DDoS Victim Service Containment to Minimize the Internal Collateral Damages in Cloud Computing

Gaurav Somani[a,b], Manoj Singh Gaur[b], Dheeraj Sanghi[c], Mauro Conti[d], Muttukrishnan Rajarajan[e]

[a] *Central University of Rajasthan, Ajmer, India*
[b] *Malaviya National Institute of Technology, Jaipur, India*
[c] *Indian Institute of Technology, Kanpur, India*
[d] *University of Padua, Padua, Italy*
[e] *City University of London, London, UK*

Abstract

Recent Distributed Denial of Service (DDoS) attacks on cloud services demonstrate new attack effects, including collateral and economic losses. In this work, we show that DDoS mitigation methods may not provide the expected timely mitigation due to the heavy resource outage created by the attacks. We observe an important Operating System (OS) level "internal collateral damage", in which the other critical services are also affected. We formulate the DDoS mitigation problem as an OS level resource management problem. We argue that providing extra resources to the victim's server is only helpful if we can ensure the availability of other services. To achieve these goals, we propose a novel resource containment approach to enforce the victim's resource limits. Our real-time experimental evaluations show that the proposed approach results in reduction in the attack reporting time and victim service downtime by providing isolated and timely resources to ensure availability of other critical services.

Keywords: Cloud Computing, Cloud Security, Distributed Denial of Service (DDoS) attack and Economic Denial of Sustainability (EDoS) attack

1. Introduction

Distributed Denial of Service (DDoS) attacks result in fatal attack effects to enterprises for last many years. DDoS attacks are also visible by service downtimes faced by important services, and remain among the top cyber security threats for the last several years. There are reports which state that one out of five enterprises across the world is affected by DDoS attacks [1]. The growth of DDoS attacks can be visualized by the progress of these attacks made from the perspective of maximum attack bandwidth each year. The peak DDoS attack bandwidth reached more than 500 Gbps in 2015 from just 8 Gbps in year 2000 [2]. Major motivation behind DDoS attacks includes business rivalry, political ideology, and cyber war among countries. The most common outcome of DDoS attacks is "unavailability of target service". In addition to the unavailability, there are many short term and long term business and reputation losses, which are actually a set of consequences of the service downtime.

In recent times, cloud computing has been adopted across the globe to support the major IT requirements of organizations from all industry sectors. As highlighted in [3], majority of the organizations (>87%) across the globe are using cloud infrastructure to run their mission-critical applications. This adoption trend is due to the profound resources and availability of on-demand resources in the cloud. However, the emergence of cloud computing has also led to the shift of DDoS attackers more towards the cloud driven services. As highlighted in [2], more than 33% of the overall reported attacks in year, 2015 were targeted towards cloud services. In addition, cloud features such as profound resources and pay-as you go accounting are also becoming attractive to the attackers.

The DDoS attacks in cloud computing are also termed as Economic Denial of Sustainability (EDoS) attacks, due to the substantial economic losses both from resource usage and business disruption. These losses are directly proportional to the downtime incurred by the attack. Most of the reported attacks usually last between few minutes to few hours [2] and some "major" attacks may last few days to even weeks. There are many recent DDoS attacks on cloud services among which the attacks on Amazon EC2 services, RackSpace and Linode are major incidents resulting into considerable service outages.

There are a large number of DDoS mitigation solutions available today, which are summarized in recent survey articles [4] for traditional fixed and cloud infrastructures [5]. On the other hand, there are solutions used by service providers, which mostly work on traffic filtering and quick attack absorption by resource scaling [6]. The cost factor is one of the major features to attract the service providers to migrate to the cloud infrastructures. The cost to eliminate the DDoS attack effects is an important aspect to consider during mitigation, as it directly comes from the economic sustainability of the victim's enterprise. There are also recent DDoS attack incidents, where the "Service Denial" does not seem to be the main focus of the attackers. These hidden attacks usually launch a DDoS attack in order to achieve the whole attention of the victim and simultaneously perform other severe activities such as data breaches. These attacks are termed as "Smoke-screening attacks" [7]. These attack effects are possible due to the heavy resource investment (both in terms of manpower, server and network resources) in DDoS mitigation leaving other important activities unattended.

Similarly, multi-tenancy, auto scaling and migration of services lead to some additional effects of DDoS attacks in cloud computing. These attacks are termed as "collateral damages" on co-hosted cloud services and network components [8]. The above discussion makes it necessary for availability of efficient solutions in the direction of DDoS attack prevention, detection and mitigation. DDoS attacks are usually considered resource intensive requests, which stress overload one or a combination of target service resources. These resources are usually the basic resources like CPU cycles, memory and swap usage, I/O operations or network bandwidth. Additional application level resources are number of simultaneous connections, ports, sessions, application buffers or other temporary identifiers. Most of the server resources are shared among many of the co-located processes to achieve their goals. Victim service, DDoS mitigation service, logging and scheduling processes are few examples of such processes.

In this work, we provide a novel observation towards operating system level resource race and contention among co-located services. We argue that the services co-located with the DDoS victim service (say a web-server process) may not provide the expected processing and timely outcome due to the extensive resource contention by the DDoS attack. These co-located services include all the important services, such as DDoS mitigation service, firewall and internal security policy services (e.g. SELinux) to other system processes and remote login processes. We show this phenomenon in the experiments and term it as "Internal Collateral Damage" as the services other than the victim service, is severely affected. We show that DDoS Mitigation methods may not provide the expected outcome and delivery due to this factor. To approach the "internal collateral damage" problem, we provide an analysis of DDoS attacks from the OS level resource management perspective. Usually, virtual machines (VMs) are used across clouds to provide strong performance and resource isolation. However, the internal operating system level isolation cannot be governed by the virtual machines.

We argue that resource isolation among co-located services, may provide quick and efficient DDoS mitigation. Based on these requirements, we propose a novel approach, "Victim Service Containment" to achieve resource contention of victim service resources which is under attack. We perform real-time attack experiments to show that the proposed approach provides resources availability to all the important services and help to minimize the overall attack effect and cost. We provide attack cases and model resource requirements to provide solutions based on resource control groups [9] to provide internal isolation without affecting the performance. We also extend the discussion to limit DDoS effects to just the target service, to eliminate or minimize additional collateral damages.

The rest of the paper is organized as follows. Section 2 provides the details about DDoS attack protection at various levels of the cloud architecture. Section 3 provides details of the experiments showing the novel "Internal Collateral Damages". We provide a detailed resource management model of the problem in Section 4. Section 5 provides the proposed design to eliminate the internal collateral damages. Section 6 provides the details of the evaluation to show the efficacy of the proposed solution. We also discuss the features of the proposed solution and various aspects related to DDoS mitigation. We provide a discussion of the related work in Section 7 and finally conclusions in Section 8.

2. DDoS Attack and Protection in the Cloud

DDoS attacks are mostly coordinated attacks planned by a command and control (C & C) server with the help of a malware infected network of computers. These computers are also known as "bots". Cloud infrastructure can also be used in the networks of attack computers with the emergence of pay-as-you-go "DDoS for hire" services. The impact of these attacks depend upon various attack dimensions.

These attack dimensions include attack frequency, attack duration, type of attack packets, number of sources, target victim services, etc. However, the main objective of the victim service, is to remain available to serve the customers. The DDoS attacks targeted at cloud services are mostly similar to the attacks targeting non-cloud fixed infrastructures from the perspective of an attacker. The major variation comes when we see the outcome of the attacks. Cloud DDoS might result in effects such as economic losses due to resource usage during attacks, collateral damages to the co-hosted VM and other non-targets. These effects are in addition to the effects of service denial [8]. Cloud hosted services usually run inside isolated environments of the virtual machines. These virtual machines are hosted on top of the physical server machines in a cloud. Each physical server may host VMs on the basis of the resource requirements of hosted VMs and available resources. The VM owner/administrator accesses the VM using a remote connection over the network.

The VM placement over geographically spread clouds also facilitates the complete control of the VM to its owner. DDoS attacks usually target victims service availability. In cloud computing infrastructure, economic losses due to fake resource usage and subsequent resource acquisition, have been reported in recent security studies [5]. We consider three important cloud architecture junctions, where the DDoS detection and mitigation mechanisms are usually employed (Figure 1). These three junctions are cloud network edge, host physical server's virtualized network and the victim VM's own network and application.

