matrix. This is a review paper helps to understand currently available solutions.

Trung V. Phan et al. [17], offered a hybrid machine learning model to improve the classifications and history-base filtering method combined togather to detect the DDoS attacks in SDN-based cloud environment. This is mainly used for the cloud environment build based on the SDN.

The proposed solution enhanced the OpenStack framework discussed in [15] and implemented the solution to detect the different type of slow HTTP DDoS attacks in the cloud environment. The slowHTTPTest tool is used to generate the variants of slow HTTP DDoS attack to the NGINX web server running in the OpenStack cloud.

The literature review of the numerous work studied and discussed shows that still we have the gaps to define the effective solution to detect the slow HTTP DDoS attack in the cloud.

**4. The Proposed Solution.** The proposed solution helps to detect the slow HTTP DDoS attack in the cloud environment and notifies to the administrator in case of an attack. The system architecture of the proposed defence model is depicted in Fig. 4.1.

The offered solution captures the multi-tenancy of cloud computing. The multiple tenants are located across the cloud service provider environment. Each tenant runs their own virtual machines (VM). The cloud has the following components

- Compute Controller: The cloud controller component contains the web interface to access and control the cloud environment, metering services used for calculating pay per usages, cloud monitoring services to look for the health of the systems, and most importantly the cloud scheduler to allocate the resources to the various tasks. This component is controlled and managed by the cloud service provider.
- Networking Node: This part provides networking services across the different components of the cloud environment. The important work of the networking node is to provide internal and external IP services and connectivity to the multiple tenants of the cloud services.
- Cloud Compute Cluster: The compute cluster is the place where each of the cloud users work is carried out. This component runs the various virtual machines of the tenants. Multiple tenants are kept in parallel in the cloud compute clusters. The cloud compute cluster has an entry point to access the virtual machines of the tenants. The proposed DDoS detection solution is implemented in this cloud

