Ann. Telecommun. manuscript No.
(will be inserted by the editor)

Service Resizing for Quick DDoS Mitigation in Cloud Computing Environment

Gaurav Somani · Manoj Singh Gaur · Dheeraj Sanghi · Mauro Conti · Rajkumar Buyya

Received: March 15, 2016 / Accepted: date

Abstract Current trends in distributed denial of service (DDoS) attacks show variations in terms of attack motivation, planning, infrastructure, and scale. "DDoS-for-Hire" and "DDoS mitigation as a Service" are the two services, which are available to attackers and victims respectively. In this work, we provide a fundamental difference between a "regular" DDoS attack and an "extreme" DDoS attack. We conduct DDoS attacks on cloud services, where having the same attack features, two different services show completely different consequences, due to the difference in the resource utilization per request. We study various aspects of these attacks and find out that the DDoS mitigation service's performance is dependent on two factors. One factor is related to the severity of the "resource-race" with the victim web-service. Second factor is "attack cooling down period" which is the time taken to bring the service availability post detection of the attack. Utilizing these two important factors, we propose a supporting framework for the DDoS mitigation services, by assisting in reducing the attack mitigation time and the overall downtime. This novel framework comprises of an affinity based victim-service resizing algorithm to provide performance isolation, and a TCP tuning technique to quickly free the attack connections, hence minimizing the attack cooling down period. We evaluate the proposed novel techniques with real attack instances and compare various attack metrics. Results show a significant improvement to the performance of DDoS mitigation service, providing quick attack mitigation. The presence of proposed DDoS mitigation support framework demonstrated a major reduction of more than 50% in the service downtime.

Keywords Cloud Computing · Distributed Denial of Service Attack (DDoS) · Economic Denial of Service Attack (EDoS) · Security and Protection

1 Introduction

DDoS attack is one of the most notorious attacks, among the list of major cyber attacks in the recent past. There is a large number of attack incidents that make the Internet-based businesses unavailable and riskier. Cloud computing based services and infrastructures are among the favorite targets of DDoS attackers [44]. The growth of cloud computing attributes to the features like a utility-based business model, high availability, low pricing, and no maintenance costs. Similarly, the rise in the frequency of cloud targeted attacks is also attributed to the nature of cloud hosting services and the business model. Cloud business model works on the "Pay-as-you-Go" basis, which enables the hosted services to acquire as many resources, as they need. Interestingly, the same set of features and facilities of cloud computing

Gaurav Somani
Central University of Rajasthan, India
Malaviya National Institute of Technology, India
E-mail: gaurav@curaj.ac.in

Manoj Singh Gaur
Malaviya National Institute of Technology, India E-mail: gaurms@mnit.ac.in

Dheeraj Sanghi
Indian Institute of Technology, Kanpur, India E-mail: dheeraj@cse.iitk.ac.in

Mauro Conti
University of Padua, Padua, Italy E-mail: conti@math.unipd.it

Rajkumar Buyya
The University of Melbourne, Australia E-mail: rbuyya@unimelb.edu.au

are also available to DDoS attackers. There is a number of "DDoS-for-Hire" services also known as booters/stressers [33] which provide attack infrastructure as a service, in the form of botnets or cloud platform (also known as BotClouds) [24]. The representation of DDoS attacks in the form of "arms-race" is getting realistic after the emergence of cloud-based services and subsequently cloud targeted attacks and cloud originated attacks [23].

The model of utility computing adds a direction of economic losses attributed to the fraudulent resource consumption [11] (also known as Economic Denial of Sustainability (EDoS [5]) attacks). Recent attack incidents on Linode, Amazon EC2 services, and Rackspace cloud services are prominent examples of DDoS attack on cloud services [44]. There are some surveys, which provide DDoS attack statistics and studies in the recent past [3][31]. In these studies, the cloud targeted attacks are getting larger share among the total DDoS attacks [27]. There are detailed surveys and recent contributions on the solutions to DDoS attacks in cloud computing environment [41]. These works and attack incidents also necessitate the requirement to understand the DDoS attack dynamics and designing specific solutions for cloud computing. Many of the recent mitigation solutions utilize a profound amount of resources available in the cloud [19][45][49][51]. Service providers also recommend and use the resource scaling techniques to perform efficient and quick DDoS mitigation [4]. In these techniques, more and more resources are given to the affected service so that it can handle the incoming attack and carry out the mitigation.

Resource scaling comes with the cost of additional resources which subsequently has impact on the budget/sustainability of enterprises. Controlling cost is more important for the SMEs (Small and Medium Enterprises) as they have many limitations on their budgets. We argue that the DDoS attack mitigation costs should ideally be less than the losses arising out due to the attack without mitigation. Additionally, the resource scaling strategies on cloud should be able to identify the real resource requirements. The auto-scaling service should be able to discard the fake resource alarms generated due to DDoS attacks. Authors in [42] have proposed a DDoS aware resource allocation algorithm to scale when there is a real requirement.

In this work, we critically ponder on much finer grain performance issues of DDoS mitigation process at the level of the victim service run in a virtual machine (VM). These issues are related to the resource availability during the DDoS attacks and the difference in resource usage of various attacks. We show the DDoS attack dynamics by doing real attack experiments on cloud services. Based on these experiments, we observe that DDoS attacks may take different shapes based on the attack features and available resources on the server. In most of the real incidents, DDoS attacks take a form of "extremely unavailable DDoS (extreme DDoS)", in which case, all the services (including the victim service) are inaccessible to do the mitigation and recovery. In these cases, due to the heavy resource contention among attack requests and other system services, the DDoS mitigation methods may not get a chance to act and perform mitigation in time. Adding more and more resources to the victim service, may not always help the mitigation. In cloud computing infrastructure, resources always come with a cost, hence the mitigation methods should spend resources carefully.

With these observations, we provide a novel DDoS mitigation support and resource management framework, which provides support services to the DDoS mitigation mechanisms to perform quick and sustainability aware mitigation on cloud services. By "quick", we aim to minimize the service downtime, time to mitigate and attack cooling down period. The "sustainability" or the cost aspect is addressed by providing mitigation by using the available resources within the service instead of resource scaling. We argue that during "extreme DDoS", sacrificing the resources by the victim web-service and utilizing those freed resources for the DDoS mitigation service can provide a quick, sustainability aware in-resource mitigation. The proposed framework achieves these goals by providing two important features, (i) Resource Shrinking and Expanding and (ii) TCP tuning.

The rest of the paper is organized as follows: We detail the DDoS attack mitigation and attack dynamics in cloud computing in Section 2. In the Section 3, we illustrate the various attack instance on a cloud-based service and different aspects of DDoS attacks in cloud computing. Our novel contributions are highlighted in detail in Section 5. In Section 6, we evaluate the proposed technique with various experiments and analyse the results. In Section 7, we present the related work and contributions in the area. Finally in Section 8, we conclude and discuss the possible future directions.

2 DDoS Attack and Mitigation in Cloud Services: State of the art

We show the DDoS attack scenario in Figure 1. Cloud computing infrastructure consists of multiple high capacity physical servers connected with high speed networking. The physical servers are managed by a cloud middle-ware framework to support resource allocation

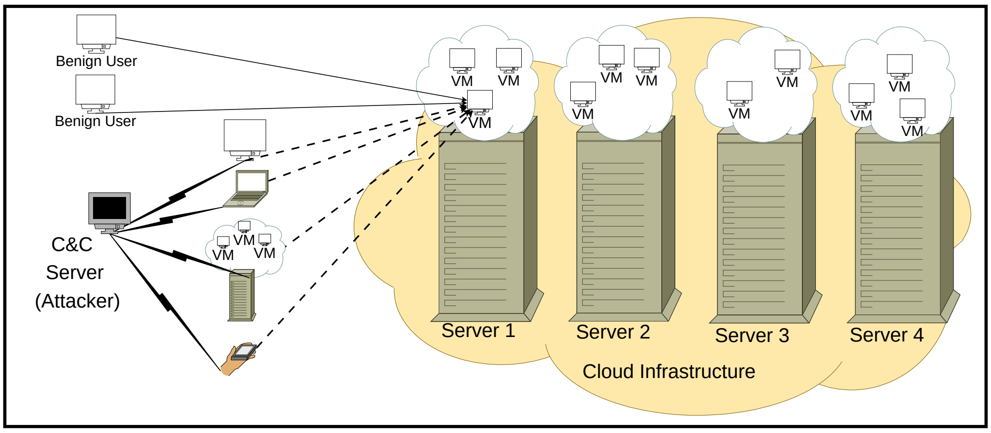


Fig. 1: DDoS Attack in Cloud

and management, fault tolerance, and resource accounting. In most of the cases, the physical servers are virtualized to host and run virtual machines. These VMs are resource abstractions of physical servers to support easy deployment, multi-tenancy, improved resource utilization and other services like service migration, backup-recovery and cloning. Cloud VMs run a variety of services such as web, ftp, and database servers to HPC nodes as part of large computing platform. Attackers targeting a DDoS attack on these services may follow the traditional DDoS attack model, where a command and control (C&C) server directs large number of bots to send the attack traffic. The consequence of such attack is usually "Service Denial" for the legitimate users. There are incidents of using the "DDoS-for-Hire" services [33] as an inexpensive attack infrastructure to launch the attack. In Figure 1, we also include cloud originated attacks to show the attackers utilizing cloud capabilities [24]. Other attack infrastructures include malware infected computers, phones and servers [41].

