O'REILLY®

Compliments of A10

Distributed Denial of Service (DDoS)

Practical Detection and Defense

Eric Chou & Rich Groves

Your DDoS Protection Solution

– *A10 THUNDER TPS* –

> *"We will continue to partner to mitigate future threats leveraging DCU's expertise and A10's advanced threat protection technologies."*
>
> – Microsoft Digital Crimes Unit

> *"The flexibility, scalability and integration with different architectures that Thunder TPS offers is a clear benefit to our customers."*
>
> – Leading computer security services company

> *"We deployed A10 Thunder TPS in a proactive, symmetric configuration which allowed continuous, comprehensive detection and faster mitigation."*
>
> – A prestigious University in Asia

SEE THE DIFFERENCE: A10NETWORKS.COM/DDOS

Distributed Denial of Service (DDoS)

Practical Detection and Defense

Eric Chou and Rich Groves

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Distributed Denial of Service (DDoS)
by Eric Chou and Rich Groves

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Editor: Courtney Allen
Production Editor: Nicholas Adams
Copyeditor: Gillian McGarvey
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest
Tech Reviewers: Allan Liska, JR Mayberry, and Nick Payton

March 2018: First Edition

Revision History for the First Edition
2018-02-27: First Release

978-1-492-02615-0

[LSI]

Table of Contents

Foreword v

1. DDoS Attacks: Overview 1
What Are DDoS Attacks? 2
Why Are DDoS Attacks Effective? 4
Who Is Behind the Attacks and What Is Their Motivation? 5
Common Types of DDoS Attacks 9
Botnets and IoT Devices 12
Summary 14

2. DDoS Detection 15
Poll-Based Monitoring and Detection 16
Flow-Based Network Parameter Detections 18
Network Mirrors and Deep Packet Inspection 21
Anomalies and Frequency-Based Detections 24
Summary 27

3. DDoS Mitigation and Countermeasures 29
DDoS Terms and Traffic Flow 31
DDoS Mitigation Topology 34
Network-Level Mitigation Tools 37
Session-Level Mitigation Tools 39
Example 1: Combating the Classic Flood 41
Example 2: Combating State Exhaustion 46
Emulate DDoS Attacks for Better Response 49
Summary 50

4. Evaluating Cloud-Based Mitigation Vendors. 51
Why Use Cloud-Based DDoS Mitigation? 52
When Not to Use Cloud-Based DDoS Mitigation 55
Cloud-Based DDoS Mitigation Methods 59
DDoS Mitigation Mechanism in the Cloud 60
Summary 64

5. DDoS Focused Threat Intelligence. 67
IP Blocklists 68
Community Supported Efforts 70
Honeypots 74
DDoS-as-a-Service 76
Summary 77

6. Final Thoughts. 79

Foreword

Humans need to be connected to one another for society to flourish. The internet is an essential connector in today's world. By 2020, it is projected that there will be 50 billion internet-connected devices in use. With the rise of new technologies in our lives, new cyber threats and attacks regularly occur. We're seeing politically motivated DDoS attacks, and a new twist on cyberattacks—the 2017 attempt to cash in on the soaring price of Bitcoin. We need cyber-warriors to continually out-think and out-smart those who are using IoT devices, cloud infrastructures, and other technologies against us.

As we implement the next generation of security solutions, intelligent automation that leverages machine learning is the weapon we need to win the cyber war. But technology alone is not enough. We all need the tenacity and dedication of our security experts to ensure our digital life not only endures, but thrives for all, as it should.

Working with Rich and Eric at A10, I've witnessed their tenacity and dedication to winning the cyber war. They have been key warriors architecting next-generation security solutions and working with third parties to develop systems to take down and dismantle massively damaging global botnets. Their efforts have benefited millions of users.

I'm honored to write this foreword for them, and I'm excited to have this book as a resource for fellow warriors.

— Lee Chen, A10 CEO

CHAPTER 1
DDoS Attacks: Overview

It is the morning of Christmas in 2014, a day on which, in many areas of the world, kids and adults alike awake to cheerful Christmas music and gift-wrapped presents underneath the Christmas tree. Smiling from ear to ear, many eagerly unwrap the gift of a new game console such as a Microsoft Xbox or Sony PlayStation. Others jump for joy for the latest and hottest release of online games. As they rush to fire up the new console or game, they wait patiently for the game to register online and start. They wait and wait, only to be greeted with a "Service Unavailable" error.

Upon further research, news that the gaming sites are under a Distributed Denial of Service attack, or DDoS, starts to surface. The companies' social media outlets, shown in Figure 1-1 with over 1,000 retweets, begin to fill with angry comments from frustrated users. Rumors on the web start to swirl around as to who were the malicious actors, what their motivations were, and when the service will be restored.