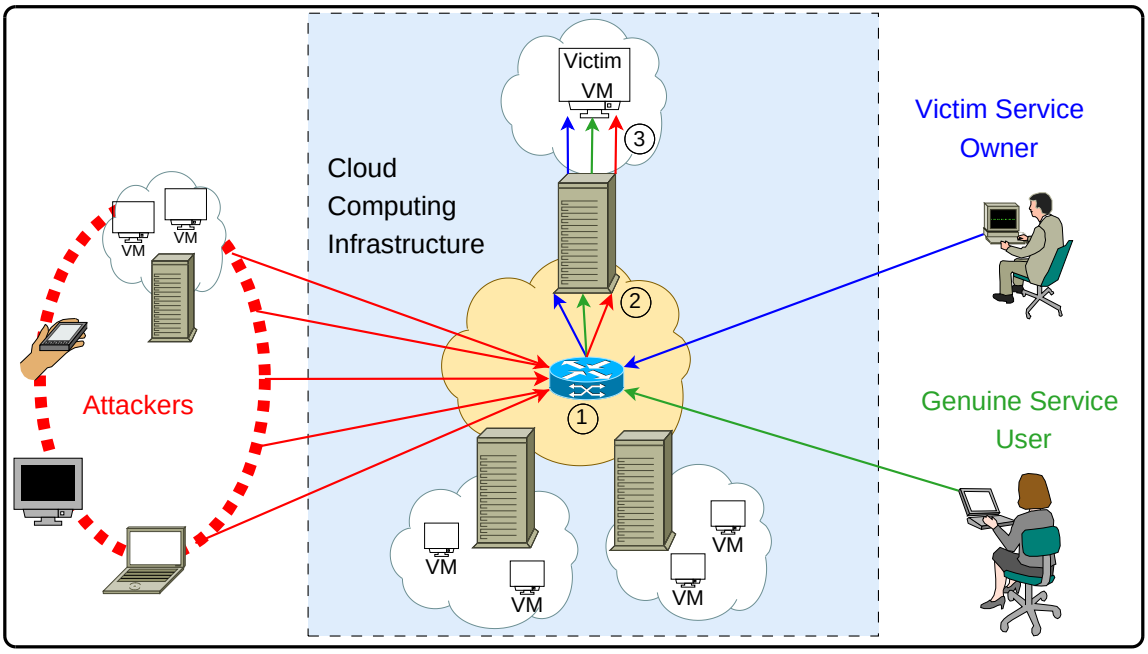


Figure 1: DDoS attacks targeted at cloud services

Most of the DDoS attack detection mechanisms work on top of the traffic filters, access pattern anomaly detections or other detection mechanisms based on attack related behaviors [5]. However, there are many recent detection methods, which, in addition to relying on traffic filters, also utilize the resource management and auto scaling capabilities [10] [11] [12]. Most of these methods add more resources to the victim service for a timely mitigation. Similarly, there are production environment solutions from industry relying on dynamic resource allocation to victims' service to overcome the DDoS traffic [6]. The on-demand and elastic nature of cloud computing platform provided by the cloud necessitated the re-look on the legacy DDoS protection solutions available in the non-cloud traditional infrastructures. We propose the following essential requirements for effective DDoS mitigation of cloud services:

R1. Quick attack detection and mitigation with minimum downtime
R2. Sustainability/budget aware mitigation
R3. Minimum collateral damages

Requirements R1 and R2 are highly important, as the whole notion of cloud computing is based on the cost and availability of resources. The victims' organizations are much more concerned about the costs of DDoS mitigation solutions as the mitigation cost directly affects their IT and security budgets. On the other hand, minimization of downtime and subsequent savings can be achieved through quick mitigation. Requirement R3 is important from the perspective of minimizing the effects to non-targets and attack spread. Authors in [8] show that DDoS attacks in cloud computing may also affect the co-hosted VMs and the associated services run by these VMs. Subsequently, these attacks also increase the overall throughput and energy consumption of the whole cloud infrastructure. Authors show that the effective solutions in this direction can be designed by minimizing the overall attack effect to the victim service by stronger isolation and resource separation from other services [8].

We extend the notion of resource isolation from hypervisor level to individual VM or guest operating system level and discuss its applicability in Section 3. Each server operating system usually run a number of utilities to support the overall working of the victims' service in addition to its protection, maintenance and recovery. Typical set of services provided by each server operating system are shown in Figure 2.

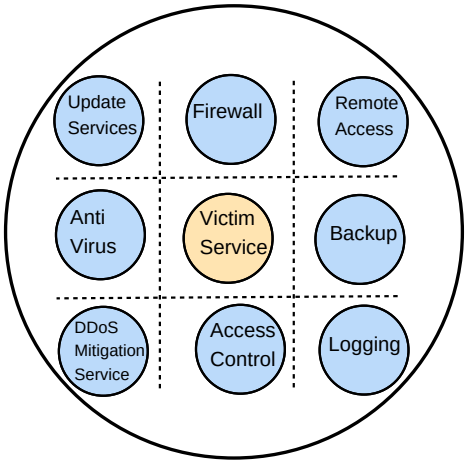


Figure 2: Various essential services on a typical server operating system

Most of these services are provided to support the main service, for example a web-service is supported by other services related to service health, backup and monitoring. Other than the web-service, none of the other services are directly accessible by a web-service user or an attacker. The VM owner accesses required other services using a "remote access" service. These services include update services for server software, security software and operating system patches, firewall, DDoS mitigation service, backup services, logging services and all the other services/resources available on the operating system. Many of these services are critical to the system availability, security and fault tolerance. Ideally, these services should always be available and remain unaffected by the DDoS attacks on victims' web-service. The availability of these co-located services even during the DDoS attack period would assist in the following activities:

I. System state awareness: With the help of all the other services, the real cause of web-service unavailability can be identified. If the services such as remote access becomes unresponsive and unavailable, the owner has no other methods available to access the VM. Additionally, if the services such as DDoS mitigation service is unavailable due to heavy resource usage by victim's web-service, we may observe unwanted delay in the overall mitigation activity leading to increased losses. Therefore, the system state awareness is the most important and primary requirement to achieve any mitigation objectives.

II. Helping the critical situation: The co-located services are also required to help the critical state created by the attack or other faults. Subsequently, additional assistance such as fault tolerance by backup servers, additional resources using auto scaling, VM migration, load balancing and application backups can be provided. Achieving these goals, may also require manual intervention to recover from the attack.

There are many recent DDoS attacks, known as "Smoke-screening attacks" [7], which are used to launch other severe attacks behind the massive DDoS attacks. The success of smoke-screening attacks lies in the heavy consumption of victim resources during the attack. These resources can be computing resource or manpower resources to help the attack mitigation. The above requirements may also help in minimizing the smoke-screening attacks if other services looking after security primitives such as data breaches, remain available during the DDoS attack and react to the situation. For example, assume DDoS attack `A1` is targeting a service `s1` running on a VM. Service `s2` is there to take care of some other important attacks such as data breaches. There is a smoke-screening attack `A2` which may bypass the service `s2` on the same VM as `s2` is no more available due to the internal collateral damages. In this case, the attack `A2` might become successful as the service s2 which is responsible for stopping the attack `A2`, is unavailable for protection. We discuss this scenario in further detail using resource contention in Section 4.

3. DDoS Attack Mitigation: A Closer Look

In this section, we extend our discussion on DDoS attack mitigation and its resource requirements. To quantify this, we perform experiments to critically analyze the attack launch and subsequent attack mitigation. The experimental setup is as shown in Figure 3 and the detailed experimental configuration settings are shown in Table 1. In the experimental setup we have a victim VM hosted on a physical server and an attack VM hosted on another physical server. The major motivation to conduct this experimental attack study is to see the effects of DDoS attacks on important and critically required co-located services during DDoS attack periods.These co-located services are running on the same VM as operating system processes. We prepare target service by giving resources equivalent to an Amazon EC2 "C4 Extra Large" instance [13]. Our target services is representative service of most of the dynamic web services, which converts images from JPEG format to PNG format. For this experiment, we are using the input image of size 1 MB. We have used other image sizes to perform detailed comparison with the proposed solution in Section 5. These sizes are representative sample of websites across the globe [14].

