Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework

Opeyemi Osanaiye [a,b], Kim-Kwang Raymond Choo [b,c,*], Mqhele Dlodlo [a]

[a] *Department of Electrical Engineering, University of Cape Town, South Africa*
[b] *Information Assurance Research Group, University of South Australia, South Australia, Australia*
[c] *INTERPOL Global Complex for Innovation, Singapore*

ARTICLE INFO

Article history:
Received 23 September 2015
Received in revised form
14 December 2015
Accepted 11 January 2016
Available online 18 January 2016

Keywords:
Anomaly-based
Change point detection
Cloud distributed denial of service (DDoS) attacks
Cloud distributed denial of service (DDoS) survey
Distributed denial of service (DDoS) attack taxonomy
Flooding attacks

ABSTRACT

Despite the increasing popularity of cloud services, ensuring the security and availability of data, resources and services remains an ongoing research challenge. Distributed denial of service (DDoS) attacks are not a new threat, but remain a major security challenge and are a topic of ongoing research interest. Mitigating DDoS attack in cloud presents a new dimension to solutions proffered in traditional computing due to its architecture and features. This paper reviews 96 publications on DDoS attack and defense approaches in cloud computing published between January 2009 and December 2015, and discusses existing research trends. A taxonomy and a conceptual cloud DDoS mitigation framework based on change point detection are presented. Future research directions are also outlined.

Contents

[*] Corresponding author at: Information Assurance Research Group, University of South Australia, South Australia, Australia.
E-mail address: raymond.choo@fulbrightmail.org (K.-K.R. Choo).

http://dx.doi.org/10.1016/j.jnca.2016.01.001
1084-8045/© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Cloud computing has become a convenient way of accessing services, resources and applications over the internet. This model has shifted the focus of industries and organizations away from the deployment and day-to-day running of their IT facilities by providing an on-demand, self-service, and pay-as-you go business model. Cloud computing has continued to increase in popularity in recent times. The National Institute of Standard and Technology (NIST) defines the essential characteristics of cloud computing as on-demand self-service, resource pooling, rapid elasticity and measured service (Mell and Grance, 2011). The service model can be broadly categorized into Software-as-a-service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS), and it can be deployed as either a private, public, community or hybrid cloud (Tsai et al., 2010). While cloud computing provides various benefits to users, there are also underlying security and privacy risks (Khorshed et al., 2012; Christof et al., 2009; Gonzalez et al., 2012; Wayne, 2011; Quick and Choo, 2014; Chonka et al., 2011). For example, multi-tenancy, resource pooling and shareability features can be exploited by cybercriminals and anyone with a malicious intent. This is to the detriment of both the cloud users and providers to deplete resources which result in denial of services.

In the current Internet-connected society, there is an expectation that cloud and other Internet services are always available and any downtime can result in user dissatisfaction. For example, telecommunication industries in some countries require cloud providers to meet the five 9's availability requirements (Hormati et al., 2014) (i.e. cloud providers are allowed only five minutes of downtime per year). Unavailability in cloud can be a result of a number of factors, such as failure of cloud infrastructural component or software application and distributed denial of service (DDoS) attacks directed towards the cloud system. For example, cybercriminal groups such as 'Vickingdom2015' had reportedly brought down cloud services (SC Magazine, 2015), resulting in financial loss to both cloud users and providers.

In this paper, we survey common DDoS attacks targeting cloud computing, and categorize such attacks into application-bug level and infrastructural level attacks. We discuss the various tools that could be used to conduct or facilitate DDoS attacks, as well as reviewing mitigation strategies (e.g. how are DDoS mitigation strategies for cloud computing different from those designed for traditional computing). Two key differences in DDoS mitigation strategy between cloud computing and traditional computing identified by Wang et al. (2015) are as follows:

(i) The cloud provider is in control of the network and computational resources, rather than the user in the case of a traditional DDoS mitigation strategy.
(ii) Network infrastructure and resources are shared by users in cloud, which result in a reliability network segregation requirement (differing from traditional computing).

This paper reviews 96 publications between January 2009 and December 2015 on ScienceDirect, Springer, IEEE Xplore, Google Scholar and ACM digital Library using keywords such as "DDoS in Cloud", "Detecting DDoS in Cloud Computing", "Cloud Availability", and "Cloud Computing Security". A small number of surveys have been published on the topic, but our survey differs from previous surveys in the following ways:

(i) In Wong and Tan (2014), for example, the focus is on DDoS attacks targeting cloud infrastructure and applications, whilst we focus on both attacks and mitigation strategies.
(ii) The surveys in both Prabadevi and Jeyanthi (2014) and Darwish et al. (2013) (conference papers) are of a limited scope.

The rest of the paper is organized as follows. Section 2 discusses DDoS attacks and presents an attack taxonomy. Section 3 describes existing DDoS mitigation techniques and its taxonomy. Sections 4 and 5 present a general discussion and a conceptual framework for cloud DDoS defense, respectively. Finally, Section 6 concludes the paper and provides suggestions for future research.

2. DDoS attacks

As remarked by Bruce Schneier, "the only secure computer is one that's turned off, locked in a safe, and buried 20 feet down in a secret location – and I'm not completely confident of that one either" (Liebeskind, 2007). Ensuring the security of information and communications technologies (ICT) is a continuous rat race between the attackers and the defenders. This has attracted the attention of security experts both in academia and industry. Due to advances in technology and tools for launching this attack, the defenses proffered are not static; therefore, defenders need to stay up-to-date on the most recent attack trends and the state-of-art defenses. As opposed to other network security attacks that seek to exfiltrate or alter information, DDoS attack is perpetrated by one or more compromised systems controlled by an attacker to flood a predetermined target using series of malformed or malicious packets that overwhelm the allocated resources. The consequences of a successful attack will result in the unavailability of cloud services (Choi et al., 2013). A recent survey (Zargar et al., 2013) identified revenge, extortion, political issues, proficiency testing by cybercriminals and competition between cloud providers as common motivations of DDoS attacks – see Fig. 1. Anwar and Malik (2014) highlight the consequences of DDoS attacks against a cloud data center.

2.1. DDoS attack in cloud computing

Cloud security deployment policy is guided by the confidentiality, integrity and availability (CIA) triad model.

In its simplest form, DDoS can be conducted using compromised vulnerable nodes on the internet (also known as zombie

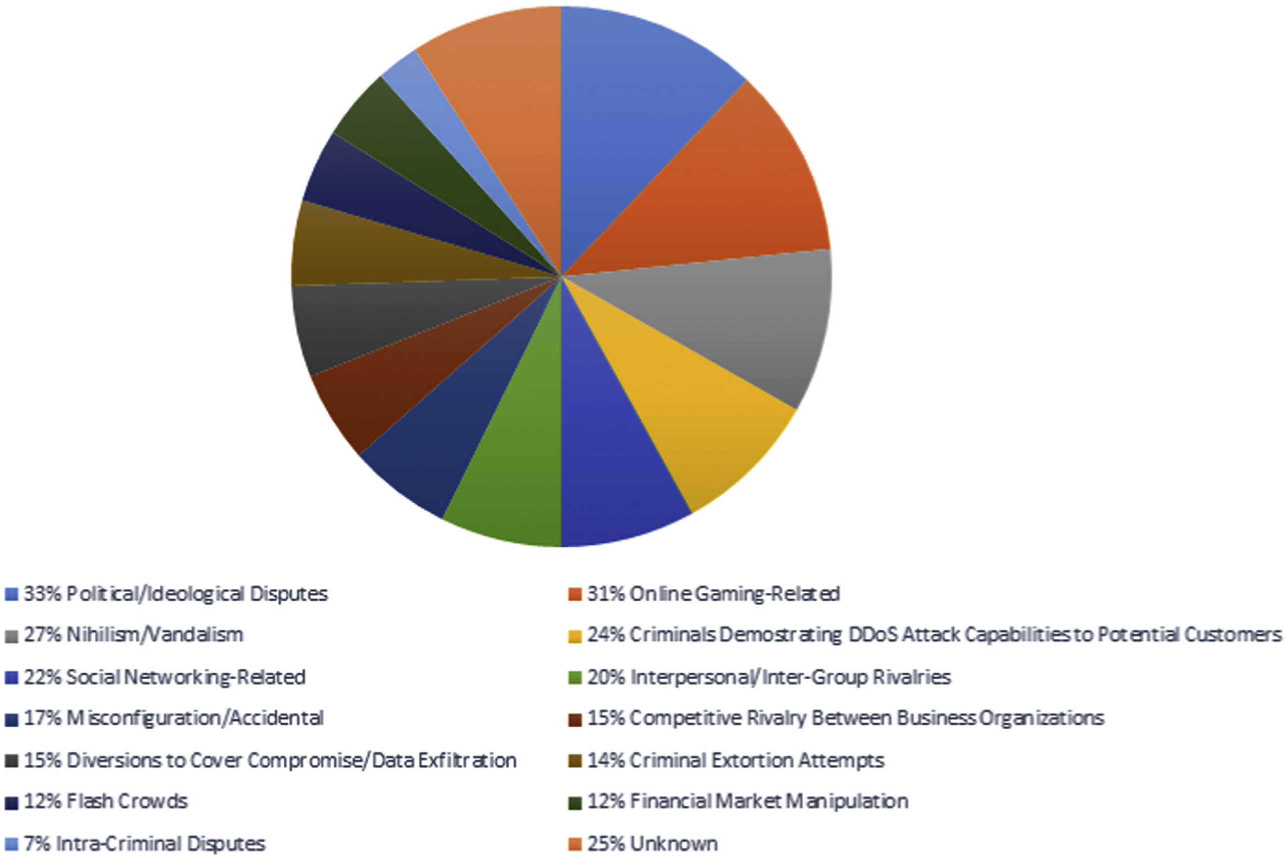

Fig. 1. Motivations behind DDoS attacks (adapted from Online The Truth about DDoS Attacks (2013)).

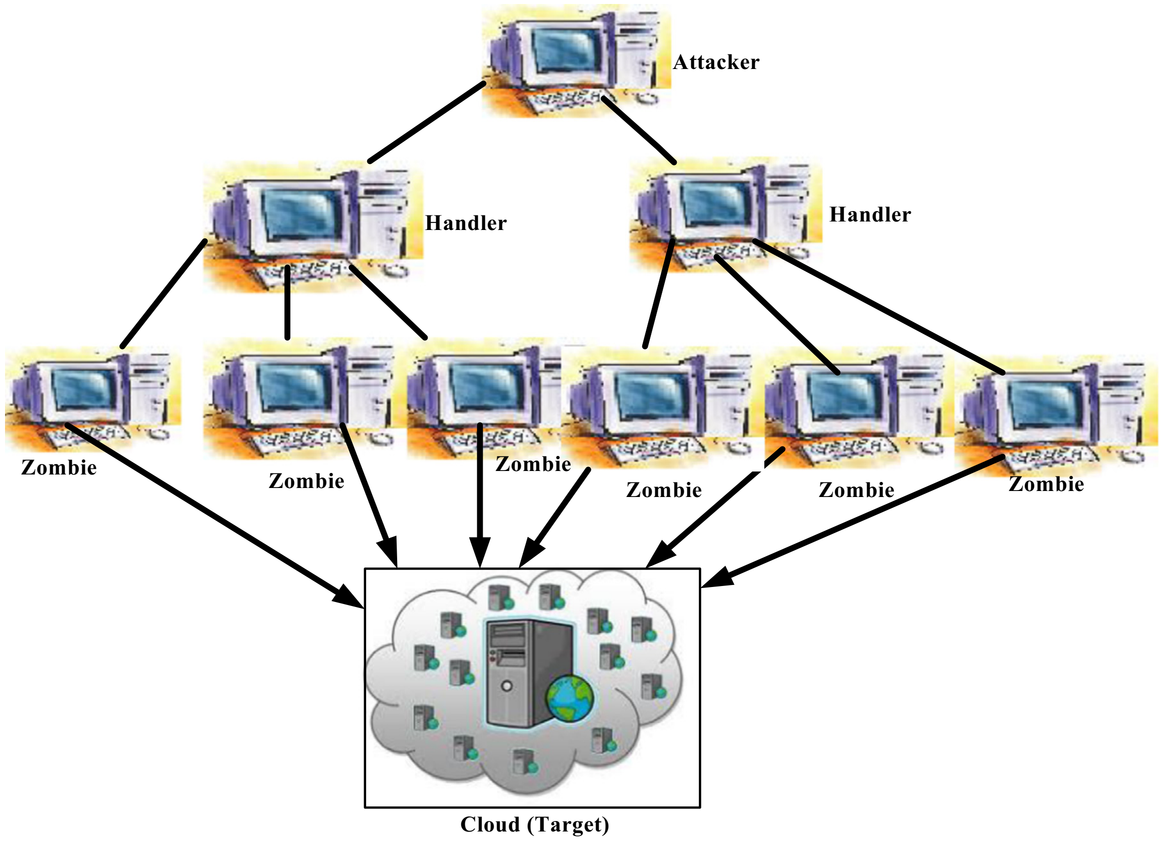

Fig. 2. DDoS attack in cloud (adapted from Osanaiye (2015)).

computers) (Deshmukh and Devadkar, 2015) – see Fig. 2. Upon receiving malformed packets, the targeted system may not know how to handle such packets and it eventually freezes or reboots (Mishra et al., 2011). When this occurs, the cloud user will be denied access to the respective cloud services and resources.

Several cases of such DDoS attacks have been reported in recent months (e.g. Swathi), and can take different forms, targeting different cloud components. Merlo et al. (2014) explain how DDoS attacks can be carried out on a cellular network which use properly crafted SIM-less device. The attacks can result in service degradation in universal mobile telecommunications system (UMTS) network and disrupt the mobile network coverage. Ficco and Rak (2015) describe a low rate DDoS attack that affects the pricing model of the cloud by evading early detection to incur cost. A similar type of attack, fraudulent resource consumption (FRC) that could exploit the utility and pricing model (pay-as-you-go) of cloud service has been suggested in Idziorek et al. (2013). A new and subtle denial of service attack on the cloud datacenter infrastructure has also been reported. Palmieri et al. (2013, 2014) describe this attack that exploits computing resources to waste energy and increase costs. In the worst case scenario, it achieves denial of service due to power outage having exhausted the power budget. A similar type of DDoS attacks against cloud computing infrastructures was presented in Ficco and Palmieri (2015), which affects both the quality of service delivered and the energy consumption. DDoS attacks on web service hosted in cloud using HTTP and XML were described in Chonka and Abawajy (2012).