It was later confirmed that the service disruption was due to a group of malicious actors called Lizard Squad launching the DDoS attack on the gaming companies. The gaming services were interrupted on one of the biggest holidays of the year and a large sum of revenue was lost. More importantly, the reputation of the companies was severely damaged and consumer confidence in the service took a punishing hit that took the companies years to regain.

Figure 1-1. Sony PlayStation "Service Unavailable" Twitter message from December 25, 2014

In this chapter, you will find answers to questions such as what DDoS attacks are and why they are effective. You will also learn about who is behind the attacks and what their motivations are, as well as common types of DDoS attacks.

Let's get started by looking at what DDoS attacks are.

What Are DDoS Attacks?

Let's start by separating "Distributed" from "Denial of Service" and looking at them separately. Simply put, a Denial of Service is a way to make the service unavailable, thus denying the service to users. Often times, this is done by blocking the resources required for providing the service. One of the most effective ways of doing this is to generate lots of bogus requests from different, or "Distributed," sources, which drowns out legitimate requests.

Imagine for a minute that you own a corner bakery. As a merchant, you need certain elements to happen before you can transfer goods into the hands of customers. In order to complete the transaction, many elements are required; three of them are shown in Figure 1-2:

1. The customers need to know how to access your store. They will need a way to look up your store address, such as by calling the local directory service.
2. The customers need to take some kind of transportation to your store and access the goods by walking into your store through the door.

3. The customers need to pay for the goods they wish to purchase. On the merchant side, you will need a mechanism to document the transaction so you can calculate any necessary taxes and fees as well as the price of the goods. You might also need a form to process electronic payments such as credit card transactions.

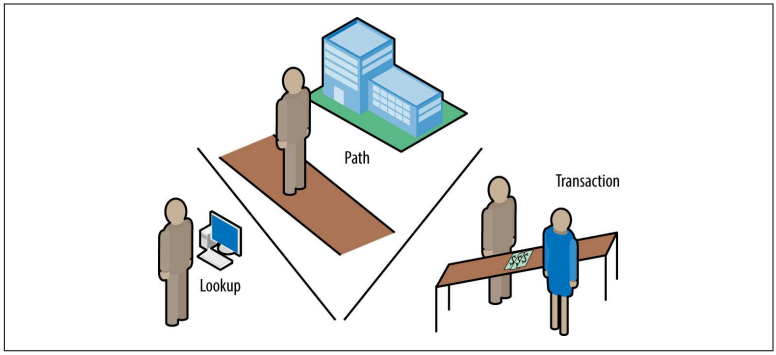

Figure 1-2. Required elements of a business transaction

Now let's assume that I am a bad guy who does not want the transaction to succeed, or that I am somebody who is simply curious if I can stop that transaction from happening. By carefully observing the three elements above, the DDoS equivalent of blocking the service are shown in Figure 1-3:

1. I can disallow the address lookup for your store. For example, if the address lookup is done by an operator-directed service, I can place a lot of calls to the operator, which will block new calls from coming in.
2. I can hire a lot of people to block the street or your store entrance so the customer cannot get into your store.
3. I can place a lot of low-level transactions to your credit card service (e.g., buying a lot of one-cent candies) thus delaying the transaction for higher dollar value items. I can also distract the cashier by asking them to do something else such as answer phone calls.

As you can see, the act of denying service usually requires a large volume of a partially legitimate act. In the analogy just given, at least in the beginning, it is hard to tell if somebody standing in front of your door is a legitimate potential customer or if their intention is to block other customers.

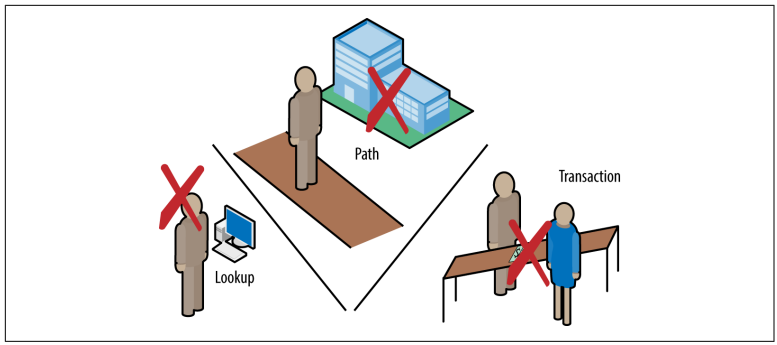

Figure 1-3. DDoS for different business elements

The example of the corner bakery can be extrapolated to our digital world today. The store could be your e-commerce store, the public street that leads to your store could be the various internet connections, and the cash register could be the web server that handles your check-out process. The address-lookup of the store is analogous to the domain-name-to-IP-address translation, which is a service that historically has been a target of DDoS attacks.

In the next section, we will take a look at what makes DDoS effective.

Why Are DDoS Attacks Effective?