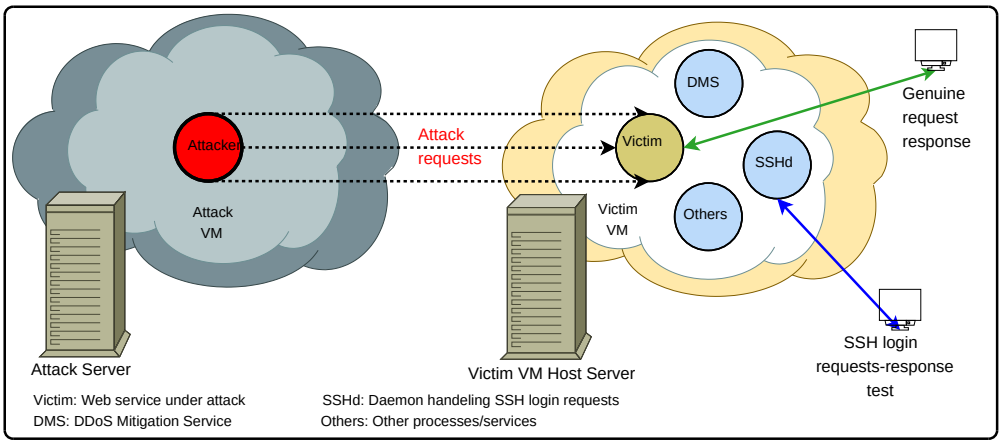


Figure 3: Experimental setup for DDoS attack analysis

To see the impact on the availability of other services inside the victim VM, we carry out a DDoS mitigation service to detect and report the attack. Instead of using any sophisticated DDoS mitigation service, we use DDoS-Deflate [15], which is a connection count based attack filtering service. We use this tool with its default settings. These settings allow this tool to identify attackers, who are establishing more than 150 connections with the server. The reason behind using this naive detection mechanism is to detect the attack and maintain the applicability of our experiments to any other tool. Therefore, any other DDoS mitigation service can also be used in place of DDoS-Deflate. Major motivation of this attack experiment is to see the attack detection time, overall service downtime and effects on other services. Instead of sending attack requests from large number of sources, we are sending attack requests from a single source. This enables us to see invariability of attack detection time values reported in the experiments.

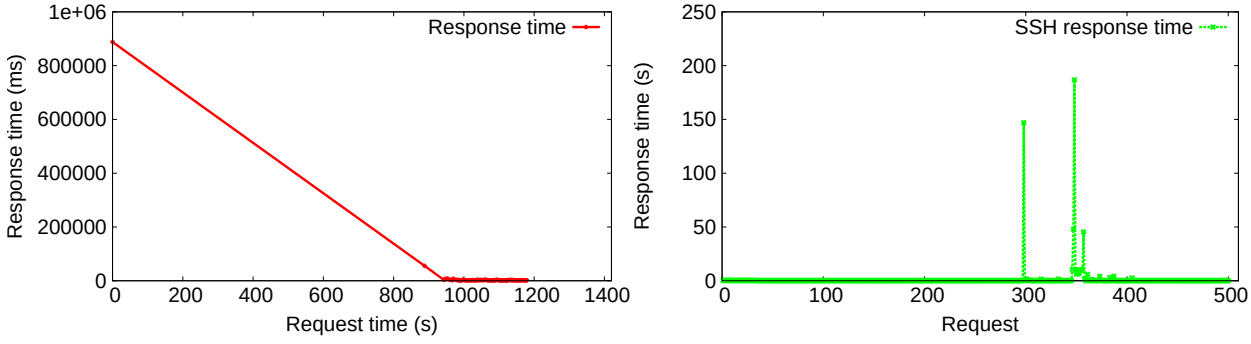
Component/Resource	Configuration Settings
Victim host physical server	Dell PowerEdge R630, Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz
Resources on physical server	8 processors (4 cores each), total memory 96GB
Virtual Machine Monitor/Hypervisor	XenServer Version 6.5
Victim, attacker, benign operating systems	Ubuntu Version 14.04
Victim service	Image conversion web service
Victim VM configuration	Processors 4 vCPUs and 8 GB Memory
Attacker VM configuration	Processors 2 vCPUs and 4 GB Memory
Benign VM configuration	Processors 2 vCPUs and 4 GB Memory
Attacker and benign traffic launch application	ApacheBench2
Attack traffic rate	500 concurrent requests (for a total of 5000 requests)
Benign traffic rate	1 concurrent request (for a total of 100 requests)
Network	1Gbps
Other services for availability test	SSH to have remote login-logout one after another during attack period

Table 1: Configuration settings for the attack experiments

In case of multiple sources the attack detection times vary depending upon the connection establishment times and share of established connections by each source. We consider "SSH" service as an important co-located service for impact evaluation. In SSH, the VM owner sends a SSH session request to the victim server and if the session is granted immediately, it logs out from the session. This test is done for 500 SSH login-logout cycles during the attack period. We tried to keep the attack scenario real by using a dynamic service attacked by a DDoS attack and the victim owner trying to access the victim's machine using SSH service. To check the availability of the target machine, a genuine user sends a request, one after another to the victim's service for a total of 100 requests during the attack period. We launch the DDoS attack on the victim's service by sending 500 concurrent attack requests. We create the attack traffic on the basis of the requirements given in [16]. The attack traffic, genuine traffic and SSH traffic are sent simultaneously to the victim's VM. We show the attack results and associated quantified metrics in Table 2. Figure 4a, illustrates the response time behavior of victim web service. We show the SSH login-logout completion time in Figure 4b.

Attack Reporting Time	Victim Service Unavailability Time	SSH Unavailability Time	Maximum Response Time (SSH)	Minimum Response Time (SSH)	Average Response Time (SSH)
39s	945s	517s	186.830s	0.129s	1.226s

Table 2: Various Metrics: DDoS attack experimental study



(a) Service response time during attack

(b) SSH response time during attack

Figure 4: Behavior of co-located services during attack

The DDoS mitigation technique could detect the attack source based on its policies after 39s of the attack being launched. Subsequently, the victim service becomes unavailable for a period of 945s. The graph shown in Figure 4 (a) represent the request-response behavior at the benign senders end. Benign sender sends a total of 100 requests with one concurrent request at a time. These graphs show the times when each request (1^{st} to 100^{th} request) is served one after another. The X-axis in Figure 4(a) show the "Request time" which is the time the request was sent from the benign sender. The major activities performed by DDoS mitigation service during the attack period includes:

1. Count based filtering based on connections
2. Adding rules to firewall for rejecting the attack traffic from the detected sources
3. Cleaning the established connections involving detected attack sources

After detecting the attack, the mitigation service adds rules to firewall and drop all the tcp connections involving the attackers. The SSH service was not available for more than 500s during the attack period (See spikes in Figure 4b). This gives a direct indication on the unwanted effects on other important services. In addition to all these observations, the victims' VM was not responsive on the XenCenter interface, which makes the service unavailable even for direct interactions in the data center. We attribute resource contention/race among services as the major reason behind such attack effects.

We term these attack effects as "Internal Collateral Damages". These services are co-located at the level of operating system and the fair sharing provided by CPU or device scheduler does not provide the required resource isolation among these services. The severity of resource race created by the DDoS attack is such that, even the VM interface is not available. We attribute the observed resource contention to the most basic resources such as CPU, memory, network and disk resources, which are required by all the services, though in different proportions at different times. In Section 4, we model the operating system level service requirements that will be used in the design of our proposed solution.

4. DDoS Attack Mitigation: Modeling OS level requirements

In this section, we provide an abstraction for operating system level resource race and contention among the running OS services. We elaborate resource exhaustion states and the resulting resource contention. This discussion is also applicable to traditional operating systems which are not running on top of virtual machines/clouds. A cloud infrastructure runs a number of physical servers (P_i, i = 0, 1, ... n), these servers host various services (S_k, k = 0, 1, ... m). Service S_k in turn runs various VM instances (I_j, j =0, 1, ... p) to host and manage cloud based services. Victim's VM which is affected by a DDoS attack would result into a stressing condition of one or more of its resources. The exhaustive set of these bottleneck resources include CPU, memory, swap, disk I/O operations, bandwidth, and number of connections/ports.