Resources in the form of bandwidth, compute, application and infrastructure are the target of DDoS attack to alter service models (e.g. pricing model, business model and security model) to the detriment of both cloud providers and users.

2.2. *DDoS attack taxonomy*

There had been limited attempts to draw up a taxonomy for DDoS attacks in the cloud. Deshmukh and Devadkar (2015) grouped DDoS attacks into bandwidth and resource depletion. In Cha and Kim (2011), DDoS attacks targeting cloud web services were categorized into oversized payload, coercive parsing and flooding attacks, while (Wong and Tan, 2014; Bhuyan et al., 2015) categorized DDoS attacks into infrastructural level (OSI Layers 3 and 4) attacks and application level (OSI Layer 7) attacks. In this paper, we categorized DDoS attacks into application-bug level attacks and infrastructural level attacks, similar to the approach undertaken in Beitollahi and Deconinck (2012). We argue that classification based on a layered structure will simplify the reader's understanding of the attack process – see Fig. 3.

2.3. *Application-bug level DDoS*

In carrying out application-bug level attacks, attackers exploit system vulnerabilities or weaknesses to render cloud resources unavailable for users. Among the common attack vectors are protocol vulnerability, system weakness, outdated patches and misconfiguration. For example, vulnerabilities in protocol used by target applications can be exploited by attackers, by sending specially crafted packets to overload the application thereby crashing it. Dantas et al. (2014) discussed two types of such attacks, namely: HTTP PRAGMA and HTTP POST attacks. Beitollahi and Deconinck (2012) also described the ping-of-death attack, which uses a ping packet size of 65,535 bytes. The latter exceeds the maximum IPv4 packet size. When most modern operating systems try to handle such packets, they generally freeze, crash or reboot due to buffer overflow.

2.4. *Infrastructural level DDoS*

Infrastructure attacks (also known as flooding attacks) target cloud components, such as storage, network bandwidth, CPU circles and TCP buffers, to make them unavailable to legitimate cloud users. In infrastructural level DDoS attacks, the attackers only need the IP address of the target without the need to exploit any vulnerability. DDoS flooding attack can be carried out in two different forms, namely a direct attack and a reflector attack.

2.4.1. *Direct attack*

A direct attack involves the use of compromised victim hosts/ zombie computers to send massive malicious packets aiming to overwhelm the target system by consuming all available resources, resulting in the system being unavailable to legitimate users. Such attacks can be further classified into network layer DDoS and application layer DDoS attacks.

2.4.1.1. Network layer DDoS attack. In carrying out a network layer DDoS attack, research has shown that protocols that exist in the network and transport layers can be used to flood the target host (Zargar et al., 2013). Examples of these type of attacks are TCP SYN flood, UDP flood, and ICMP flood.

2.4.1.1.1. TCP SYN flooding attack. Transmission Control Protocol (TCP) is a connection oriented protocol that exists on the transport layer of the TCP/IP model stack. The connection oriented feature is derived from the three-way handshaking established prior to packet transmission between hosts. During the connection process, SYN message is sent by the connecting host and is acknowledged by the remote host by sending a SYN+ACK message. To complete the handshaking process, the connecting host responds with a final ACK

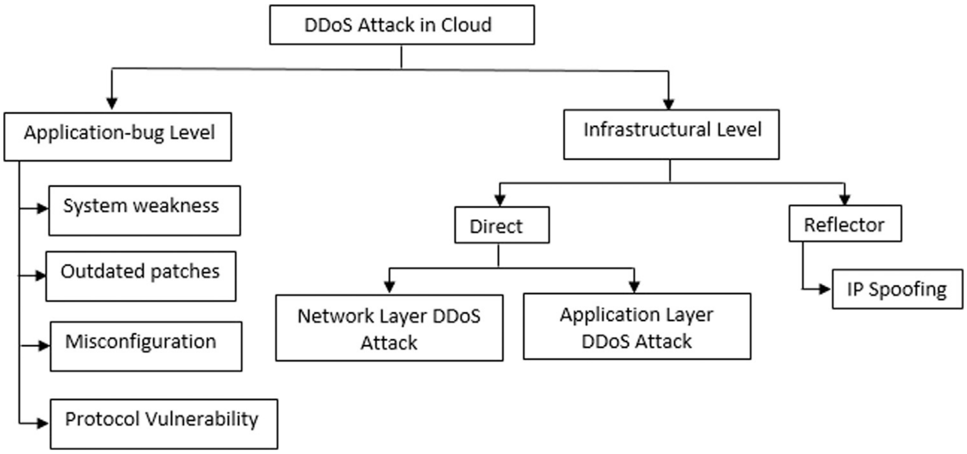

Fig. 3. DDoS attack taxonomy in cloud.

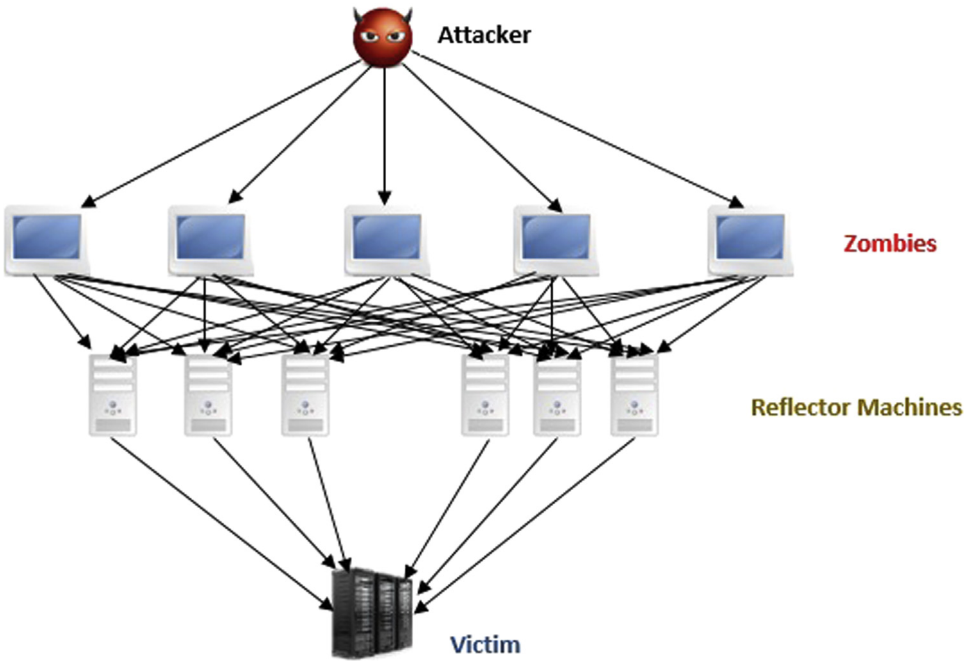

Fig. 4. DDoS reflector attack (adapted from Beitollahi and Deconinck (2012)).

and the connection is established between the two hosts. Attackers have exploited this connection feature by initiating half-opened connection, which exhausts the kernel memory by creating too many transmission block allocations (Wong and Tan, 2014). This can be accomplished by infiltrating vulnerable nodes on the internet to carry out a coordinated attack. DDoS attacks using TCP SYN flooding can also be facilitated using spoofed IP addresses. During a spoofed attack, the final ACK required to complete the connection process will not be received, as the host whose IP address was spoofed will respond with a RST flag or the host might not exist. Cha and Kim (2011) reported a case of successful TCP SYN-flooding attacks affecting Amazon's cloud services.

2.4.1.1.2. UDP flooding attack. The UDP protocol is also a transport layer protocol. It is connectionless and is often used when reliability of the packet transfer is not mandatory. An example of this is during the transfer of real time applications such as voice and video. On the internet, UDP can also be used for online gaming and instant messages (Rui et al., 2009). The protocol vulnerability can be exploited to launch DDoS attacks, such as flooding attacks. UDP flooding can be initiated by generating excessive amount of UDP packets to random ports of the cloud target (Wong and Tan, 2014). The attack exploits UDP's connectionless and unreliability feature by directing high volume of malicious traffic towards the target to fill up the response queue; thereby, preventing responses to legitimate users (Rui et al., 2009). The unreliable feature in UDP does not allow the target system to regulate the attackers' sending rate (Wong and Tan, 2014).

2.4.1.1.3. ICMP flooding attack. ICMP is an IP protocol which can be used to check the current status of a host's network connectivity. Attackers have used ICMP to launch DDoS attack in form of smurf and ping flood attacks (Wong and Tan, 2014). It is carried out by directing enormous ICMP packets to a target with an attempt to consume the bandwidth and crash the target. Consequently, the target will not be able to respond to incoming request from legitimate users.

2.4.1.2. Application layer DDoS. Application layer DDoS attacks in cloud computing have continued to increase over the years, both in volume and complexity. These attacks adversely impact the productivity, quality of service, quality of experience, reputation and the revenue of the cloud provider. Attacks on the application layer target cloud services using flood packets, and typically use significant HTTP flood at high rates to overwhelm a target webserver hosted in the cloud. This consumes the target cloud webserver's resources and prevents legitimate users from accessing the target. Application layer DDoS type of attacks are challenging to mitigate as such attacks consume less bandwidth and are stealthier in nature. The attacks flood the target server with what appears as legitimate requests (Wong and Tan, 2014). Common among such attacks are HTTP flood attack and XML flood attack.

2.4.1.2.1. HTTP flood attack. HTTP flood attacks (also known as H-DoS) are designed to flood web servers and applications in the cloud by using malformed HTTP packets ('impersonating' HTTP GET or POST requests) (Choi et al., 2014). Such attacks do not necessarily require a high rate of traffic flow. For example, HTTP GET attack can be carried out by compromising several nodes on the Internet to create several request sessions to the victim in other to disable the victim. A recent report on global DDoS attack reveals that close to a quarter of current DDoS attacks target the application layer (Wong and Tan, 2014), and one-fifth of the HTTP DDoS attacks are HTTP GET floods.

2.4.1.2.2. XML flood attack. When requesting for resources, cloud users and providers use SOAP message to start the communication. SOAP messages work with HTTP and are written in XML because the latter is a universally acceptable language that runs on any platform (Karnwal et al., 2012). X-DoS, an Extensible Markup Language DoS attack, can be carried out using less sophisticated tools due to its ease of implementation. The distributed version of X-DoS is known as DX-DoS. In the XML-wrapping attack on Amazon EC2 services described by Gruschka and Iacono (2009), SOAP message request validations are exploited by changing the XML tags. Hence, any unauthorized user can have access to Amazon's EC2's services, which can be abused to send spam mails using multiple virtual machines.

2.4.2. Reflector attack

In a reflector based DDoS attack, the attacker spoofs an IP address and sends the request to a large number of reflector hosts (see Fig. 4). When the requests are received, the reflector hosts

Table 1
DDoS attacks, features and tools.

Attack name	Characteristics				Tools
	Application	Infrastructure	Direct	Reflector	
SYN flooding		✓	✓		LOIC, XOIC
ICMP flooding		✓	✓		TFN, XOIC
UDP flooding		✓	✓	✓	LOIC, XOIC
HTTP (H-DoS) flooding	✓		✓		DDoSIM
XML (X-DoS) Flooding	✓		✓		DAVOSET
Ping of Death (POD)		✓	✓		Ping
Slowloris	✓		✓		PyLoris, Goloris
Zero-day[a]	✓	✓	✓	✓	Any tool
Smurf		✓		✓	Nemesis, Ping

[a] Unknown or new DDoS attack that exploits vulnerabilities without patch or fix.

send the response to the target, resulting in the flooding of the target (Bhuyan et al., 2015). An example of this attack is a smurf attack, which is carried out by sending an ICMP echo request as a broadcast message to hosts on the Internet with a spoofed IP address (the target's IP address) (Darwish et al., 2013). These hosts amplify the attack by directing their ping response to the target. In the study of reflector attacks, Arukonda and Sinha (2015) proposed mitigation strategies. Other examples of reflector attack are SYN ACK RST flood, and DNS flood (Bhuyan et al., 2015).

2.5. *Other cloud resource consumption*

Flash crowd is when a group of legitimate packets attempt to access a resource concurrently. Ali-Eldin et al. (2012) describe flash crowd as a surge in traffic known as the Slashdot effect, which results in the service being unreachable to other legitimate users during the period. Flash crowd is not a new phenomenon, and a high profile example is the attack targeting FIFA website during the world cup in 1998. Other related attacks targeting cloud resources without necessarily denying legitimate user access include Economic Denial of Sustainability (EDoS) and Fraudulent Resource Consumption (FRC). An EDoS attack targets the cloud service provider's billing system by fraudulently consuming resources (Sqalli et al., 2011). FRC is another attack variant (Idziorek et al., 2012), which exploits the utility pricing model of operating in the cloud over an extended period of time.

2.6. *DDoS attack tools*

Recent tools that have been used by attackers include the following:

- Low Orbit Ion Canon (LOIC): LOIC, a popular tool used to launch flooding attack, is readily available over the internet, and has been used by cybercriminal groups such as Anonymous (http://www.troyhunt.com/2013/01/what-is-loic-and-can-i-be-arrested-for.html). The coordinated attack was perpetrated by sending requests to other internet users to join the attack via Internet Relay Chat (IRC). As the tool is automated, all that is required for an internet user to participate is the URL or IP address of the victim. The tool will use the participating user's bandwidth to send UDP, TCP or HTTP requests to flood the target (http://www.troyhunt.com/2013/01/what-is-loic-and-can-i-be-arrested-for.html).
- XOIC: XOIC, a tool closely related to LOIC, has a user friendly graphical user interface (GUI) that can be used during DDoS attack to a specified target IP address or port number. XOIC has three attack modes, namely: test mode, normal mode and DoS mode with an in-built TCP/HTTP/UDP/ICMP messages (http://resources.infosecinstitute.com/dos-attacks-free-dos-attacking-tools).
- DDoSIM: This is a DDoS facilitating tool that creates zombie hosts with random IP addresses to launch a full TCP connection against a target host. The tool is written in C++, and can conduct HTTP DDoS, SMTP DDoS and Application-Layer DDoS (https://stormsecurity.wordpress.com/2009/03/03/application-layer-ddos-simulator).
- DAVOSET: This is a tool that abuses the vulnerability of target to carry out DDoS attacks, and its latest version includes features such as support for cookies (http://www.hackerschronicle.com/2014/03/davoset-most-powerfull-ddos-tool.html).
- PyLoris: This DoS tool uses SOCKS proxies and SSL to perform DoS attacks on a target. It allows the attacker to own the HTTP request header, which will help keep the connection opened for as long as possible to exhaust server resources and deny legitimate users (http://security.radware.com/knowledge-center/DDoSPedia/Pyloris).