We are living in a world that is more digitized than ever. "Software is eating the world," declared Marc Andreessen in a 2011 Wall Street Journal article. For many people, the first thing that comes to mind when discussing cybersecurity is software bugs. Software is created by humans, and humans introduce bugs to the applications. Even software widely used by thousands of people every day can have bugs that are only discovered years after its release; a good example is the Heartbleed OpenSSL vulnerability in CVE-2014-0160. Fortunately, even though bugs exist, if the software was written using best practices by top software developers, they are difficult to catch. You have to be an expert in the given field in order to catch them. Top technology companies, like Google and Microsoft, have the so-called "bug bounties" programs that reduce the likelihood of a zero-day threat even more.

DDoS attacks are different from software bugs in that an understanding of the underlying mechanism of the software or infrastruc-

ture is not required to carry out a successful attack. An attack can be even more potent if the attacker understands the architecture, but some of the more successful attacks that we have seen were carried out by industry outsiders. The complexity of the attack relies on the ability of the attacker to control a lot of administered sources. In today's connected world where everybody carries a smartphone in their pocket, lives in a home where every lightbulb and thermostat have embedded computers, and travel in self-driving cars with supercomputers for brains, it is not difficult to see where such hosts can be found. Later in this chapter, we will discuss the botnets and Internet-of-Things (IoT) that can be used as seemingly legitimate sources in DDoS attacks.

The simplicity of the process and the proliferation of the ever-expanding connected world we live in is what make DDoS attacks so effective, in our opinion. If anyone with a relatively small amount of money can rent a botnet and launch DDoS attacks, the chances of a successful attack increase tremendously. In defending your network against these attacks, it is worth noting that the good guys need to defend almost all attacks while the bad guys only need to succeed once to achieve their goal. For the entities needing to defend against DDoS attacks, there is a real cost in the area of equipment, knowledge, operations, and lost productivity associated with the attacks.

In Chapter 5, we will examine how to turn a passive defense into a more active offense by using honeypots and threat intelligent systems.

Who Is Behind the Attacks and What Is Their Motivation?

You might be wondering who the people are behind the DDoS attacks and what their motivations are. In general, they can be divided into several categories. We will look at some of them.

Criminals

Perhaps the easiest group to understand is the criminals who seek financial gain from the DDoS attacks they conduct. The most straightforward way for the criminals to earn money from an attack is to make themselves available to be hired to attack designated targets on demand. This is often disguised as stress testing sites. Gran-

ted, some vendors do offer legitimate stress test services, but rogue stress test sites often do not verify the identity and source of the requester, no question is asked by the stressor regarding the target, and certainly no advance warnings are given to the target. When these conditions occur, it is often understood that they are DDoS-for-hire guys.

Often the attack is done automatically without the buyer ever being in contact with the person or group providing the attack service. The transaction is often paid for in untraceable currency, such as Bitcoin. Interestingly enough, nowadays DDoS-for-hire is a very competitive market; it is our experience when we hire some of them for attack research (we attack targets that we own, of course) that they often provide good customer service. If the attack target failed to go down, they would even offer a refund. Figure 1-4 shows an example of a self-service DDoS-for-hire website.

Another way for a criminal to earn money from DDoS attacks might be to demand ransom from institutions in exchange for *not* launching a DDoS attack against them. The attackers might demonstrate that they can successfully bring down the target at a smaller scale, making it inaccessible for a short period of time, before demanding a larger ransom from the victim to stop a larger attack down the road.

Figure 1-4. DDoS for Hire Botnet (source: http://bit.ly/2rXJ3NZ)

How Easy Is It to Pay for a DDoS?

A question that people often ask is, "How easy it is to pay for a DDoS?" From our experience, it is extremely easy to find a potential provider, although the results of the attacks will vary. In one instance, we paid for a five-minute attack via Bitcoin and saw the spike in traffic on our attack target immediately (in this case, our cloud-based instance). In another instance, we were only able to observe a limited amount of incoming traffic spike.

If you operate an internet-facing business and someone threatens to DDoS attack you, we recommend that you be cautious but do not give in to the threat, even if they have conducted a small-scale proof of attack. It is always a good idea to start collecting data from the threat to prepare for possible legal actions and to start preparing your infrastructure and staff by increasing visibility and operating procedures. But keep in mind that it is always a slippery slope once you start to cave in to the attackers.

Thrill Seekers and Status Seekers

There are of course people who launch DDoS attacks for the thrill of having done something that is disruptive so they feel they are in control and powerful. Besides DDoS-for-hire sites, in the world of open source projects and knowledge sharing, DDoS attack tools can often be obtained easily. Thrill seekers do not need in-depth knowledge of the tool, as many of the open source tools have simple point-and-click interfaces to successfully launch an attack. Since the attack tools can often be as simple as a programming script, sometimes we refer to thrill seekers as "script kiddies." The ease of getting such a script might surprise some—it can be as simple as a digital trip to a hacker forum (Figure 1-5) to obtain the necessary scripts and instructions.

