Network Security of Internet Services: Eliminate DDoS Reflection Amplification Attacks

Todd Booth[1]* and Karl Andersson[2]
Division of Computer Science, Luleå University of Technology, Sweden
[1]Information Systems
[2]Pervasive and Mobile Computing Laboratory
{Todd.Booth, Karl.Andersson}@ltu.se

Abstract

Our research problem is that there are a large number of successful network reflection DDoS attacks. Via a UDP Reflection Attack, an attacker can send just 1 Gb/s of payload to innocent servers, and it is these servers which then can send over 4,600 times the payload to the victim! There are very expensive and complex solutions in use today, however most all of these on premise solutions can be easily circumvented. The academic community has not adequately addressed this research problem. We have created a new Internet services network security surface attack mitigation methodology. Our novel design patterns will help organizations improve the price/performance of their anti-network reflection solution by 100 times, as compared to common on premise solutions. Our analysis and results confirm that our solution is viable. Our novel solution is based on stateless IP packet header filtering firewalls (which can be implemented mostly in hardware due to their simplicity). We have reduced and in some cases eliminated the need for researchers to even try and find new ways to filter the same traffic via more complex, software driven stateful solutions.

Keywords: Internet Services, Information Systems, Network Security, Firewall, Cloud, Distributed Denial of Service

1 Introduction

The focus of our research is on DDoS bandwidth Reflection Attacks, which use the UDP protocol. We will include material on how to defend both UDP and TCP Internet services, from these attacks. In order to frame our research, we will use set theory and will refer to the following different attack sets, which are each a proper subset of network attacks:

1. DoS attacks
2. DDoS attacks
3. Bandwidth attacks
4. Reflection attacks
5. UDP attacks
6. TCP attacks

We will exclude the term "attacks" when referring to the above sets. The focus of our research is on the following, which we will simply refer to as "Reflection Attacks".

Journal of Internet Services and Information Security (JISIS), volume: 5, number: 3 (August 2015), pp. 58-79
*Corresponding author: Luleå University of Technology, Todd Booth, Skeria 3, 93187 Skellefteå, Sweden, Tel: +46-72-519-7773

$$x \mid x \in \{DDoS \cap Bandwidth \cap Reflection \cap UDP\}$$

We present an architecture model, which includes our proposed security design patterns. In this article, we provide a way to mitigate several types of Reflection Attacks. The acronyms, terms and definitions used in this article are found below in table 1.

Term	Definition
AS	IP Autonomous System
BotNet	Robot Network of compromised hosts
CI	Critical Infrastructure
CIP	Critical Infrastructure Protection
DDoS	Distributed Denial of Service (attack)
DoS	Denial of Service (attack)
ICT	Information and Communications Technology
IP	Internet Protocol (in this article IPv4)
IS	Information Systems
Mb/s	Megabits per second
NAT	Network Address (or port) Translation
NG	Next Generation (Firewall)
NTP	Network Time Protocol
OS	Operating System
SLA	Service Level Agreement
TCP	IP Transmission Control Protocol
UDP	IP User Datagram Protocol
US-CERT	USA Computer Emergency Readiness Team
VM	Virtual Machine
Zombie	Infected computers in a BotNet

Table 1: Acronym and term definition table

In the industry, the term NG firewall is used to refer to a next generation “firewall” which has many more security features than a traditional firewall. In this article, we use the term “firewall” loosely, to refer to a traditional firewall or “NG firewall”, as needed.

1.1 Motivation

A summary of our motivation is found in the abstract. Reflection Attacks against Internet services are now very common. We are focusing on protecting vulnerable Information System components such as public facing servers running either TCP or UDP Internet services. This includes DNS, Web, database, NTP, and email Internet services. The cost to launch these Reflection Attacks is extremely low and for at least thousands of important servers, it is quite easy to launch a successful attack. These Reflection Attacks cause the services to be unavailable to legitimate users. Based on current industry solutions, the cost to the victim to protect against these Reflection Attacks, is often quite high and requires strong expertise to install, maintain and manage. The cost to the victim, concerning a successful Reflection Attack, can often be significantly high per hour. Sometimes these Reflection Attacks are successful for hundreds of hours. If the attacked site is part of the Critical Infrastructure (CI), this can have a life threatening effect

on people in society! Our focus in this article, includes defending very important Internet services, which are part of the CI. Therefore, our article discusses mitigation strategies for the more severe Reflection Attacks. It has been reported that the cost per year, to defend 5Mb/s of valid Internet bandwidth is over 100,000 USD/year [8]. However, one actual Reflection Attack has sent 400 Gb/s to the victim server, [10] so it may often cost companies much more than 100,000 USD/year to properly defend against 400 Gb/s attacks. In the near future, we expect attacks of more than a magnitude greater (4,000 Gb/s or 4 Tb/s). There are 7,000 DDoS attacks observed daily [18]. Most of the prior work has come up with lots of technical algorithmic solutions, which can be run at the organization's premises. However, as we will show, any organization premise side solution is simply inadequate against a high bandwidth Reflection Attack. Also there are lots of designs, based on stateful packet inspection. However, we have not yet found any design, which can stop most of the Reflection Attacks, via simply packet filter firewalls, which are the most efficient type of firewall and can be implemented in hardware.

Our research solution is concerning how to implement the network security protection in the cloud, via packet filter firewall techniques, which will eliminate these Reflection Attacks before they reach the organizations' Internet services.

1.2 Specification

Our specific desired research outcome is to find a cloud based security solution, which will allow all organizations to greatly mitigate any and all Reflection Attacks easily and at a very low cost. The first generation of firewalls is based on stateless packet filters. The next generation of firewalls is based on stateful filtering. However stateless packet filters are often simpler to implement, requires no state to be remembered, can be more easily implemented in hardware, so it is normally extremely fast and scales well. Stateful filtering is often the opposite. However, it is possible to have stateful filtering performed, with just a small amount of state, which will allow that stateful firewall to operate extremely fast. Our research is concerning the possibility to use stateless packet filters to eliminate most or all of the Reflection Attacks.

Our approach is to consider moving the Information Systems from the on premise organization location to the Microsoft Azure cloud. To run our experiments on all major clouds is outside the scope of this article. Even though our experiments were performed with the Microsoft Azure cloud, most other cloud providers should have or in the future will have similar Internet service security features. So we will try to find a novel way to perform Internet services security, via Cloud based firewalls. There exist many Cloud firewalls which can be purchased and operated by organizations. However, we will try to perform all security protection, via the Azure firewalls, which are included at no additional cost when implementing a VM server.

The specification for our research constraints is the following:

1. The organization is running very important Internet services, where the monetary or other costs of downtime is very high.
2. The organization is running IPv4 UDP and TCP Internet services on their servers, which are using public IP addresses. We consider these as public facing servers.
3. The organization wishes to protect their UDP and TCP Internet services, from Reflection Attacks, which are up to 400 Gb/s.
4. Our research is focused more on generalized attacks (by any and all of the higher layer 4 UDP protocol attacks), as compared to focusing on the layer 7 application protocols.
5. Our focus is to help organizations, but we are not focused on helping ISPs or cloud providers.
6. The Information System servers are initially located at the organization's own premises.

7. Our focus is to help organizations who have a 10 Gb/s or less bandwidth SLA, with their ISP.
8. The Organization's Internet services may be intended for all Internet users or may be intended for only a small subset of the Internet users (for example, only for their own organization employees).
9. The attacks are intended to perform a denial of service and the attacker's strategy is to send a large number of packets and a high volume of bandwidth. This implies that the actual attacker is able to spoof their source IP address, to be that of the reflector server.
10. The reflector is an innocent server, meaning that the operator of the reflector server is not intentionally performing malicious activities. The relevance is that this research does not cover the case where the attackers are using their own reflector servers.
11. We do not perform research on the traceback of spoofed IP address defenses. Our research is limited to simply filtering the attack on stateless packet filters.
12. Our focus on mitigation, is concerning what the organizations can do today, to protect their Internet services.

There are various ways to deal with network security risks, such as acceptance, transference, mitigation and denial. Our research is focused on transference, but there is a small amount of remaining vulnerability. So our research can also be considered as how to greatly mitigate risk, at an extremely low cost. A specification of what we will analyze, in order to try and find a novel solution, includes the following:

1. Behavior of normal UDP traffic
2. Behavior of a UDP Reflection Attack
3. Behavior of a few layer 7 protocols, based on UDP
4. Possibility to use packet filters in novel ways, in order to reduce the reflection traffic
5. Azure cloud ACL security features
6. Azure cloud Network security group features

1.3 Outline

The rest of this article is organized as follows: Section 2 introduces the network reflection problem domain, including current vulnerabilities, attacks and defenses; Section 3 describes our proposed design patterns (and related works), which greatly mitigate the vulnerabilities and then, our conclusion is found in Section 4.