Attack effects and losses distinguish a cloud targeted DDoS attack from a DDoS attack targeted at fixed on-premise service. These effects include sustainability/economic losses due to auto-scaling, collateral damages due to multi-tenancy to non-targets and additional costs of attack mitigation [42,40,51]. All these effects are in addition to the visible service downtime and other long-term and short terms business and reputation losses. A detailed analysis of DDoS attack threat model is presented in [38].

DDoS attacks and their growth can be measured by the pattern of peak attack bandwidth in the last one decade. As per report in [3], the DDoS attack peak bandwidth in the year 2015 has reached the record mark of 500 Gbps, which was only 8 Gbps in 2004. We can also see a trend of exponential growth in the number of attacks each quarter in last few years [27]. After the emergence of cloud computing, there is a growing number of attack instances originated from the cloud servers, utilizing the resources of cloud [24]. At the same time, there are multiple mitigation methods in the market, which provide cloud-based mitigation as a service. These trends are supporting the analogy of DDoS attacks as "arms-race", where the "arms" are either resources or overall spendings on attack launching or attack mitigation. There is a number of attack incidents, in which the victims face heavy losses during and post attacks. A mitigation method should always be less costly as compared to the losses suffered by the attacks. Multiple news items suggest about the massive costs of attacks and their mitigation [43]. However, for most of the small and medium enterprises (SMEs), a huge cost of DDoS attack mitigation cannot be justified. If we see the trends of 2015, more than 84% of the reported DDoS attacks were having peak bandwidth less than 1 Gbps. More than 46% of the target services are web-services offering business and enterprise services.

A similar trend is shown by "DDoS for Hire" services, where the maximum attack bandwidth is around 1 Gbps [33]. At the same time, more than 33% of the reported DDoS attacks target cloud infrastructure based services. Also, most of the organizations are running less than 50 VMs in public clouds based on their scale and adoption patterns [17]. These facts provide very important insight for designing DDoS mitigation solutions to a large portion of DDoS victims. As most the DDoS attacks with small footprint target SMEs, having a limited infrastructure based on the requirements and budgets. Therefore, we require organizational sustainability-aware solutions. These solutions should continue to provide safety from the attack related effects with minimum downtime of the services.

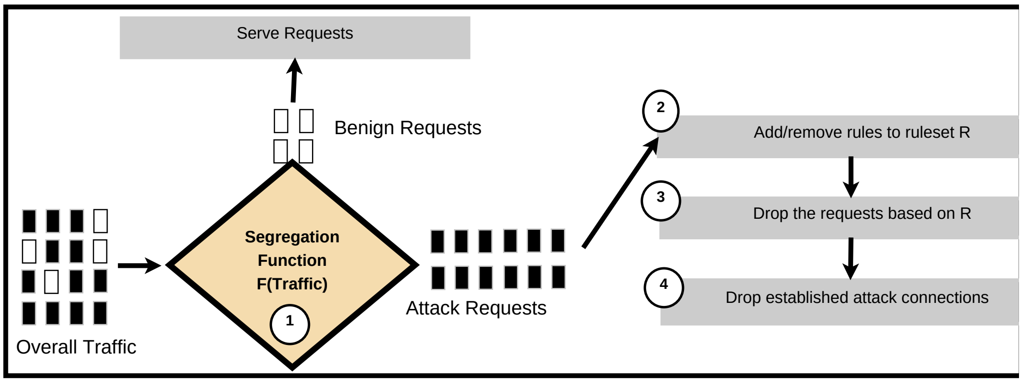

Serve Requests
Benign Requests
Segregation Function F(Traffic)
1
Overall Traffic
Attack Requests
2
Add/remove rules to ruleset R
3
Drop the requests based on R
4
Drop established attack connections

Fig. 2: Generic Architecture of DDoS Defense Mechanisms

2.1 Generic Architecture of Defense Mechanisms

DDoS solutions in the literature and the marketplace, mostly follow a generic architecture. Most of the DDoS mitigation methods are traffic evaluation based processes, where the principal aim of the process, is to differentiate attackers and victim source addresses by a segregation function `F(Traffic)`. As shown in Figure 2, a DDoS mitigation method will have following major activities.

1. Segregation: `F(Traffic)` is usually an online algorithm, which processes the incoming traffic and filters the traffic on the basis of various features including request patterns and connection patterns. These features of benign or attack traffic behavior help in filtering out the attack traffic. The remaining traffic is assumed to be legitimate traffic and served by the service. There are a large number of such segregation functions, which are discussed in surveys on DDoS mechanisms [41].
2. Add/Remove rules: When the function `F(Traffic)` identifies the attackers, mitigation mechanism will add the attacker source addresses in the block ruleset `R` to enforce the incoming attack connection drop. This activity is performed by a firewall service like `iptables` and APF [26].
3. Drop attack requests: Based on the rules maintained by `R`, the firewall, which is also a part of the segregation function, drops the attack requests.
4. Drop established attack connections: The final important activity, which is dependent upon the output of the segregation function, `F(Traffic)`, would close all the established connections involving attacker addresses. This activity is usually performed by sending a connection close (e.g. reset) packet using mechanisms offered by TCP.

The above four activities are online in nature and may run simultaneously, specially for attacks instances that change attack vectors or repetitive. We would like to highlight the fact that most DDoS mitigation solutions based on some traffic evaluation, follow the above generic architecture. We would also like to emphasize the fact that our solution framework is flexible and open to use any segregation function `F(Traffic)`.

3 DDoS Attack and its Mitigation: A Real Time Experimental Case Study

In this section, we provide a discussion on the results of few interesting attacks launched and targeted to example web services. To conduct attack experiments, we create attack instances on a service with the configuration given in Table 1. These attack instances help us in answering few important questions related to DDoS attack defense in the cloud services. We detail these questions later in this section with an effort to find out the answers in experiments, and then when we evaluate the proposed work in Section 6. The infrastructure available to the service is similar to a “C4 Extra Large”

Resource	**Configuration**
Physical Server	Dell PowerEdge Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz
Total CPUs	8 Processors (4 cores each)
Total Memory	96GB
Hypervisor	XenServer 6.5
Vicitm/Attacker/Benign OS	Ubuntu 14.04
Victim Service	Dynamic Web service Apache2-PHP
Victim Configuration	4CPUs and 8 GB
Attacker Configuration	2 CPUs and 1 GB
Benign Configuration	2 CPUs and 1 GB
Attacker/Benign Application	ApacheBench2
Attack Traffic	500 concurrent requests (Total 5000 requests)
Benign Traffic	1 concurrent request (Total 100 requests)
Network	1 Gbps

Table 1: Attack Setup

instance on Amazon EC2, which has 4 vCPUs, 7.5 GB RAM and on a 64-bit platform. We design the attack traffic by following the classical work in [25]. The web-service under attack is a representative service of most of the modern web services. This dynamic web service runs an image conversion program, which converts an image from one format to the other. We are converting a .jpeg image to .gif images for our experiments. As discussed in Section 2, we are not using a particular mitigation mechanism but a generic representative of many mitigation methods. The DDoS mitigation mechanism or segregation function (`R`) becomes a supplement to our discussions. Hence any other mitigation method can be used in place of the mechanism used in this work. We are mostly interested in the cost-resource dynamics while the mitigation approach carries its activity in the presence of an attack. In all the experiments, we are using a popular open source DDoS mitigation mechanism, DDoS-Deflate [13]. This tool is a connection count based filter working on top of `netstat` utility to count connections by each sender. We have used this in its default configuration, which flags an address "attacker" if it tries to establish more than 150 concurrent connections. Let us consider the three attack instances and their impact on a benign user, which is accessing the service. In the first three instances (figures 3, 4, and 5), the victim service is converting a 500KB file on each request. If the service is not under an attack, each request usually takes a response time of around∼900ms. Figure 3 shows the victim service behavior experienced at benign user's end when there were resources similar to C4 Extra Large instance. In all the experiments, attack and benign traffic arrival is scheduled in such a manner that they start sending the requests at the same time but with a large difference in their request frequency. Additionally, we have made the request timeout values very high (10000s) to see the attack impacts without missing any reponses to the requests. The attack starts its impact on the service from the very first request and makes the victim service unavailable instantly. The mitigation mechanism starts its work with a frequency run of each 5 seconds. The attack gets detected and reported after∼36 seconds of attack start. The total downtime of the victim service is 939s. We repeat the same attack by giving more resources in figures 4 and 5. We made this "resource increase" in the CPU resources to see if the attack mitigation gets fastened. "Resource increase" is also motivated by the recommendations made by many DDoS mitigation mechanisms about scaling the resources during the attack [42, 51]. In the 8vCPUs-8GB and 12vCPU-8GB victim instances, we see insignificant difference from the perspective of attack detection, reporting time and total victim service downtime.