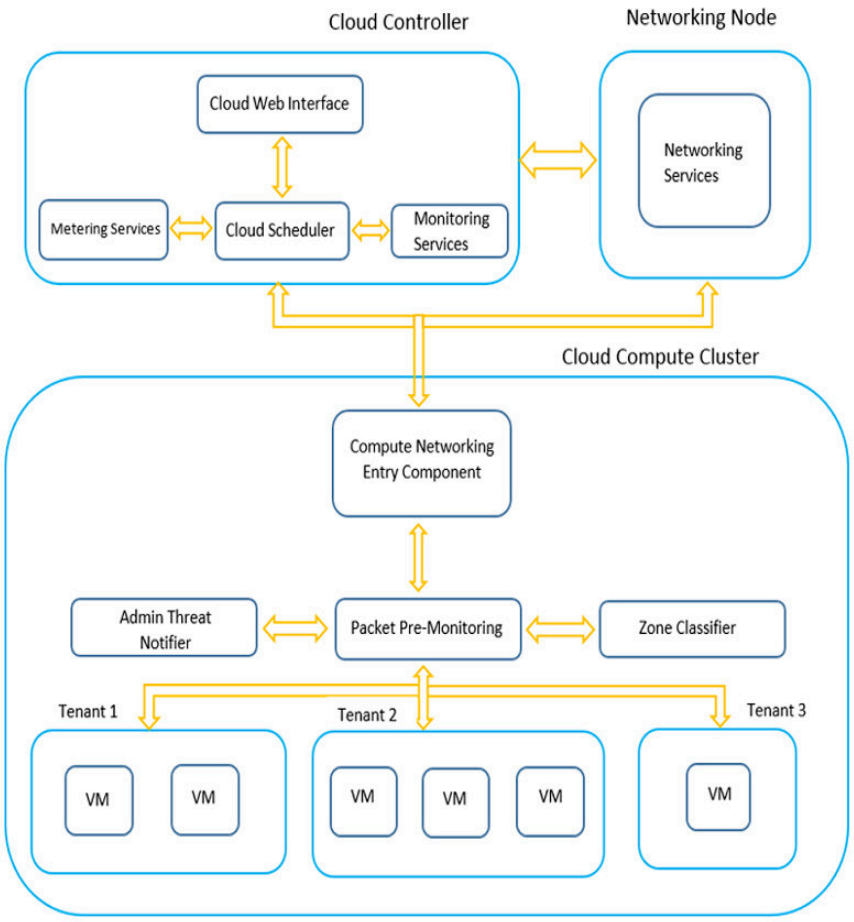


FIG. 4.1. *Proposed Solution System Architecture*

compute cluster. Every HTTP requests entering the cloud environment is monitored using packet pre-monitor module to identify the behaviour of the client and the same information is passed to the zone classifier module. The zone classifier module has two stages known as ALLOWED LIST zone and BLOCKED LIST zone. The clients are classified into either of the zones in any given point in time based on their behaviour.

Initially, all the clients requesting to the web server are kept in the ALLOWED LIST zone, and requests are forwarded to the web server for further processing. The behaviour of the client is continuously monitored by the classifier module. The client exhibits the suspecting activity are moved into BLOCKED LIST zone.

The client with the following behaviour is put into BLOCKED LIST zone.

- The average network delay to reach the client is calculated by means of sending 5 ping requests to the client. The average reply response time of those ping request is calculated and considered as network delay. This helps to identify the client mimicking the slow network.
- The client will be exhibiting the slow network. Whenever the classifier module gets the slow request continuously, five times the average network delay is compared with the client request interval. If the time between each of the HTTP request of the client exceeds more than five times of the calculated network delay, the client is treated as a bot machine and moved into BLOCKED LIST zone. The five times of the network delay selected by considering the processing required for the application.
- The client sends the HTTP Post request with the abnormal size of data more than three thousand of

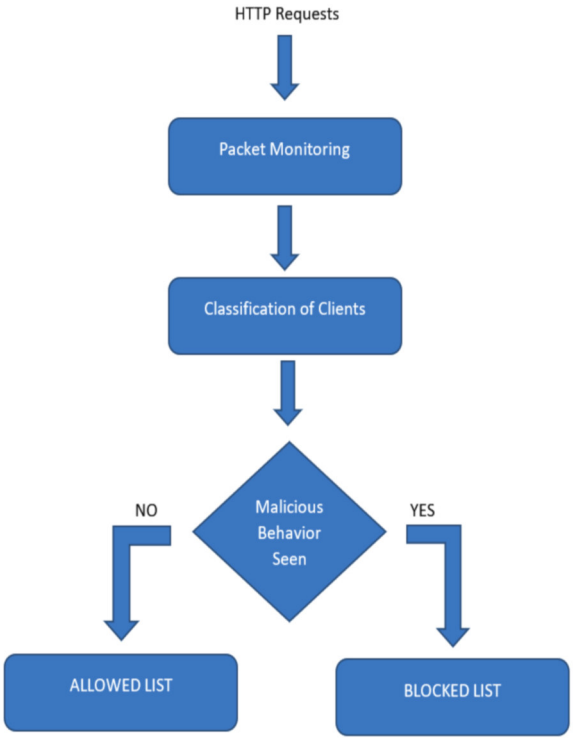


FIG. 4.2. *Flowchart of the proposed architecture*

the bytes and one or few bytes in the request with slow network behaviour is closely monitored. The same above calculated network delay is compared with the request interval time of the client. In case the request interval exceeds more than 5 times of average network delay, it is considered as a bot and moved into BLOCKED LIST zone.

- The client with a frequent request to the server saying that zero TCP Window size is monitored and moved into BLOCKED LIST if this behaviour is seen persistently.
- The client sends out an HTTP Post request with one or few bytes or Get request with an incomplete header in the time interval very close (above 80%) to the connection keep-alive time is treated as an attack and moved to BLOCKER LIST zone. For example, if the connection keep-alive time is 10 seconds and the client sends a request in the interval only after 8 seconds are suspected.