2 Background

In this section, we provide the relevant background of network Reflection Attacks and defenses, which is required to understand our contributions. This includes the current status. With a DDoS Reflection Attack, an attempt is made to use up so much of the network bandwidth, between the reflector servers and the organization, that the normal and valid network traffic is unable to properly travel to the organization's server.

Direct network attack: A direct attack, without reflection, is where the attacker sends IP packets directly to the victim server.

Indirect reflection network attack: On the other hand, a Reflection Attack, is an indirect attack, in which the attack sends request packets to the reflector server and that reflector then sends response packets to the victim.

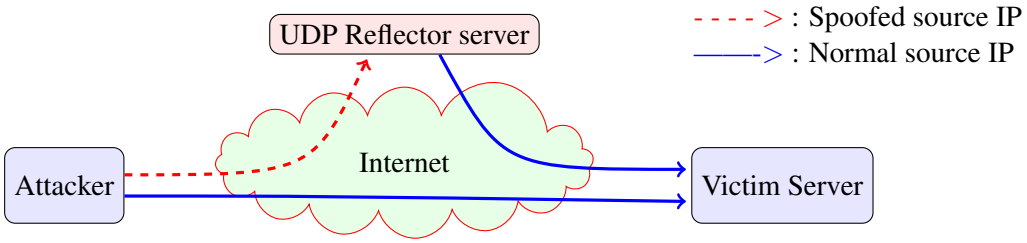

Figure 1: Direct and Reflection Attack

Direct network attacks and indirection Reflection Attacks are shown in figure 1. In the direct attack, for every request packet sent, the victim server receives one request packet. For every 1 Mb/s which the attacker sends, the victim server receives the same amount of traffic.

In the Reflection Attack, for every request packet sent, the reflector server receives one request packet. For every 1 Mb/s of payload sent by the attacker, the reflector server receives 1 Mb/sec. The reflector server then amplifies the traffic it receives, when sending the replies to the victim. For example, the attacker sends a small query and the reflect server replies (to the victim), with a very large response. For every request packet the reflector server receives, it may send more than one response packet to the victim. For every 1 Mb/s of payload which the reflector server receives, the reflector server sends more than 1 Mb/s of payload to the victim. So there is an amplification in the number of response packets and/or the bandwidth. Not all reflector servers amplify the traffic, with the same outgoing bandwidth. We will present some amplification numbers, which are only concerning the reflection servers, which performed the highest level of amplification (top 10%), during some network scans. The average of these top 10% amplification reflection servers, will be shown. Note that the amplification numbers presented, are only concerning the UDP payload amplification, and do not include the IP and UDP headers. The top five UDP amplification protocols, are shown in table 2 [11].

Protocol	**Amplification**	**Service Listening Port**
NTP	4,670	123
CharGen	358	19
QOTD	140	17
DNS	98	53
Quake 3	82	27950, 27952, 27960, and 27965

Table 2: UDP protocols with large DoS amplification effects

As shown, the payload amplification for the NTP UDP protocol, can be over 4,600 times! Public facing reflector servers, which are vulnerable to the Reflection Attacks are running UDP Internet services. For example, the reflector server may be running NTP and/or DNS[1].

2.1 UDP Reflection Attack

Without reflection, a valid and normal UDP client request and server response is shown in figure 2. However in a UDP Reflection Attack, the malicious UDP client request, reflection amplification and

[1] We are of course aware DNS uses both UDP and TCP but we are simplifying things, for just the explanation

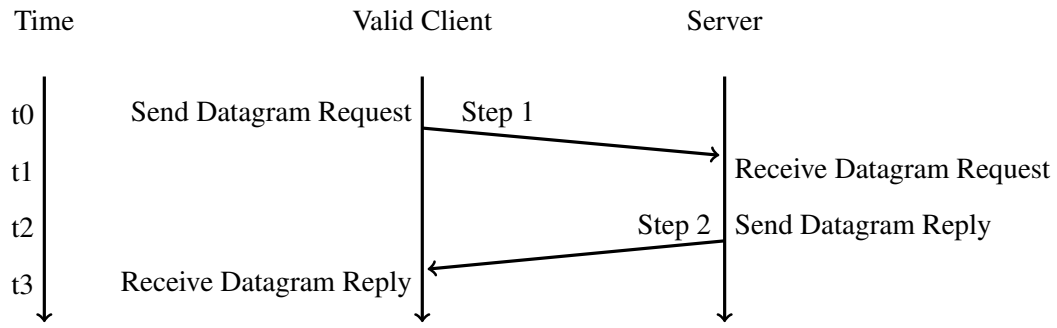

Figure 2: Valid UDP request and reply

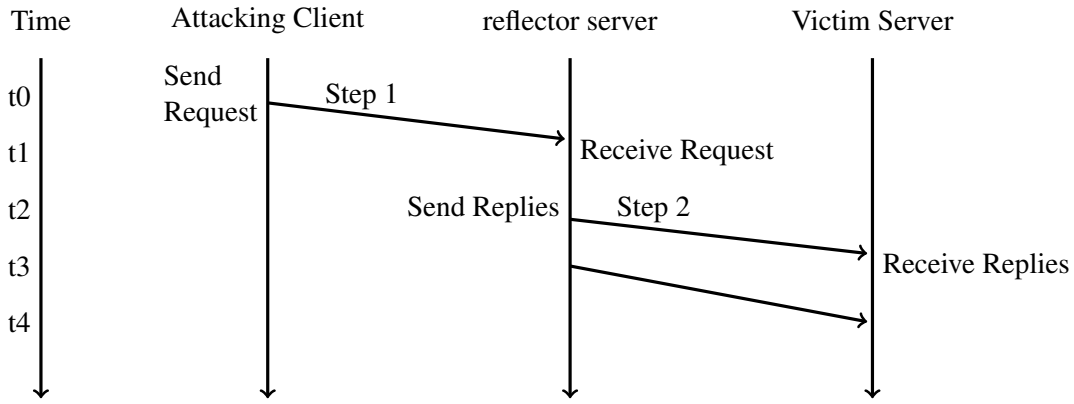

Figure 3: UDP Reflection Attack

server receives are shown in figure 3. In step 2, the reflector server may send multiple requests. If multiple requests are sent, the server will receive multiple requests.

A specific example, showing the IP UDP datagrams of a UDP Reflection Attack is now presented. It will be much easier to follow along, if you first review the corresponding figure 4. In this UDP Reflection Attack, the attacker spoofs their source IP address (1.1.1.1), to be that of the victim (3.3.3.3). The attack also changes their source port to be whatever they want to be the final destination port to be (123). In this example, they are sending a UDP datagram request to the reflector server's NTP service. The attacker sends this request to the reflector server by setting the destination IP address to that of the reflector server

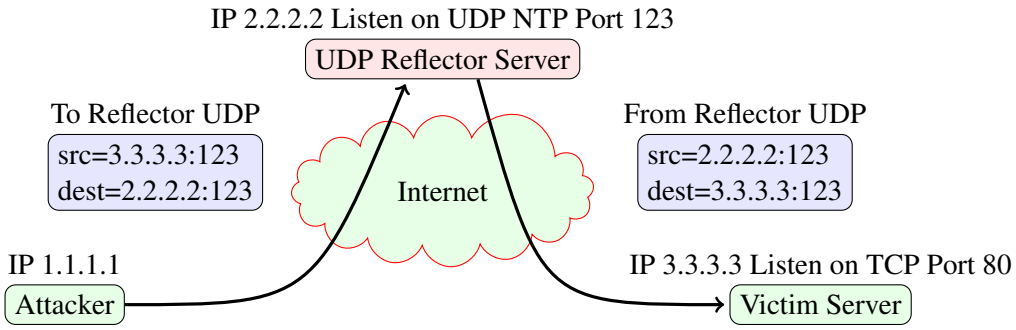

Figure 4: Packet transitions during UDP Reflection Attack

(2.2.2.2). The attacker also sets the destination port, to the reflector server's NTP service (123). The reflector server then sends multiple replies, but which are sent to the victim (since the source IP address was spoofed to 3.3.3.3). The victim server is only running a web server, their IP address is 3.3.3.3 and they are not even running the UDP NTP server. However the victim server will still receive this UDP NTP Reflection Attack traffic. Even if the server is not running the NTP service, they still must process these UDP datagrams.