The DDoS attacks may also choose to exploit any general and weak target resource or utility, which is an application level resource such as a temporary buffer created by a programmer. This buffer may be exhausted by some specific requests or a higher frequency of requests even in the presence of availability of other ample resources. Each physical server has following shared resources, CPU, memory, bandwidth, and disk, which we represent using `C`, `M`, `B`, and `D` respectively. Any other resource of interest is represented by `O`. We define a function, *Resources*(*entity*), which provides the available resources of an entity.

$$Resources(P_i) =< C_i, M_i, D_i, B_i, O_i > . \tag{1}$$

Resources available on each VM instance is a subset of the resources available on physical server.

$$Resources(I_j) =< C_j, M_j, D_j, B_j, O_j > . \tag{2}$$

The type of virtualization and resource sharing techniques, decide the actual resource allocation to VMs. To have a generic view, the allocation of the resources CPU (`C`) resources are usually represented as number of processors in case the dedicated CPUs are provided to each VM. In case, the VMs share processors, the `C` resource can be represented in the form of CPU shares or CPU time. On the other hand, resource such as memory (`M`) and disk (`D`) are divided among the hosted VM instances with clear limits on the size.

Bandwidth(B) is the total throughput on NIC which is limited and shared using network virtualization techniques. The requirements of the hosted instances can only be met, if the sum of individual instance requirements are less than or equal to the host physical server. Following Equation 3, must hold true, If there are two VM instances, I_1 and I_2, to be hosted on P_i.

$$Resources(P_i) \geq \sum_{j=1}^{2} Resources(I_j) + Resources(H). \tag{3}$$

$Resources(H)$ represent the requirements of the host operating system on each physical server. The Equation 3 should also hold true for individual resources such as the sum of the individual resource requirements (for example CPUs, C) of each instance should be less than or equal to the individual instance's CPU requirements. If this equation does not hold true, the auto-scaling algorithm will be triggered to find and provide more resources. These resources can be acquired using migration or creating more VM instance(s) at other physical servers. Now, when we can observe the effects of DDoS attack cases (Section 3), we estimate that the resource race taking place at the level of individual processes should also be taken into account. On the other hand, each VM Instance I_j runs a complete operating system on top to support the service.

The two most important primitives at the level of operating system are, processes and resources. These processes include the system processes, daemons and other application processes. We represent these processes or services by p_l, l = 0, 1,...q. Among these processes, victim service (p_v), DDoS mitigation service (p_d) and other set of processes p_o, are of primary interest to our discussion. Let us take an example of a DDoS attack, which impacts the resources, such as CPU, memory and a number of network connections. At any given point in time during the attack, the resources available with the victim process under attack are:

$$Resources(p_v) =< C_v, M_v, D_v, B_v, O_v > . \tag{4}$$

The above resource requirement does not show resources occupied by the service at all times. It just represents the need of a service during a given point in time. During the attack, the victim's service will be stressing one or more resources such as CPU, memory and connections to their maximum. At the same time, the VM owner would want the DDoS mitigation service to act on the situation and perform the detection. Similarly, the other important services like remote access (ssh) or other linked services like firewall need to act on time. The resource requirement of DDoS Mitigation Service (p_d) and other important set of processes p_o during the attack period would be

$$Resources(p_d) =< C_d, M_d, D_d, B_d, O_d > . \tag{5}$$

and

$$Resources(p_o) =< C_o, M_o, D_o, B_o, O_o > . \tag{6}$$

We can also write

$$Resources(I_j) \geq Resources(p_v) + Resources(p_d) + Resources(p_o). \tag{7}$$

If we rewrite Equation 7 for individual resources during the attack duration t.

$$C_j \geq C_v + C_d + C_o. \tag{8}$$

$$M_j \geq M_v + M_d + M_o. \tag{9}$$

$$D_j \geq D_v + D_d + D_o. \tag{10}$$

$$B_j \geq B_v + B_d + B_o. \tag{11}$$

$$O_j \geq O_v + O_d + O_o. \tag{12}$$

As the resources such as memory or CPU are usually not dedicated in their entirety to a single process (unless it is pinned in isolation). Not fulfilling one or more of the above equations, is the real cause of delayed action by DDoS mitigation service (p_d) and unavailability of other services (p_o) during the attack period.

In addition, the resource consumption behavior of a DDoS mitigation service is also multi-resource (using CPU, memory, disk and network). Therefore the resource contention between these services occurs. From the CPU scheduling perspective, it appears that the resources will be fairly given to each process based on their priority, however the multi-resource and interlinked resource requirements show the internal collateral damages. The major reason for adoption of virtualization technology as an enabler for cloud computing is due to the features such as performance isolation among VMs and multi-tenancy. However, virtualization of resources has no major role to play, while the internal performance isolation among processes is considered. The operating system scheduling strategies and resource management activities remains mostly same for operating systems running on both virtual machines and physical machines.

Cloud platforms use the auto-scaling algorithms to govern and facilitate the on-demand dynamic resource allocation in terms of horizontal and vertical scaling. These algorithms monitor various resource usage metrics and based on the resource triggers, they provide resources or withdraw the idle resources. Giving more resources to solve the problem of internal collateral damages among processes, would not be suitable for the following reasons:

I. Auto-scaling algorithms operate as cloud-level or hypervisor-level resource allocation primitives. The resource requirements of individual processes inside the VM, will not be visible to the auto-scaling methods. Additionally, auto-scaling algorithms add additional resources on top of a VM and not to individual processes or services running inside a VM.

II. Many of the existing solutions [10] advocate the application resource scaling to absorb the DDoS attack. However, the resource scaling may not necessarily help to mitigate quickly and provide timely resources to p_d and p_o. The additional resources and instances will also face a similar situation as the victim's service under attack and will also overload the added resources. The additional resource can surely become useful for the mitigation processes if the p_d process can use it readily.

DDoS attack attributes such as attack duration and attack scale also decide the attack effects and resultant impact on the victim service. Based on experimental study and the above discussions below are the major requirements for an effective solution to internal collateral damage.

I. **Strong Internal Isolation**: An effective solution should provide a strong resource and performance isolation among processes. Performance isolation should be guaranteed for processes, which are used in ensuring the availability and attack mitigation capabilities in a victim's server. Strong isolation can also be ensured by having resource usage limits.

II. **Resource Availability**: Resource requirements to maintain availability is directly related to the requirement I above on strong internal isolation. However, the availability of required resources should be ascertained for each process. Capacity planning and server consolidation areas have a large number of approaches [17] dealing with the actual resource requirements of co-located virtualized servers. Similar solutions can be extended to ensure the timely resource availability to DDoS mitigation and other related services.

III. **Service Performance**: The victim service performance for benign users should not be affected due to the solution targeting the above requirements. This metric is important as the service quality should not be hampered due to the implementations for isolation and availability.

We will use these requirements as the basis of our proposed solution design in the next section.

5. Victim Service Containment: Proposed Solution

Based on the listed effective solution requirements in the last section, we propose a novel solution based on the resource containment of victim process groups at the level of operating system. The primary idea of this solution lies in ascertaining the required resource share for each service. However, in the presence of DDoS attacks, the real requirement is to ascertain and fix the minimum resources required for services (except victim service), to remain available and produce the required and timely outcome. The resource contention situation arising out of DDoS attacks is the major cause behind the unavailability. Operating system schedulers try to achieve these requirements by employing various scheduling primitives and access primitives such as process groups. However, they do not provide the strong guarantee about the control against situations such as "Internal Collateral Damages".

The proposed method "Victim Service Containment" is detailed in Algorithm 1. The proposed algorithm is a proactive approach for resource calculation. This algorithm evaluates the resource requirements of all the services (except the victim service), while the service is not under attack. Resource containment limit is defined by keeping the resource requirements of all the services (except victim service) as minimum required resources for the rest of the system. The remaining resources, become the upper limit or resource containment limit for the victim web-service. We describe the solution using Figure 5. We show a rectangle to represent all the available resources on the virtual machine. In Figure 5(i), while the status is "no attack presence", there are ample resources available on the virtual machine.