Whenever the classifier detects the abnormal activity of the client, it moves the client into BLOCKED LIST zone, and the appropriate information is sent to the admin threat notifier module. The admin threat notifier is responsible for indicating the cloud administrator about the client's suspicious activity and corresponding classifier zone movement. This helps the cloud administrator to take necessary action against the slow HTTP DDoS attacks.

The flowchart of the proposed architecture is captured in figure Fig. 4.2.

**4.1. Implementation Details.** The solution discussed is implemented using the OpenStack IaaS cloud software. The OpenStack is freely available on the internet for anyone. The OpenStack Network Topology look as shown in Fig. 4.3.

The multiple customers are placed on the OpenStack cloud and each of the customers is associated with separate IP networks. The OpenStack network topology consists of three private networks namely Orange-Private-Network, Green-Private-Network, Private-Network and one Public-Network. Each private network associated with one customer. Every virtual instance or nodes of the customer is assigned with an internal IP address in the given address range. The OpenStack cloud provides two IP addresses to each virtual node. The IP address

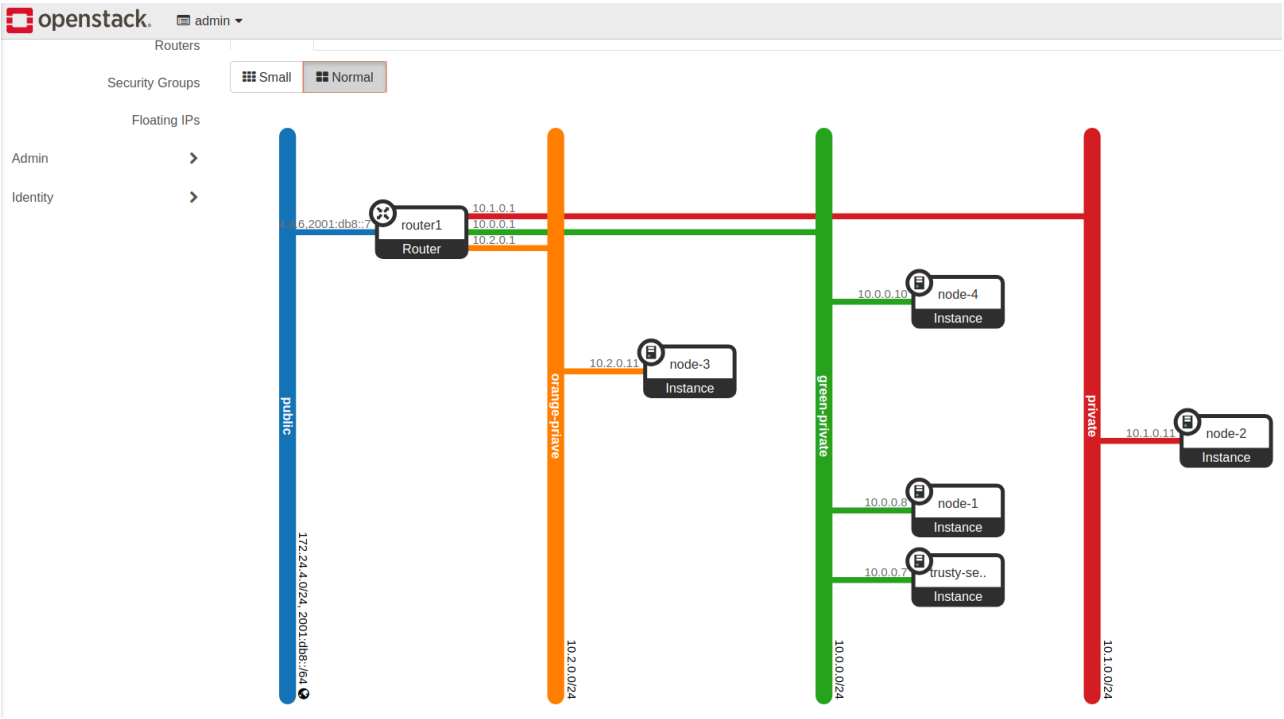


FIG. 4.3. *The OpenStack network topology*

TABLE 4.1
*The OpenStack Virtual Instance Details*

| Instance Name | Internal/Private IP address | External/Public IP address |
|---|---|---|
| Trusty-server | 10.0.0.7 | 172.24.4.13 |
| Node-1 | 10.0.0.8 | 172.24.4.4 |
| Node-2 | 10.1.0.11 | 172.24.4.11 |
| Node-3 | 10.2.0.11 | 172.24.4.14 |
| Node-4 | 10.0.0.10 | 172.24.4.12 |