Via security terminology we can say that the problem is that the reflector server is not authenticating the UDP client. The UDP client is spoofing their source IP address and the reflector server is unable to know that the source IP address is spoofed. This is a problem with the reflector servers, but it is also a very broad and general problem, with most of the UDP protocols.

2.2 Complex UDP Reflection Bandwidth Attack

In an indirect UDP Reflection Attack, there can be a payload amplification factor of 4,670 times (with the NTP). What happens is that the attacker can send a smaller stream of UDP NTP requests, for example 1 Gb/s towards one or more reflector servers. The reflector servers will reply to the spoofed IP source address (the victim), with up to over 4,670 times as much payload traffic. In summary, if the attacker sends a 1 Gb/s stream payload of spoofed requests to various reflector servers, they can send up to 4,670 Gb/s worth of payload traffic to the victim server! Therefore, whenever possible, attackers prefer to use Reflection Attacks, with high amplification factors, whenever they are performing a Reflection Attack. Note that for a Reflection Attack to work, the attacking PC needs to be able to spoof (change) their source IP address, to that of the victim. It was reported that researchers were able to "reveal up to 2,692 Autonomous Systems that lack egress filtering" [10]. We are unable to find any Internet policy, which is planning to require ISP's to prevent their customers from spoofing their source IP address.

A limitation of a single reflector, is that the reflector's outgoing bandwidth may be limited to, for example, 1 Gb/sec. Let's suppose the attacker wishes to generate 400 Gb/sec, of attack traffic. They can send a 1 Gb/s of payload traffic to a reflector server, which will then try to send the 400 Gb/s of payload attack traffic to the victim. Let's assume that the reflector server only has a 1 Gb/s bandwidth SLA with their ISP. The reflector will exceed their bandwidth allowance with their ISP. To get around this limitation, the attacker might wish to send just 1 Mb/s to this reflector server, and 1 Mb/s to 999 other reflector servers. All 1,000 reflectors will then send traffic to the reflector, which does not exceed their ISP SLA. This would still generate the huge amount of traffic to the victim, but it would no longer exceed the reflector servers' 1 Gb/s upload bandwidth limit. It has been reported that via scanning, "the time it took to identify 1,000" NTP amplifiers took very little time [10]. However, if you don't want to scan from scratch yourself, you can start with a current list of NTP servers, which is available at Scans.io [2]. For our testing, we used the Scans file 20150608-ntpmonlist-123.csv.gz We wrote an awk script to extract the NTP server addresses, which were found in the 2nd column. The NTP monlist command is used to generate a high amount of bandwidth amplification. So our script output was a series of NTP client monlist command scans. We created the NTP client monlist requests with the NMAP tool. An example scan we used follows (with and without the NMAP spoofing feature):

```
nmap -sU -pU:123 -Pn -n --script=ntp-monlist (followed by IP addresses)
nmap -sU -pU:123 -Pn -n --script=ntp-monlist -S 10.0.3.4 (IP addresses)
```

[2]Scans: `https://scans.io/study/sonar.udp`

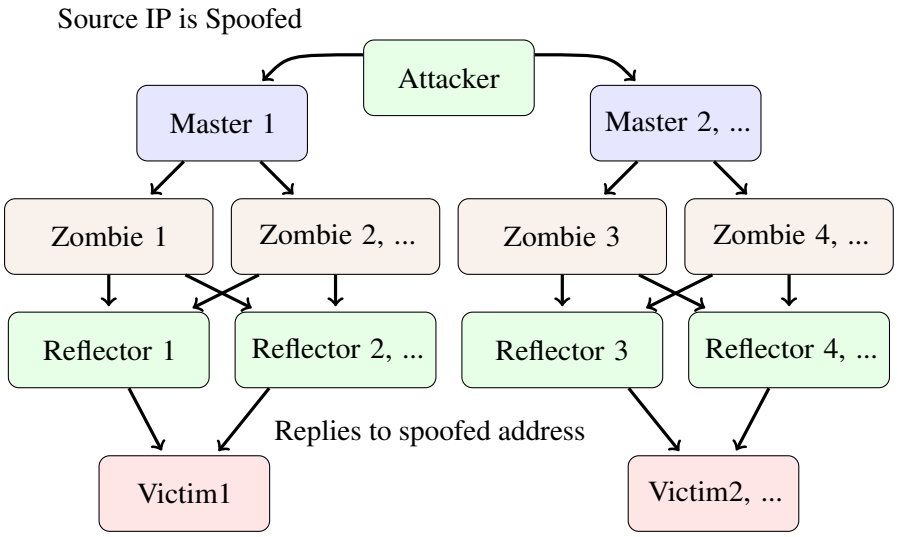

Figure 5: Detailed and complex Reflection Attack

As shown above, NMAP has a feature to perform source address spoofing. For testing we also performed our own spoofing via the Linux iptables/netfilter facility. Here is an example of our iptables spoofing rule:

```
iptables -t nat -I POSTROUTING -d 10.0.2.3 -j SNAT --to-source 10.0.3.4
```

One can then grep the output file of the "monlist" server replies, in order to find the vulnerable NTP reflection servers. The attackers also wish to minimize the chance that their attacking PC is detected. So instead of having one attacker send 1 Gb/s of payload data, they may wish to have 100 attacking PC's send just 10 Mb/s each, for an aggregate of 1 Gb/sec. In this case instead of just a DoS, it's called a distributed DoS (DDoS). There is often a master controller, which would control these 100 attacking PCs (called zombies). The zombies may be infected, via malware, to join the BotNet. A newer trend is for users to voluntarily opt in to become an attacker. They voluntarily join a BotNet (if they agree what the stated cause of the attack, is for). The master can also decide if the attack should be against just one victim server or against perhaps 100 victim servers, at the same time. So a more realistic and detailed diagram of Reflection Attacks is shown in figure 5.

2.3 Defense Against Reflector Attack at the on-site Premises

If the organization only tries to defend against Reflection Attacks, via on-site organization premises equipment, it is close to impossible to defend against. Even if the organization does prevent the Reflection Attack, the extremely high defense costs (ISP bandwidth, on premise equipment and expertise) would in effect, be a successful Reflection Attack (causing monetary damage instead). So in summary, Reflection Attacks against on premises servers can be close to 100% successful.

A simplified example will now be presented. In order to more easily follow this section, please first review our simplified scenario in figure 6 and then follow along. Suppose a organization has purchased 10 Gb/s of bandwidth which is between the ISP and the organization's premises. Assume that the attacker's reflectors send 20 Gb/s of traffic to the organization's server. The ISP will then have 20 Gb/s of traffic to send to the organization, but only 10 Gb/s was paid for. So the ISP will simply send 10 Gb/s of traffic and randomly drop the excess 10 Gb/s of traffic, which is destined for the organization's server. Some of that

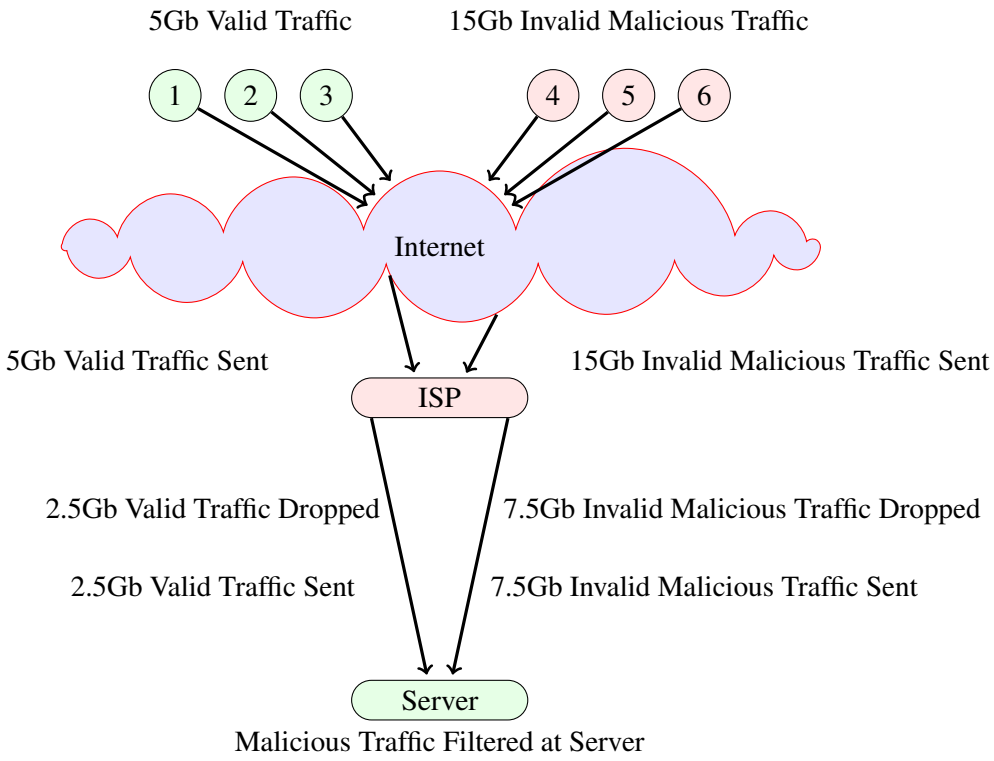

Figure 6: Network bandwidth attack

10 Gb/s traffic dropped will be valid organization traffic and some would be invalid malicious traffic. In this scenario, the excess traffic is dropped at the ISP, before reaching the organization. So by the time the 10 Gb/s traffic arrives at the organization's site, it is too late to disregard the malicious traffic. Therefore, any technical solution at the organization's premises will be useless, concerning the ISP dropped valid traffic.