Now, we initiate the same attack with a change in the victim service's behavior. Instead of processing a 500KB image, we use the image size of 2MB. This change is in consonance with the average page size of the web pages across the globe from a popular survey [8]. A single request to the web server for this image takes around 4.5s while there is no attack. We have performed two experiments with the attack configuration as shown in Table 1 with 2MB image size. The resultant request graphs are shown in figures 6 and 7. In this case, the service behavior changes completely and the response time becomes very high due to the image size. However, we were unable to know when the mitigation service detects the attack. Mitigation service reported the attack only when the attack effects are over. The resource usage in these attack instances was maximum (CPU and memory usage reaching to 100%), and no service was available during the whole attack period, till the time when all the attack effects subsides. On the other hand, in all the three attack instances (figures 3, 4 and 5), we were able to use the victim server for other services, during the victim service downtime. Another crucial fact to observe, as in many of the attack instances, the victim server's other services (e.g. *ssh*) also becomes unavailable due to the intensity of the attack and the heavy resource usage. The attack may lead to resource starvation for all the services on the machine. It is also worth highlighting that the resource starvation or the "resource race" (as pointed in [39]) was so severe that the mitigation service could not even report the detection of the attack. We configured the mitigation service to write the time of mitigation activities such as detection and reporting in a file. The reporting of these activities was very comfortably done in the initial three attack instances. However, in the case of Figure 6, the mitigation service could not even get the file in the memory and write into it. We term this attack scenario as "extreme DDoS" as there is an extreme service denial at the server.

Figure 7 shows an instance in which we repeat the attack after the first attack effects are over. Recent attack reports show attack incidents where attacks are repeated a "stop-start-cycle" with attacks repeated after few minutes to hours [27]. Repetition may come with changes in attack vector, sources and size. In this case, once the service is recovered from the attack, we start the next attack after 10 minutes. In attack repetition, we see a repetition of effects of the "extreme DDoS". In the attack repetition cases, the detection time of the attack is not known and service is only available after the attack effects are completely over. We have summa-

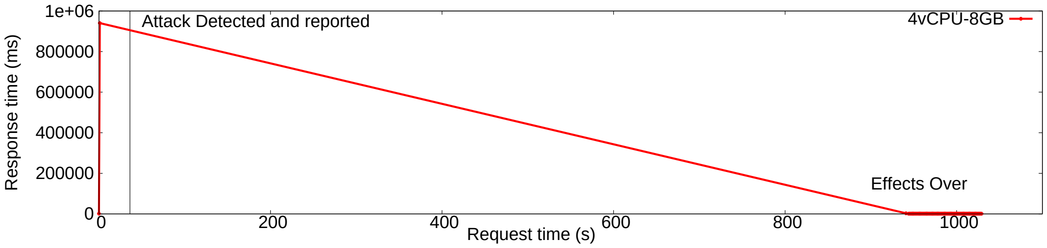


Fig. 3: Request Response Behavior during a DDoS Attack (Resources on Victim Service=4vCPU-8GB)

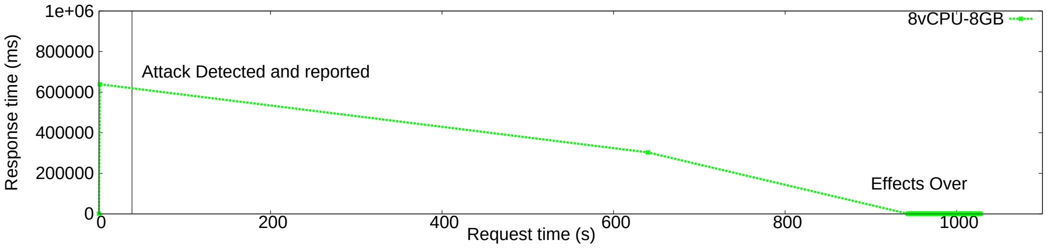


Fig. 4: Request Response Behavior during a DDoS Attack (Resources on Victim Service=8vCPU-8GB)

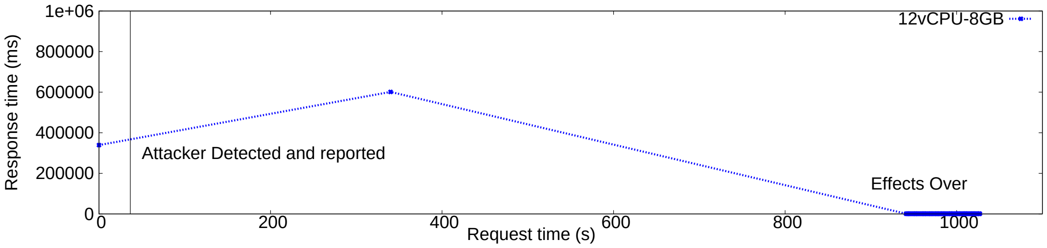


Fig. 5: Request Response Behavior during a DDoS Attack (Resources on Victim Service=12vCPU-8GB)

rized the results of all four attack instances in Table 2. Attack repetition results are given in Table 3. It is clear that in the repeated attack instances are equivalent to "two" individual extreme DDoS attacks with large attack cooling down period.

4 DDoS Mitigation Requirements: Discussion

Based on the attack instances and the outcomes, we discuss and design five important requirements related to DDoS attacks on cloud services. These questions are equally relevant to the cases of DDoS attacks to "fixed" infrastructure services. Based on these observations and design requirements, we propose our DDoS mitigation framework in the next section.

4.1 R1: DDoS mitigation in the presence of attack

DDoS attacks aim to create "denial" of the victim service. Victim service becomes unavailable due to the lack of resources and more and more incoming requests. In this case, DDoS mitigation method, as well as victim service both, need more resources, which are not available readily. Mitigation behavior is quite visible when we differentiate a "DDoS' attack and "extreme DDoS" attack instance. In "DDoS" (Figure 3), the mitigation mechanism was able to work in the presence of attack and could perform the mitigation activities like adding rules, dropping the subsequent connection and terminating the established attack connections while reporting the attack. On the other hand, in the case of "extreme DDoS", getting the required resources to perform the activity was very competitive as resources like CPU time and memory were heavily used by web service. Hence, it becomes difficult for the victim service owner to monitor the state of the victim service. Providing additional support in the form of fault-tolerance and recovery is also difficult without accessing the service. Additional support can be given by providing additional instances, other essential resources like memory, and by monitoring the situation manually for attacks or even vulnerabilities. These support and recovery mechanisms always require information from the victim server, as without knowing the state and gain-

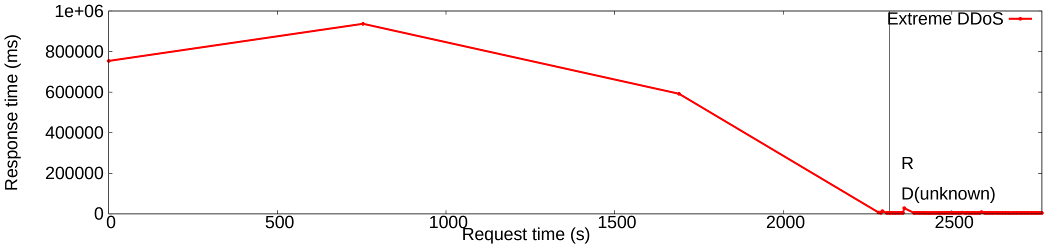


Fig. 6: Request Response Behavior during a Extreme DDoS Attack (R= Time of reporting and D = Time of detection)

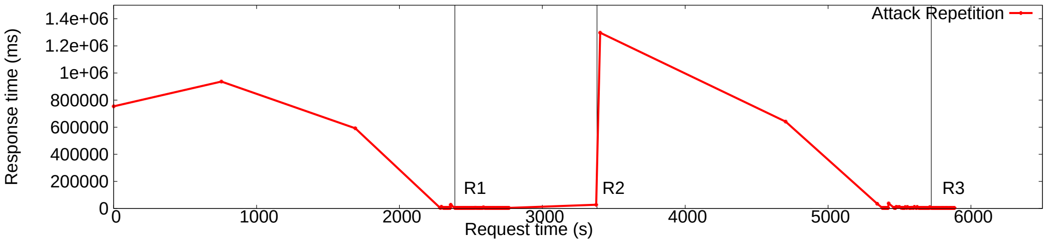


Fig. 7: Request Response Behavior during Repeated Extreme DDoS Attacks (R1= Time of reporting attack 1, R2 = Start of attack 2, and R3= Time of reporting attack 2)

Attack	**Resources**	**Time of Attack Detection**	**Time of Attack Reporting**	**Downtime (Victim Service)**	**Downtime (Other Services)**	**No. of Attack requests served before detection**
DDoS	4vCPU-8GB	36s	0m36s	939s	0s	45
DDoS	8vCPU-8GB	39s	0m39s	943s	0s	394
DDoS	12vCPU-8GB	37s	0m37s	941s	0s	452
Extreme DDoS	4vCPU-8GB	Unknown	2315s	2294s	2294s	27

Table 2: Various Attack Metrics

ing the access, no other supports are useful. We need mechanisms that can help in providing access to other services and resources for mitigation mechanisms, even in the presence of extreme DDoS attacks.