Other notable tools include HULK, R-U-Dead-Yet and GoldenEye HTTP DoS tool, which can be used to generate different forms of DDoS attacks targeting cloud services. Common DDoS attack, features and tools for perpetrating attacks is thus presented (see Table 1).

3. Cloud DDoS defenses

Several DDoS defense solutions had been proposed in the last two decades, and earlier solutions are designed to mitigate DDoS attacks against a single machine. For example, Trinoo was deployed on approximately 227 hosts to flood a single computer in University of Minnesota (Bhuyan et al., 2013). A number of DDoS defenses for cloud computing proposed recently were based on software defined network (SDN) Yan et al. (2015) reviewed recent SDN literature and the potential of using SDN to defeat DDoS attacks targeting cloud computing. Similarly, Wang et al. (2014) examined the security impact of DDoS attack defense techniques in an enterprise network where SDN and cloud computing were adopted.

Varadharajan and Tupakula (2014) examined various attack scenarios on cloud hosted services and proposed a trust enhanced security model.

In this section, we will be focusing only on DDoS defenses proposed for cloud services, categorized using the DDoS defense taxonomy outlined in Fig. 5.

3.1. *Cloud DDoS defense deployment*

DDoS defenses for cloud services can be deployed in four key locations, source-end, access point, intermediate network and distributed.

3.1.1. *Source-end deployment*

The advantages of source-end deployment include more effective protection of network resources and bandwidth. For example, defenses deployed at the source of a potential attack usually use a throttling component to limit the rate of outgoing packets during DDoS attacks (Bhuyan et al., 2013), which will preserve the resources of both the intermediate network and the target victim.

3.1.2. *Access point deployment*

Access point deployment is usually deployed in the front-end, back-end or on each virtual machines (VMs) in the cloud computing environment. The front-end is typically the administrative domain of the cloud service that serves as an interface between

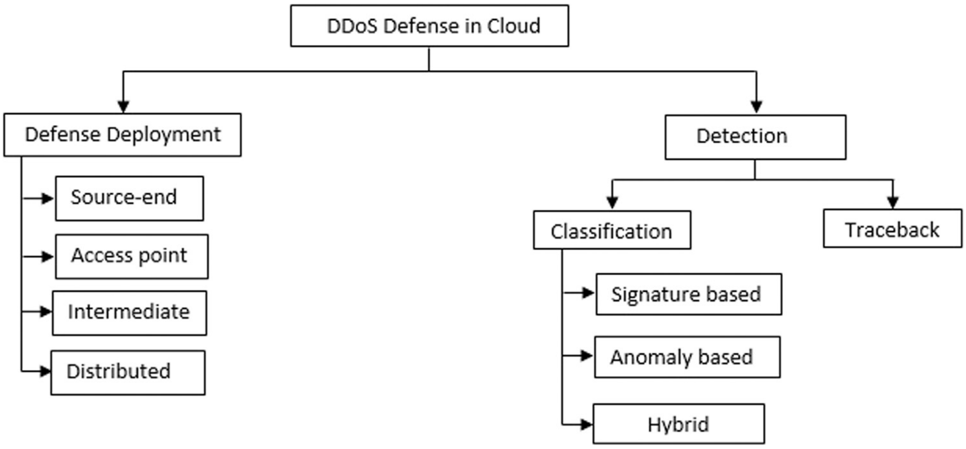

Fig. 5. Cloud DDoS defenses taxonomy.

the cloud user and the various cloud components. In Eucalyptus, for example, this is referred to as the cloud controller, while in Xen, it is known as dom0. DDoS defenses deployed at the access point distinguish legitimate packets from malicious packets before granting access to the cloud computing resource and services. A key limitation of this deployment is that the access point is generally not the most suitable place for filtering or rate-limiting as bandwidth might be saturated. However, this approach is most commonly deployed due to the ease of deployment, and SBTA (Yang et al., 2012) is one popular example.

3.1.3. *Intermediate-network deployment*

These are defenses deployed on network nodes to limit the impact of DDoS attacks on the network before the attacks affect the intended target. This is achieved by imposing rate limits on the traffic passing through the nodes after comparing the traffic against a normal profile pattern (Bhuyan et al., 2013). Such a deployment can be effective but it is impractical in a cloud computing environment as the nodes are not controlled by the same provider and are in different administrative domains. This could, perhaps, work in a private cloud deployment.

3.1.4. *Distributed defense*

Distributed defense is a hybrid deployment model comprising source-end, access point and/or intermediate network deployments. Depending on the configuration, this deployment model can be tuned to achieve a high DDoS attack detection rate (e.g. good cooperation between various administrative domains and providers). MTF (Iyengar et al., 2014) is an example of a distributed defense deployment.

3.2. *DDoS detection*

Typical DDoS detection techniques classify packet traffic as either legitimate or malicious, and can be broadly categorized into signature based, anomaly based and hybrid.

3.2.1. *Signature based detection*

Signature based detection technique use a set of rules and known signature attack patterns stored in a knowledge database. Traffic patterns are monitored and compared against existing signatures to detect malicious traffic (somewhat similar to a typical signature based anti-malware solution). Signature based is known for its accuracy in detecting known attack signatures as long as the database is always up-to-date. The major drawback is its inability to detect unknown attacks or variation of known attack signatures, which lead to high false negatives.

A signature based DDoS detection in cloud was proposed by Bakshi and Yogesh (2010). They used an intrusion detection system (IDS) in VMs to counter DDoS attacks. The IDS sensor uses SNORT, a signature based technique, and is deployed on the virtual interface of VMware virtual ESX machine to analyze both in-bound and out-bound traffic in real time. The defense is designed to counter DDoS attacks in the network/transport layer, by identifying the IP addresses used for the attacks and automatically generating an access control list to drop the entire packets from the blacklisted IP addresses. If the generated attacks are from compromised zombie machines, the approach can be configured to block such traffic and transfer the targeted application to a VM hosted in another datacenter.

Lonea et al. (2013) deploy a VM based IDS, which has a graphical interface. The researchers configured MySQL database to monitor alerts of cloud fusion unit in the front-end of the Eucalyptus cloud architecture. Their solution used the Barnyard tool to capture attacks, while signature based snort was configured with predefined DDoS rules to defend against known DDoS attacks. The captured attack packets by Barnyard are stored in binary unified file and transmitted using a secured tunnel to a centralized MySQL database at the front-end. DDoS attacks were simulated using Stacheldraht, a DDoS attack tool that generates infrastructural resource depletion attack consisting of ICMP flooding, UDP flooding and TCP SYN. Even though the system had a high detection rate with a low false positive, the major limitation of this approach is the inability to detect unknown attacks – an inherent weakness of the signature based approach.

Karnwal et al. (2012, 2013) propose the use of filter tree approach to defend against application layer flooding (see Fig. 6). The five modules in the proposed approach were designed to detect and resolve XML and HTTP based DDoS attacks that occur in requests for resources using SOAP messaging. The IP marking module uses the Flexible Deterministic Packet Marking (FDPM) scheme to mark SOAP messages at the edge router; while IP traceback is a logical file that stores a blacklist of IP addresses provided by the Cloud Defender. Finally, the Cloud Defender filters the attacks by going through five different stages: Sensor Filter, Hop Count Filter, IP Frequency Divergence Filter, Confirm legitimate user IP Filter and Double Signature Filter. The first four stages are used to detect HTTP DDoS attacks while XML DDoS attack is detected in the fifth stage.

Gul and Hussain (2011) propose an intrusion detection model to handle large flow of packets by analyzing and producing reports on these packets. These reports are distributed to the various parties in the cloud system. The proposed model employs multi-threading techniques to improve the IDS performance. The NIDS senses and monitors traffic in the network to check for malicious packets. As soon as a malicious packet is detected, an alarm is sent

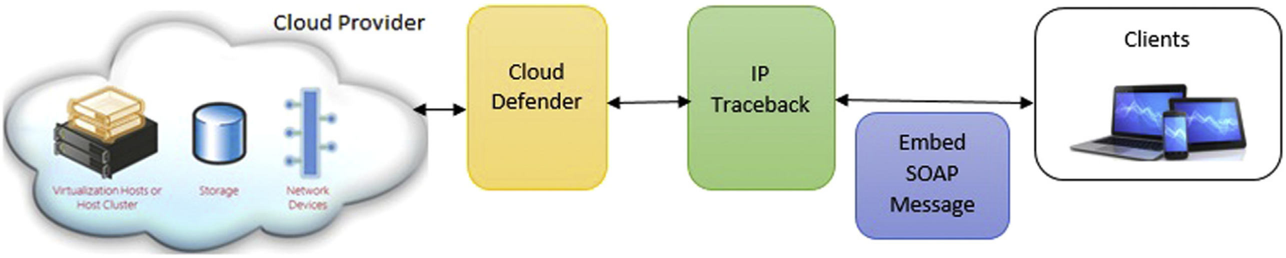

Fig. 6. Filter tree approach against XML and HTTP DDoS attack in cloud (adapted from Karnwal et al. (2012)).

to a third party monitoring system that reports to the cloud management system. The NIDS model has three modules, namely: capture and queueing module, processing/analysis module, and reporting module. This model was implemented using .NET in a Windows environment. The evaluation carried out by the researchers suggested that multiple thread deployment mode is more efficient than single threat deployment mode.

Lo et al. (2010) propose an IDS based mechanism distributed within the cloud environment. Alerts will be exchanged with the IDS nodes distributed throughout the cloud environment when an attack is detected. The IDS system comprises four components, namely, cooperative operation, response and block, threshold check and alert clustering, and intrusion detection. A cooperative agent is contained in each IDS, which is used to determine whether to accept or reject the alert sent by other IDS nodes. Such an approach can mitigate attacks detected and reported by other IDS nodes. Gupta and Kumar (2013) propose the use of an attack pattern detection scheme based on VM profile optimization. Rule based detection was used to match packet during TCP SYN flooding attacks by extracting the threshold for rule pattern during the initial rule establishment phase.

Advantages of signature based detection method include:

- Accuracy in detecting known attack signature with a low false positive rate.
- The presence of DDoS attack labels allows the system administrator to determine the exact type of DDoS attack the victim is experiencing.

Disadvantages of signature based detection include:

- Maintaining an up-to-date signature is an uphill and costly, if not impossible, task.
- Misrepresentation of signature pattern will result in a high false negative rate.
- Inability to detect unknown and zero-day attacks.

3.2.2. Anomaly based detection

Anomaly based or behavioral classification approach involves the collection of normal traffic behavioral profile pattern over a pre-determined period. Its main objective is to detect subsequent patterns that deviate from an expected behavior. Chandola et al. (2009) group anomalies into three main categories, namely: point anomalies, contextual anomalies and collective anomalies.

Point anomaly occurs when an individual data instance is considered anomalous with respect to the rest of the data. A typical example of this is an application-bug level attack (see Prokhorenko et al. (2016) for an overview of web application attacks and mitigation strategies) in packets resulting in the DoS attacks. Anomaly is referred to as being contextual if data is anomalous in a specific context but not in another context. This is mainly determined by the structure of the dataset. In collective anomaly, a group of data instances is anomalous with respect to the whole dataset. An example of this is DDoS flooding attacks, where individual data instance only becomes anomalous and harmful in coordination.

The anomaly approach is typically carried out in two phases, namely: training and detection phases.

Training phase: The efficiency of anomaly detection depends on the nature of input data during the training phase. Input is a collection of data instances in the form of patterns, samples and observations described by a set of attribute represented in binary, categorical or numerical type. Each data instance may consist of single attribute (univariate) or multiple attributes (multivariate) (Chandola et al., 2009). In a multivariate data instance, data can either be same type or combination of data types. Data labels are used to specify if a particular instance is normal or anomalous. There are research datasets consisting of different anomalous attack labels and normal data instances. A popular (but out-of-date) example is KDD'99, which consists of approximately 4,900,000 single connection vectors. Each of the vectors contains 41 features and labeled as either attack or normal. The simulated attack falls into four categories, namely: Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L) and Probing attack (Tavallaee et al., 2009).

Detection phase: Anomaly detection can exist in three modes based on the amount of labels available, namely: supervised, semi-supervised and unsupervised.

Supervised mode assumes the availability of labeled instance training datasets for both normal and anomaly classes. The approach is used to build a predictive model for normal versus anomaly classes. Previously unseen data instances are compared against the model to determine the class it falls into. There are two key issues with supervised anomaly detection. Firstly, anomalous instances are far fewer when compared to normal instances in training data; and secondly, the classification challenge for anomaly class is the lack of accurate and representative label (Bhuyan et al., 2014). Semi-supervised anomaly detection assumes the training data has only label instances for the normal class. They are much more practical compared to supervised techniques, as they do not require the use of labels for anomaly class (Chandola et al., 2009). Finally, in the unsupervised anomaly detection approach, the model does not require any training data. Hence, it is one of the most widely deployed techniques (Bhuyan et al., 2014). The latter assumes that normal instances are significantly more frequent than anomalies in a typical test dataset. If this assumption is not true, the technique suffers from a high false alarm rate.

Reporting detected anomalies is a very important aspect of anomaly detection. According to literature, the two common detection output types are scores and labels. Using scores involve assigning an anomaly score to each instance data to reflect the degree of anomaly. A cut-off threshold is set to determine either to accept or reject the instance data. Labels on the other hand involve assigning a label to each test instance indicating either normal or anomaly.