Due to the Reflection Attack exceeding the ISP to organization's bandwidth limit, to try and defend against this attack, an organization could increase their ISP bandwidth. Let's assume that the organization increases their download speed from 10 Gb/s to 20 Gb/sec. However, the attacker could just increase the attack bandwidth to be greater than the new higher ISP bandwidth (for example to 40 Gb/sec). These bandwidth attacks can consume several hundred Gb/s of bandwidth. So the organization may need to increase their ISP speed to be several hundred Gb/sec. Many ISP's do not yet offer organizations the ability to have hundreds of Gb/s of bandwidth. Therefore, in general, the organization would not be able to provision enough ISP bandwidth to defend against an high bandwidth Reflection Attack. Even if they could provision this high bandwidth, the ISP bandwidth cost would be much higher than needed for their normal operations traffic. Also the organization would need a large amount of expensive on premise defense equipment to handle the hundreds of Gb/s invalid traffic streams. Last, this expensive solution would only be needed during an attack, which can be just a few hours per year. So this solution is very inefficient. In summary, an on-premise defense solution is not considered adequate, against a bandwidth attack.

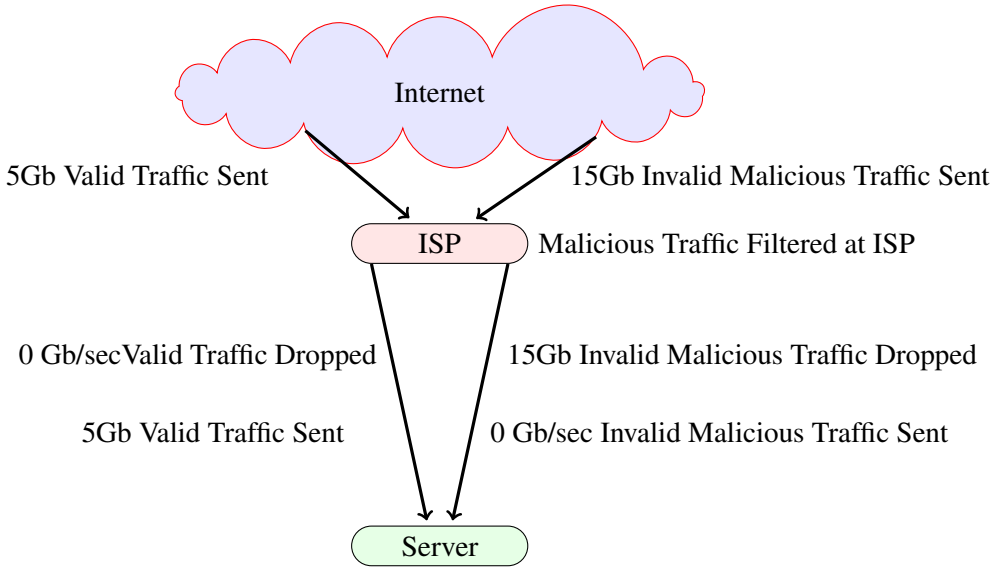

Figure 7: Malicious traffic removal at the ISP

2.4 Defense Against Bandwidth Attack at the ISP

An alternate defense is to have the organization implement a security solution at their ISP. With any ISP, an attempt is made to remove the invalid and malicious traffic, prior to it being sent from the ISP to the organization. Assuming that we can remove 100% of the malicious traffic, at the ISP, the resulting traffic characteristics are shown in figure 7.

This ISP located equipment solution somewhat overcomes some issues. For example, it should be quicker and perhaps can even be done automatically, to reconfigure the bandwidth from the ISP to the organization's on ISP premises defense equipment. However, if several hundred Gb/s of traffic is sent to the ISP, this may be more bandwidth than the ISP is provisioned to receive from other ISPs. If so, we have the same type of problem as we had when locating the defense equipment at the organization's site. Even if hundreds of Gpbs can be sent to the ISP and forwarded to the organization's ISP premises defense equipment, we also have the same type of problems as we had with the on-site organization premises (expensive equipment, engineers, etc.). This is also an extremely inefficient solution, since the organization covers all the costs of the attack year round, even if there are no attacks during the entire year. So, like the on organization premises solution, this ISP defense is still very costly and inefficient.

2.5 Current Example of Reflection Attack

In 2014, there was a UDP Reflection Attack, of over 400 Gb/s [10]. This amount of traffic will not only cause a denial of service at an organization's on premise location, but it may cause a denial of service to many ISPs.

2.6 Current Example of Alternative Prices

The low end Azure VM guest server costs about 11 USD / month [3]. This includes unlimited incoming traffic and includes the Azure stateful and stateless firewalls. An example of an alternative defense solution, is the CloudFlare enterprise service which costs an average of 5,000 USD / month. [4]. However with CloudFlare, they don't include a VM guest web server. Let's assume that it costs you 10 USD / month, to run your own server (when using CloudFlare). In this case, the Azure only costs you an extra 1 USD / month, for the included security features. So the Azure solution costs you only 1 USD / month compared to the CloudFlare 5,000 USD / month solution. So in this case, the CloudFlare solution is about 5,000 times as expensive, as compared to our proposed solution. Note that both CloudFlare and Azure include security protection, other than from Reflection Attacks.

3 Architecture Model to Mitigate Reflection Attacks and Related Works

This section includes our architecture model, which describes different general defense approaches and their effects. Our model includes our newly created and proposed Reflection Attack defense design patterns. We of course used our model and design patterns to propose how to mitigate the Reflection Attack effects. However, our model and design patterns are also applicable, to network attacks, which are outside the scope of our defined Reflection Attacks.

We first analyzed the initial scenario, where the organization was trying to defend against Reflection Attacks, and their IS equipment was located on premises. We also performed a detailed analysis of the protocols and the Azure cloud features. We then found came up with our architecture model the following design patterns. We included the related works, next to our design pattern contributions. The most closely related work is our previous work [3]. However, in this article we continue where we had left off, with our past research.

3.1 Research Challenges

In the background section, we have presented the current state of Reflection Attacks, including some serious vulnerabilities. One of the research challenges we faced was how to collect vulnerability data, concerning the current status of Internet services. In most countries it is illegal to perform vulnerability scanning of Internet services, unless of course you have permission. One reason that scanning is illegal, is that the scanning of a server may result in the server crashing. To get around these legal issues, we mainly relied on other papers which presented detailed information, concerning their collected scanning data, such as in [10, 11, 13]. We did however, perform some limited actual scanning and setup our own test lab so that we could run into and solve all the same problems that attackers would likely run into. We used Docker[5] containers to create the attackers, the reflectors, the routers and the victims. We then used an Azure VM guest and local VM guest as our Docker hosting servers. A simplified diagram of our test environment is found in figure 8.