4.2 R2: Victim service availability after the attack mitigation/attack duration

Attack mitigation has multiple facets such as attackers identification, blocking and clearing established connections. The time taken by each of these activities is important to estimate the overall downtime and subsequent service availability time. We see in Table 2, that even though the attack was detected at 38s the service became available after a much longer time (940s). We term this time period as "Attack Cooling Down Period, T_C", which is the total time taken by the services to recover after the attack is detected. There are a number of contributions available to perform quick DDoS mitigation; however, there are no contributions towards quantifying or reducing the T_C. Our work makes novel contributions in the directions of minimizing T_C.

4.3 R3: Availability of other services during the attack period

Most administrators report about the unavailability of any access channel (including manual terminal access) to the victim service during the attack. Even cloud-based mitigation methods require a channel to perform the mitigation at the victim side. In all the extreme attack instances, the interactive services to access the victim server were unavailable. We are also interested to consider the performance of other critical services in the presence of an attack to the victim service. Performance of these services can be monitored by considering the availability (or intermittent availability) and the response time. We detail the availability aspect of other services in Section 6.1.

Attack in Repetition	Time of Attack Detection	Time of Attack Reporting	Downtime (Victim Service)	Downtime (Other Services)	No. of Attack requests served before detection
Attack 1	Unknown	2388s	2290s	2290s	40
Attack 2	Unknown	2345s	2259s	2259s	42

Table 3: Attack Repetition Results

4.4 R4: Effect of scaling on mitigation and sustainability/costs

Most of the cloud hosting service providers propose to mitigate DDoS attacks by scaling the service [45] [53] [51] to quickly mitigate the effects utilizing the enhanced resources. In DDoS attack cases, we did not observe significant change in overall downtime, attack detection and reporting time even with scaled up resources. On the other hand, an interesting statistics relates the total "attack" requests served during an onslaught. If we have more and more resources available on the server, then the attack requests entering the service queue will rise up before they get classified as "attack". In this case, the service will try to respond to more and more attack requests and make the detection difficult due to the "resource race". Additionally, if we look at the size of outgoing bandwidth, the victim service spends to serve these attack requests. The bandwidth will be directly proportional to the no. of attack requests entered into the system. In any sophisticated attack, if the attack detection does not take place even after a long time, and if we infuse more and more resources (anticipating a quick detection and mitigation), the system will have large number of requests resulting in massive attack mitigation costs. The attack mitigation cost include the cost of additional resources in terms of additional physical resources or VM instances and the cost of the mitigation software. These direct costs exclude the other costs such as business losses and penalties due to downtime. Incoming bandwidth to a cloud-based service is free up to an extent; however, the more expensive outgoing bandwidth may result in severe economic losses [4].

There is a high probability of detecting the attack quickly with very large amount of resources [51]. Though cloud computing utility models follow hourly pricing model, still, it may reach to a multi-fold sum of plain hosting costs without attacks. Costs of losses become significant if the attacks continue for a significantly long period or repeated or launched with additional sophistication. On the other hand, a secure remote accessibility to the victim machines is still needed to employ the mechanisms. In the case of network layer/overlay based detection this might not be entirely true; however, most of the application layer methods will surely need the access.

4.5 R5: Repeated/prolonged attacks and variable attack vector

There are many attack incidents, where the attacks continue for longer duration and bring variations in attack features [27]. At times, sophisticated attacks try to defeat the mitigation mechanism by stealth [7]. As per the attack reports by Arbor Networks [27], repeated attacks may come in a "start-stop cycle" after some intervals. We have shown a case of repeated attack instance in Figure 7 and Table 3. It is very difficult to anticipate the attack repetition to prepare the defense. The mitigation mechanisms should be able to circumvent the attacks as quickly as possible in the presence of repetition. Repeated attacks may bring variations from the perspective of attack rate, type, packets, sources, and the attack duration.

5 Proposed Mitigation Framework

In the attack experiments, we observed that an attack with the same "frequency" and "intensity" may bring completely different manifestations on two different victims. On a victim, running a light service, it was mere DDoS, where the mitigation was quick and without many hurdles. On the other hand, a web service, which does more "resource utilization (work) per request" gets deteriorated to have "extreme DDoS" attack. Most of the DDoS attacks target with a peak attack bandwidth of∼1 Gbps and most attack targets of are SMEs [27]. These small targets are very cautious about sustainability and concerned about the cost of the security solutions. Additional resources and scaling should only be used, when they are required and used in the mitigation. In the following, we propose a solution having the following objectives. We decide these objectives considering the design requirements detailed in the Section 4.

1. Minimizing "Attack Cooling Down Period", T_C.
2. Mitigation with the available resources. Acquiring more resource only when needed.
3. Handling prolonged attacks and repetitions.
4. Providing quick resources from the available resources for mitigation in the presence of an attack.
5. Minimizing attack consequences like bandwidth costs and isolation penalties.

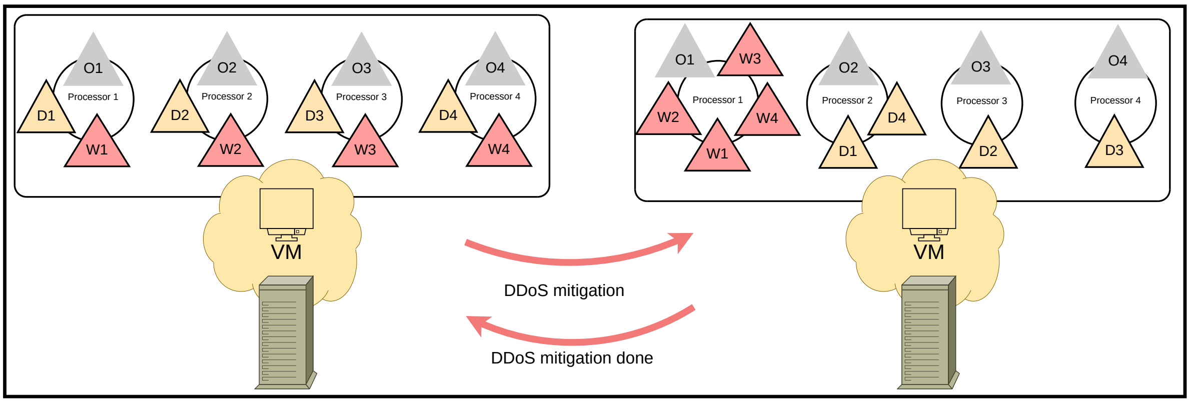


Fig. 8: Resizing Services for DDoS mitigation

5.1 Shrink-Expand Based Service Resizing

CPU time is one of the necessary resources to offer a service on a machine. "Resource race" created due to DDoS attacks is mostly due to CPU and Memory resources. To achieve the five objectives listed as above, we propose a novel "service resizing" method, which provides more resources to DDoS mitigation methods, in the presence of an attack. Our proposed method frees the resources by shrinking the "resource intensive" victim web-service to minimal resources thus reducing attack surface. We use OS level processor affinity methods [15][21] to affine the services to few or more processors dynamically. These methods are accessible using the affinity utilities such `taskset`. We illustrate the algorithm of service resizing in Algorithm 1 and a related function, RESIZE() in Algorithm 2. We show the process of service resizing in Figure 8. This algorithm works by regularly monitoring two important service parameters, (i) Request response time, T_{req} and (ii) Number of established connections = N_{est}. We assume that the service has a defined set of service capacity parameters, which show the maximum supported connections as per the capacity, N_{max} and acceptable request timeout at the client end, T_{to}. Algorithm checks these parameters at regular intervals. Attack detection time would decide the time taken by overall attack mitigation time. The algorithm checks if both the parameters are under control using a condition, which tests them ($T_{req} >= T_{to}$ && $N_{est} >= N_{max}$). If this attack condition becomes true (as the attack is present), in that case, the algorithm will immediately go for service resizing.