3.2.2.1. Anomaly detection techniques. In categorizing anomaly detection of cloud DDoS attacks, we group existing techniques into different "ring" classes based on the algorithm(s) used. Coming up with the ring scheme is not a straightforward process as proposed

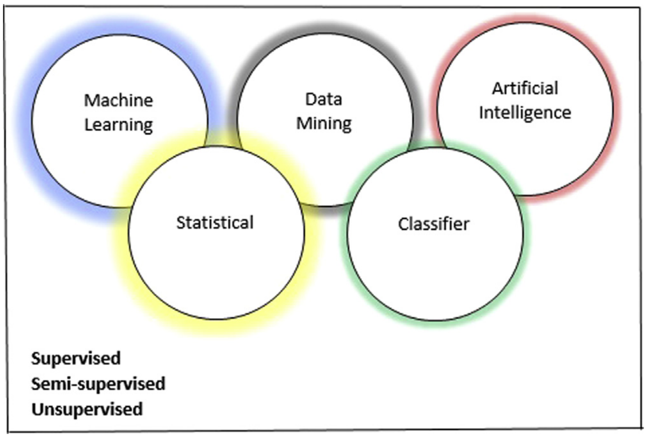

Fig. 7. Ring classification of anomaly detection methods.

classes are likely to overlap. The five classes are machine learning, statistical, data mining, classifiers and artificial intelligence (see Fig. 7).

3.2.2.1.1. Statistical anomaly detection. In statistical anomaly detection, the statistical features of a normal traffic are compiled to generate a normal traffic pattern, which will be compared with incoming traffic to detect anomaly packets. During detection, statistical inference tests, either parametric or non-parametric techniques, are applied to determine the legitimacy of the behavior (Modi et al., 2013).

Vissers et al. (2014) propose a parametric technique using the Gaussian model to defend against application layer DDoS attacks on cloud services, which use malicious XML content contained in a Simple Object Access Protocol (SOAP). Normal profile model is constructed from the dataset during the initialization stage prior to activating the proxy to listen to requests. During detection, there will be several phases. In the first phase, HTTP header inspection will be carried out to prevent HTTP flooding. It also undertakes SOAP action check and size outlier inspection. In the next phase, the XML content is processed before checking if RSOAPAction is/are spoofed by consulting previous maps. Finally, the SOAP feature outlier detection (a filter process that evaluates each feature to its corresponding Gaussian Model) is carried out. The drawback of this model is its inability to detect request schematics resulting from new DDoS techniques, without implementing additional features.

Shamsolmoali and Zareapoor (2014) use a statistical-based filtering system – Cloud confidence DDoS Filtering, which uses two levels of filtering. It first removes the header field of the incoming packet, and compares the TTL value with the stored value in the IP to hop-count (IP2HC) table. If these values are not equal, the packet is dropped and categorized as spoofed. The second level is based on the Jensen–Shannon divergence concept, which uses a stored normal profile stored in a database to compare incoming packet header information. This helps to check for information divergence. Zakarya (2013) introduces an entropy based detection technique that uses attack packet dropping algorithm to detect DDoS attacks in the cloud computing environment. The entropy rate is used to identify attack flow based on the distribution ratio. Anomaly detection system is deployed on each edge router which subsequently transfers the flow to an adjacent router for a confirmatory check when DDoS is detected. If confirmed, DDoS packets are dropped. Findings from the CloudSim simulation suggested an accuracy rate of 90%.

Girma et al. (2015) propose a hybrid statistical model which uses the covariance matrix and entropy based system to classify DDoS attack pattern, by measuring heightened dependency in the data. Ismail et al. (2013) use a mathematical model, covariance matrix approach to detect flooding based DoS attacks against cloud services. In the first phase of their approach, a model for profiling normal traffic pattern was used as the baseline, by mapping captured normal traffic into a matching covariance matrix. The second phase is the intrusion detection stage where the covariance matrix obtained from the first phase is compared with the covariance of the currently captured traffic. The last phase is the prevention stage that implements the two earlier phases.

A confidence-base filtering (CBF) method that uses correlation characteristics was proposed by Dou et al. (2013), which could be deployed in both attack and non-attack periods. During the non-attack period, a nominal profile is created by extracting an attribute pair from the network and transport layers. The frequency of occurrence of these value pairs will be extracted and used to calculate their confidence value. The correlation characteristics that exist between these two layers were used to determine the legitimacy of a packet during the traffic flow. During the attack period, attribute value pairs of incoming packets will be collected and compared with the nominal profile to determine their confidence value in legitimate flow. Packet discarding strategy uses CBF score and filtering criterion to ascertain the legitimacy of a packet by checking if the CBF score is above the pre-defined threshold. This will be used to determine whether the packet will be granted access to the cloud environment or not. Negi et al. (2013) presented an enhanced CBF packet filtering method of Dou et al. (2013) to improve the processing speed and utilization of storage based on correlation pattern.

More recently in 2015, Wang et al. (2015) propose a cloud DDoS attack defense (DaMask). DaMask employs a highly programmable network monitoring technique that detects attack and responds using a flexible control structure. DaMask has three layers (i.e. network switches, network controllers and network application) and two modules (i.e. anomaly based network attack detection module – DaMask-D, and an attack mitigation module – DaMask-M) – see Fig. 8. When an attack is detected by DaMask-D, an alert is issued. Both the alert and the packet information will then be forwarded to the DaMask-M module. If the packet is determined to be legitimate, it will be forwarded to its destination. DaMask-M performs two functions, namely: countermeasure selection and log generation. After receiving an alert, DaMask-M decides on the countermeasure to undertake. A typical countermeasure is to drop the packet. Evaluations using Amazon Web Service (EC2) public cloud and a private cloud running Ubuntu 12.10 with the UNB ISCX dataset suggested that DaMask's performance was similar to detection schemes based on continuous time Bayesian network, but DaMask reportedly has a lower computational cost.

Bedi and Shiva (2012) propose a mechanism to secure cloud infrastructures from co-resident DoS attacks using game theory. The mechanism is used to model both legitimate and malicious VM behaviors that co-reside on a physical machine, and a game inspired firewall defense was also modeled.

Marnerides et al. (2015) present an anomaly detection technique, Ensemble Empirical Mode Decomposition (E-EMD), which can be used to conduct statistical characterization and decomposition of measured signals. E-EMD can be implemented on the hypervisor level and functions by jointly considering systems and network information from every VMs. The proposed approach was motivated by the fact that most monitored traffic exhibit non-linear and non-stationary properties.

In summary, the advantages and disadvantages of statistical anomaly detection approaches are as follows:

Advantages

- Statistical anomaly detection approaches allows the learning of expected behavior from observations without prior knowledge of normal activities of the target system. This can potentially result in more accurate detection of malicious activity.
- Anomaly scores associated with statistical detection can be used as a confidence interval during decision making.

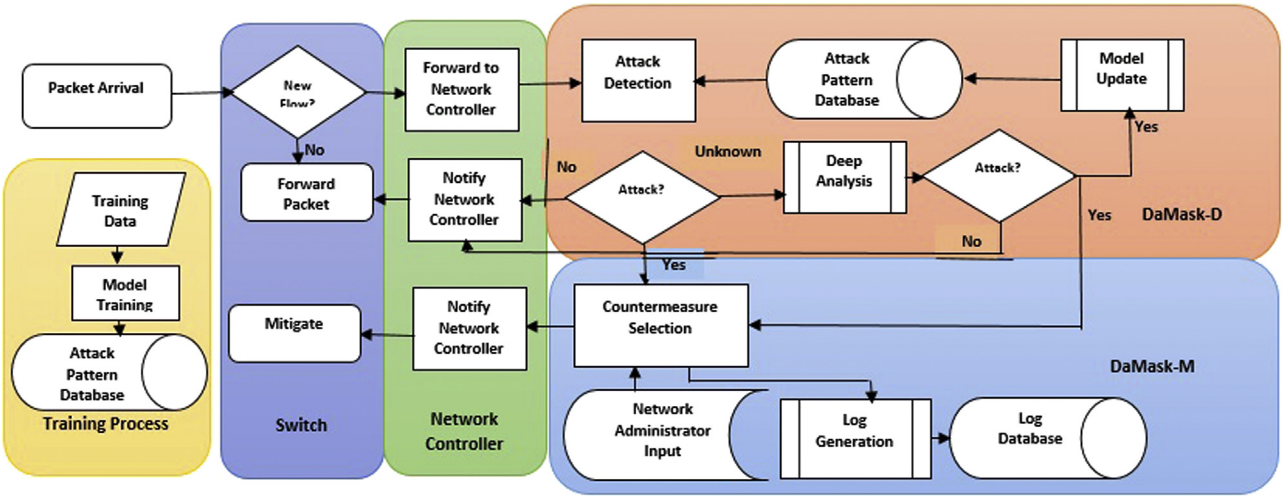

Fig. 8. Workflow of DaMask. Adapted from Wang et al. (2015).

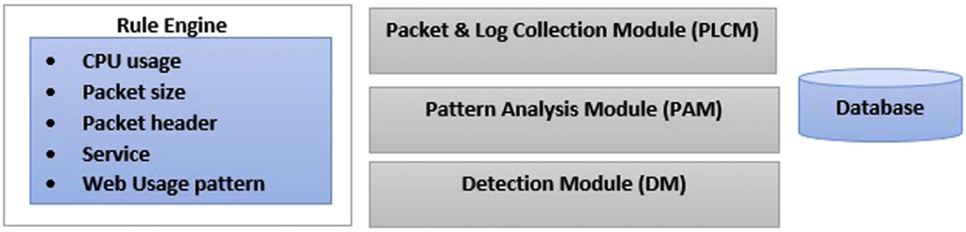

Fig. 9. Packet analysis and control function block. Adapted from Choi et al. (2014).

Disadvantages

- Setting an optimal threshold without resulting in an extreme false positive or false negative can be challenging.
- Statistical anomaly detection techniques involve assumptions and hypotheses. If not justified reasonably, this can lead to high misclassification rate.

3.2.2.1.2. Data mining. The significant increase in Internet traffic complicates efforts to detect DDoS anomaly pattern. To address this challenge, Choi et al. (2014) present a data mining approach that uses map reduce model to mitigate application layer HTTP GET DDoS attacks. Map reduce is a parallel processing model that has been used to expedite batch job operations. The proposed framework consists of three parts as shown in Fig. 9. The packet and log module analyses packet transmission and web server logs, and the pattern analysis module creates the attack pattern for DDoS detection. The parameters to be analyzed include CPU usage, packet size, load, and information distribution of the packet header. The detection module uses a normal behavioral pattern to detect DDoS attacks. To evaluate the model, a map reduce algorithm was used to measure rates between pattern rule and detection time of proposed system to external signatures. The results suggested that the proposed method performs better than Snort, as it can identify new attack profile and has a shorter processing time.

In another related work to detect application layer HTTP GET flooding attacks, Choi et al. (2013) use map reduce to classify parameters (i.e. CPU usage, load, packet size, protocol distribution and information distribution of packet header) prior to employing entropy statistics to measure reliability of the parameters. Alqahtani and Gamble (2015) propose a solution to detect DDoS attacks at the service level, tenant level, application level, and cloud level. The anomaly detection is performed locally at the service level using a hash map to summarize the data stream. During the monitoring process of the service level detection, an alarm is raised at the cloud interface when the ingress flow increases significantly. The requesters contributing to the high flow rate are identified and tagged. Their flow rate is measured using an abstract information distance metric to compare with a predetermined threshold. If the measured rates are higher than the pre-defined threshold, then the activity is classified as an attack. In the tenant level detection, detectors identify potential attack by combining hash maps received from the local services. Application level detection correlates DDoS attacks, flow rate and performance degradation of web services to detect the spread of DDoS attacks. Results from the tenant and application levels are sent to the cloud level for further verification. Key performance metrics were used to monitor the detection rate and extent of damages.

Kwon et al. (2011) propose a lightweight IDS based on the self-similarity feature. It was determined that behavioral patterns of normal traffic are similar and can be differentiated from behavioral patterns of malicious traffic (i.e. the outlier). The cosine similarity was used in the approach, and optimal time interval was used to estimate self-similarity. During the evaluation, the authors use a pre-processor to extract events from the windows security event log. If the self-similarity is not valid, a system alert is generated and the deployed IDS examines the outlier points and the IP address of the source. The incident is reported to the system administrator. A key feature of this approach is that it does not require a lengthy learning process and the self-similarity can be determined in real time. Chen et al. (2015) propose a network monitoring and threat detection system to secure critical infrastructure in cloud computing. This system is made up of three components; monitoring agents, cloud infrastructure and operation center, and uses Hadoop MapReduce and Spark to enhance the speed of data processing.

From the review, it is observed that DDoS anomaly detection based on data mining is increasingly popular, and the main benefits and limitations are as follows:

Advantages

- This approach is able to address limitations of other (non-data mining) approaches in dealing with large databases by extracting information sets and transforming them into an understandable structure.
- It adds a level of focus that helps to improve the process of detecting DDoS anomaly.
- It enhances the network administrator's ability to differentiate between attack and normal traffic by identifying bounds for valid network activity.

Disadvantages

- Cases of missing or bad values from the dataset will affect detection efficiency.
- Attribute selection can be an issue for large datasets as selection of all attribute may worsen performance.

3.2.2.1.3. Artificial Intelligence. Artificial intelligence based approach (also referred to as soft computing approach), such as Genetic Algorithm, Artificial Neural Network and Fuzzy Sets, requires a continuous learning process to effectively detect new anomalies. Joshi and Joshi (2012), for example, propose a Cloud TraceBack model (CTB) and cloud protector to deal with DDoS attack on web service by using back propagation neural network. The CTB applies the service-oriented architecture (SOA) approach to the traceback methodology to determine the true source of the attack. CTB is deployed at the edge router closer to the cloud and uses deterministic packet marking algorithm to mark the reserved flag and ID fields of the IP header packets. However, CTB does not directly eliminate DDoS attacks, as the elimination is undertaken by the cloud protector. The implementation phase consists of five stages, namely: dataset training/testing, processing dataset, determining the NN architecture, training the system, and testing the system.

The entropy approach of Jeyanthi et al. (2013a) is similar to Jeyanthi et al. (2013b), but the former uses enhanced entropy to detect the cause of overload by determining the network condition. Simulation results suggested a reduction in traffic and a better response time.