We solved a few issues, which we will describe in order to help others reproduce our lab environment. The current Ubuntu version didn't have the vulnerable UDP NTP server monlist feature. So

[3] Azure: `http://azure.microsoft.com/en-us/pricing/details/virtual-machines/`

[4] CloudFlare: `https://support.cloudflare.com/hc/en-us/articles/200170326-How-much-does-the-Enterprise-Plan-cost-which`

[5] Docker: `http://Docker.io`

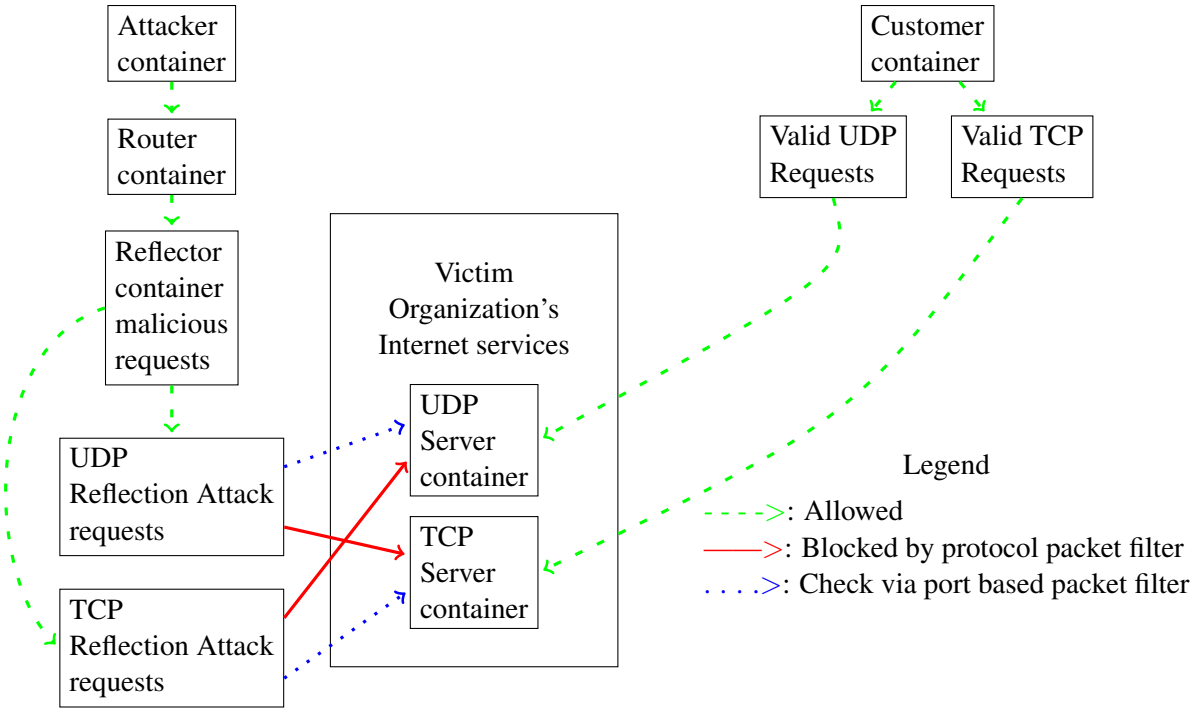

Figure 8: Simplified figure of our Azure and Docker container based DDoS attack lab.

for the Docker reflector container, we based it on Ubuntu version 11.04. The current Ubuntu version would not forward spoofed traffic. Even after we configured it to allow martian (spoofed) traffic, via net.ipv4.conf.default.rp_filter=0, spoofed traffic was not forwarded. So for the Docker router container, we based it on Ubuntu version 10.04.

It is not illegal to scan your own servers. So we of course considered scanning our own servers, which are located in the Microsoft Azure cloud. Microsoft does allow certain penetration testing against your own Azure resources [6]. Note that you must first obtain approval. However, Microsoft does not allow any DDoS testing, even against your own VM servers (not even with a very low bandwidth attack rate). So in our Azure application, we only requested the ability to perform validation of the Azure firewall and our own vm guest firewall, which of course overlaps somewhat with performing DDoS testing. Our application was approved by Microsoft.

An issue is that Microsoft requires you to state which specific source IP address will be used for the vulnerability testing. By default, Microsoft will only assign temporary public IP addresses to your VM guests. So if you need to do any vulnerability scanning, from your own Azure VM scanning client to your own Azure VM server, you will need to use a static public IP source address. To get a static public address for your scanning client, you will need to reserve a public static IP address. Here is an example of the Azure PowerShell command we used to reserve a static public IP address:

```
New-AzureReservedIP -ReservedIPName Scanning_Client_12 -Location "East US"
```

[6] Azure: `https://security-forms.azure.com/penetration-testing/terms`

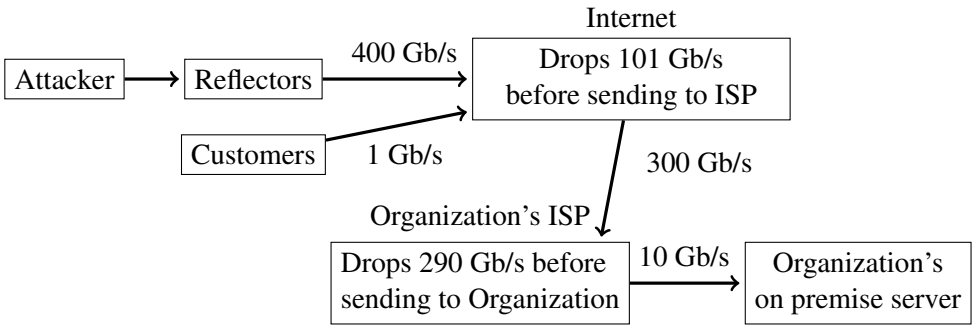

Figure 9: On premise bandwidth bottleneck from the ISP

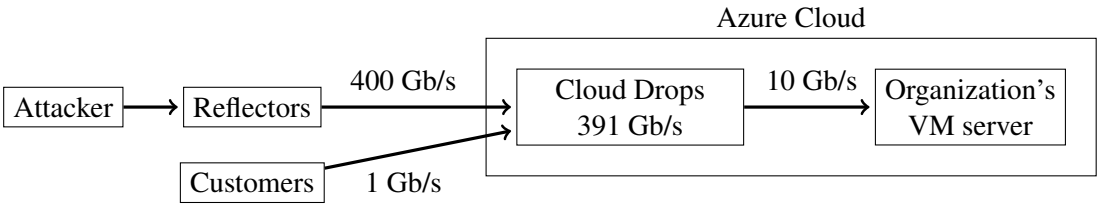

Figure 10: Moving the on premise bandwidth bottle to the cloud

3.2 Create Bandwidth Defense in Cloud

As we stated, a research challenge is how to mitigate the Reflection Attack, when an organization's Internet services, are located on premises. The specific issue is that the amount of bandwidth can easily saturate the organization's ISP link. We will perform a traffic analysis walk through. Let's assume we now wish to defend against a 400 Gb/s Reflection Attack. Most of the prior research and proposed solutions have been concerning ones which can be implemented at the organization's site. However, as we previously stated, it is not possible to efficiently defend against such an attack, with only on organization premises equipment.

Let's assume that the customer has 10 Gb/s bandwidth from the ISP, the ISP only has 300 Gb/s bandwidth from the Internet, there is a 400 Gb/s Reflection Attack, and that the organization only has 1 Gb/s of valid traffic, from their customers. To simplify the discussion, let's assume that the ISP has no other customers. The issues are that traffic is dropped before reaching the ISP and additional traffic is dropped before reaching the organization, as shown in figure 9. The excess dropped traffic will randomly include both malicious traffic and valid customer traffic.

However, cloud solutions, such as Microsoft Azure cloud do have plenty of incoming bandwidth and allow customers to use their own virtual defense equipment, within the Azure cloud. We would like to try and use cloud security solutions to overcome these issues. So our first step is to move the organization's Internet services to the cloud. This overcomes the dropped traffic incoming to the ISP issue. The resulting traffic pattern is shown in figure 10.

So by using the cloud, we are no longer limited to the incoming bandwidth at the ISP. So we have solved one issue. However, even if the cloud provider has 400 Gb/s of incoming bandwidth to their premises, your virtual machine will have a limited amount of incoming bandwidth. In our previously discussed figure 9, the same amount of 391 Gb/s traffic is randomly being dropped before reaching our VM server. The good news is we can easily increase our bandwidth. The previous figure shows the default network interface which only allows 10 Gb/s. However, in the cloud, we can have up to 16 network interfaces on

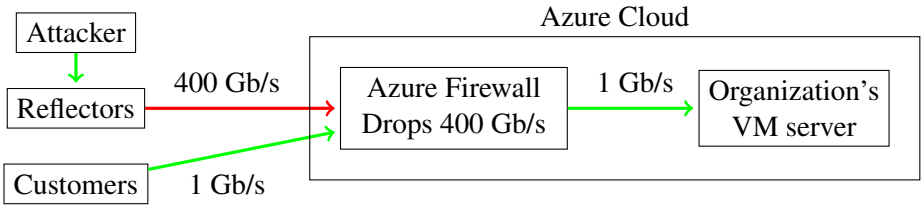

Figure 11: Moving the on premise bandwidth bottle to the cloud

our VM server, for a total of 160 Gb/s. Or we can spin up 50 VM servers, with just one network interface each, for a total of 500 Gb/s. Note that with the Microsoft Azure cloud, there is no charge for incoming traffic [7]! So even if you have a 400 Gb/s Reflection Attack for weeks, you don't need to pay for any of the incoming traffic which the Azure firewalls filter.