starting with 10.x.x.x are known as private IP address and used for communication across the virtual nodes within the cloud environment. Another IP address is known as public or floating IP address. This floating IP address starts with 172.24.x.x and used for communicating with the outside world through the internet. The Public-Network has an IP address in this range to communicate to the external world.

The customer-1 on OpenStack runs only one virtual node known as node-3, and the customer is placed on the Orange-Private-Network with IP address range of 10.2.0.0/24.

The customer-2 is placed on the Green-Private-Network with IP address range of 10.0.0.0/24. This customer runs three virtual nodes namely node-1, node-4 and trusty-server. The trusty-server node runs the NGINX web server in the OpenStack cloud.

Similarly, the customer-3 placed on the Private-Network has IP network 10.1.0.0/24. The node-2 is virtual instance launched for customer-3.

Each of the virtual nodes internal, as well as external IP address details are captured in Table 4.1.

The NGINX web server is running over the trusty-server of the customer-2. The web pages are accessed using the URL http://172.24.4.13/WebPage.html. The opensource tool named slowHTTPTest is used to generate three different types of slow HTTP DDoS attacks against the NGINX web server running in the cloud.

**5. Experimentations Results and Discussions.** Slow HTTP DDoS test carried out using slowHTTP-Test tool with following options:

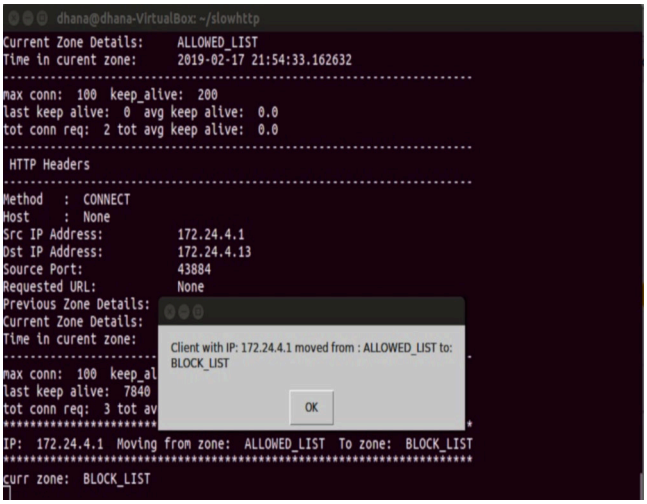


FIG. 5.1. *The detection of slow HTTP DDoS attack*

c is number of connections.
H indicates the slowloris Header attack mode, -B for slow HTTP Body attack mode, -X slow HTTP read attack mode.
g says generate statistics and output with file -o file name.
i time interval between requests.
r number of connections per seconds.
t type of request (GET/POST)
u target URL
l test length in seconds.