Besides people who DDoS attack others for fun, sometimes the motivation can be to obtain a certain status within the community they belong to. People who are seeking status often pick well-known sites that are more difficult to bring down. There is a me-against-them mentally from the attacker to the establishment. They are often eager to claim credit and brag about the event online.

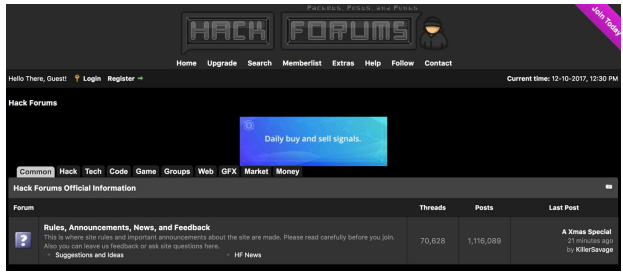

Figure 1-5. Hackerforum.net for scripts

The line between thrill seekers and status seekers is often blurred. A classic example can be that of the Lizard Squad case that we mentioned earlier. The group was clearly amused by the amount of attention they got, even demanding that other Xbox and PlayStation users write Lizard Squad on their foreheads to stop the attack. They were also eager to claim their status as "the group that brought down Xbox Live and Sony PlayStation Network."

Angry and Disgruntled Users

Quite surprising to us when we initially looked into the DDoS security space, the most common DDoS attacks were not done by one group to another, but rather from one user to another. This is especially common in the gaming community as it consists of passionate users who are deeply invested in the environment with their time and money. It stands to reason that when one party is losing during a competition, sometimes that party would try to take a shortcut by knocking the other user offline. It is so common in the industry that there are FAQs and established standard procedures that companies direct their users to if they feel they are under a DDoS attack.

The angry and disgruntled user could also be ex-employees or angry customers who had a bad experience. It really goes to show how little friction exists today to launch a DDoS attack, therefore making it a common tool for angry and disgruntled users to turn to.

Hacktivist

The angry user scenario does not stop at the gaming industry for taking recreational activity a bit too far. Angry users can also be those who are protesting a certain company policy or value. It can

also be political motivation and beliefs with no financial or criminal intentions associated with these individuals. The infamous group Anonymous was a strong hackivist group. You still see hacktivist attacks toward official government establishments, as well as the likes of North Korea and ISIS.

DDoS as a Distraction

We are focusing on DDoS attacks in this book. However, DDoS attacks can sometimes serve as a distraction while the malicious hackers work on other security compromises. "Go look at this loud noisy thing while we backdoor you over here unnoticed because your hair is on fire." It is well published that a lot of DDoS attacks have resulted in additional compromise (source: *http://bit.ly/2GBfAgd*).

Common Types of DDoS Attacks

In this section, we will look at the most common types of DDoS attacks. New attacks happen often, and most of the time they can be generalized and put into existing categories. By separating one type of attack from another, we can then devise generalized mitigation strategies for each of them. Though there are different types of DDoS attacks, they all rely on traffic volume. It is worth mentioning that the attack can succeed as long as they can break the weakest link in the network since there are many different elements in the network.

The Weakest Link

The saying "A chain is as strong as its weakest link" couldn't be truer in the case of DDoS attacks. There are many interconnected components in the computer network today, such as Domain Name Service (DNS), upstream internet service providers, wireless access points, and web servers, to name a few. If you can flood the web server and bring down the service, even if you have the strongest DNS system, the impact is still the same for the user.

Volumetric Floods

The attacker can simply flood the network with traffic to starve out the legitimate requests and render the service unavailable. The target can be any of the network components, such as a flood of requests to the DNS or web server. The DNS and web server need to be public in order for people to request service from them, and they can be a direct target for the attacker. It is worth noting that in the case of flooding, the request does not need to be properly formatted. In other words, as long as the request packet makes its way to the target the attack can potentially succeed.

Network Protocol–Level Attacks

The internet is built on common layers of technologies; this is part of the fundamental bedrock that allows different systems to communicate with each other. You might be familiar with the OSI model that standardized the communication model among computer systems. The transport layer consists of the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) that most modern applications are built on. For example, the HTTP protocol that serves web pages is built on TCP while the DNS protocol is built on UDP.

The TCP and UDP protocols are built on the idea of openness and inclusivity, just like the internet itself. Though this idealism made the internet what it is today, it also gave the attackers the same level ground as everybody else. The operation of the protocol, as well as their possible vulnerabilities, can be gleaned easily from publicly accessible documents and then used in a DDoS attack.

For example, the TCP protocol relies on a three-way handshake where the receiver keeps the state of the connection after the initial contact, known as SYN. One of the oldest DDoS attacks consists of the attacker sending the server a flood of TCP SYN packets that exhausts the server's resources.