Guenane, et. al. [8] and Salah, et. al. [14] propose a cloud based firewall, where the organization is managing their own cloud based VM guest firewalls. As attacks increase, they propose more virtual servers are provisioned. All of our architecture design patterns will be formatted, as shown in the following boxed format. Here is our first design pattern.

Design Pattern 1: To defend against high bandwidth Reflection Attacks, a defense located in the cloud (such as the Microsoft Azure cloud) can overcome some incoming bandwidth issues, which is not possible with ISP or organization located defenses [3].

3.3 Use Cloud Based Stateless and Stateful Firewalls

We could improve our solution if we could find a more efficient way to filter out malicious traffic (as opposed to spinning up lots of VM servers). So we need to find a way to reduce the amount of reflection traffic which reaches the organization's vm servers. It will now be assumed that the Microsoft Azure cloud, or a similar cloud service is being used. We will use the Azure cloud terminology since that is the one we tested.

A research challenge is that it is very expensive for companies to install, maintain and manage their own cloud based firewalls. So we propose that the organization use Microsoft's Azure cloud firewalls instead of processing the malicious traffic on their own VM servers. Our solution will dramatically reduce the costs. The Azure cloud has both a stateless packet filter firewall and a stateful firewall, which are both included for no extra charge. If these firewalls are filtering, for example, 400 Gb/s of traffic, this traffic is never seen or processed by the organization's virtual equipment. Therefore, by using the cloud's firewall, there is no longer a need to spin up more virtual servers and this cost is eliminated. Our proposal is shown in figure 11.

The Azure cloud is now receiving 401 Gb/s, filtering all of the malicious 400 Gb/s of traffic, and just delivering the valid 1 Gb/s traffic to the organization's vm server. Previously, in figure 10 the 301 Gbps was dropped randomly, which means that some of the organization's valid customer traffic was being dropped. By using the Azure firewalls, if the firewalls are configured correctly (as we will soon show how to do), only the reflection malicious traffic is dropped. In our proposed figure 11, we see that the Organization's VM server no longer receives any malicious reflection traffic.

[7] Azure: http://azure.microsoft.com/en-us/pricing/details/data-transfers/

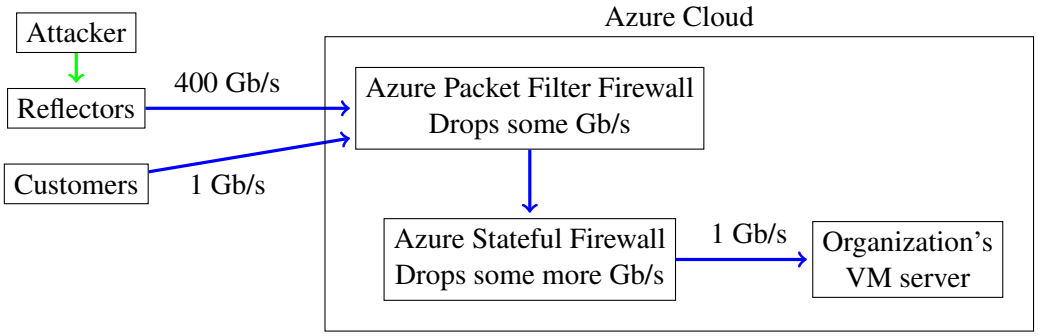

Figure 12: Filter malicious traffic with packet filtering rules before using stateful filtering rules.

Design Pattern 2: Whenever possible, instead of using your own Azure VM guest firewall, use the Azure cloud stateless and stateful firewalls, which are both included for no extra charge.

3.4 Use Packet Filter Firewalls, as 1st Defense

The organization does not really care if the Azure cloud eliminates the malicious reflection traffic via a stateless or stateful firewall. However, a properly designed architecture will have a better chance of being successfully implemented in the Azure cloud. Some people incorrectly believe that 2nd generation stateful firewalls have made the 1st generation packet filter firewalls obsolete. However, this is not the case. Stateful firewalls are often at least somewhat more complex and are at often implemented mostly in software. The stateless packet filter firewalls are extremely simple and just look at the IP fields, in order to make their drop or allow decisions. So stateless packet filter firewalls can be more easily implemented in hardware, which makes them incredibly fast. Our architecture includes the design pattern which proposes that a stateless packet filter firewall is the first line of defense and that the stateful firewall is only used as a second line of defense. From a practical point of view, both the stateless and stateful firewalls are often implemented on the same device. Our more theoretical design pattern is shown in figure 12.

Related works by Ye, et. al. [17], and Dou, et. al. [6], like many other papers, propose stateful algorithms to mitigate DDoS network attacks. With regard to reflection DDoS attacks, we reject that suggestion and we propose to use stateless packet filters as the very first line of defense, which are much simpler, require far less lines of code, can execute more quickly and are easier to implement in hardware. The default Azure stateful firewall rules start at priority number 65000, so you just need to set your packet filter rule numbers to be something lower, for example 200-299. The following design pattern is a more practical pattern, for use with the Azure cloud.

Design Pattern 3: Whenever possible, use the Azure cloud stateless packet filter firewall first. Then only use the Azure cloud stateful firewall for the remaining traffic (which you can't filter with just the packet filter).

3.5 Run UDP and TCP Services, on Different Servers

Another research challenge is how can we use stateless filters to filter Reflection Attacks? Our novel solution is to run different types of Internet services, such as UDP and TCP services, on different servers.

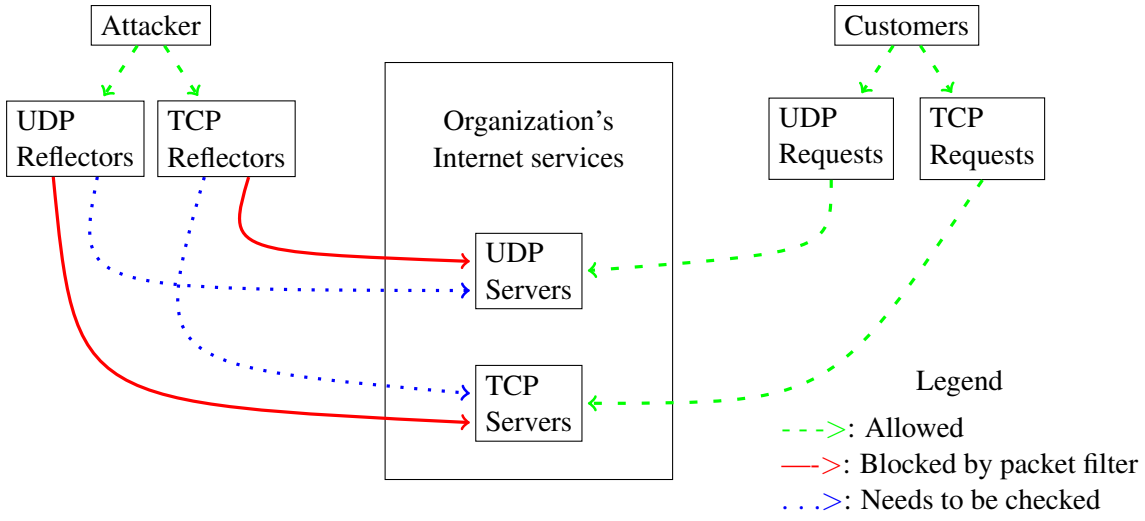

Figure 13: Run TCP and UDP Internet services, on different hosts.

This will in turn increase the amount of traffic that can be filtered with just a stateless packet filter firewalls. If you have, for example, a TCP and UDP service, on the same virtual machine, you can't filter all UDP Reflection Attack traffic with just a packet filter. However, if you only have TCP services on a given computer, and have no UDP traffic, you can use the Azure UDP stateless packet filter, to filter any and all possible UDP Reflection Attack traffic. We were unable to find any related work papers, other than our own [3], where they recommend running TCP and UDP services on different computers, so that packet filtering can eliminate many Reflection Attacks. There is no longer a need to consider the figure's traffic in green or red. So, from this point forward, we will focus on the traffic, in blue, which still needs to be checked and filtered when possible. This is shown in figure 13.