The slow HTTP Header DDoS attacks, as well as slow HTTP Read attacks are launched against the web server URL http://172.24.4.13/products.html

The command used for slow HTTP Header attack:

slowhttptest -c 10000 -H -g -o slowhttpheader -i 10 -r 500 -t GET -u http://172.24.4.13/products.html -l 600

The slow HTTP read attack done using:

slowhttptest -c 10000 -X -g -o slowhttpread -i 10 -r 500 -t GET -u http://172.24.4.13/products.html -l 600

The slow HTTP Body attacks are carried out for the URL http://172.24.4.13/register.html

The command used for slow Body attack is:

slowhttptest -c 10000 -B -g -o slowhttpbody -i 10 -r 50 -t POST -u http://172.24.4.13/register.html -l 600

The proposed DDoS detection model efficiently identify all possible slow HTTP DDoS attacks and classify the attacking client to BLOCKED LIST zone. Fig. 5.1 shows that notification of the slow HTTP DDoS attack launched from client 172.24.4.1 is detected and moved to BLOCKED LIST zone from ALLOWED LIST zone.

The common parameters used for generating different slow HTTP DDoS attacks by the slowHTTPTest tool are:
The total number of connections: 10000;
The content-length of the header: 4096 bytes;
Timeout for probe connection: 5 seconds.

The performance results of the slow HTTP Header attack captured in Fig. 5.2.

The service available line Fig. 5.2 indicates the availability of the web services to the slowHTTPTest tool. The implemented system detected the attack when the connection request crossed above 4000, and the client is

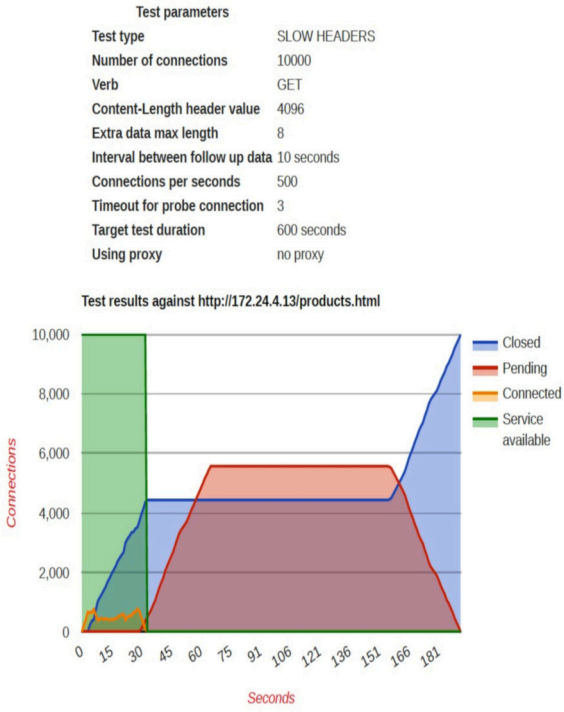


FIG. 5.2. *The slow HTTP Header attacks result*

moved into BLOCKED ZONE list. Any further request from the client is blocked, so the attacker thinks that the web server is down, but the requests from the attacker are not processed. The pending request from the attacker is growing in the very high rate as shown in Fig. 5.2.

The slow HTTP Body attack results are shown in Fig. 5.3, and the system detected the attack during when connection request reached $\tilde{1}700$. After this point of time, any request from the attacker is not serviced by the web server. The pending requests are increasing very sharply as displayed in Fig. 5.3.

The performance result of the slow HTTP Read attack is depicted in Fig. 5.4. Figure 5.4 explains that read attacks are generating frequent zero window size to receive the response from the web server. The same behaviour has been detected by the system and moving such clients into the BLOCKED LIST zone. The incoming requests from those clients are not given to the web server for any services. In Fig. 5.4, the slow HTTP Read DDoS attacks are detected when the number of connection reached $\tilde{1}000$. The detection happened sooner because of the clients continuous malfunctioning of reading the response from the server. The pending requests moved to peak value suddenly $\tilde{8}100$, due to not serving any request from the client. The total number of connection accepted throughout the experiment is $\tilde{2}000$. The efficient and early detection of slow HTTP Read attacks help the availability of the web server to the legitimate requests.