Amplification and Reflection

While TCP is vulnerable in that the host requires more resources to be tied up and easily exhausted in a flood situation, the connectionless nature of UDP is also susceptible to DDoS attacks and more often misused. In particular, because the UDP-based server does not

verify the source in favor of a faster connection, the UDP protocol is often leveraged in an amplification and reflection attack. The amplification and reflection usually go hand in hand.

Consider the analogy in Figure 1-6 of a prank that is sometimes played by teenagers: the prankster, Bill, calls a pizza shop pretending to be Mike and orders 100 pizzas to be delivered to his house.

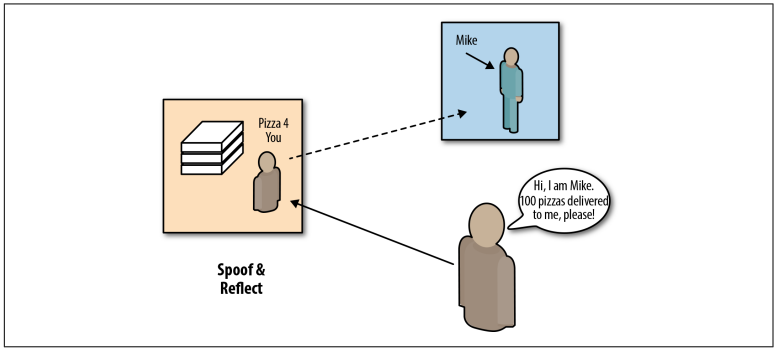

Figure 1-6. Pizza delivery prank

If the pizza shop does not verify that the source of the call was indeed from Mike (instead of Bill pretending to be Mike), and goes ahead and makes and delivers the 100 pizzas, both the pizza shop and Mike will be left with an ugly situation.

In the world of UDP, unlike TCP, by design it does not verify the request IP source. Therefore, the attacker can easily spoof the victim as the source by making a UDP request to a server, and reflect the response of the server toward the victim. In Figure 1-7, we illustrate a simple packet flow from a spoofed source, amplifier, and the victim.

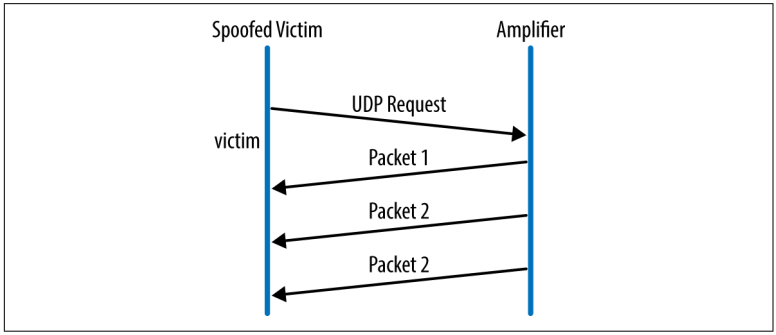

Figure 1-7. UDP amplification and reflection

If you couple the reflection with a small size of requests that result in a large response, the amplification effect would take place. This is precisely the type of attack that would result in the victim being DDoS attacked. Some examples of such an attack include DNS amplification and NTP reflection attacks.

Application-Level Attacks

The application-level attack requires more application-level knowledge but not necessarily in-depth knowledge. For example, if you understand the basics of the HTTP protocol POST, you can launch a low-and-slow POST operation by posting one out of thousands of characters at a time to an HTTP server before the session times out. Or you can perform an HTTP GET flood knowing that the server might not have enough resources to handle the burst of GET requests.

The difference between application- and network-level attacks is the volume of traffic involved. Usually, the network-level attack is very obvious because it takes a lot more traffic to exhaust the network services, whereas the application-level attack requires a much lower volume of traffic and might be able to disguise itself until somebody familiar with the application is able to diagnose the problem.

Multivector Attacks

Of course, since the goal of the attacker is to make the service unavailable to other users, the attack can be a combination of the different types for a multivector attack. In several instances, we have seen the attack incident start out as a flood of traffic toward the network consisting of classic floods, then morphing into various other forms of attacks such as protocol-level attacks.

Botnets and IoT Devices

It is clear that the techniques of DDoS are simply a blockage of service by using a large number of distributed sources. But what are these devices? Are people knowingly giving up their computer to participate in a DDoS attack? The answer is no. Oftentimes the hosts used in the attack are unknowingly affected via malware or some kind of Trojan horse software that disguises itself as something useful or interesting to the user but in reality provides a backdoor for another computer to take control.

These infected hosts are often called *bots*, and the cluster of bots are referred to as *botnets*. The unaware users who open mail attachments that are executable programs or who download pirated movies that are actually malware often unknowingly become part of the botnets. This problem is sometimes lessened by more educated users who understand the risk and do not perform any of these actions.