In case of C4 instance, the resources available to the VM are 4 vCPUs-8GB. We resize the services utilizing the affinity utilities available for compute resources (vCPUs). We will assign the minimum resource `MinR` =1 vCPU to victim service and `R-MinR`=3 vCPUs to DDoS Mitigation Sevice (DDoSMS) and other services.

Algorithm 1: Service Resizing Algorithm (Shrink-Expand)

```
SHRINK-EXPAND;
Data: Request response time = $T_{req}$,
Request timeout =$T_{to}$,
Number of established connections = $N_{est}$,
Maximum connections as per the capacity = $N_{max}$,
Victim service = $WebService$
DDoS mitigation service = $DDoSMS$;
Total resources = $R$,
Minimum resources to run the WebService = $MinR$,
Result: Attack Mitigation Successful
initialization;
while ($T_{req} < T_{to}$ && $N_{est} <= N_{max}$) do
    Nothing
end
RESIZE(WebService, MinR);
RESIZE(DDoSMS, R-MinR);
while ($T_{req} >= T_{to}$ && $N_{est} >= N_{max}$) do
    Nothing;
end
RESIZE(WebService, R);
RESIZE(DDoSMS, R);
Show "Attack Mitigation Successful"
```

Algorithm 2: Resize() function

```
RESIZE();
Data: Service $S$
Resources = $M$,
Result: Service S's new affinity is M
initialization;
Find out all the instances of S;
Change affinity of S to M;
```

We argue that the resource shrinking to minimum resources (in this case 1 vCPU) provides free resources to the mitigation methods. Presence of the extreme attack is an important information to proceed with the decision of shrinking. Resource shrinking and expansion will allow the DDoS mitigation service `DMS` to get maximum compute power, which is also isolated from the

No. of Rules	Shared Resources		Separate Resources	
	Addition	Deletion	Addition	Deletion
100	0m0.381s	0m0.547s	0m0.090s	0m0.097s
500	0m2.095s	0m2.069s	0m0.582s	0m0.590s
1000	0m5.479s	0m4.450s	0m1.505s	0m1.614s
5000	0m50.811s	0m49.552s	0m30.045s	0m30.200s
10000	3m5.560s	2m59.750s	2m22.881s	2m22.743s
20000	11m28.439s	11m29.298s	9m50.675s	9m55.632s

Table 4: Firewall: Shared Resources vs. Separate Resource

victim service. After resizing, the attack requests will be limited to MinR resources, and DMS will be able to perform its activities comfortably using dedicated resources. Once DMS detects the attack, it performs all the related activities. Once the attack cooling down period T_C passes, the algorithm will succeed in mitigation and resize the services back to their original form (resource R) To support the claims, we have conducted the experiments again with the proposed algorithms (Section 6). We also performed additional experiments to demonstrate how an "operating system level resizing" with separate resources using affinity, helps. As discussed in the Section 2, adding rules to the firewall is an important activity during the overall DDoS mitigation activity. In the presence of an attack (similar to Figure 3), we add and remove a number of rules to the firewall. First, we perform this operation on shared resources, with the victim web service. To see the impact of isolation and separation, we perform the same operation on separate resources. We show the results of this experiment in Table 4. It is clearly visible that sharing resources with the victim service under attack results in heavy performance penalties, which are as high as four times the actual time taken using separate resources (e.g. time required to add/remove 500 rules).

5.2 Minimizing Attack Cooling Down Time using TCP Tuning

We saw in the attack instance of Section 3 that the attack cooling down period is an important part of the overall service downtime. Clearing up "established" attack connections is an important part of the overall mitigation activity. These established connections may have both attack connections and benign user connections. However, in the extreme attack cases, the downtime results into successive timeouts for benign users. Usually, the connection removal activity is performed by identifying the sequence number and sending an RST (e.g. tcpkill). We have supported the connection removal activity by tuning two important TCP parameters to clear the established connections involving attackers quickly. However, to maintain the service quality, we unset the parameters to their original values once the attack downtime is over. These two parameters are, *tcp_fin-timeout* and *tcp_retries2* [2]. We set their values to "10s" and "1 retry" respectively.
1. *tcp_fin-timeout=10*: This parameter decides the time for which sockets will be in state FIN-WAIT-2. It is an important parameter to assist in early removal as the victim service has closed the connection.
2. *tcp_retries2*: This parameter decides the number of retries to be performed before killing an alive connection.
By setting the above parameters, we may lose some benign connections; however, loosing some benign connections during attack downtime is reasonable. Victim service looses the benign connections during the extreme attacks. By employing the proposed techniques we show that the reduction in overall downtime, also results in reduction in the loss of benign connections.

6 Evaluation and Results

The detailed results are shown in Figure 9, Figure 10 and Table 5. To evaluate the effectiveness of proposed service resizing algorithm, and TCP tuning technique, we perform following attacks.

1. Extreme DDoS attack with Shrink-Expand and TCP Tuning
2. Extreme DDoS with Shrink-Expand and without TCP Tuning
3. Extreme DDoS without Shrink-Expand and with TCP Tuning
4. Repeated extreme DDoS attack with Shrink-Expand and TCP Tuning

We compare the results of attacks mentioned above (point 1,2 and 3) with the extreme DDoS attack instance, discussed in Section 3 and Figure 6. Similarly, we compare the attack described in point 4 with the attack incident shown in Figure 7. In Figure 9a, we show the attack outcome without applying any of our proposed techniques. In Figure 9b, we use both "Shrink-Expand" and "TCP Tuning" techniques to support the

DDoS mitigation process. The results show improvement in all the important parameters such as attack detection time, attack reporting time, downtime for both victim service as well as other services, and total number of attack requests served by the victim. When we only use “Shrink-expand”, the attack detection time is increased to 901s from 845s when both the techniques were employed. We can also see in the Figure 9b about the response time downfall during the downtime for few requests which is not available in Figure 9c. On the other hand, in case of Figure 9d, the attack detection time is increased to more than 950s due to the unavailability of “Shrink-expand” mechanisms. However, “Only TCP Tuning” setting results into very small improvement (2118s as compared to 2294s in Table 5) in downtime of victim service which signifies the requirement of isolated resources for the DDoS mitigation service. We see that quick connection release during the attack (using TCP tuning) support the isolated resource availability in the presence of extreme DDoS attacks.

For the cases of repeated attack incidents, we compare the performance of the proposed techniques (Figure 10b) with its counterpart in which we do not employ these techniques (Figure 10a and Table 3). In these attack cases, the performance of the mitigation process is boosted and a quick attack detection and reporting is achieved. On the other hand, the case of repeated attacks one after another (shown in Table 6), show similar performance metrics to the extreme attack cases shown in Table 5.

6.1 Discussion and Issues

Extreme DDoS attacks occur due to the heavy resource sharing at the level of operating systems. Performance and resource issues among VMs are highly isolated as compared to the process isolation at the level of an operating system [40]. The proposed techniques should not be used and are rather not useful where these resource contentions are not severe. Considering the evaluation results, following are few important issues about the proposed scheme concerning the DDoS mitigation in cloud computing.

Deciding when to resize : Resizing is only needed when the service is facing an extreme attack. Anticipating a DDoS attack to take the shape of extreme attack depends on service and the amount of efforts it spends on each request. Resizing may also result in downtime for benign users in the presence of low rate DDoS attacks. In this case, we would like to adopt step-wise resizing (e.g. freeing just 1vCPU for the DDoS Mitigation).

Attack requests in the system : Number of attack requests entered into the system, are directly proportional to the time it takes to detect the attack source. Therefore, attack detection time will lead to the downtime, network bandwidth spent on attack and attack cooling down time.

Network bandwidth : A control on the attack requests will also result in network isolation, which will lead into minimization of collateral damages and energy consumption.

Attack strength : Resizing tries to help in one critical aspect, which is the impact of attack strength. If the attack comes with a minimum rate and achieves the “extreme DDoS”, increasing the attack rate further will not have any adverse impact on the service under the attack. Resizing will always bring the victim web service to the `MinR` resources.

Availability issues : In the cases of extreme DDoS attacks without using the proposed techniques, victim service faces a huge downtime. With the help of “Shrink-Expand” and TCP Tuning, the downtime is reduced to achieve availability. After the attack is over the services are resized to the original resources to maintain the availability.