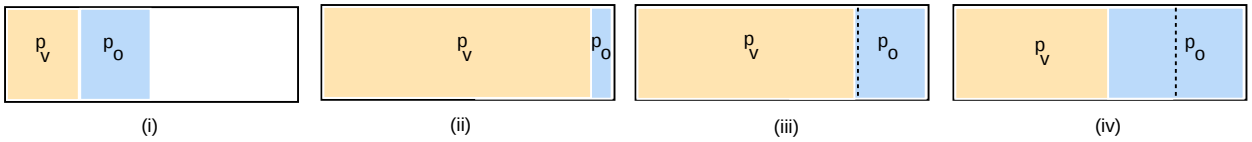


Figure 5: Victim Service Containment (V is victim service and O is all other services), (i) Normal operation, (ii) During DDoS attack, (iii) After "Victim Service Containment" during DDoS attack, and (iv) No containment for other services during normal operations

During a DDoS attack, the resource usage reaches its maximum for one or more resources and the victim's service occupies or uses most of the resources and hence the other services do not get the necessary resources in a timely manner. Once the resource containment method is applied, we define a maximum limit of resource usage for the victim's service, which it will not be allowed to exceed (dashed line in Figure 5(iii)). It is important to note that this limit has no effects on the other services and they may still use the available resources (Figure 5(iv)).

For our DDoS attack experiments, we have evaluated resource requirements for CPU, memory, disk and network bandwidth. Similarly, the proposed algorithm 1, can be extended to evaluate the resource requirements for other fine grain resources. The most important phase of the algorithm is to identify the resource requirements of all the services prior to a real DDoS attack. Server consolidation strategies in cloud computing are there to provide optimization algorithms to allocate resource to VMs. Our major concern is to provide performance isolation and resource availability inside a VM, therefore, we calculate the requirements (limit $L = Resources(I_j) - ResourcesNoAttack(I_j)$ in the algorithm) taking into consideration these aspects. Each resource can be controlled and shared using multiple dimensions. For example, a CPU can be distributed using "Shares" and also as dedicated CPUs. Similarly a disk can be divided from the perspective of disk size based distribution as well as maximum read and write speed shares. We take the specific example of the VM under consideration in the attack experiment shown in Section 3. We also show the specific resource limits, we enforced.

I. CPU: The CPU requirements of the complete operating system are around 5%-20% for each of the four CPUs, while there is no attack. This sums to around 20%-80% out of the total 400% covering all the four processors which equates to one processor out of every 4 processor for other services. Therefore, to have a stronger isolation, we give a limit of 3 processors to the victim's service so that one processor is always available for all the other services.

II. Memory: The memory limit for the victim's service is defined by keeping the minimum memory usage by the operating system, when there is no attack. We have not included the requirements of victim web-service into consideration while calculating the memory requirements of the complete system. The total VM memory is 8GB. The memory usage while there is no attack is 1.6 - 2GB, which adds to an oveall memory limit of 6GB.

III. Disk: We have controlled the disk read and write speeds for the victim's service. The maximum read and write speeds of the storage disks using IO benchmarks is around 75-80 MBps. Looking at the memory occupancy (75%), we have decided to limit the speeds to 60 MBps which is 75% of the total 80 MBps.

IV. Network: Network traffic can only be prioritized using the control group facilities. Therefore, we have given top priority to all the other traffic and the next priority to victim's service' traffic.

Algorithm 1: Victim Service Containment

Victim Service Containment;
Data: Total available resources $Resources(I_j) =< C_j, M_j, D_j, B_j, O_j >$
Victim Service p_v
Result: Resource limit, L for victim service (p_v)
initialization;
Show $L = Resources(I_j) - ResourcesNoAttack(I_j)$;
end

Applying the proposed algorithm also changes the the overall service performance. Once the limit L is defined by the victim service containment algorithm, the resource limit can be enforced in the following two ways.

A. Limiting from the available resources: We have used this approach to limit the available resources in the resource limits we have calculated previously. One disadvantage of this approach is that it may limit the performance of the victim's service due to the resource constraints, which restricts to lower amounts of resources as compared to the earlier limit (or no limit) for all the resources.

B. Scale to get more resources: In this case, the resources required for the whole operating system including all the services for attack free cases can be added over and above the VM resources. The victim's web service can be limited to the resources, which were already available and the added resources on top, can be used by the other services as a minimum resource guarantee. This will provide both the advantage of victim's service performance as well as resource containment. The additional cost of added resources is a factor while opting for this approach.

6. Evaluation Results and Discussion

We evaluate the "Victim Service Containment" algorithm by observing various metrics related to victim service and co-located service performance during the DDoS attack. We employ the experimental setup as shown in Figure 3. Our major focus is towards DDoS mitigation service and SSH service. To achieve this, we first launch the attacks against target VM, while there is no victim service containment in place. We use three different services (500KB, 1MB and 2MB image size for conversion) to see these effects. In addition, we have employed the algorithm 1 to apply the resource containment on victim's web-service and tested against the same set of services once again. We show the collective results of these attack experiments in Table 3 and Table 4.

Target Web Service Type	**Attack Reporting Time** (No VSC)	**Attack Reporting Time** (With VSC)	**Service Down Time** (No VSC)	**Service Down Time** (With VSC)	**SSH Down Time** (No VSC)	**SSH Down Time** (With VSC)
500KB	37s	40s	943s	346s	0s	0s
1MB	39s	38s	945s	375s	517s	0s
2MB	2040s	41s	2383s	2801s	1959s	0s

Table 3: Various performance metrics: before and after the "Victim Service Containment (VSC)"

We show the victim's service response time and SSH response time in Figure 6 and Figure 7 respectively. We discuss the results and observations with respect to important evaluation metrics in the following.

A. DDoS mitigation service performance: To see the performance of DDoS mitigation service, we monitor the attack reporting time and victim service availability time. Attack reporting time is when the DDoS mitigation service detects the attacker IPs and adds them to the firewall. Attack reporting time in these attack cases should be small as the detection criteria is immediately fulfilled by the number of connection requests launched by the attacker.

Target Service Type	**Max. Response Time** No VSC	**Max. Response Time** With VSC	**Min. Response Time** No VSC	**Min. Response Time** With VSC	**Avg. Response Time** No VSC	**Avg. Response Time** With VSC	**VM Interface** No VSC	**VM Interface** With VSC
500KB	0.363s	0.267s	0.127s	0.131s	0.148s	0.152s	Not available	Available
1MB	186.8s	0.284s	0.129s	0.130s	1.226s	0.149s	Not available	Available
2MB	765.2s	0.298s	0.129s	0.134s	4s	0.154s	Not available	Available

Table 4: SSH performance: before and after the "Victim Service Containment (VSC)"

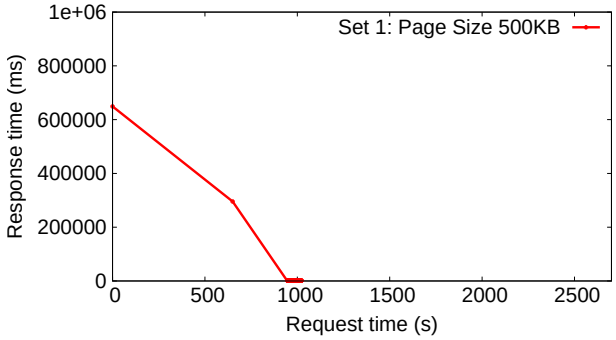
However, the attack reporting time is quite high (2040s) for page size 2MB, which gives a clear indication of resource contention faced by DDoS mitigation service (Figure 6e and Figure 6f). As the amount of resource contention generated by victim's service, serving image conversion for 500KB and 1MB is not significant, the attack reporting time for these services is only around 40s. The victim's service availability time is greatly minimized in the cases of victim service serving 500KB and 1MB conversions (figures 6a, 6b, 6c and 6d). However, for 2MB image size, the victim's web service is unavailable for an additional period as compared to the case of "no service containment". This is due to the fact that the resource limits imposed on victim's service provides limited amount of resources compared to the resources available in the case of "no service containment". This issue can be resolved by selecting for approach "Scale to get more resources", which is discussed in Section 5. On the other hand, the victim's service becomes available after the attack reporting time and by terminating the attack connections in all the cases. Additionally, the victim service availability time for the cases of 500KB and 1MB image conversion services has reduced from 943s to just 345s after the service containment.