However, one scary trend lately is the rise of Internet of Things (IoT) devices. The term often refers to connected homes that contain the internet-connected thermometer, doorbell, DVR, and light switches. Though they provide useful functions to benefit our lives, one problem is that these devices are relatively powerful and large in number, often unmanaged, and many times shipped with exploits that cannot be patched for some time—if ever. The most recent Mirai attack is a good example of IoT devices that are being used in a DDoS attack.

Regardless of the type of botnets, they are dormant without external instructions that direct them to send bogus requests to the attack targets. There is a controlling host that is aware of the botnets and places instructions in them when the time is right. The controlling host is referred to as the Command and Control (C&C) server. It is essentially the brain of the bots and critically important to the operations of the botnets. There are many ways a C&C server(s) or cluster of them can exist; different layers of C&C can also exist to avoid detection.

Shift to Cloud Computing

Another component is the shift towards cloud computing. Sometimes companies and end users will leave unpatched virtual machines exposed to malware and subsequently leveraged as part of a botnet.

It is worth noting that many of the botnets consist of home routers and other embedded devices. Keeping your home router firmware updated will not only keep your device out of the reach of C&C, it will also protect your digital devices at home. In Figure 1-8, you can see that only a single C&C machine can control a large number of bots.

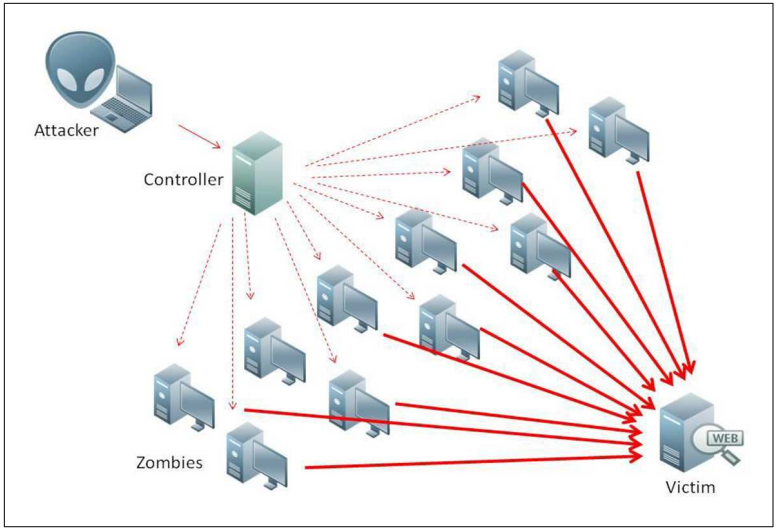

Figure 1-8. Botnet Command and Control server (source: http://bit.ly/2BKHFh7)

Botnet Takedown Efforts

There are many entities working jointly to take down the botnets. One of them is the Microsoft Digital Crimes Unit. Along with its partners around the globe, they have been successful in various botnet takedowns.

Summary

In this chapter, you have seen an overview of the DDoS attacks—from the actors to the techniques used. In the next chapter, we will take a deeper look at how to detect DDoS attacks.

CHAPTER 2
DDoS Detection

The first step in mitigating a DDoS attack is to know the attack is happening. This might sound obvious, since a volumetric attack will by nature tie up computing resources, such as bandwidth, CPU, buffer, memory, or a combination of all of those. But just as DoS, distributed or otherwise, comes in many shapes and sizes, our detection needs to match the ever-increasing types of attacks.

There are many ways to stop an ongoing or potential attack, some of them are obvious, some are less known. Our goal for detection is to quickly and accurately diagnose the attack and lower the mean time to mitigation.

In this chapter, we will look at some of the common ways to detect DDoS attacks using information gathered in poll-based and flow-based monitoring. When needed, there are instances where we need to perform packet inspection using network mirrors. We can also use anomalies and a frequency-based detection mechanism for possible DDoS attacks.

It is our opinion that there is no single detection mechanism that can detect all types of DDoS attacks. In our experience, whenever possible, all of the detection technologies mentioned in this chapter should be set up in advance and continuously validated with ongoing feedback from live traffic. The machine needs to be trained to recognize potential signals of attack from actual attacks in order to accurately predict the next one.

Tools in Your Detection Toolbelt

It is our opinion that there is no single detection mechanism that is able to detect all of the DDoS attacks! If possible, all of the detection technologies mentioned in this chapter should be set up in advance and continued to be validated with ongoing feedback with live traffic. We should leverage all data sources with the intention to help identify and understand the impact of any given attacks.

Let's begin by looking at the poll-based network detection.

Poll-Based Monitoring and Detection

The first place to start in your detection strategy is to examine the current reporting capabilities of the hardware and software in your infrastructure. Simple Network Management Protocol (SNMP) is a mature internet standard protocol defined in RFC 3411–3418 for collecting and organizing information about networked devices. It is widely supported on routers, switches, servers, workstations, and more.