Huang et al. (2013) propose a system that uses Multi-stage detection and text-based turning test to mitigate HTTP request flooding attacks. The proposed system is made up of five modules, namely: source checking, counting, attack detection, Turing test, and question generation. The source checking and counting module intercepts incoming packets. The checked packet will be challenged by the Turing test module if the packet is deemed suspicious by answering several text-based questions. The DDoS attack detection module retrieves and record the traffic behavior of each virtual cluster (VC); this will be used as a profile to analyze every VC's instant traffic to check for possible abnormal traffic behaviors by malicious packets. The text-based Turing testing module receives redirected blocked packets and randomly selects a question to be answered by the requester. Access to the destination is only granted if the question is correctly answered. Question generation modules periodically update the pool of questions. The system was implemented in Linux kernel and user spaces according to requirements. Performance test suggested a low reflection ratio and high efficiency.

The advantages and disadvantages associated with the deployment of artificial intelligence are as follows:

Advantages

- The use of neural network for unsupervised learning can be relatively effective in detecting DDoS attack packet.
- The adaptive nature of artificial intelligence techniques allows the training and testing of instances in an incremental fashion.

Disadvantages

- Such approaches may not be easily scalable.
- Over-fitting can occur during the training phase.
- Inadequate access to required amount of normal traffic data compounds training of the underlying algorithm; thus, impacting on the algorithm's effectiveness and efficiency.

3.2.2.1.4. Classifier. Classifiers are techniques that learn from a set of labeled data instances in order to classify a test instance into one of the classes. Such techniques generally operate in two phases, namely the training phase and the testing phase. The training phase classifier learns by using available training data labels while the testing phase classifies a test instance as either normal or anomalous using the classifier (Chandola et al., 2009). Chonka and Abawajy (2012) and Chonka et al. (2011) propose a decision tree classification technique – Pre-Decision, Advanced Decision, and Learning System (ENDER) – to detect and mitigate HX-DoS attacks against cloud web services. HX-Dos is an application layer attack that combines both HTTP and XML message to flood the resources of the target cloud provider service. ENDER detects and marks attack traffic using two decision theory methods. In the first method, a rule set (CLASSIE) that has been built over time using decision tree is used to look for both known and unknown attributes. The second method introduces Added Decision Making and Update (ADMU) that decides on the likelihood of a previously classified message. When an attack is detected, a '1' bit mark from CLASSIE will be appended to the message to allow Reconstruction and Drop (RAD) to discard the message before it harms the victim. RAD is located one-hop away from the victim. The authors used Gaussian Distribution Traffic Model (GM-TM) to demonstrate that ENDER has a 99% detection rate, compared to GM-TM's 88% detection with 15% false positive.

Lonea et al. (2013) suggest an IDS based technique. The IDS was deployed in the cloud's VMs, and a data fusion methodology is hosted at the front-end server. During detection, the alert generated by the VMs will be stored on a MySQL database located in the cloud fusion unit of the front-end server. Analysis of alert generated by each of the VM-based IDS uses a quantitative solution classifier; Dempster–Shafer theory (DST) in 3-valued logic and fault-tree analysis (FTA) for the mentioned flooding attack. Evaluations suggested that the proposed solution can reduce the false negative rate and increase detection rate, without the associated complexity.

The multilevel thrust filtration (MTF) mechanism of Iyengar et al. (2014) contains four detection and prevention modules designed to shield attackers from gaining access to the cloud environment. These modules are traffic analysis, abnormality detection, abnormality classification, and attack prevention. MTF functions by authenticating incoming packets and detecting four types of traffic congestion (i.e. spoofed attack, DDoS attack, flash crowd, and aggressive legitimate traffic) that affect cloud availability at different levels. The proposed technique uses both host based and router based techniques to detect attacks in the early stage to avoid influx of malicious traffic in the data center. In the MTF architecture, an intermediate web server that resides in the cloud acts as a look-up server, which offers a unique ID to cloud users on request. The authorized scrutinizing node's IP address will be sent to registered users.

The escape-on-sight mechanism proposed by Jeyanthi and Iyengar (2013) is designed to provide an efficient scalable mechanism for escape during DDoS attacks. Incoming traffic to the cloud are verified by the traffic analyzer which determines if a

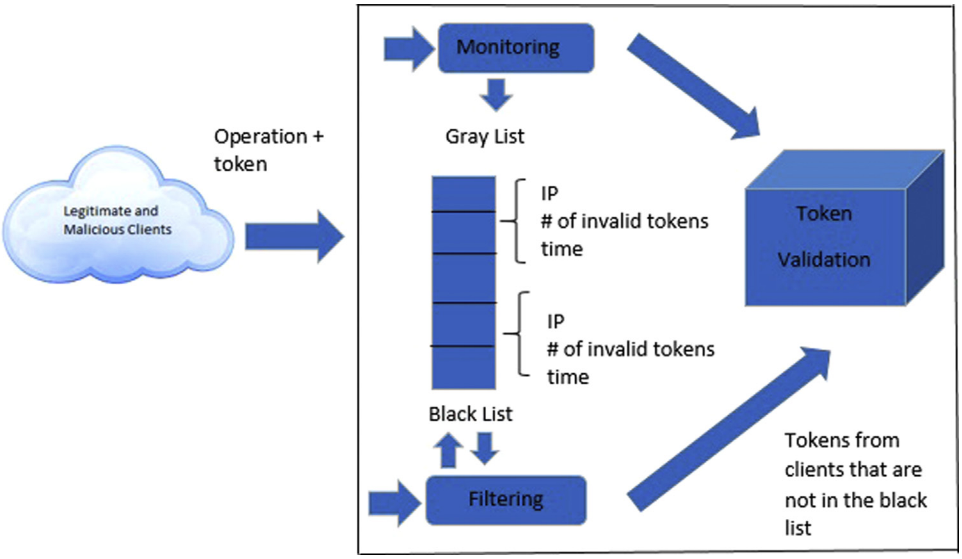

Fig. 10. Client control solution architecture (adapted from Michelin et al. (2014)).

traffic is normal or an attack. Once an attack is detected, the firewall filters and block attack traffic to prevent the cloud service from being overwhelmed.

Application layer DoS attacks exploiting Representational State Transfer (REST) API in cloud services have been reported in recent times. REST, an abstraction for distributed communication over a network, can be used as an alternative to SOAP. The vulnerability is due to its exposure, as services using REST do not require authentication. This allows (cloud) services to be overloaded, resulting in resource depletion during an attack. To defend against this attack, Michelin et al. (2014) propose a defense strategy that uses authentication token. This approach has two different modes, namely: monitoring and filtering (see Fig. 10). During the monitoring mode, a stress test is carried out on the system to check for overloading. It verifies the token from each user, and if an invalid token is detected, the user will be placed in a gray list as a potential attacker. During overloading of the service, the mechanism moves from monitoring to filtering mode and the gray list is changed to black. Users on this black list will have their REST dropped. The system will revert to monitoring mode when the system is decongested. The mechanism was evaluated on OpenStack Grizzly running on VMware workstation 10.0.1.

In summary, classifier anomaly detection presents some advantages and drawbacks.

Advantages

- Classifier based anomaly detection technique have a high detection rate subject to precise threshold settings.
- It is characterized by high adaptation rate for updating detection strategies.

Disadvantages

- They require adequate information training to detect unknown attack event.
- Resource consumption is high when compared to other existing techniques.

3.2.2.1.5. Machine learning. Deploying machine learning to detect cloud DDoS attacks encompasses techniques, such as statistics and data mining, but these techniques have a subtle difference from statistical techniques. The latter requires understanding the process that generates the data; while machine learning involves building a system to improve the performance based on past results. Gupta et al. (2013) propose a profile based network intrusion detection and prevention system that secures the cloud against malicious insiders and outsiders. It combines both fine-grained data analysis and Bayesian technique approach to detect DDoS attacks using unsupervised learning algorithm. The latter aims to detect network based attacks, such as TCP SYN flooding. Just as in the case of Lonea et al. (2013), the network profile is deployed on each VM in the cloud environment. The first stage of the system entails passing a newly created VM through rigorous attack test using the attack signature database. Thereafter, a profile is created for the VM that identifies possible attacks against other VMs. The second stage, the anomaly detection stage, uses normal traffic pattern obtained from the virtual bridge of the VM to check for consistency against other probable attack behavioral patterns. The final stage is network intrusion detection, which analyzes the normal traffic flow path obtained from virtual bridge for attack pattern from VM profile database. Any detected anomaly will be reported to the alert module before going for further analysis.

A generic approach to detecting network anomaly through independent component analysis was proposed by Palmieri et al. (2014). The distributed approach employed a two phase machine learning scheme, which consists of Blind Source Separation (BSS) and rule-based classifier to detect zero-day attacks that alters flow characteristics and traffic volume rate. BSS extracts traffic features from distributed sensors to be used by decision tree classifier to build a baseline traffic profile.

Advantages of machine learning technique include:

- Relatively high efficiency in detecting DDoS attack pattern.
- Ability to change their execution strategy during detection based on additional acquired information.
Disadvantages of machine learning technique include:
- Such an approach requires significant computing resources during both training and testing phases.
- Bottleneck can be introduced to the system as a result of high overhead, which leads to performance degradation of the monitored system.

3.2.3. Hybrid detection

Hybrid-based detection approach involves the use of both signature-based and anomaly-based techniques. This approach uses the complementary features of both techniques to achieve a higher detection rate. For example, Krishnan and Chatterjee (2012) propose an adaptive distributed IDS that combines anomaly based

Table 2
Comparative summary of DDoS defense approaches.

Approach	Efficiency	Overhead	Adaptive	Overfitting	Scalability issues
Statistical	✓				
Data mining	✓				
Artificial intelligence	✓		✓	✓	✓
Classifier	✓	✓	✓		
Machine learning	✓	✓	✓		
Signature-based		✓			✓
Hybrid	✓	✓	✓		

and knowledge based techniques to defend against cloud DDoS attacks. The solution has a service agent, an alert agent and a storage agent, which communicates with each other and the pair nodes. This adaptive hybrid solution is designed to improve the detection rate by lowering the false positives. The system also implements an alert clustering and analyzer that helps all cooperating nodes to differentiate between false alarm and malicious nodes. Cha and Kim (2011) design a three-stage anomaly detection. The first stage is the monitoring stage, which uses a rule-based system to pre-process known DDoS attack patterns. The second stage presents a lightweight anomaly detection that predicts the expected future load on each customer interface using time-series modeling. The traffic volume over network is divided into large and small volumes along the time-axis, and Bayesian technique is used to analyze DDoS attack candidate on the network topology. The last stage uses a focused anomaly to detect both known and unknown DDoS attack patterns using an unsupervised learning algorithm.

Modi et al. (2012) design a hybrid network based intrusion to detect cloud DDoS attacks. Snort, an open source signature based detection method that stores rules of known DDoS attack patterns, and Bayesian classifier, a statistical classifier that predicts the probability of a network event belonging to a class such as normal or malicious with high accuracy, were used in this solution. Eucalyptus, an open source cloud was used for the experimental setup where intrusion detection system was installed on each node controller and all ports were opened for testing purpose. Scapy was employed to generate custom packets while performance and quality were evaluated using KDD'99 dataset.

Teng et al. (2014) propose a cooperative intrusion detection architecture modeled with E-CARGO to defend against cloud DDoS attacks. The proposed architecture is made up of four layers. The first layer, an event generator, collects network packets and generates suspicious intrusion events. The feature detector works in a similar fashion as Snort, and is used to separate events according to network protocols (e.g. ICMP and TCP). Statistical detector uses data packets from the feature detector to determine the attack event. If the number of packets obtained within a certain time range is higher than the threshold set, it is considered an attack. Finally, the fusion center uses agents to perform different roles, such as data pre-processing, space-time fusion and content fusion. Findings from the experiment showed that the proposed method is a viable solution to detect DDoS and slow scanning attacks. However, it has a limitation of differentiating flash crowd traffic from malicious attacks.

A hybrid hierarchical correlated approach, proposed in Ficco (2013), uses security probes to collect and analyze information at different cloud architectural levels. The correlations of intrusion symptoms to the identified cause and target are driven by a knowledge-based, and represented by an ontology.

While a hybrid approach provides the advantages offered by both signature and anomaly based, they are associated with overheads and complexity in getting different algorithms to interoperate efficiently and effectively. A comparative summary of the reviewed detection approaches is presented in Table 2.

3.3. Traceback and IP spoofing detection

Traceback technique can help to locate the true source of DDoS attacks, as these attacks tend to spoof their addresses (e.g. launching a reflector attack). In proposing a defense strategy for application layer DDoS attacks against cloud services, Yang et al. (2012) propose a SOA based technique called SOA-Based Traceback Approach (SBTA) and a cloud filter. SBTA performs DDoS attack traceback by deploying the technique before the web server. SBTA uses advance packet marking based on Compressed Edge Fragment Sampling (CEFS) to determine path reconstruction. The cloud filter, on the other hand, is deployed on the edge router and used as a control mechanism for filtering and rate limiting purpose. The cloud filter gathers cloud traceback mark tags and source IP addresses during the attack, and uses the database to filter out packets with spoofed IP addresses. A significant drawback of this technique is its reactive approach and high rate of false negatives.

Defense against spoofed IP addresses in cloud DDoS attacks that hide the true source has also been proposed. In Jeyanthi et al. (2013b), a technique is proposed to identify spoofed IP addresses in DDoS attacks. The authors also proposed an algorithm, which is activated whenever there is a sudden rise in the packet traffic greater than a pre-defined threshold. The approach also consists of a cloud authentication system (CAS) that verifies the legitimacy of connecting cloud user. CAS, hosted in the cloud environment, has two tables and three procedures. The tables are spoofed address table and current connection table. The procedure tables comprise check flood, packet check and final check, and are collectively used to determine whether the traffic is flooding traffic for subsequent filtering. OPNET Modeler 14.0 was used to simulate the approach, and performance was improved when a buffer was employed for the overloaded packet. A similar approach using IP spoofing to defend against DDoS attacks in the cloud environment was proposed in Osanaiye (2015). Motivated by the fact that most DDoS attacks are characterized by the spoofing of IP addresses, Osanaiye proposed an operating system (OS) fingerprinting technique that monitors incoming packets to the cloud environment to determine its source OS. The algorithm has two stages, namely: active and passive stages. During the passive stage, the packet headers of incoming traffic are captured and analyzed to determine the running OS. In the active stage, specially crafted packets are sent to the source IP address of the connecting packet. A matching is performed to compare the passive OS and probed OS. If both OSes are not the same, packets will be considered spoofed and dropped. Due to OS distribution, an extension of Osanaiye (2015) was presented in (Osanaiye and Dlodlo, 2015) where the final TTL value is used to determine spoofed DDoS attacks when spoofed and true source run similar OS.