B. SSH service performance: Proposed solution has solved the issues with the availability of co-located services during the attack period. The availability aspect is shown by the performance of SSH service "login-logout" response time in Figure 7. We show the average, minimum and maximum response times observed for the SSH service in Table 4. In the extreme resource contention case (2MB image conversion service), the SSH service was not available for more than 1959s out of the total downtime of 2383s. Similarly, for the case of 1MB service, the SSH unavailability time is around 517s. On the other hand, the SSH unavailability time for attack case on 500KB image service is 0s, which means that the SSH services was available throughout the attack period. However, the SSH service has been affected in having peaks in the response time during start of the attacks (Figure 7a). These peaks have been resolved up to a certain extent after service containment (Figure 7b).

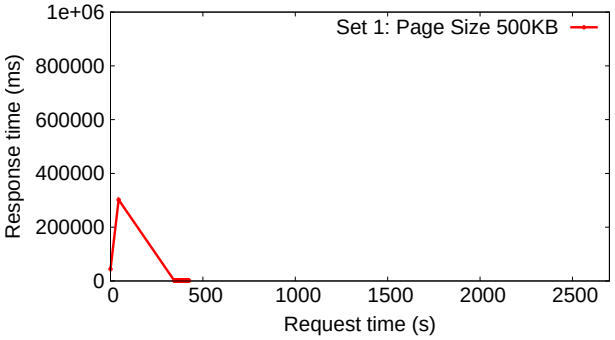
C. Availability of VM interface: We also use the responsiveness of the VM interface on the server side (XenCenter Interface) as an important criteria during the attack period. As the attack mitigation often requires remote as well as manual intervention to the victim's computer to see the real cause of the situation. We have shown the availability/responsiveness of the VM interface during various attack cases in Table 4. The availability issue of VM interface is completely solved by the resource limits posed by the victim's service containment.

Now, we discuss the various aspects related to the DDoS mitigation process in connection with the proposed service containment algorithm.

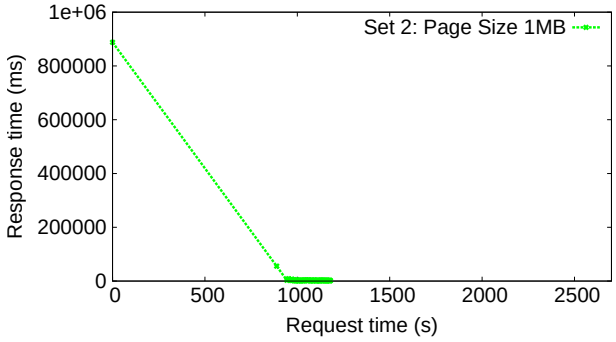
I. Disadvantages of resource limits: The resource limits we put during containment, might not be the most appropriate limit for the different applications. The proposed algorithm is flexible to use with different requirements for different applications, however, the resource limit will be independent of the incoming attack features. One obvious disadvantage of the containment is on the performance of the victim's service, if the resource limit is applied on the available resources. However, this can be easily resolved by having additional resources on top of the VM resources already available with additional cost. On the other hand, all the other co-located services have no resource limits as they can use any amount of the available resources. The resource limits for other process groups can also be decided, if there are incidents of resource contention by them. Previous overhead studies show that there is a small overhead of memory containment limits due to the fine grain monitoring [18].



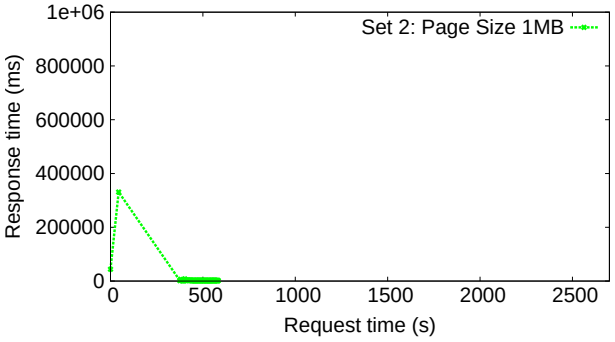
(a) Service response time during attack



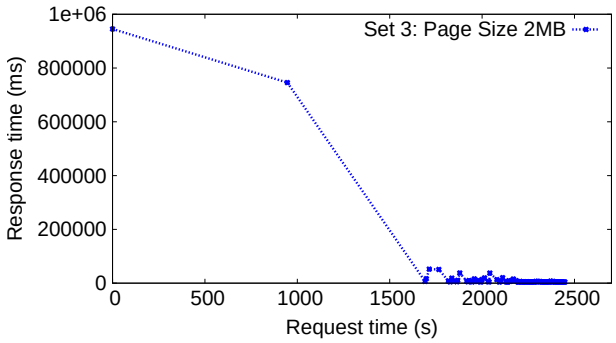
(b) Service response time during attack (After Containment)



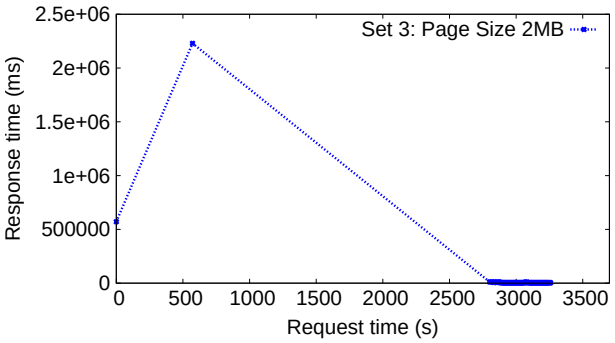
(c) Service response time during attack



(d) Service response time during attack (After Containment)