The basic operation of SNMP consists of one or more management stations responsible for collecting the data from a group of hosts and devices. The managed node typically has an SNMP agent that is responsible for returning the data to the manager in a standardized format conforming to the RFC. The agent serves as a proxy that in turn queries the subagent in each device. This setup subsequently hides the proprietary components that make monitoring different proprietary systems easier.

The poll-based information retrieval can be handy because it is likely that it already exists in your devices. Once you have a management station in place, the incremental effort involved in adding a new managed node is minimal.

In terms of DDoS, SNMP can generally reveal device health information that shows signs of stress at points in your network, such as the following:

- Saturated interfaces
- High CPU

- High packets-per-second
- High rate of packet losses

Generally, when the device is under a DDoS attack, you would see a significant deviation of the metric you are tracking from the normal usage, such as the spike in network traffic shown in Figure 2-1. As mentioned, this is usually an indication of stress, and the administrator should perform further investigation in order to determine the cause of the stress. The result could have been caused by a DDoS attack but does not have to be.

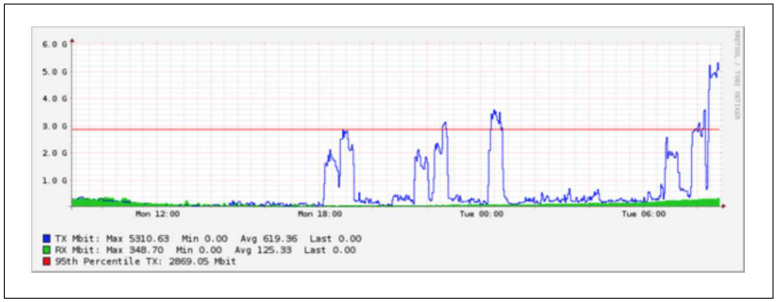

Figure 2-1. Bandwidth spike (source: http://bit.ly/2EurjMI/)

The poll-based detection mechanism is handy and useful, but the operation tends to be control-plane based and CPU-intensive. We have been in an environment where multiple management stations were polling information from a network device at a high frequency. When we reduced the number of pollers, the CPU level dropped by 30%.

NOTE

First Layer of Detection: SNMP

SNMP is a mature protocol that serves as a common denominator among network and computing devices. It is a great first response detection mechanism and should be a starting point of reference for network behavior. However, it is less likely to provide more meaningful insight other than the fact that your network is under stress.

Imagine a time when your device is under stress, such as during a DDoS attack, and the only way to retrieve more information will add even more CPU cycle to the device such as SNMP poll, thus

adding more stress to the device. SNMP might not be the best choice of tools and care needs to be taken when using SNMP. But since they are so widely used and adapted, they can be a useful first alert detection tool in your DDoS detection toolbelt.

Flow-Based Network Parameter Detections

Compared to a poll-based detection mechanism, a flow-based network detection is push-based. Shown in Figure 2-2, the device information is collected on the device itself and pushed to the collector. The basic operation consists of flow exporters and collectors. Similar to SNMP, the collector is a central aggregation point for multiple exporters. Unlike SNMP, the exporter on the device is responsible for aggregating the information before export to the collector. This task delegation allows the exporter, usually the network and system devices, to place a higher priority (if necessary) on more critical operations, such as processing BGP control packets.

The flow-based monitoring mechanism was first introduced by Cisco in the form of NetFlow; many vendors have similar mechanism but with different names, such as JFlow or CFlowd for Juniper Networks, and NetStream for Huawei Technologies. RFC 7012 is the latest IETF standard that tracks IPFIX based on NetFlow v9.

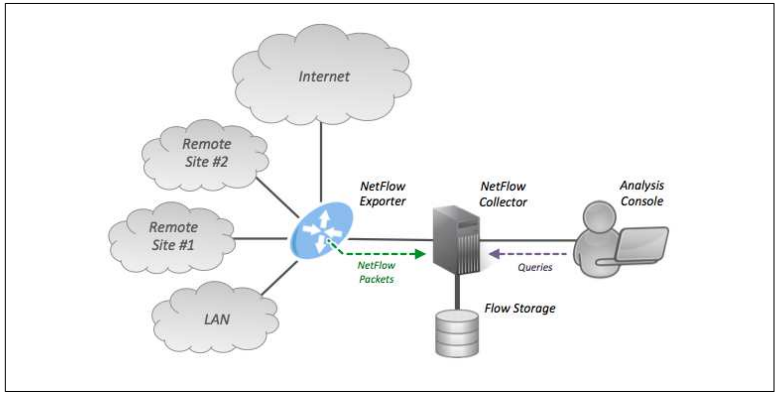

Figure 2-2. NetFlow architecture (source: http://bit.ly/2E3C2Qp)

Flow-based technologies can often perform the same function as SNMP with less CPU cycle. Although mainly used as a flow observer, in the newer version of IPFIX, the exporter can export more relevant information than its SNMP counterpart with

template-based configuration that allows more agile adaptation to newer information.