3.4. Other forms of DDoS attack defenses

In solving DDoS attack issue in cloud computing, Yu et al. (2014) consider the scenario where an individual cloud user is being targeted. In their approach, an intrusion prevention system (IPS) was deployed at different access points of the cloud environment to monitor incoming packets during DDoS attacks. This is a reactive method that dynamically allocates available resources during DDoS attacks to compensate for the attacks. A queuing model is developed to establish a relationship between resource allocations and various attack strengths. The proposed DDoS mitigation algorithm extracts non-attack parameters for a protected server. It identifies resources for current IPS and the available resource for the cloud. IPS is cloned from the original IPS when an attack is detected. All IPSes work in coordination to filter out attack packets to provide quality of service (QoS) for users. When the volume of attack decreases, the system will automatically reduce the number of IPS and de-provision the resources previously allocated back to the pool.

Guenane et al. (2014) present a firewall based approach that reduces the effect of cloud DDoS attacks, based on a Security-as-a-Service model (SecaaS). The hybrid feature in the proposed solution is derived from its architecture which consists of two parts, namely: virtual and physical. The virtual part is made up of virtual firewalls deployed as VMs that execute firewall functionalities, such as monitoring, analyzing and reporting with dynamic resource provisioning. The physical part is the organization's physical IT resource infrastructure that agrees to acquire a security service offered by the cloud provider. The DDoS mitigation system redirects and loads balance traffic on overloaded physical firewalls based on its hybrid architecture operation. The localized firewall in the cloud receives redirected traffic, which is managed by the virtual part of the hybrid architecture. The two key objectives are decision management and availability performance.

Iyengar et al. (2015) propose a trilateral trust based defense mechanism, which identifies different attack groups prior to separating legitimate packets from incoming traffic. The TTM authenticates incoming cloud user requestor as either trusted client or threat, using three sequential traffic threat notification levels. Each level detects different threats; thereby, reducing the threat traffic for successive levels. Early stage detection reduces traffic congestion at the data center, which improves its availability for legitimate cloud users. TTM protocols include Client ID acquisition, mutual trust establishment, historical behavior monitoring, credit points update, service provision and attack exclusion.

Chapade et al. (2013) propose an average distance estimation technique to defend against cloud DDoS attacks. Exponential smoothing estimation was used to determine the mean value of the distance in the next time period. Minimum Mean Square Error (MMSE), a linear predictor, was used to determine the distance-based traffic separation for DDoS detection by estimating traffic rates from different distances. The distance value was determined from the TTL field of an IP packet. The technique was implemented using NS2 on more than 100 nodes, and the evaluation suggested that the technique has a high detection rate with low false positives.

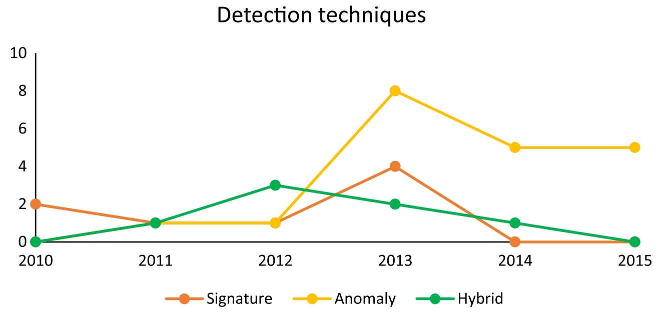

Fig. 11. DDoS detection techniques for cloud computing trend between January 2010 and December 2015.

Liu (2010) describes a new form of DoS attack, coined Loaded Spread Pair estimator (LSP), which exploits the under provisioning of network resources in cloud infrastructure. The author proposes a solution that does not detect nor prevent the attack but dynamically transfer the cloud servers to a different infrastructure to achieve the desired QoS. The technique does not only cater for DoS attacks but also compensate for performance degradation due to resource constraints.

Aishwarya and Malliga (2014) use IDS to defend against transport layer DDoS attacks, using SYN cookies. In this approach, connecting users with malformed ACK are ignored and packets are checked to determine if they are spoofed using hop count filtering (HCF). This serves as a first layer of security. In the second layer of security, the sequence number of SYN packet is encoded; therefore, only legitimate cloud users can decode it. An open source distributed architectural framework that provides application program interface and tools to develop multiple probes was proposed in Ficco et al. (2013). The framework can reportedly be dynamically deployed to collect security information at different cloud architectural levels for analysis of intrusion in the cloud system.

A shuffling-based moving target approach for DDoS cloud defense was proposed by Jia et al. (2014). In their work, the system architecture uses selective server replication and intelligent client reassignment to turn victim servers into a moving target to isolate DDoS attacks.

4. Discussion

In the preceding section, we categorized DDoS attacks into application-bug level and infrastructural level attacks. Most research efforts appear to be directed towards infrastructural level DDoS attacks, which includes both network and application flooding attacks. The reason for several reported cases is because of the ease at which infrastructural level DDoS attacks are carried out. In infrastructural level DDoS attacks, the attacker does not seek to exploit cloud vulnerability as malicious flood packets are merely directed towards the target to clog and consume its resources to the detriment of legitimate users. Application-bug level DDoS attacks, on the other hand, have also been reported (Karnwal et al., 2013; Chonka

Table 3
Summary of commonly used metrics and datasets.

Reference	Performance evaluation metrics	Benchmark/datasets
Zargar et al. (2013)	Detection rate and computation time	Snort[a]
Shamsolmoali and Zareapoor (2014)	Detection rate and false alarm rate.	CAIDA "DDoS Attack 2007"[b]
Choi et al. (2014)	Detection rate and average detection time	Simulated
Huang et al. (2013)	Throughput during attack and performance overhead.	Simulated
Lonea et al. (2013)	Detection rate and computational time	DARPA 1999 Dataset[c]
Wang et al. (2015)	Detection rate and computational cost.	UNB ISCX dataset[d]
Michelin et al. (2014)	Response time and CPU usage.	Simulated
Chonka and Abawajy (2012)	Detection rate	StuPot project dataset[e]
Modi et al. (2012)	Detection rate, computational cost, scalability.	KDD'99 dataset[f]

[a] https://www.snort.org.
[b] http://www.caida.org.
[c] http://www.ll.mit.edu/ideval/data/.
[d] http://www.unb.ca/research/iscx/dataset/.
[e] http://www.deakin.edu.au/~ashley/.
[f] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

and Abawajy, 2012). In these attacks, system vulnerability(es) is/are exploited to carry out the attack. Typically exploited vulnerabilities are misconfiguration, outdated patches, protocol vulnerability and system weakness. However, very few publications on application-bug level DDoS attack mitigation have been proposed.

The most common deployment location of DDoS defense in the cloud environment is the access point (similar to IDS deployment – see Table 4), and it is observed that some have proposed a distributed deployment for efficiency. The choice of the access point is, probably, due to the distributed nature of the Internet and cloud architecture. Source-end deployment would have been the most ideal location to detect DDoS attacks, but it would be challenging to enforce all hosts on the Internet to adopt a general policy.

It also appears that earlier techniques generally use signature based classification to identify known DDoS signatures. While this is an effective solution against known DDoS attack patterns, such solution is increasingly irrelevant in today's threat landscape due to their inability to detect unknown DDoS attack signatures. For example, there are abundance of tools that can be used to generate DDoS attacks to defeat signature based approaches; thereby, leading to a high false negative rate. Anomaly based solutions are increasingly popular, as such approaches have been shown to be effective against both unknown and derivative of known attack patterns. Normal traffic behavior is modeled to obtain a normal behavioral profile using modeling techniques, such as data mining, machine learning, artificial intelligence and statistical methods. This involves extracting packet attributes during the non-attack period and profiling normal behavior. During the attack period, incoming packets events are analyzed with the profiled normal behavior to detect DDoS attacks against cloud services. Hybrid solutions attempt to take advantage of the complementary nature of these approaches by integrating both signature and anomaly

Table 4
Summary of some existing DDoS attack defense mechanisms in cloud computing.

Year	Reference	Detection technique			Deployment location					DDoS attack type		
		Signature	Anomaly	Hybrid	S[a]	AP[b]	I[c]	D[d]	NS[e]	App[f]	Infra[g]	NS[e]
2010	Bakshi and Yogesh (2010)	✓				✓					✓	
	Lo et al. (2010)	✓				✓					✓	
2011	Gul and Hussain (2011)	✓				✓						✓
	Kwon et al. (2011)		✓			✓						✓
	Cha and Kim (2011)			✓		✓					✓	
2012	Karnwal et al. (2012)	✓						✓		✓		
	Bedi and Shiva (2012)		✓			✓						✓
	Krishnan and Chatterjee (2012)			✓		✓						✓
	Chonka and Abawajy (2012)			✓						✓		
	Modi et al. (2012)			✓		✓						✓
2013	Lonea et al. (2013)		✓			✓					✓	
	Karnwal et al. (2013)	✓						✓		✓		
	Gupta and Kumar (2013)	✓				✓					✓	
	Modi et al. (2012)			✓		✓						✓
	Zakarya (2013)		✓									✓
	Huang et al. (2013)		✓			✓					✓	
	Lonea et al. (2013)	✓				✓					✓	
	Choi et al. (2013)		✓						✓		✓	
	Ismail et al. (2013)		✓			✓						✓
	Dou et al. (2013)		✓			✓						✓
	Negi et al (2013)		✓			✓						✓
	Jeyanthi et al (2013)		✓					✓				✓
	Gupta et al (2013)	✓				✓					✓	
	Gupta et al. (2013)			✓		✓					✓	
2014	Shamsolmoali and Zareapoor (2014)		✓			✓					✓	
	Vissers et al., (2014)		✓			✓				✓	✓	
	Choi et al (2014)		✓					✓			✓	
	Iyengar et al (2014)		✓					✓				✓
	Michelin et al (2014)		✓			✓				✓		
	Teng et al (2014)			✓		✓						✓
2015	Alqahtani and Gamble (2015)		✓			✓					✓	
	Girma et al (2015)		✓					✓				✓
	Wang et al (2015)		✓			✓						
	Marnerides et al. (2015)		✓			✓						
	Chen et al. (2015)		✓			✓				✓		

[a] Source-end.
[b] Access point.
[c] Intermediate.
[d] Distributed.
[e] Not stated.
[f] Application-bug level.
[g] Infrastructural level.

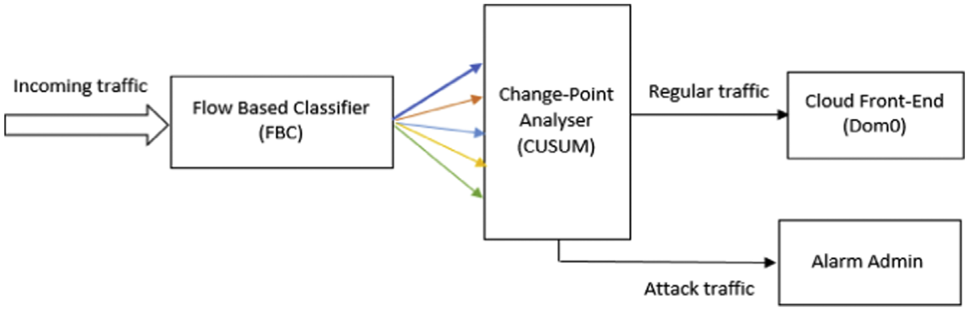

Fig. 12. Our proposed conceptual cloud DDoS defense framework based on change-point detection.

based techniques to achieve a better detection rate. Fig. 11 depicts the research trend between January 2010 and December 2015.

In DDoS defense, most techniques used for generating normal profile patterns fail to consider low rate DDoS attacks, which can result in a high false negative rate. An example is the work of Teng et al. (2014) and Jeyanthi et al. (2013b), where a pre-defined threshold is used in determining the presence of a DDoS attack. In such settings, low rate DDoS attacks will not be detected.

During the evaluation phase, it is apparent that evaluating the proposed solutions remains a challenge due to the lack of up-to-date real-world datasets for training. Commonly used datasets (see Table 3) in the literature include the UNB ISCX 2012 dataset, CAIDA DDoS 2007 dataset, DARPA 2000 LL-DDoS from Lincoln laboratory, MIT and KDD'99 dataset. As an example, CAIDA DDoS dataset (one of the most current datasets) is made up of an hour of anonymized trace from DDoS attack on August 4, 2007 between the hours of 20:50:80 UTC to 21:56:16 UTC split into 5 min pcap files. Another key issue is the dearth in the availability of labeled datasets; this is evident as KDD'99 represents one of the few publicly available labeled datasets currently in use today by researchers.

Identification of normal and attack patterns is crucial as this is a key metric to determine the utility and efficiency of the proposed techniques. Other commonly used metrics that have been considered in literature include detection rate, average response time, percentage bandwidth utilization and normal packet survival ratio.

- Detection rate: Detection rate measures the accuracy of the defense technique, in terms of identifying attack patterns in a traffic flow.
- Average response time: This is the average time it takes a legitimate cloud user to request and receive cloud services during an attack. The service can be in the form of an application hosted in the cloud or infrastructural service.
- Percentage bandwidth utilization: This represents the number of bits transmitted or received by the cloud user per unit time in bits per seconds. During a DDoS attack, this may include both legitimate and attack traffic; however, only legitimate traffic will be considered.
- Normal packet survival ratio: This is determined by measuring the ratio of the total number of legitimate packets that successfully access the cloud environment against the total number of attempted access.

A summary of reviewed DDoS attack defense categories in cloud computing is presented in Table 4.

5. A conceptual cloud DDoS change-point detection framework

It is clear from the discussions in the preceding sections that most defense methods are designed for specific DDoS attack types. We believe that the packet inter-arrival time (IAT) feature of a traffic flow can be used to detect different DDoS attack types and design a generic DDoS defense solution. In general, a DDoS attack is successful if the attack consumes substantial bandwidth and resources, which overwhelms the target. The consumed bandwidth and resources can be considered a function of the packet IAT. In this section, we present our conceptual cloud DDoS defense framework (see Fig. 12), which uses change point detection method to identify statistical anomaly.