**6. The conclusion and future directions.** The authors discussed cloud computing and classification of cloud computing in details. The DDoS attacks and its types, the important application layer attack of the HTTP protocol is explained. The variation of HTTP DDoS attacks also captured. The slow HTTP DDoS attacks and various forms of slow HTTP attacks are described appropriately. The literature survey of related work and their gaps in addressing the slow HTTP DDoS attacks in the cloud is discussed. The proposed solution to detect such attacks in the cloud environment defined. The offered solution has been implemented and integrated into the OpenStack IaaS software. The details of the OpenStack cloud environment have been explained. The different types of slow HTTP DDoS attacks are carried out to the web server in the OpenStack using the slowHTTPTest tool. The performance and the result of the implemented solution are captured and

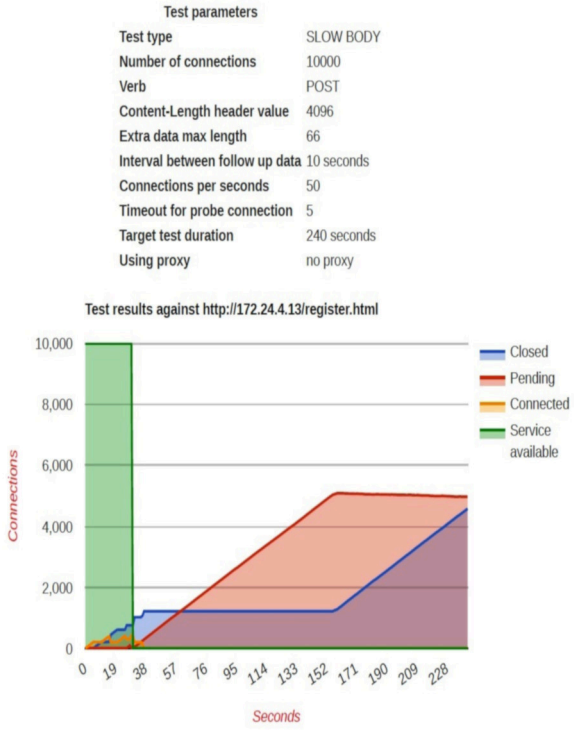


FIG. 5.3. *The slow HTTP Body attacks result*

discussed in detail.

The future enhancement of this work is to implement a model to detect, mitigate and prevent any form of the slow HTTP DDoS attack in the cloud environment. Evaluate the performance of the same against the multiple tools available on the internet for generating such slow HTTP DDoS attacks.

## REFERENCES

[1] SALESFORCE, *https://www.salesforce.com/uk/blog/2015/11/why-move-to-the-cloud-10-benefits-of-cloud-computing.html*, 09-Jan-2019.
[2] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST), *https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp*, 09-Jan-2019.
[3] WHATISCLOUD.COM, *http://whatiscloud.com/cloud_delivery_models/index*, 09-Jan-2019.
[4] WHATISCLOUD.COM, *http://whatiscloud.com/cloud_deployment_models/index*, 09-Jan-2019.
[5] RADWARE, *https://security.radware.com/ddos-knowledge-center/ddospedia/dos-attack/*, 09-Jan-2019.
[6] IMPERVA INC., *https://www.incapsula.com/blog/top-10-cloud-security-concerns.html*, 09-Jan-2019.
[7] CLOUDFARE, LU-*https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris/*, 09-Jan-2019.
[8] CLOUDFARE, *https://www.cloudflare.com/learning/ddos/ddos-attack-tools/r-u-dead-yet-rudy/*, 09-Jan-2019.
[9] CLOUDFARE, *https://www.cloudflare.com/learning/ddos/ddos-low-and-slow-attack/* , 09-Jan-2019.
[10] TANISHKA SHOREY, DEEPTHI SUBBAIAH, ASHWIN GOYAL, ANURAAG SAKXENA, ALEKHA KUMAR MISHRA, *Performance Comparison and Analysis of Slowloris, GoldenEye and Xerxes DDoS Attack Tools*, 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), DOI: 10.1109/ICACCI.2018.8554590, (2018).
[11] CLIFFORD KEMP, CHAD CALVERT, TAGHI M. KHOSHGOFTAAR, *Utilizing Netflow Data to Detect Slow Read Attacks*, 2018 IEEE International Conference on Information Reuse and Integration (IRI), DOI: http://doi.ieeecomputersociety.org/10.1109/IRI.2018.00023, (2018), pp. 108–116.
[12] KIWON HONG, YOUNJUN KIM, HYUNGOO CHOI, AND JINWOO PARK , *SDN-Assisted Slow HTTP DDoS Attack Defense Method* , IEEE Communications Letters, Vol. 22, Iss. 4 , DOI: 10.1109/LCOMM.2017.2766636, (2018), pp. 688–691.
[13] SHUNSUKE TAYAMA, AND HIDEMA TANAKA , *Analysis of Slow Read DoS Attack and Communication Environment*, Mobile and Wireless Technologies 2017, Lecture Notes in Electrical Engineering 425, DOI 10.1007/978-981-10-5281-1_38, (2018), pp. 1718–1724.