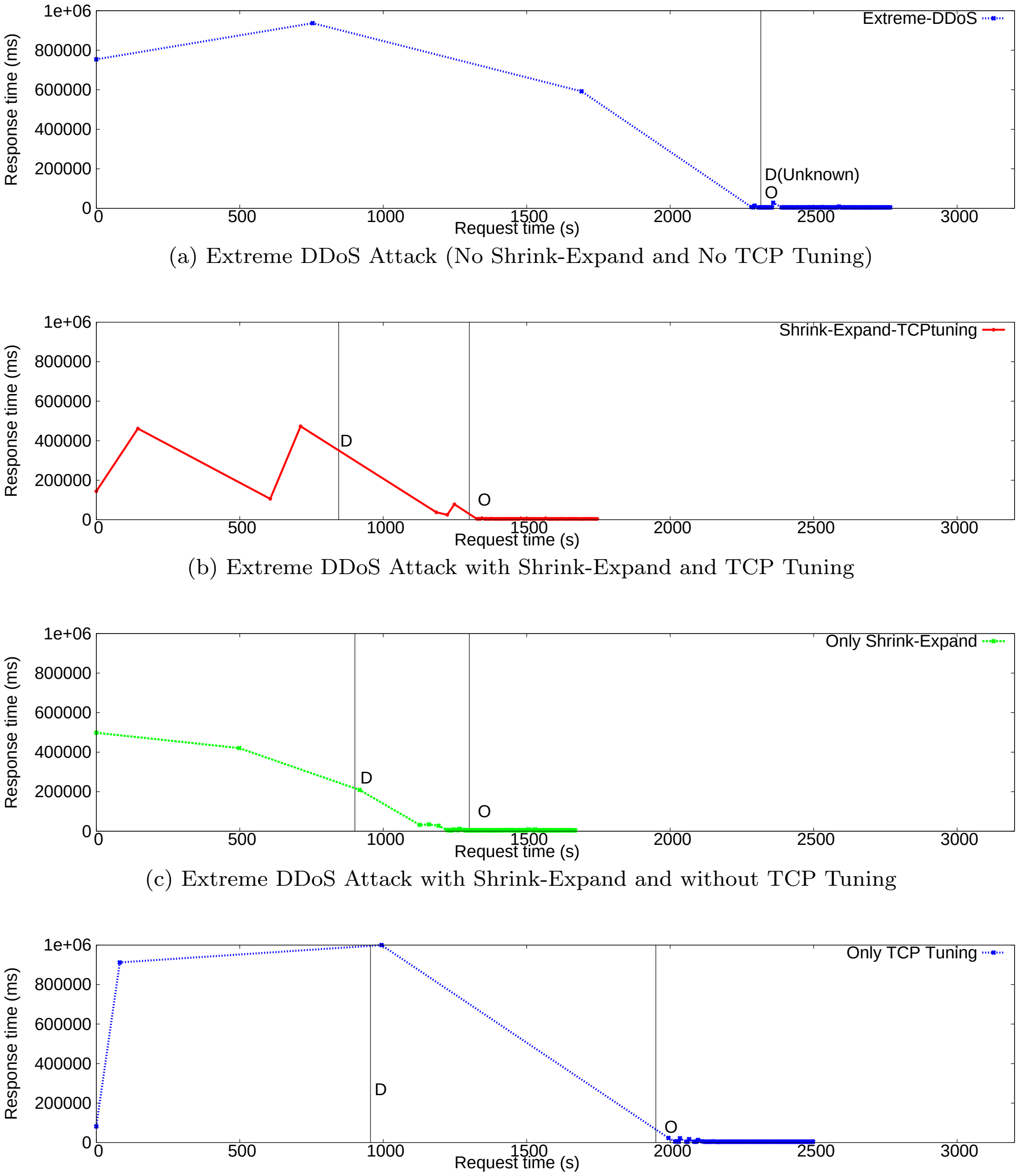
Attack repetition : There is no clear way by which we can know that an attack is going to repeat in future. Therefore, after completing the mitigation requirements through shrinking, we will again expand the resources. If there is another attack before this expansion than the web service will remain with minimum sized resources.

Attacks during downtime: There may be other DDoS attacks or changed attack vector once the service downtime is reached. In both the situations the attack mitigation will be quick as compared to the case where “Shrink-Expand” or TCP Tuning are not utilized because of the resource availability. Additional resource requirement in case the attack detection or mitigation is not possible within the available resources, in that case the traditional auto-scaling methods are required to scale the service.

Overhead of resizing : Shrink-Expand overhead will be similar to the overhead of moving tasks from one CPU to another CPU using context switches used in preemption and global load balancing.

Availability of other services : Other services were not completely available (intermittently available) in the case of resizing of applications. This is mostly due to heavy memory usage. Memory level resizing can ensure availability of other services. However due to a large decrease in downtime, the services are restored quickly.

Resources available : Resource requirement has not been a primary thought while designing DDoS mitigations solutions. We could see that giving more resources may not help in few attack instances unless the resource



(a) Extreme DDoS Attack (No Shrink-Expand and No TCP Tuning)

(b) Extreme DDoS Attack with Shrink-Expand and TCP Tuning

(c) Extreme DDoS Attack with Shrink-Expand and without TCP Tuning

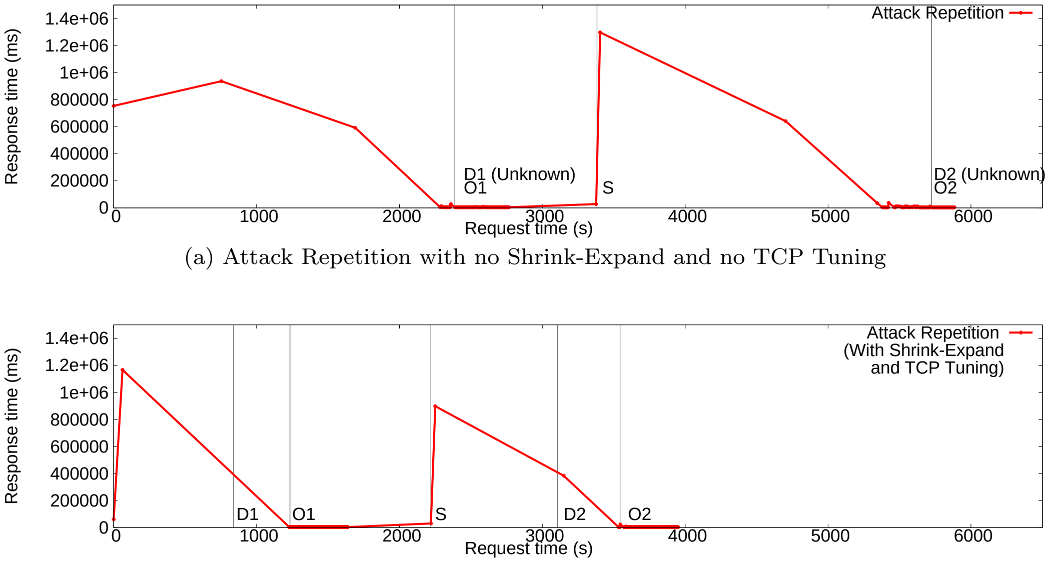
(d) Extreme DDoS Attack with TCP Tuning and without Shrink-Expand

Fig. 9: Evaluation and Results of Shrink-Expand and TCP Tuning (D= Time of Attack Detection and O= Attack Effects Over)

contention issues are solved. On the other hand by resizing, we could spare 3 vCPUs for the mitigation which is equivalent to having 75% of C4 Compute resources (4 vCPUs) without costs.

7 Related work

There are a large number of contributions in the area of DDoS mitigation in a variety of computing environments. A detailed list of these contributions is available in popular surveys in [6][22][30]. Similarly, there is a number of surveys related to DDoS mitigation mechanisms in cloud computing environment [35][41][48]. We see that there are other related contributions in the areas of attack impact studies in the cloud computing environment. Authors in [29] show a study of network-level DDoS and its impact on services running in the cloud. Authors have shown that the power usage is affected due to the heavy impact on CPU and I/O usage. Similarly, Shea et al. in [36] show DDoS attack and its effects on various kinds of virtualization tech-



(a) Attack Repetition with no Shrink-Expand and no TCP Tuning

(b) Attack Repetition with Shrink-Expand and TCP Tuning

Fig. 10: Evaluation and Results of Shrink-Expand and TCP Tuning in Case of Repeated Attacks (D1= Attack 1 Detection Time, O1= Attack 1 Effects Over, S= Start of Attack 2, D2= Attack 2 Detection Time, and O2= Attack 2 Effects Over)

Attack	**Resources**	**Time of Attack Detection**	**Time of Attack Reporting**	**Downtime (Victim Service)**	**Downtime (Other Services)**	**No. of Attack requests served before detection**
Extreme	4vCPU-8GB	Unknown	2315s	2294s	2294s	27
Extreme	4vCPU-8GB (Shrink-Expand and TCP Tuning)	845s	845s	1326s	1326s	10
Extreme	4vCPU-8GB (Only Shrink-Expand)	901s	901s	1278s	1278s	12
Extreme	4vCPU-8GB (Only TCP Tuning)	956s	956s	2118s	2118s	34

Table 5: Attack Results after applying Shrink-Expand and TCP Tuning

Attack in Repetition	**Time of Attack Detection**	**Time of Attack Reporting**	**Downtime (Victim Service)**	**Downtime (Other Services)**	**No. of Attack requests served before detection**
Attack 1	840s	840s	1234s	1234s	8
Attack 2	851s	851s	1321s	1321s	12

Table 6: Attack Repetition Results after applying Shrink-Expand and TCP Tuning

niques. Authors in [11], show effects of a DDoS variant in the cloud, known as Fraudulent Resource Consumption (FRC attacks), where the attackers plan the attack in such a manner that it forces fake resource usage and billing. Other authors in [10][37][47] show similar attack instances and their studies. Author in [39] demonstrate the impact of DDoS attack on non-target services in the multi-tenant cloud environment. Authors argue that the shared resources and poor isolation result in attack effects to the services and components, which are not on the target of the attack. A range of sophisticated, stealthy attacks traffic patterns has been shown by authors in [7] when the attackers to remain undetected, change attack patterns. Authors also show energy losses due to these stealthy DDoS attacks on cloud services.