We now describe the three key components of the framework.

5.1. Change point detection

Statistical methods used in detecting anomaly involve the use of sequential change point algorithm that makes observation and stores the observation to be used as an input (De Oca et al., 2010). An alert is flagged upon the detection of an unusual trend in the monitored event, which signifies a change has occurred.

Change point detection is, perhaps, first used in 1920–1930s to ensure quality control (Polunchenko and Tartakovsky, 2012). Shewhart chart was the early dominant method before sequential statistics was developed by Wald. The seminal works of Shewhart and Wald were widely cited and adopted in the sequential change detection literature (Cha and Kim, 2011). A number of change point detection methods, such as exponential weighted moving average (EWMA), Shiryayev–Roberts and cumulative sum (CUSUM) (Zhang et al., 2011), have been proposed over the years. For example, Wang et al. (2004) propose Change-Point Monitoring (CPM), which uses the nonparametric CUSUM method. In their approach, the strong positive correlation between requests and responses acknowledgment (ACKs) in the internet is regarded as the modification of a normal protocol behavior due to a DDoS attack. SYN flooding was used as a case study and the differences between SYN and FIN (or SYN/ACK) pair were used to identify an attack. In a similar vein, Peng et al. (2004) adapted the detection scheme proposed in Wang et al. (2004) using an advance nonparametric change detection scheme Instead of relying on SYN and FIN (or SYN/ACK) relationship used in Wang et al. (2004), they use the arrival rate of new source IP addresses to detect DDoS attacks. Takada and Hofmann (2004) pointed out that the proposed scheme in Peng et al. (2004) suffers from issues such as a delayed detection of constant rate DDoS attacks and the inability to accurately determine the termination of an attack. Ahmed et al. (2009) employ a dynamic sliding window CUSUM algorithm to detect nested changes in large repository, and Zhou et al. (2013) propose a distributed detection scheme which uses adaptive CUSUM at the source end to compute the mean and the variance in order to detect traffic anomalies. We refer the interested reader to Srivastava and Wu (1993) for a detailed comparative summary of change point detection methods.

In our conceptual framework (see Fig. 12), we use the change point detection which leverages the statistical feature of packets in detecting anomaly behavior. Similar to other statistical anomaly detection methods, our framework monitors and compares features of observed packet sequence with the normal behavioral profile pattern obtained over a pre-determined period, with the aim of detecting any significant deviation. However, we present a new traffic flow pattern feature, packets IAT, to improve the detection of DDoS attacks.

5.2. Packet inter-arrival time (IAT)

Packet IAT has been used in network performance monitoring (Garsva et al., 2014) and identification of application over the internet (Jaber et al., 2011). For example, Arshadi and Jahangir (2011) used TCP flow IAT distribution to describe a normal traffic which conforms to a Weibull distribution against an anomaly that deviates from this.

In a typical DDoS attack, the traffic exhibits a seemingly predictable inter-arrival pattern due to the use of automation. In other words, the traffic pattern is likely to be exponentially distributed; therefore, occurring at a fixed interval of time exhibiting a Poisson distribution. Therefore, in our approach, we monitor the packet IAT of a flow by determining the time between the receipt of the first and subsequent packets. The inter-arrival distribution is then used to determine the probability of a DDoS attack occurrence in the traffic flow.

Let the observed packet P be defined by its packet inter-arrival time P_t in a traffic flow. The packet inter-arrival time P_t can, therefore, be expressed as $P_t = A_{t+1} - A_t$ which is the time between two consecutive packets, A_{t+1} and A_t, of a flow, for $t=0, 1, 2, 3, 4, \ldots, N$.

5.3. Flow based classifier (FBC)

In a cloud computing environment, traffic patterns from legitimate users are likely to vary between users and the time of the day or week; therefore, the TCP packet IAT will vary and are non-uniform. On the other hand, during DDoS attacks, we are likely to have constant rate attack (i.e. the attack traffic overwhelms the available resources, which will continue for the remaining of the attack), pulsating attack (i.e. attack traffic oscillating between zero and maximum), gradual pulse attack (i.e. attack traffic that attains a maximum rate gradually, maintains it and decreases gradually), or increasing rate attack (i.e. attack traffic gradually attains the maximum rate which is maintained until the termination of the attack) (Takada and Hofmann, 2004). These different forms of DDoS attack pattern follow a predictable pattern, in relation to packet IAT due to its automation.

Therefore, we use a flow based classifier (FBC) to classify incoming traffic from different sources, by inspecting the packet's header content. We define a flow as successive packets with the following tuples, namely: source IP address:port, destination IP address:port (Nguyen and Armitage, 2008). From the traffic flow, the IAT is determined and used to detect changes in packet sequence. To achieve this, a measure of normal IAT pattern is determined a priori. This is used to detect the presence of a pattern change during DDoS attack on a per packet basis using a nonparametric CUSUM algorithm. If an attack traffic is detected, an alarm is triggered and the packets in the flow are dropped while regular traffic gain access into the cloud environment.

6. Conclusion and future research

In this paper, we reviewed academic literature on DDoS attacks against cloud services, and the mitigation strategies published between January 2010 and December 2015. We presented a taxonomy of the different types of cloud DDoS attacks, and the corresponding DDoS defense taxonomy. Anomaly based detection and access point deployments were identified as the most popular defense mechanisms proposed. A conceptual framework was also presented, where change point detection of packet IAT was used to detect different forms of DDoS attack in the cloud. Future work will include the evaluation and refinement of the framework using real-world data.

Despite the amount of research efforts in this area, there are a number of challenges that need to be addressed. For example, the need for a defense solution to detect both application-bug level and infrastructural level DDoS attacks, as contemporary DDoS attack tools are capable of launching such attacks targeting different cloud components. More research should also be directed towards efficient defense solutions that can detect both high and low level DDoS attacks. There is also a need for an effective approach for selecting optimal thresholds in determining DDoS attack patterns for existing and future defense techniques, otherwise it can lead to high rate of false positive and false negative.

Efforts should be expended into having a common standardized deployment policy by different ISPs and cloud service providers. This will foster cooperation between different ISPs and cloud service providers, and rate limit DDoS traffics between nodes using normal profile traffic patterns. This will help to reduce the impact of DDoS attacks against cloud services.

For proper learning and evaluation, producing labeled dataset with optimal feature selection in line with current DDoS attack pattern should be addressed. This will ensure the availability of up-to-date training and testing datasets that are representative of current attack patterns.

Acknowledgments

Part of the research was undertaken when the second (and corresponding) author was with INTERPOL Global Complex for Innovation, Singapore. The views and opinions expressed in this article are those of the authors alone and not the organizations with whom the authors are or have been associated. The authors would also like to thank the Editor-in-Chief, and the five anonymous reviewers for providing constructive and generous feedback. Despite their invaluable assistance, any errors remaining in this paper are solely attributed to the authors.

References

Arukonda S, Sinha S. The innocent perpetrators: reflectors and reflection attacks. ACSIJ Adv Comput Sci: Int J 2015;4(1):94–8.

Ali-Eldin A, Johan T, Erik E. An adaptive hybrid elasticity controller for cloud infrastructures. In: Proceedings of the IEEE network operations and management symposium (NOMS), Maui, Hawaii; 2012. p. 204–12.

Alqahtani S, Gamble R. DDoS attacks in service clouds. in: Proceedings of 48th IEEE international conference on system sciences (HICSS), Kauai, Hawaii; 2015. p. 5331–40.

Aishwarya R, Malliga S. Intrusion detection system—an efficient way to thwart against Dos/DDos attack in the cloud environment. in: Proceedings of IEEE international conference on recent trends in information technology (ICRTIT), Chennai, India; 2014. p. 1–6.

Anwar Z, Malik A. Can a DDoS attack meltdown my data center? A simulation study and defense strategies IEEE Commun Lett 2014;18(7):1175–8.

Ahmed E, Clark A, Mohay G. Effective change detection in large repositories of unsolicited traffic. In: Proceedings of the fourth IEEE international conference oninternet monitoring and protection (ICIMP'09), Venice/Mestre; May 24–28 2009. p. 1–6.

Arshadi L, Jahangir A. On the TCP flow inter-arrival times distribution. In: Proceedings of the fifth IEEEUKSim European symposium on computer modelling and simulation (EMS), Madrid; Nov. 16–18 2011. p. 360–5.

Bhuyan MH, Bhattacharyya DK, Kalita JK. An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. Pattern Recogn Lett 2015;51:1–7.

Beitollahi H, Deconinck G. Analyzing well-known countermeasures against distributed denial of service attacks. Comput Commun 2012;35(11):1312–32.

Bhuyan MH, Kashyap HJ, Bhattacharyya DK, Kalita JK. Detecting distributed denial of service attacks: methods, tools and future directions. Comput J 2013 bxt031.

Bakshi A, Yogesh B. Securing cloud from ddos attacks using intrusion detection system in virtual machine. In: Proceedings of second IEEE international conference on communication software and networks (ICCSN'10), Singapore; 2010. p. 260–4.

Bedi HS, Shiva S. Securing cloud infrastructure against co-resident dos attacks using game theoretic defense mechanisms. In: Proceedings of ACM international conference on advances in computing, communications and informatics (ICACCI'12), Chennai, India; 2012. p. 463–9.

Bhuyan MH, Bhattacharyya DK, Kalita JK. Network anomaly detection: methods, systems and tools. IEEE Commun Surv Tutor 2014;16(1):303–36.

Christof W, Anandasivam D, Blau B, Borissov D, Menin D, Michalk D, Stober J. Cloud computing—a classification, business models, and research directions. Bus Inf Syst Eng 2009;1(5):391–9.

Choi J, Choi C, Ko B, Choi D, Kim, P P. Detecting web based DDoS attack using MapReduce operations in cloud computing environment. J Internet Serv Inf Secur 2013;3(3–4):28–37.

Cha B, Kim J. Study of multistage anomaly detection for secured cloud computing resources in future internet. In: Proceedings of IEEE ninth international conference on dependable, autonomic and secure computing (DASC), Sydney, Australia; 2011. p. 1046–50.

Choi J, Choi C, Ko B, Kim P. A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. Soft Comput 2014;18(9):1697–703.

Chonka A, Abawajy J. Detecting and mitigating HX-DoS attacks against cloud web services. In: Proceedings of 15th IEEE international conference on network-based information systems (NBiS), Melbourne, Australia; 2012. p. 429–34.

Chapade SS, Pandey KU, Bhade DS. Securing cloud servers against flooding based ddos attacks. In: Proceedings of IEEE international conference on communication systems and network technologies (CSNT), Gwalior, India; 2013. p. 524–8.

Chandola V, Arindam B, Vipin K. Anomaly detection: a survey. ACM Comput Surv (CSUR) 2009;41(3):1–58.

Chonka A, Xiang Y, Zhou W, Bonti A. Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks. J Netw Comput Appl 2011;34(4):1097–107.

Chen Z, Xu G, Mahalingam V, Ge L, Nguyen J, Yu W, Lu C. A cloud computing based network monitoring and threat detection system for critical infrastructures. Big Data Res 2015, http://dxdoi.org/10.1016/j.bdr.2015.11.002.

Darwish M, Ouda A, Capretz LF. Cloud-based DDoS attacks and defenses. In: Proceedings of the IEEE international conference on information society (i-Society), Toronto, Canada; 2013. p. 67–71.

Deshmukh RV, Devadkar KK. Understanding DDoS attack & its effect in cloud. Environ Procedia Comput Sci 2015;49:202–10.

Dantas YG, Nigam V, Fonseca IE. A selective defense for application layer DDoS attacks. In: Proceedings of IEEE joint intelligence and security informatics conference (JISIC), Hague, Netherlands; 2014. p. 75–82.

Dou W, Chen Q, Chen J. A confidence-based filtering method for DDoS attack defense in cloud environment. Future Gener Comput Syst 2013;29(7):1838–50.

De Oca V, Jeske D, Zhang Q, Rendon C, Mazda M. A cusum change-point detection algorithm for non-stationary sequences with application to data network surveillance. J Syst Softw 2010;83(7):1288–97.

Ficco M, Rak M. Stealthy denial of service strategy in cloud computing. IEEE Trans Cloud Comput 2015;3(1):80–94.

Ficco M, Venticinque S, Di Martino B. An advanced intrusion detection framework for cloud computing. Comput Syst Sci Eng 2013;28(6):401–11.

Ficco M. Security event correlation approach for cloud computing. Int J High Perform Comput Netw 2013;7(3):173–85.

Ficco M, Palmieri F. Introducing fraudulent energy consumption in cloud infrastructures: a new generation of denial-of-service attacks. IEEE Syst J 2015;99: 1–11.

Gonzalez N, Miers C, Redigolo F, Naslund M, Pourzandi M. A quantitative analysis of current security concerns and solutions for cloud computing. J Cloud Comput 2012;1(1):1–18.

Gruschka N, Iacono LL. Vulnerable cloud: Soap message security validation revisited. In: Proceedings of IEEE international conference on web services (ICWS 2009), Los Angeles, USA; 2009. p. 625–31.

Gul I, Hussain M. Distributed cloud intrusion detection model. Int J Adv Sci Technol 2011;34:71–82.

Gupta S, Kumar P. Vm profile based optimized network attack pattern detection scheme for ddos attacks in cloud. In: Proceedings of international symposium of security in computing and communications (SSCC 2013), Mysore, India; 2013. p. 255–61.

Girma A, Garuba M, Li J, Liu C. Analysis of DDoS attacks and an introduction of a hybrid statistical model to detect DDoS attacks on cloud computing environment. In: Proceedings of 12th IEEE international conference on information technology—new generations (ITNG), Las Vegas, USA; 2015. p. 212–7.

Gupta S, Kumar P, Abraham A. A profile based network intrusion detection and prevention system for securing cloud environment. Int J Distrib Sensor Netw 2013:1–12.

Guenane F, Nogueira M, Pujolle G. Reducing DDoS attacks impact using a hybrid cloud-based firewalling architecture. In: Proceedings of IEEE global information infrastructure and networking symposium (GIIS), Montreal, Canada; 2014. p. 1–6.

Garsva E, Paulauskas N, Grazulevicius G, Gulbinovic L. Packet inter-arrival time distribution in academic computer network. ElektronikairElektrotechnika 2014;20(3):87–90.