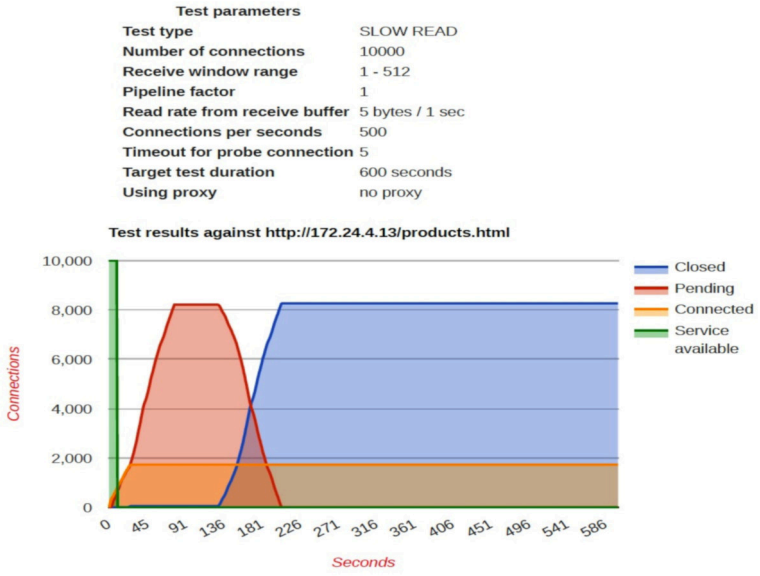


FIG. 5.4. *The slow HTTP Read attacks result*

[14] AANSHI BHARDWAJ, ATUL SHARMA, VEENU MANGAT, KRISHAN KUMAR AND RENU VIG , *Experimental Analysis of DDoS Attacks on OpenStack Cloud Platform* , Proceedings of 2nd International Conference on Communication, Computing and Networking, Lecture Notes in Networks and Systems 46, (2019)

[15] A.DHANAPAL, AND P. NITHYANANDAM , *An OpenStack based cloud testbed framework for evaluating HTTP flooding attacks*, Wireless Networks - Springer, DOI: 10.1007/s11276-019-01937-4, (2019), pp. 570–575

[16] GHAFAR A. JAAFAR, SHAHIDAN M. ABDULLAH, AND SAIFULADLI ISMAIL, *Review of Recent Detection Methods for HTTP DDoS Attack*, Journal of Computer Networks and Communications - Hindawi, DOI: https://doi.org/10.1155/2019/1283472, (2019), Volume 2019, Article ID 1283472, 10 pages

[17] TRUNG V. PHAN, AND MINHO PARK, *Efficient Distributed Denial-of-Service Attack Defense in SDN-Based Cloud*, IEEE Access, DOI: 10.1109/ACCESS.2019.2896783, (2019)

*Edited by:* Anand Nayyar
*Received:* Mar 7, 2019
*Accepted:* Apr 4, 2019