There is a number of contributions related to attack prevention techniques, which advocate the usage of challenge-response protocols like CAPTCHAs [9][16][45]. On the other hand, there are techniques, which work on detecting the attacks using traffic patterns and anomaly detection [12][18][28][52]. Authors in [1] show novel cloud service targeted attacks which are running critical health care services. In this work, authors propose an algorithm to detect the DDoS attacks in Body Area Networks based on wireless medium. Now, we discuss the contributions related to attack mitigation in cloud computing environment. There is a number of recent contributions in the area of DDoS mitigation in the cloud, which use resource management. As our work focuses on the resource management aspects of cloud computing, we will now detail the works related to resource scaling and attack mitigation in the cloud. Authors in [14], proposed a moving server based technique in which the incoming requests are redirected to different servers and their replicas based on the request behavior. The proposed method is a costly technique, which has an additional overhead of change management among replicas and cost of the servers. Authors in [34] proposed a multilevel solution based on cloud level, tenant level, and VM level DDoS detection. Authors in [51] proposed a detailed solution concerning the resource allocation techniques in cloud computing. Authors recommend using multi-instance scaling to acquire more and more instances, running intrusion prevention system to mitigate attacks quickly. Authors have proposed a DDoS mitigation algorithm, which works on resource scaling to give more and more resources while there is an attack. Authors have also conducted experiments to show the efficacy of their scheme from the perspective of cost of attack mitigation. However, this solution does not consider the cost evaluation for the repetitive and prolonged attacks for hours with heavy intensity. Additionally, the cost evaluation does not consider the cost of outgoing bandwidth. Another contribution in this area [50], proposed a low-cost cloud-based firewall to perform quick mitigation based on the trade-off between service quality and resources.

Authors in[53] uses a hypervisor level DDoS detection, which is an external detection in which after the detection, victim VM is transferred to a backup server and once the attack gets over, they bring it back to the original server. The cost of the backup server and overhead of migrations are major issues with this technique. Authors in [19] propose a broad range of DDoS attacks and their identification based on the resource usage and traffic. Additionally, this method also provides migration and scaling based DDoS mitigation and recovery to maintain service availability. Similarly, authors in [49] provides a collaborative solution based on resource acquiring from untrusted Content Delivery Network (CDN) clouds. Authors argue that these inexpensive resources can be used in quick mitigation of DDoS attacks with inexpensive resource driven scaling. Externally supportive mitigation methods are proposed by authors in [32]. Authors in this work do ISP level mitigation to help the overall mitigation activity. Authors in [46] shows Software Defined Networking (SDN) based mitigation methods. Other important contributions include [20], where authors provide an elastic intrusion prevention system based on SDN technologies.

Most of the works related to DDoS mitigation in cloud computing are similar to the methods employed on traditional infrastructures. Few methods which contribute in the direction of resource mitigation are also not considering the aspects on operating system level “resource race” among processes and services. We see that there are few solutions, which consider the resource scaling as a supporting mechanism to DDoS mitigation. However, cost and access to victim service during extreme DDoS attack is not considered, while designing these solutions. One very high commonality in all the past contributions is that the mitigation methods treat attack identification as the final stage of mitigation. However, we have shown in our experiments that post-attack detection, there is a significant time taken by the attack requests to cool down the resources and become available for future requests. We would also like to highlight on the resource management at the operating system level and isolation required among services during an attack. Resource management is a novel factor which is considered by our work looking at various attack instances and mitigation within the available resources.

8 Conclusions and Future Work

Recently reported attacks prove that the DDoS attacks are much fatal than they appear for both the victim service and the organization. DDoS attacks on the cloud services see various trends due to the nature of the business model supported by cloud computing. “Pay-as-you-Go” models are becoming real for both attackers and victim enterprises. We see a conversion of DDoS “arms-race” into a “resource-race” due to the emergence of these services. There is an immense need of methods to characterize and mitigate these attacks on the cloud environment.

We conduct real attack instances on cloud services to critically see the overall mitigation activity at fine grain level, i.e., at the resource level. DDoS attacks being resource based attack turn into “extreme DDoS” attacks for services with high resource utilization per request. We characterize these extreme DDoS attacks and observe that the resource contention created by the victim service under an attack may also compromise the DDoS mitigation service itself. Additionally, in these extreme DDoS attacks, availability after the attack detection is also affected due to a longer attack cooling down period.

To circumvent these problems, we provide a framework to support the overall mitigation activity desirable from any mitigation tool. Our supporting framework puts efforts to provide enough resources such that the mitigation mechanism can perform its task even in the presence of extreme attacks. For this purpose, we perform attack experiments and highlight the need for methods to minimize the downtime, post-attack detection. We propose a novel supporting framework which employs processor affinity-based service resizing and TCP tuning techniques during the attack period to serve two important purposes, (i) providing required resources to mitigation activity and (ii) minimizing overall downtime.

We perform detailed experiments to show the efficiency and efficacy of our scheme. The novelty of our scheme opens up multiple directions of research to visualize the inter-service relationship on an operating system. Additionally, the behavior of other unrelated services and providing access to attack mitigation techniques involve scaling, are few other directions which are open and relevant. Isolation and separation of victim services concerning other basic resources such as memory, disk, and bandwidth, is a direction, which may extend our work.

9 Acknowledgment

This work is supported by a Teacher Fellowship under Faculty Development Program funded by University Grants Commission, Government of India, under XII Plan (2012-2017). Experimental setup for this work is supported by SAFAL (Security Analysis Framework for Android Platform) project funded by Department of Electronics and Information Technology, Government of India. Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission (agreement PCIG11-GA-2012-321980). This work is also partially supported by the EU TagItSmart! Project (agreement H2020-ICT30-2015-688061), the EU-India REACH Project (ICI+/2014/342-896), the Italian MIUR-PRIN TENACE Project (agreement 20103P34XC), and by the projects "Tackling Mobile Malware with Innovative Machine Learning Techniques", "Physical-Layer Security for Wireless Communication", and "Content Centric Networking: Security and Privacy Issues" funded by the University of Padua.