Hormati M, Khendek F, Toeroe M. Towards an evaluation framework for availability solutions in the cloud. In: Proceedings of IEEE international symposium on software reliability engineering workshops (ISSREW); 2014. p. 43–6.

Huang VS, Huang R, Chiang M. A DDoS mitigation system with multi-stage detection and text-based turing testing in cloud computing. In: Proceedings of 27th IEEE 27th international conference on advanced information networking and applications workshops (WAINA), Barcelona, Spain; 2013. p. 655–62.

Idziorek J, Tannian M, Jacobson D. Attribution of fraudulent resource consumption in the cloud. In: Proceedings of 5th IEEE international conference on cloud computing (CLOUD), Honolulu, Hawaii; 2012. p. 99–106.

Iyengar NC, Ganapathy G, Kumar PM, Abraham A. A multilevel thrust filtration defending mechanism against DDoS attacks in cloud computing environment. Int J Grid Utility Comput 2014;5(4):236–48.

Ismail MN, Aborujilah A, Musa S, Shahzad A. Detecting flooding based DoS attack in cloud computing environment using covariance matrix approach. In: Proceedings of the 7th ACM international conference on ubiquitous information management and communication (ICUIMC'13), Kota Kinabalu, Malaysia; 2013. p. 36.

Iyengar N, Ganapathy G. Trilateral trust based defense mechanism against DDoS attacks in cloud computing environment. Cybern Inf Technol 2015;15(2): 119–40.

Idziorek J, Mark T, Tannian, Jacobson D. The insecurity of cloud utility models. IEEE IT Prof 2013;15(2):22–7.

Jeyanthi N, Iyengar NC. Escape-on-sight: an efficient and scalable mechanism for escaping DDoS attacks in cloud computing environment. Cybern Inf Technol 2013;13(1):46–60.

Jeyanthi N, Iyengar NC, Kumar PM, Kannammal A. An enhanced entropy approach to detect and prevent DDoS in cloud environment. Int J Commun Netw Inf Secur ((IJCNIS)) 2013a;5(2):119–40.

Jeyanthi N, Barde U, Sravani M, Tiwari V, Iyengar NC. Detection of distributed denial of service attacks in cloud computing by identifying spoofed IP. Int J Commun Netw Distrib Syst 2013b;11(3):262–79.

Joshi B, Joshi BK. Securing cloud computing environment against DDoS attacks. In: Proceedings of IEEE international conference on computer communication and informatics (ICCCI), Coimbatore, India; 2012. p. 1–5.

Jia Q, Wang H, Fleck D, Li F, Stavrou A, Powell W. Catch me if you can: a cloud-enabled DDoS defense. In: Proceedings of 44th annual IEEE/IFIP international conference on dependable systems and networks (DSN), Atlanta, GA; 2014. p. 264–75.

Jaber M, Cascella R, Barakat C. Can we trust the inter-packet time for traffic classification? In: Proceedings of the IEEE international conference on communications (ICC), Kyoto; June 5–9 2011. p. 1–5.

Khorshed Md Tanzim, Shawkat Ali ABM, Wasimi Saleh A. A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing. Future Gener Comput Syst 2012;28(6):833–51.

Karnwal T, Sivakumar T, Aghila G. A comber approach to protect cloud computing against XML DDoS and HTTP DDoS attack. In: Proceedings of IEEE students' conference on electrical, electronics and computer science (SCEECS), Bhopal, India; 2012. p. 1–5.

Karnwal T, Thandapanii S, Gnanasekaran A. A filter tree approach to protect cloud computing against xml ddos and http DDoS attack. Intell Inf 2013:459–69.

Kwon H, Kim T, Yu SJ, Sim HK. Self-similarity based lightweight intrusion detection method for cloud computing. In: Proceedings of 3rd international conference on intelligent information and database systems (ACIIDS), Daegu, Korea; 2011. p. 353–62.

Krishnan D, Chatterjee M. An adaptive distributed intrusion detection system for cloud computing framework. In: Proceedings of international conference of recent trends in computer networks and distributed systems security (SNDS), Trivandrum, India; 2012. p. 466–73.

Liebeskind JP. Keeping organizational secrets: protective institutional mechanisms and their costs. Ind Corp Change 2007;6(3):623–63.

Lonea AM, Popescu DE, Prostean Q, Tianfield H. Soft computing applications evaluation of experiments on detecting distributed denial of service (DDoS) attacks in eucalyptus private cloud; 2013. p. 367–79.

Lo CC, Huang CC, Ku J. A cooperative intrusion detection system framework for cloud computing networks. In: Proceedings of 39th IEEE international conference on parallel processing workshops (ICPPW), San Diego, USA; 2010. p. 280–4.

Lonea AM, Popescu DE, Tianfield H. Detecting DDoS attacks in cloud computing environment. Int J Comput Commun Control 2013;8(1):70–8.

Liu H. A new form of DOS attack in a cloud and its avoidance mechanism. In: Proceedings of the 2010 ACM workshop on cloud computing security workshop, Chicago, Illinois; 2010. p. 65–76.

Mell P, Grance T. The NIST definition of cloud computing; 2011.

Mishra A, Gupta BB, Joshi RC. A comparative study of distributed denial of service attacks, intrusion tolerance and mitigation techniques. In: Proceedings of the IEEE European intelligence and security informatics conference (EISIC), Athens, Greece; 2011. p. 286–9.

Modi C, Patel D, Borisaniya B, Patel H, Patel A, Rajarajan M. A survey of intrusion detection techniques in cloud. J Netw Comput Appl 2013;36(1):42–57.

Michelin R, Zorzo AF, De Rose C. Mitigating DoS to authenticate cloud REST APIs. In: Proceedings of 9th IEEE international conference for Internet technology and secured transactions (ICITST), London, UK; 2014. p. 106–11.

Modi CN, Patel DR, Patel A, Muttukrishnan R. Bayesian Classifier and Snort based network intrusion detection system in cloud computing. In: Proceedings of 3rd international IEEE conference on computing communication & networking technologies (ICCCNT), Coimbatore, India; 2012. p. 1–7.

Merlo A, Migliardi M, Gobbo N, Palmieri F, Castiglione A. A denial of service attack to UMTS networks using SIM-less devices. IEEE Trans Depend Secure Comput 2014;11(3):280–91.

Marnerides A, Spachos P, Chatzimisios P, Mauthe A. Malware detection in the cloud under ensemble empirical mode decomposition. In: Proceedings of IEEE International conference on computing, networking and communications (ICNC) and information security symposium, Garden Grove, CA; 2015. p. 82–8.

Negi P, Mishra A, Gupta BB. Enhanced CBF packet filtering method to detect DDoS attack in cloud computing environment. Int J Comput Sci Issues 2013;2(4): 142–6.

Nguyen T, Armitage G. A survey of techniques for internet traffic classification using machine learning. IEEE Commun Surv Tutor 2008;10(4):56–76.

[online] 〈http://www.troyhunt.com/2013/01/what-is-loic-and-can-i-be-arrested-for.html〉.

[online] 〈http://resources.infosecinstitute.com/dos-attacks-free-dos-attacking-tools/〉.

[online] 〈https://stormsecurity.wordpress.com/2009/03/03/application-layer-ddos-simulator/〉.

[online] 〈http://www.hackerschronicle.com/2014/03/davoset-most-powerfull-ddos-tool.html〉.

[online] 〈http://security.radware.com/knowledge-center/DDoSPedia/Pyloris/〉.

Osanaiye O. IP spoofing detection for preventing DDoS attack in cloud computing. in: Proceedings of 18th IEEE international conference on intelligence in next generation networks (ICIN), Paris, France; 2015. p. 139–41.

Online The Truth about DDoS Attacks: Part 1 〈http://www.carbon60.com/the-truth-about-ddos-attacks-part-1/〉; 2013. [Accessed: 12.07.15].

Osanaiye O, Dlodlo M. TCP/IP header classification for detecting spoofed DDoS attack in Cloud environment. In: Proceedings of 16th IEEE international conference on computer as a tool (EUROCON 2015), Salamanca, Spain; 2015. p. 1–6.

Prabadevi B, Jeyanthi N. Distributed denial of service attacks and its effects on cloud environment—a survey. In: Proceedings of the IEEE international symposium on networks, computers and communications; 2014. p. 1–5.

Palmieri F, Fiore U, Castiglione A. A distributed approach to network anomaly detection based on independent component analysis. Concurr Comput: Pract Exp 2014;26(5):1113–29.

Palmieri F, Ricciardi S, Fiore U, Ficco M, Castiglione A. Energy-oriented denial of service attacks: an emerging menace for large cloud infrastructures. J Supercomput 2013;71(5):1620–41.

Prokhorenko V, Choo K-KR, Ashman K. Web application protection techniques: a taxonomy. J Netw Comput Appl 2016;60:95–112.

Palmieri F, Ficco M, Castiglione A. Adaptive stealth energy-related DoS attacks against cloud data centers. In: Proceedings of 8th IEEE international conference on innovative mobile and internet services in ubiquitous computing (IMIS), Birmingham, England; 2014. p. 265–72.

Polunchenko A, Tartakovsky A. State-of-the-art in sequential change-point detection. Methodol Comput Appl Probab 2012;14(3):649–84.

Peng T, Leckie C, Ramamohanarao K. Proactively detecting distributed denial of service attacks using source IP address monitoring. In: Proceedings of 3rd international networking conference (IFIO-TC6), Athens, Greece; May 9–14 2004. p. 771–82.

Quick D, Choo K-KR. Google drive: forensic analysis of data remnants. J Netw Comput Appl 2014;40:179–93.

Rui X, Wen-Li M, Wen-Ling Z. Defending against UDP flooding by negative selection algorithm based on eigenvalue sets. In: Proceedings of IEEE fifth international conference on information assurance and security (IAS'09), Xi'an, China, vol. 2; 2009. p. 342–5.

SC Magazine, Website, 〈http://www.scmagazineuk.com/new-hacking-group-ddos-attacks-amazons-twitch-us-state-websites/article/405796〉; 2015.

Sqalli MH, Al-Haidari F, Salah K. Edos-shield-a two-steps mitigation technique against edos attacks in cloud computing. In: Proceedings of the fourth IEEE international conference on utility and cloud computing (UCC), Victoria, Australia; 2011. p. 49–56.

Shamsolmoali P, Zareapoor M. Statistical-based filtering system against DDOS attacks in cloud computing. In: Proceedings of international conference on advances in computing, communications and informatics (ICACCI), New Delhi, India; 2014. p. 1234–9.

Swathi RM. DDoS attacks on the rice: increased 132% in Q2 2015, HTTPS most targeted. Available from: 〈http://dazeinfo.com/2015/08/28/internet-security-ddos-attacks-china-australia-us-uk-akamai/〉.

Srivastava M, Wu Y. Comparison of EWMA, CUSUM and Shiryayev–Roberts procedures for detecting a shift in the mean. Ann Stat 1993:645–70.

Tsai WT, Sun X, Balasooriya J. Service-oriented cloud computing architecture. In: Proceedings of the seventh IEEE international conference of information technology: new generations (ITNG); 2010. p. 684–9.

Teng S, Zheng C, Zhu H, Liu D, Zhang W. A cooperative intrusion detection model for cloud computing networks. Int J Secur Appl 2014;8(3):107–18.

Tavallaee M, Bagheri E, Lu W, Ghorbani A. A detailed analysis of the KDD CUP 99 data set. In: Proceedings of the second IEEE symposium on computational intelligence for security and defense applications (CISDA 2009), Ottawa, Canada; 2009. p.1–6.

Takada H, Hofmann U. Application and analyses of cumulative sum to detect highly distributed denial of service attacks using different attack traffic patterns; 2004. 〈http://www.ist-intermon.org/dissemination/newsletter7.pdf〉.

Vissers T, Somasundaram TS, Pieters L, Govindarajan K, Hellinckx P. DDoS defense system for web services in a cloud environment. Future Gener Comput Syst 2014;37:37–45.

Varadharajan V, Tupakula U. Counteracting security attacks in virtual machines in the cloud using property based attestation. J Netw Comput Appl 2014;40: 31–45.

Wayne J. Cloud hooks: security and privacy issues in cloud computing. In: Proceedings of the IEEE 44th Hawaii international conference on system sciences (HICSS); 2011. p. 1–10.

Wang B, Zheng Y, Lou W, Hou Y. DDoS attack protection in the era of cloud computing and software-defined networking. Comput Netw 2015;81:308–19.

Wong F, Tan CX. A survey of trends in massive DDoS attacks and cloud-based mitigations. Int J Netw Secur Appl (IJNSA) 2014;6(3):57–71.

Wang B, Zheng Y, Lou W, Hou Y. DDoS attack protection in the era of cloud computing and software-defined networking. In: Proceedings of the 22nd IEEE international conference on network protocols (ICNP), Raleigh, NC; 2014. p. 624–9.

Wang H, Danlu Z, Kang G. Change-point monitoring for the detection of DoS attacks. IEEE Trans Depend Secur Comput 2004;1(4):193–208.

Yang L, Zhang T, Song J, Wang J, Chen P. Defense of DDoS attack for cloud computing. In: Proceedings of IEEE international conference on computer science and automation engineering (CSAE), vol. 2, Zhangjiajie, China; 2012. p. 626–9.

Yu S, Tian Y, Guo S, Wu DO. Can we beat DDoS attacks in clouds? IEEE Trans Parallel Distrib Syst 2014;25(9):2245–54.

Yan Q, Yu R, Gong Q, Li J. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges. IEEE Commun Surv Tutor 2015;99:1–23.

Zargar ST, Joshi J, David T. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun Surv Tutor 2013;15(4):2046–69.

Zakarya M. DDoS verification and attack packet dropping algorithm in cloud computing. World Appl Sci J 2013;23(11):1418–24.

Zhang J, Zou C, Wang Z. An adaptive Shiryaev–Roberts procedure for monitoring dispersion. Comput Ind Eng 2011;61(4):1166–72.

Zhou Z, Chen X, Wang J, Li X. A distributed detection scheme based on adaptive CUSUM and weighted CAT against DDoS Attacks. In: Proceedings of the 3rd international conference on multimedia technology (ICMT 2013), Guangzhou, China; 2013. p. 97–105.