(e) Service response time during attack

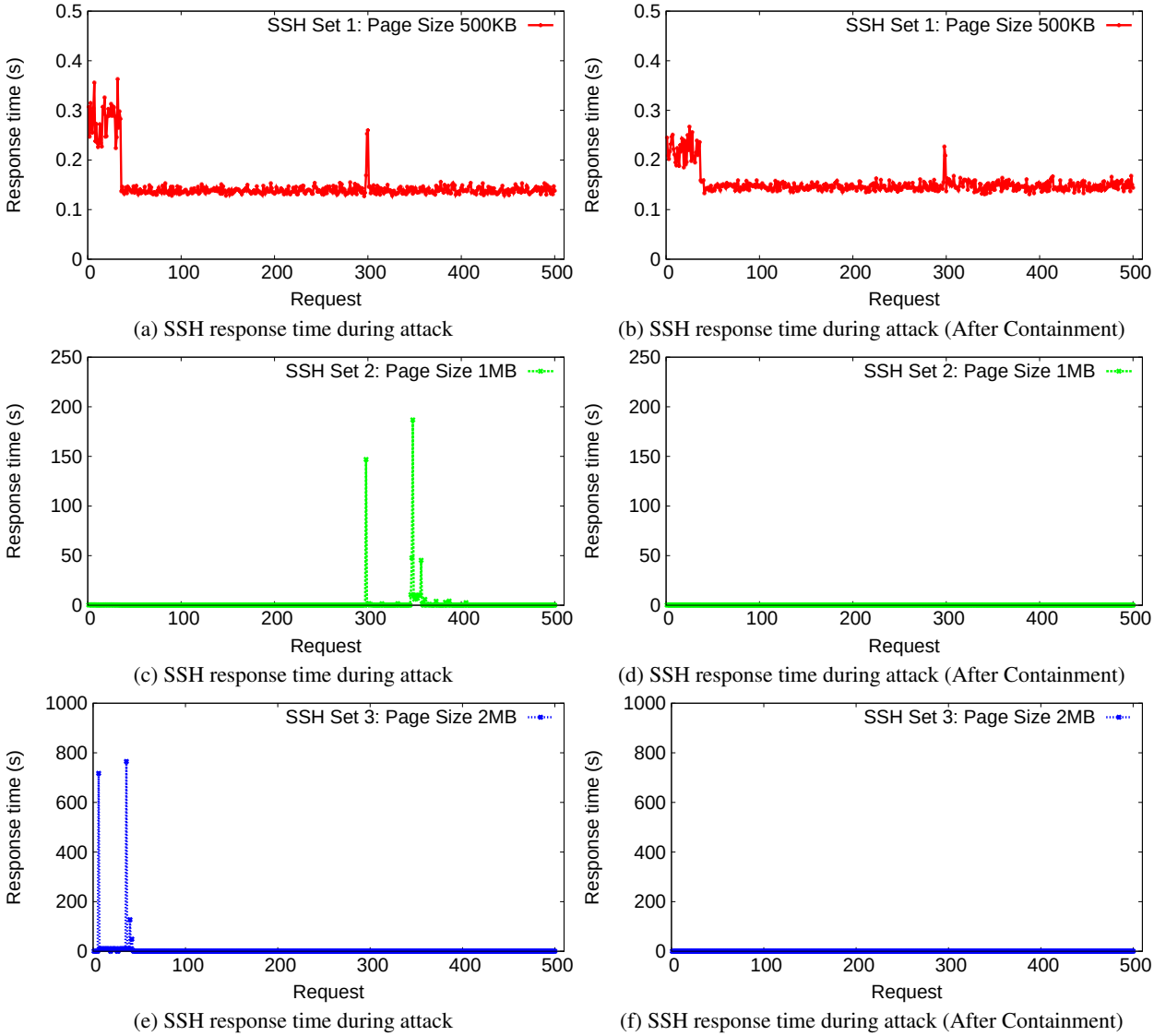


(f) Service response time during attack (After Containment)

Figure 6: Service response time during attack before and after containment

II. Cloud level collateral damages: The collateral damages shown by studies in [8] are due to the resource race and contention among co-hosted VMs. Most of the resource contentions are due to the shared or virtualized nature of the resources. We have shown that the elimination or minimization of resource contention at operating system level, may also help in achieving positive results outside the VM. Additionally, we have provided the attack cases, where the resource contention is extreme for CPU, memory and disk resources, where resolving internal collateral damages, may minimize the collateral damages among co-hosted VMs.

III. Victim separation: The collateral damages due to the DDoS attacks on cloud computing, are also extended due to the victim's service running multiple VM instances of services on different server hardware such as load balanced servers. Victim's VM separation may give the desired results of minimization of collateral damages due to the removal of resource contention among VMs. However, in the proposed approach, the effects are minimized to the victim's VM itself. We have extended this separation and minimized the effects to the other services.



(a) SSH response time during attack

(b) SSH response time during attack (After Containment)

(c) SSH response time during attack

(d) SSH response time during attack (After Containment)

(e) SSH response time during attack

(f) SSH response time during attack (After Containment)

Figure 7: SSH response time during attack before and after containment

IV. Fixed Infrastructure Protection: The proposed solution also applies to services not running on virtualized or cloud platforms. Resource containment can be directly applied to the host operating system to perform quick DDoS mitigation and availability of all the other services.

V. Proactive vs. Reactive Implementation: In the proposed victim's service containment approach, we have used the proactive implementation, where the resource containment limits are calculated and imposed on the victim's service. Hence, the method "Victim Service Containment" is enabled all the time. Triggering the "Victim Service Containment" approach just during the attack (reactive implementation) can also be used if the resource limits are already calculated for other services. However, applying containment limits while the resource in consideration (such as memory) is already at the maximum utilization is difficult to achieve as it requires the usable memory for the victim process to be shrunk. Therefore, we have used a proactive approach in the contribution.

VI. Smoke-screening attacks: There are multiple recent incidents, where DDoS attacks are used to bring down the organizational resources in the DDoS mitigation and other severe cyber attacks are launched on targeted services. We relate the problem of "internal collateral damages" with these attacks, as in the presence of DDoS attacks almost all the resources are used up to their maximum limits and create a contention for services. We have firewall and other access control primitives such as SELinux to stop and report these attacks. However, these services may not be available and may not give the desired and timely outcome due to the heavy resource contention due to the attack. The proposed resource containment methods take care of these attacks, if there is a service available, which has a capability to stop/detect the real attack underneath the DDoS attack.

VII. Contention formed by other services: In the proposed approach, we have aimed at the resource containment of the victim's service. However, there might be instances where the resource contention is developed by the services other than the victim's service. This might be due to the faulty code, resources or a real attack on the other co-located service. There are also recent attack incidents, where DDoS attacks are targeted at DDoS detection tools [19] to slowdown the mitigation. In this case, multiple resource containment groups may be formed to help in this situation by anticipating the possible target services.

7. Related Work

DDoS attack trends and mitigation methods have always been seeing a notable growth since the inception of Internet based computing. There are a number of surveys available to list the DDoS attack and solution hierarchies [4]. Recently, there are also a number of these surveys and taxonomies which are available in the area of DDoS attacks in the context of cloud computing [5]. Most of the DDoS solutions rely on differentiating the attacker request patterns from the genuine request patterns. These methods work towards identifying anomalies in the access pattern or similar features. Similarly, there are a number of contributions, which provides proactive attack prevention methods by using the challenge response protocols or similar cryptographic authentication techniques [20].

There are many methods, which provide the DDoS mitigation based on the traffic evaluation without considering any specific features of cloud computing. Authors in [21] provide the notion of Fraudulent Resource Consumption attacks on cloud services and gave methods to deter these attacks. Additionally, they provide the characteristics of these attacks. Next we provide the details of the methods that are based on resource management aspects of DDoS mitigation. Authors in [22] provide the effects of DDoS attacks on virtualized servers. The authors show how DDoS attacks affect a target VMs performance using various virtualization techniques. Authors also compare the behavior and performance degradation due to the DDoS attacks on virtualized and non-virtualized environments.

Authors in [23] provide a DDoS detection and defense method by measuring the utilization thresholds of VMs. Authors propose to mitigate the DDoS attacks by migrating the victim's VM to the reserved network isolated server resources. Authors also proposed to bring the victim's VM back at its original place once the attack is mitigated. Performance of this technique is highly dependent upon the attack duration and repetitions. As for longer and repetitive attacks, there will be longer service downtime and overhead of repetitive migrations. Authors in [12] provide cases of DDoS attacks targeted at cloud services and also attacks originated by cloud services. In this work, the authors propose to use reserved backup resources for DDoS mitigation.

Authors in [20] proposed moving the target based defense strategies. Authors proposed to have multiple hidden servers and dynamically assign clients to these servers. In case of attacks, they proposed to shut down the existing proxy servers and move the target to other available servers. In this contribution, scalability and overhead of managing large numbers of servers is an issue.

Moving onto other resource management based DDoS mitigation solutions, authors in [24], proposed a collaborative multi-level DDoS mitigation solution based on the attack detection and control at VM level, application level and the hypervisor level resource usage. Cloud resource scaling based DDoS mitigation method is proposed by authors in [10]. Authors show that the quick resource scaling of Intrusion Detection Systems (IDS) in terms of their instances, during the DDoS attacks, may help in the quick DDoS mitigation. Authors proposed to utilize the dynamic resource allocation provided by the cloud computing infrastructures to harness these solutions. There are a number of industry solutions, which also propose to use the resource scaling as a mechanism to absorb the DDoS attack as quickly as possible [6].

Obviously, there are cost considerations of these scaled resources, which must be considered before deciding the scaling limit. Authors in [25] provide an extension to these solutions and proposed to use the untrusted resources available at Content Delivery Network (CDN) clouds to help the mitigation with the resources.

Authors in [11] provided a DDoS aware resource allocation method, which consults a DDoS detection module before directly scaling the resources to save the costs and attack effects. Authors in [8] provided collateral damages to non-target VMs due to the DDoS attacks on co-hosted VMs. Authors also extended these collateral damages to cloud scale by the DDoS effect spread due to these attacks.

We see that most of the contributions related to DDoS attack areas consider the DDoS mitigation activity as an activity, which works in parallel with the victim's service. Most of the solutions working on the resource management aspects of DDoS mitigation also work on fastening the DDoS mitigation by adding more resources in terms of additional instances. However, we do not see contributions working towards finding the problems with DDoS mitigation and other services during the attack. Our observations related to "Internal Collateral Damages" among co-located services and subsequent availability concerns are the first novel attempt to see the resource contention and availability as a problem. "Victim Service Containment" method proposed in this work, is the first novel solution in this area to combat DDoS attacks while minimizing the "Internal Collateral Damages".