References

1. Abbas, H., Latif, R., Latif, S., Masood, A.: Performance evaluation of Enhanced Very Fast Decision Tree (EVFDT) mechanism for distributed denial-of-service attack detection in health care systems. Ann. Telecommun. pp. 1–11 (2016)
2. Andreasson, O.: Ipsysctl tutorial 1.0.4. `https://www.frozentux.net/ipsysctl-tutorial/chunkyhtml/tcpvariables.html` (Retrieved on August 4, 2016)
3. Arbor Networks: Understanding the nature of DDoS attacks. `http://www.arbornetworks.com/asert/2012/09/understanding-the-nature-of-ddos-attacks/` (2014)
4. AWS Discussion Forum: https://forums.aws.amazon.com. `https://forums.aws.amazon.com` (2006)
5. Cohen, R.: Cloud Attack: Economic Denial of Sustainability (EDoS). `http://www.elasticvapor.com/2009/01/cloud-attack-economic-denial-of.html` (2009)
6. Douligeris, C., Mitrokotsa, A.: DDoS attacks and defense mechanisms: classification and state-of-the-art . Computer Networks **44**(5), 643 – 666 (2004)
7. Ficco, M., Rak, M.: Stealthy denial of service strategy in cloud computing. Cloud Computing, IEEE Transactions on **3**(1), 80–94 (2015)
8. HTTP Archive: httparchive.org/compare.php. `httparchive.org/compare.php` (2016)
9. Huang, V., Huang, R., Chiang, M.: A DDoS Mitigation System with Multi-stage Detection and Text-Based Turing Testing in Cloud Computing. In: Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on, pp. 655–662. IEEE (2013)
10. Idziorek, J., Tannian, M.F., Jacobson, D.: The insecurity of cloud utility models. IT Professional **15**(2), 22–27 (2013)
11. Idziorek et al.: Exploiting cloud utility models for profit and ruin. In: Proc. IEEE International Conference on Cloud Computing (4th IEEE CLOUD'11), pp. 33–40. IEEE Computer Society, Washington, DC, USA (2011)
12. Ismail, M.N., Aborujilah, A., Musa, S., Shahzad, A.: Detecting flooding based dos attack in cloud computing environment using covariance matrix approach. In: Proc. of the 7th International Conf. Ubiquitous Information Management and Communication, p. 36. ACM (2013)
13. Jefferson Gonzlez: DDoS Deflate. `https://github.com/jgmdev/ddos-deflate` (2016)
14. Jia, Q., Wang, H., Fleck, D., Li, F., Stavrou, A., Powell, W.: Catch Me If You Can: A Cloud-Enabled DDoS Defense. In: Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on, pp. 264–275. IEEE (2014)
15. Kerrisk, M.: SCHED_SETAFFINITY. `http://man7.org/linux/man-pages/man2/sched_setaffinity.2.html` (Retrieved on July 11, 2016)
16. Khor, S.H., Nakao, A.: spow: On-demand cloud-based EDDoS mitigation mechanism. In: HotDep (Fifth Workshop on Hot Topics in System Dependability) (2009)
17. Kim Weins: Cloud Computing Trends: 2015 State of the Cloud Survey. `http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2015-state-cloud-survey` (2015)
18. Koduru, A., Neelakantam, T., Bhanu, S., Mary, S.: Detection of Economic Denial of Sustainability Using Time Spent on a Web Page in Cloud. In: Cloud Computing in Emerging Markets (CCEM), 2013 IEEE International Conference on, pp. 1–4 (2013)

19. Latanicki, J., Massonet, P., Naqvi, S., Rochwerger, B., Villari, M.: Scalable Cloud Defenses for Detection, Analysis and Mitigation of DDoS Attacks. In: Future Internet Assembly, pp. 127–137 (2010)
20. Lopez, M.A., Ferrazani Mattos, D.M., Duarte, O.C.M.B.: An elastic intrusion detection system for software networks. Ann. Telecommun. pp. 1–11 (2016)
21. Love, R.M.: Taskset Command. `http://www.linuxcommand.org/man_pages/taskset1.html` (Retrieved on July 11, 2016)
22. Mirkovic, J., Reiher, P.: A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. SIGCOMM Comput. Commun. Rev. **34**(2), 39–53 (2004). DOI 10.1145/997150.997156
23. Mirkovic, J., Robinson, M., Reiher, P.: Alliance formation for DDoS defense. In: Proceedings of the 2003 workshop on New security paradigms, pp. 11–18. ACM (2003)
24. Mohammad, R.M., Mauro, C., Ville, L.: EyeCloud: A BotCloud Detection System. In: In Proceedings of the 5th IEEE International Symposium on Trust and Security in Cloud Computing (IEEE TSCloud 2015), Helsinki, Finland. IEEE (2015)
25. Moore, D., Shannon, C., Brown, D.J., Voelker, G.M., Savage, S.: Inferring internet denial-of-service activity. ACM Transactions on Computer Systems (TOCS) **24**(2), 115–139 (2006)
26. Netfilter/iptables project home page: www.netfilter.org. (2016)
27. Networks, A.: Worldwide Infrastructure Security Report Volume XI. (2015)
28. Osanaiye, O., et al.: IP spoofing detection for preventing DDoS attack in Cloud Computing. In: Intelligence in Next Generation Networks (ICIN), 18th International Conf on, pp. 139–141. IEEE (2015)
29. Palmieri, F., Ricciardi, S., Fiore, U.: Evaluating Network-Based DoS Attacks under the Energy Consumption Perspective: New Security Issues in the Coming Green ICT Area. In: BWCCA, International Conference on, pp. 374–379 (2011)
30. Peng, T., Leckie, C., Ramamohanarao, K.: Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems. ACM Comput. Surv. **39**(1) (2007)
31. Prolexic: http://www.prolexic.com/. `http://www.prolexic.com/` (2014)
32. Sahay, R., Blanc, G., Zhang, Z., Debar, H.: Towards Autonomic DDoS Mitigation using Software Defined Networking. SENT 15 (2015)
33. Santanna, J.J., van Rijswijk-Deij, R., Hofstede, R., Sperotto, A., Wierbosch, M., Granville, L.Z., Pras, A.: BootersAn analysis of DDoS-as-a-service attacks. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 243–251. IEEE (2015)
34. Sarra, A., Rose, G.: DDoS Attacks in Service Clouds. In: 48th Hawaii International Conference on System Sciences. IEEE Computer Society (2015)
35. Shameli-Sendi, A., Pourzandi, M., Fekih-Ahmed, M., Cheriet, M.: Taxonomy of distributed denial of service mitigation approaches for cloud computing. Journal of Network and Computer Applications pp. – (2015)
36. Shea, R., Liu, J.: Understanding the impact of denial of service attacks on virtual machines. In: Proc. 20th International Workshop on Quality of Service, p. 27. IEEE Press (2012)
37. Sides, M., Bremler-Barr, A., Rosensweig, E.: Yo-yo attack: Vulnerability in auto-scaling mechanism. SIGCOMM Comput. Commun. Rev. **45**(4), 103–104 (2015). URL `http://doi.acm.org/10.1145/2829988.2790017`
38. Somani, G., , Gaur, M.S., Sanghi, D.: DDoS Protection and Security Assurance in Cloud. In: Guide to Security Assurance for Cloud Computing, Computer and Communications and Networks, Springer (2015)
39. Somani, G., Gaur, M.S., Sanghi, D.: DDoS/EDoS Attack in Cloud: Affecting Everyone out There! In: Proceedings of the 8th International Conference on Security of Information and Networks, SIN '15, pp. 169–176. ACM, New York, NY, USA (2015)
40. Somani, G., Gaur, M.S., Sanghi, D., Conti, M.: DDoS attacks in Cloud Computing: Collateral Damage to Non-targets. Computer Networks (2016)
41. Somani, G., Gaur, M.S., Sanghi, D., Conti, M., Buyya, R.: DDoS Attacks in Cloud Computing: Issues, Taxonomy, and Future Directions. arXiv preprint arXiv:1512.08187 (2015)
42. Somani, G., Johri, A., Taneja, M., Pyne, U., Gaur, M.S., Sanghi, D.: DARAC: DDoS Mitigation using DDoS Aware Resource Allocation in Cloud. In: 11th International Conference, ICISS, Kolkata, India, December 16-20, 2015, Proceedings (2015)
43. SPAMfighter News: Survey - With DDoS Attacks Companies Lose around 100k/Hr. `http://www.spamfighter.com/News-19554-Survey-With-DDoS-Attacks-Companies-Lose-around-100kHr.htm` (2015)
44. Tara Seals: Q1 2015 DDoS Attacks Spike, Targeting Cloud. `http://www.infosecurity-magazine.com/news/q1-2015-ddos-attacks-spike/` (2015)
45. Wang, H., Jia, Q., Fleck, D., Powell, W., Li, F., Stavrou, A.: A moving target DDoS defense mechanism. Computer Communications **46**, 10–21 (2014)
46. Wang, X., Chen, M., Xing, C.: SDSNM: A Software-Defined Security Networking Mechanism to Defend against DDoS Attacks. In: Frontier of Computer Science and Technology (FCST), 2015 Ninth International Conference on, pp. 115–121. IEEE (2015)
47. Xu, Z., Wang, H., Xu, Z., Wang, X.: Power Attack: An Increasing Threat to Data Centers. In: Proc. of NDSS, vol. 14 (2014)
48. 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. Communications Surveys Tutorials, IEEE **PP**(99), 1–1 (2015)
49. Yossi, G., Amir, H., Michael, S., Michael, G.: CDN-on-Demand: An Affordable DDoS Defense via Untrusted Clouds. In: NDSS 2016 (2015)
50. Yu, S., Doss, R., Zhou, W., Guo, S.: A general cloud firewall framework with dynamic resource allocation. In: ICC, pp. 1941–1945. IEEE (2013)
51. Yu, S., Tian, Y., Guo, S., Wu, D.O.: Can we beat ddos attacks in clouds? Parallel and Distributed Systems, IEEE Transactions on **25**(9), 2245–2254 (2014)
52. Zhang, Jian, et al.: A Robust and Efficient Detection Model of DDoS Attack for Cloud Services. In: Algorithms and Architectures for Parallel Processing, pp. 611–624. Springer International Publishing (2015)
53. Zhao, S., Chen, K., Zheng, W.: Defend against Denial of Service Attack with VMM. In: Grid and Cooperative Computing, 2009. GCC'09. Eighth International Conference on, pp. 91–96. IEEE (2009)