Design Pattern 4: We recommend that organizations separate their Internet service servers, as follows: 1) servers running only UDP services, 2) servers running only TCP services, and 3) servers which must run both UDP and TCP services. On TCP only servers, use the Azure stateless packet filter to filter UDP and all other protocol traffic. On UDP only servers, use the Azure packet filter to filter TCP and all other protocol traffic.

Here is an example of the Azure packet filter firewall rules, which should be used after separating the UDP and TCP services. The following stateless rules are for a TCP server. We'll assume you are running a web server and a MySQL server. The following rule will allow HTTP TCP traffic to the Web Server:

```
Get-AzureNetworkSecurityGroup -Name "MyVNetSG" ‘
 | Set-AzureNetworkSecurityRule -Name Web_80 -Type Inbound -Priority 220 ‘
 -Action Allow -SourceAddressPrefix ’INTERNET’ -SourcePortRange ’*’ ‘
 -DestinationAddressPrefix ’*’ -DestinationPortRange ’80’ -Protocol TCP
```

The following rule will allow HTTPS TCP traffic to the Web Server:

```
Get-AzureNetworkSecurityGroup -Name "MyVNetSG" ‘
| Set-AzureNetworkSecurityRule -Name Web_443 -Type Inbound -Priority 230 ‘
-Action Allow -SourceAddressPrefix ’INTERNET’ -SourcePortRange ’*’ ‘
-DestinationAddressPrefix ’*’ -DestinationPortRange ’443’ -Protocol TCP
```

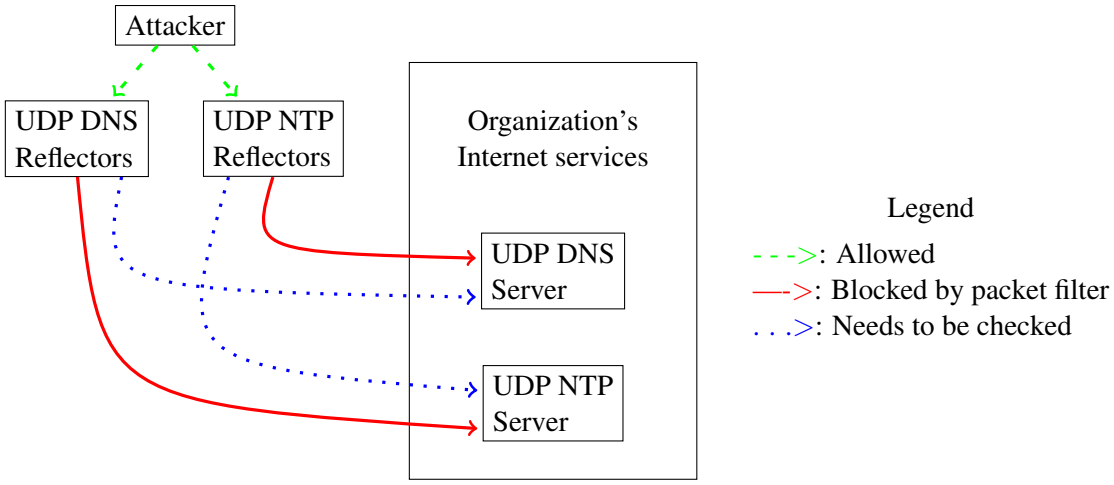

Figure 14: Separating UDP services onto different servers.

The following rule will allow SQL TCP traffic to the MySQL Server:

```
Get-AzureNetworkSecurityGroup -Name "MyVNetSG" ‘
| Set-AzureNetworkSecurityRule -Name MySQL -Type Inbound -Priority 240 ‘
-Action Allow -SourceAddressPrefix ’INTERNET’ -SourcePortRange ’*’ ‘
-DestinationAddressPrefix ’*’ -DestinationPortRange ’3306’ -Protocol TCP
```

The Azure firewall includes a default deny all. So you don't need to add this deny rule.

3.6 Run Each Service on a Separate Server

Based on our previous design patterns, the only possible remaining reflection malicious traffic is 1) from the TCP reflectors to TCP servers and from the UDP reflectors to the UDP servers (which is shown in figure 13 as the blue line traffic). As we stated in our research problem constraints, we are limiting this research article to attacks from UDP reflectors, since this is the most common and most serious threat to Internet services. So we will now focus on just the UDP reflector malicious attacks being received by the UDP servers. So let's now analyze the scenario, where a UDP server is running multiple UDP Internet services. For example, let's assume the Internet server is running both the NTP and DNS UDP Internet services. This will result in additional needless vulnerabilities, as we will show, which can be eliminated. Under these assumptions, the server is vulnerable to both a NTP Reflection Attack and a DNS Reflection Attack. We will now show how to mitigate these vulnerabilities. Our design pattern is to separate the UDP Internet services, onto different servers, as show in figure 14.

Note that we assume that on the DNS server there is no needed NTP traffic and on the NTP server there is no needed DNS traffic. Both of these vulnerabilities can be completely eliminated, via our following design pattern proposal.

1. On the UDP DNS server, you should perform an Azure stateless packet filter, and drop all UDP NTP traffic, with the source port of NTP/123 (since all NTP reflection traffic will use the server port of NTP/123). Note that we are aware of some limited issues, where the DNS needs to also run via TCP.

2. Likewise, on the UDP NTP server, you can perform an Azure stateless packet filter, and drop all UDP DNS traffic, with the source port of DNS/53.

The idea of running each service on each separate server is considered as creating micro-services and works very well when implemented with Docker containers. We were unable to find any related work papers, where they recommend running each UDP service on different computers, so that packet filtering can eliminate Reflection Attacks.

Design Pattern 5: On servers, the minimum number of Internet services should be run on each server. So, for example, if you have two UDP Internet services, such as NTP and DNS, they should be run on separate servers.

Here is an example of the Azure packet filter firewall rules, which should be used after separating the services. The following stateless rules are for the NTP server. First we need to accept UDP traffic which is being sent to our service.

```
Get-AzureNetworkSecurityGroup -Name "MyVNetSG" ‘
 | Set-AzureNetworkSecurityRule -Name UDP -Type Inbound -Priority 202 ‘
 -Action Accept -SourceAddressPrefix ’INTERNET’ -SourcePortRange ’*’ ‘
 -DestinationAddressPrefix ’*’ -DestinationPortRange ’123’ -Protocol UDP
```

The Azure firewall’s default deny which will drop all incoming traffic, to any UDP port other than 123. It will also drop all unsolicited incoming TCP traffic.

3.7 Example of Cloud Firewall Configuration

With the Azure cloud, there are two ways to configure firewall rules, the access control lists and the network security groups (NCG). The NCG firewall configuration has more stateless firewall configuration options. We showed some NCG firewall rules earlier. Here is the main process we used which shows the firewall algorithm. We first created the Azure NCG, as follows:

```
New-AzureNetworkSecurityGroup -Name "MyVNetSG" -Location "North Europe" ‘
       -Label "Security group for my Vnet in North Europe"
```

Then we need to add some packet filter rules, which we showed how to do earlier.

We then need to associate our NSG with our VM server.

```
Get-AzureVM -ServiceName "src-reflect" -Name "Web" ‘
 | Set-AzureNetworkSecurityGroupConfig -NetworkSecurityGroupName "MyVNetSG" ‘
 | Update-AzureVM
```

We can then view our firewall rules and configuration.

```
Get-AzureNetworkSecurityGroup -Name "MyVNetSG" -Detailed
```

3.8 Additional Related Work

We have analyzed some related works and created table 3. We have stated which authors have proposed something very close to our Design Patterns (DP 1-5). We have also included a column, as to if their

proposal is available today from major security vendors (as opposed to a proposal which must first be implemented by security vendors). The Background column states if we consider their paper as one of the best documents, explaining the Reflect Attack background. The first row is referring to this article.