8. Conclusions and future scope

Many of the enterprises across the world today, shifted their mission critical applications to cloud. On the other hand, "DDoS for Hire" services are also available to the attackers. Hence, both attackers as well as victim enterprises are using cloud infrastructures. In this work, we show a novel resource contention phenomenon, "Internal Collateral Damage", which is observed in the presence of DDoS attacks on the cloud services. Real-time experiments demonstrate that these attacks may severely affect the timely and expected outcome of critical services such as the DDoS mitigation service. We observed the attack effects on co-located SSH and VM interface services which are mandatory tools for last resort manual access for control. Both the services also become unavailable due to the resource contention generated by the victim service. The external auto-scaling methods working from outside the target VM, would not help in the mitigation due to the resource problems at the process level.

We model the DDoS mitigation activity as OS level resource management problem. We show that the victim service is responsible for the resource contention due to its heavy resource usage for one or more resources like CPU, memory, disk and bandwidth including other application resources. To overcome these issues, we propose a novel victim resource containment by which the rest of OS services including critical services like DDoS mitigation service and remote log-in service remain available irrespective of the presence of any severity of DDoS attacks. We develop an illustrative example of "Victim Service Containment algorithm" and address the service unavailability problem. Experimental attack results demonstrated the efficacy of our proposed solution leading to the overall improvement of attack reporting and service availability. This novel attack characterization also opens up multiple issues related to DDoS mitigation, resource management activity and availability of other services.

9. Acknowledgement

Gaurav Somani is supported by a Teacher Fellowship by University Grants Commission, Government of India, under the XII Plan (2012-2017). Experimental setup for this work is supported by Security Analysis Framework for Android Platform (SAFAL, Grant 1000109932) by Department of Electronics and Information Technology, Government of India. Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission (PCIG11-GA-2012-321980). This work is also partially supported by the EU TagItSmart! Project (H2020-ICT30-2015-688061), the EU-India REACH Project (ICI+/2014/342-896), the Italian MIUR-PRIN TENACE Project (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] Kaspersky Labs, Global IT Security Risks Survey 2014 - Distributed Denial of Service (DDoS) Attacks, http://media.kaspersky.com/en/B2B-International-2014-Survey-DDoS-Summary-Report.pdf (2014).

[2] Arbor Networks, Worldwide Infrastructure Security Report Volume XI. (2015).

[3] C. Burt, Majority of Organizations Run Mission-Critical Apps in the Cloud: Verizon Report, http://www.thewhir.com/web-hosting-news/majority-of-organizations/-run-mission-critical-apps-in-the-cloud-verizon-report (2015).

[4] T. Peng, C. Leckie, K. Ramamohanarao, Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems, ACM Comput. Surv. 39 (1). doi:10.1145/1216370.1216373.

[5] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, R. Buyya, DDoS Attacks in Cloud Computing: Issues, Taxonomy, and Future Directions, arXiv preprint arXiv:1512.08187.

[6] Amazon Web Services, AWS Best Practices for DDoS Resiliency, https://d0.awsstatic.com/whitepapers/DDoS_White_Paper_June2015.pdf (2015).

[7] I. Kolochenko, DDoS Attacks: a Perfect Smoke Screen for APTs and Silent Data Breaches, http://www.csoonline.com/article/2986967/advanced-/persistent/-threats/ddos-attacks-a-perfect-smoke/-screen-for-apts-and-/silent/-data-/breaches.html (2015).

[8] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, DDoS attacks in Cloud Computing: Collateral Damage to Non-targets, Computer Networks 109 (2) (2016) 157–171.

[9] P. Menage, CGroups, https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt (2016).

[10] S. Yu, Y. Tian, S. Guo, D. O. Wu, Can We Beat DDoS Attacks in Clouds?, Parallel and Distributed Systems, IEEE Transactions on 25 (9) (2014) 2245–2254.

[11] G. Somani, A. Johri, M. Taneja, U. Pyne, M. S. Gaur, D. Sanghi, DARAC: DDoS Mitigation using DDoS Aware Resource Allocation in Cloud, in: 11th International Conference, ICISS, Kolkata, India, December 16-20, 2015, Proceedings, 2015, pp. 263–282.

[12] J. Latanicki, P. Massonet, S. Naqvi, B. Rochwerger, M. Villari, Scalable Cloud Defenses for Detection, Analysis and Mitigation of DDoS Attacks, in: Future Internet Assembly, 2010, pp. 127–137.

[13] Amazon EC2 Instance Types, https://aws.amazon.com/ec2/instance-types/ (2016).

[14] HTTP Archive, Compare stats, httparchive.org/compare.php (2016).

[15] DDoS Deflate, https://github.com/jgmdev/ddos-deflate (2016).

[16] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, S. Savage, Inferring Internet Denial-of-Service Activity, ACM Transactions on Computer Systems (TOCS) 24 (2) (2006) 115–139.

[17] S. Mohan, F. M. Alam, J. W. Fowler, M. Gopalakrishnan, A. Printezis, Capacity Planning and Allocation for Web-Based Applications, Decision Sciences 45 (3) (2014) 535–567.

[18] J. Petazzoni, Gathering LXC and Docker Containers Metrics, https://blog.docker.com/2013/10/gathering-lxc-docker-containers-metrics/ (2013).

[19] P. Muncaster, DDoS-ers Take Down Mitigation Tools in Q1, http://www.infosecurity-magazine.com/news/ddos-ers-take-down-mitigation/ (2016).

[20] H. Wang, Q. Jia, D. Fleck, W. Powell, F. Li, A. Stavrou, A Moving Target DDoS Defense Mechanism, Computer Communications 46 (2014) 10–21.

[21] J. Idziorek, M. F. Tannian, D. Jacobson, The Insecurity of Cloud Utility Models, IT Professional 15 (2) (2013) 22–27.

[22] R. Shea, J. Liu, Performance of Virtual Machines Under Networked Denial of Service Attacks: Experiments and Analysis, Systems Journal, IEEE 7 (2) (2013) 335–345.

[23] S. Zhao, K. Chen, W. Zheng, Defend Against Denial of Service Attack with VMM, in: Grid and Cooperative Computing, 2009. GCC'09. Eighth International Conference on, IEEE, 2009, pp. 91–96.

[24] A. Sarra, G. Rose, DDoS Attacks in Service Clouds, in: 48th Hawaii International Conference on System Sciences, IEEE Computer Society, 2015.

[25] G. Yossi, H. Amir, S. Michael, G. Michael, CDN-on-Demand: An Affordable DDoS Defense via Untrusted Clouds, in: Network and Distributed System Security Symposium (NDSS), 2016.

Gaurav Somani is an Assistant Professor at Department of Computer Science and Engineering, Central University of Rajasthan, India. He completed B.E. in Information Technology from University of Rajasthan and M.Tech. in Information and Communication Technology from DAIICT, Gandhinagar. He is pursuing his PhD from MNIT, Jaipur, India. His research interests include Distributed Systems and Security Engineering.

Manoj Singh Gaur is a Professor in the Department of Computer Science and Engineering at Malaviya National Institute of Technology Jaipur, India. He has obtained his Ph.D. from University and Southampton, UK. He has supervised research in the areas of Networks on Chip and Information Security. He has published over 150 papers in peer-reviewed reputed conferences and journals.

Dheeraj Sanghi is a Professor of Computer Science and Engineering at IIT Kanpur. He has a B. Tech from IIT Kanpur, and MS and Ph.D. from University of Maryland, USA. From 2008 2010, he served as the Director, LNM Institute of Information Technology (LNMIIT), Jaipur. His research interests include network performance optimiza-

tion, security and distributed systems.

Mauro Conti is an Associate Professor at the University of Padua, Italy. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. He was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. His main research interest is in security and privacy. He published more than 150 papers in topmost international peer-reviewed journals and conferences.

Muttukrishnan Rajarajan received his BEng and PhD degrees from City University London in 1994 and 1999 respectively. After a few years in the industry Raj is now a Professor of Security Engineering. He also sits on the Editorial boards of Springer/ACM Journal on Wireless Networks, Elsevier Journal of Health Policy & Technology and Emerald Journal of Information Management & Computer Security.