Item	**Cite**	**Available Today**	**Background**	DP 1	DP 2	DP 3	DP 4	DP 5
0	N/A	✓	✓	✓	✓	✓	✓	✓
1	[3]	✗	✓	✓	✓	✗	✓	✗
2	[4]	✗	✓	✗	✗	✗	✗	✗
3	[7]	✓	✗	✓	✗	✗	✗	✗
4	[8]	✓	✓	✓	✗	✗	✗	✗
5	[4]	✗	✓	✗	✗	✗	✗	✗
6	[11]	✗	✓	✗	✗	✗	✗	✗
7	[13]	✗	✓	✗	✗	✗	✗	✗
8	[14]	✓	✓	✓	✗	✗	✗	✗
9	[17]	✓	✗	✓	✗	✗	✗	✗
10	[6]	✓	✗	✓	✗	✗	✗	✗
11	[9]	✓	✓	✓	✗	✗	✗	✗
12	[18]	✓	✓	✓	✗	✗	✗	✗
13	[16]	✗	✓	✗	✗	✗	✗	✗
14	[12]	✗	✓	✗	✗	✗	✗	✗
15	[1]	✗	✓	✓	✗	✗	✗	✗
16	[15]	✗	✓	✗	✗	✗	✗	✗
17	[2]	✗	✓	✗	✗	✗	✗	✗
18	[5]	✗	✓	✗	✗	✗	✗	✗

Table 3: Analysis of research papers concerning our design patterns

The related work which provides some great additional UDP Reflection Attack technical details, which is by Rossow [13]. There is also a great related work as to how TCP Reflection Attacks operate by Kührer, et. al. [11], while our paper is focused on UDP Reflection Attacks. There is an excellent related work which discusses which types of systems are vulnerable to Reflection Attacks and does a great job to show how to discover and mitigate the attacking systems, by preventing these systems from spoofing their IP address, by Kührer, et. al. [10].

There are other general related works, however we were unable to find any related works, which specifically discussed a cloud based, almost free, anti-Reflection Attack defense via simple and high performance packet filters. Most of the related papers provide recommendations and algorithms which can't be immediately implemented by organizations (and are intended for implementation by security equipment manufacturers). As stated in our introduction, we limited our research to what organizations can do today, to immediately mitigate reflection bandwidth attacks, at an extremely low cost.

4 Conclusions

The security attackers should in general, be interested to find the lowest cost methods, which will allow their network reflection DDoS attacks to be successful. That is why Reflection Attacks are so popular. At the same time, the security defenders should be interested to defend in such a way, which will require their attackers to have the highest cost possible. The defenders should also try to minimize their costs. This security description is nothing new and is quite general. Via our novel cloud firewall design patterns, we have met all of these low cost defense goals. We have shown how simple stateless packet filters can be used to eliminate many Reflection Attacks, as opposed to using more complex and slower stateful firewall filters.

In summary, for under 1% of the cost of competing anti-Reflection Attack security solutions, such as CloudFlare, we have provided an architecture which includes several specific design patterns, and can be used to very efficiently eliminate most of the Reflection Attacks. We have shown a way to transfer the defense costs to the Microsoft Azure cloud, starting at just 11 Euro / month, which can otherwise cost 5,000 Euro / month. Our design pattern contributions are extremely easy to implement. There is no longer a need for researches to find new stateful firewall solutions to eliminate the same attack traffic, which we can now eliminate via simple stateless packet filters. Our recommended future work, which we have started on, is to enhance our architecture to defend against TCP network based DDoS, bandwidth, and/or reflection attacks.

References

[1] P. Arun Raj Kumar and S. Selvakumar. Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. *Computer Communications*, 36(3):303–319, 2013.

[2] M. Barati, A. Abdullah, N. Udzir, R. Mahmod, and N. Mustapha. Distributed Denial of Service detection using hybrid machine learning technique. In *Proc. of the 2014 International Symposium on Biometrics and Security Technologies (ISBAST'14), Kuala Lumpur, Malaysia*, pages 268–273. IEEE, August 2015.

[3] T. Booth and K. Andersson. Elimination of DoS UDP Reflection Amplification Bandwidth Attacks, Protecting TCP Services. In R. Doss, S. Piramuthu, and W. ZHOU, editors, *Proc. of the 1st International Conference on Future Network Systems and Security (FNSS'15), Paris, France*, volume 523 of *Communications in Computer and Information Science*, pages 1–15. Springer International Publishing, June 2015.

[4] A. Brucker, L. Brügger, and B. Wolff. Formal firewall conformance testing: An application of test and proof techniques. *Software Testing Verification and Reliability*, 25(1):34–71, 2015.

[5] Y. Chen, X. Ma, and X. Wu. DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory. *IEEE Communications Letters*, 17(5):1052–1054, 2013.

[6] W. Dou, Q. Chen, and J. Chen. A confidence-based filtering method for ddos attack defense in cloud environment. *Future Generation Computer Systems*, 29(7):1838–1850, 2013.

[7] F. Guenane, B. Jaafar, M. Nogucira, and G. Pujolle. Autonomous architecture for managing firewalling cloud-based service. In *Proc. of the 5th International Conference on the Network of the Future (NOF'14), Paris, France*, pages 76–80. IEEE, December 2014.

[8] F. Guenane, M. Nogueira, and G. Pujolle. Reducing DDoS attacks impact using a hybrid cloud-based firewalling architecture. In *Proc. of the 2014 Global Information Infrastructure and Networking Symposium (GIIS'14), Montreal, Quebec, Canada*, pages 1–6. IEEE, September 2014.

[9] A. Khakpour and A. Liu. First step toward cloud-based firewalling. In *Proc. of the 31st IEEE Symposium on Reliable Distributed Systems (SRDS'12), Irvine, California, USA*, pages 41–50. IEEE, October 2012.

[10] M. Kührer, T. Hupperich, C. Rossow, and T. Holz. Exit from hell? reducing the impact of amplification ddos attacks. In *Proc. of the 23rd USENIX Security Symposium (USENIX Security'14), San Diego, California, USA*, pages 111–125. USENIX Association, August 2014.

[11] M. Kührer, T. Hupperich, C. Rossow, and T. Holz. Hell of a handshake: Abusing tcp for reflective amplification ddos attacks. In *Proc. of the 2014 USENIX Workshop on Offensive Technologies (WOOT'14), San Diego, California, USA*. USENIX Association, August 2014.

[12] P. Priya, V. Akilandeswari, S. Shalinie, V. Lavanya, and M. Priya. The Protocol Independent Detection and Classification (PIDC) system for DRDoS attack. In *Proc. of the 2014 International Conference on Recent Trends in Information Technology (ICRTIT'14), Chennai, India*. IEEE, April 2014.

[13] C. Rossow. Amplification hell: Revisiting network protocols for ddos abuse. In *Proc. of the 2014 Network and Distributed System Security Symposium (NDSS'14), San Diego, California, USA*. Internet Society, February 2014.

[14] K. Salah, J. M. A. Calero, S. Zeadally, S. Al-Mulla, and M. Alzaabi. Using cloud computing to implement a security overlay network. *IEEE security & privacy*, (1):44–53, 2013.

[15] T. Spyridopoulos, G. Karanikas, T. Tryfonas, and G. Oikonomou. A game theoretic defence framework against DoS/DDoS cyber attacks. *Computers and Security*, 38:39–50, 2013.

[16] W. Wei, F. Chen, Y. Xia, and G. Jin. A rank correlation based detection against distributed reflection DoS attacks. *IEEE Communications Letters*, 17(1):173–175, 2013.

[17] X. Ye and Y. Ye. A practical mechanism to counteract DNS amplification DDoS attacks. *Journal of Computational Information Systems*, 9(1):265–272, 2013.

[18] S. Zargar, J. Joshi, and D. Tipper. A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks. *IEEE Communications Surveys and Tutorials*, 15(4):2046–2069, 2013.

Author Biography

Todd Booth has over 20 years experience in computer networking and computer security. He has a B.Sc. in Mathematics/Computer Science from UCLA with honors and has taught networking and programming classes at UCLA. He has a M.Sc. in Computer Science related Information Security from the University of Technology (in Luleå, Sweden). He has been teaching masters level information security courses full time, for five years, and has received the teacher of the year award. He has been recognized by IBM as a visiting scientist. He is now a Ph.D. student at Luleå University of Technology, Sweden with a research discipline of Computer Science related Information Systems (IS), a research area of network security and his research interests include network attacks. Contact information: Research ID: `http://www.ResearcherId.com/rid/C-3576-2015` and Orc. ID: `http://OrcId.org/0000-0003-0593-1253`

Karl Andersson (Senior Member of IEEE) has a M.Sc. degree in Computer Science and Technology from Royal Institute of Technology, Stockholm, Sweden and a Ph.D. degree in Mobile Systems from at Luleå University of Technology, Sweden. After being a postdoctoral fellow at the Internet Real-time Laboratory at Columbia University, New York, USA and a JSPS Fellow with National Institute of Information and Communications Technology, Tokyo, Japan, he is now Associate Professor of Pervasive and Mobile Computing at Luleå University of Technology, Sweden. His research interests include mobile computing, Internet of things, cloud technologies, and information security. Contact information: Research ID: `http://www.ResearcherId.com/rid/E-3611-2010` and Orc. ID: `http://OrcId.org/0000-0003-0244-3561`