Being newer, vendor-introduced technology, NetFlow and its variants take longer to sort out and set up; however, given its usefulness, it is an invaluable tool in the DDoS detection and should be used whenever possible. The most useful nature of NetFlow is its ability to identify high offenders individually. For example, the SNMP data is usually collected on a per-interface level where you see the total bytes and packets per time interval on a network interface. When drilling down, NetFlow can be used to identify which source IP is the offender. This information is critical for mitigation, which we will cover in Chapter 3.

NOTE

Flow Information Identifies Individual Offenders

Flow information can identify the top-N traffic usage by source and destination IP. Since infrastructure devices are typically shared among many resources, this information is critical to our mitigation strategy. Figure 2-3 shows an example output.

```
Source          Destination     Packets
-----------------------------------------
192.168.0.201   192.168.0.1     235
192.168.0.202   192.168.0.1     42
```

Figure 2-3. IPFIX screen output

In a typical flow, such as a client web browser downloading a webpage, the number of packets is not known in advance. The exporter will take the first packet unique to the 5-tuple network header and identify the subsequent packets matching the information. When the flow is deemed finished, such as by timeout value or TCP FIN or RST, the number of packets and byte count is tallied and exported.

As such, the exporter needs to keep track of the flow information, record the flow information, and export it at the end. It is important to note that the exporter uses onboard resources, such as TCAM, to keep track of the flows before exporting. Because the network today can process thousands of flows per second, the flow information is generally taken in samples due to resource constraints. Therefore the information is typically expressed in "1 in N packets" sampling with the degree of error in an inverse relationship with the N pack-

ets. The higher the N, the less accurate the flow information is. When designing a NetFlow architecture, it is always a balancing act between accuracy and device overhead.

Sampled Flow (RFC 3176), or sFlow, on the other hand, try to lessen the exporter resource burden by placing the calculation and flow state information to the collector. It does so by doing a "1 in N" sampling as well as the interface counter for the same time period while exporting the sampling packet right away without keeping flow state information on the device. By doing a simple calculation of correlating the two numbers, the collector can analyze the data and derive an estimate of the individual flow usage.

sFlow was originally developed by InMon but aims to be open source, multivendor supported, and in a scaled-out design. The technology proves to be popular with so-called "white box" or newer vendors who need to lower overhead on network devices by focusing their limited resources on core functions, such as routing and switching. In Figure 2-4, we see an example of sFlow in operation.

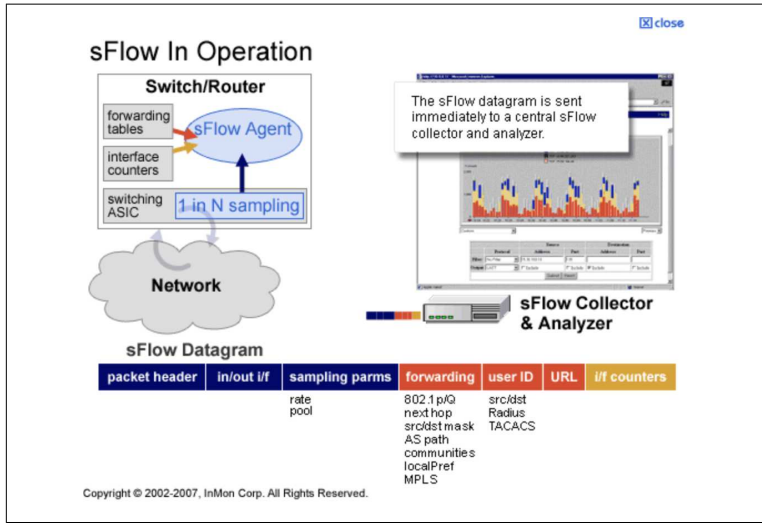

Figure 2-4. sFlow in operation (source: http://bit.ly/2nsbbUI)

Compare to SNMP, flow-based detection technology is newer and more fragmented. For example, the operator might need to implement different collectors for NetFlow and sFlow. However, because it is one of the only technologies that can identify individual usage

information, it is critical in DDoS detection. Besides immediate mitigation needs, this information is often used if you need to take legal actions against the attackers.

Time from Detection to Mitigation

In both the SNMP and flow-based detection, there is a trade-off between detection overhead versus time-to-detect. The more frequent you set the interval, the faster you can detect a potential attack. However, the additional frequency adds to the general overhead of device resources, network bandwidth, and data storage.

There is no one-size-fits-all solution to the frequency of flow export or SNMP poll interval; it is best to conduct a smaller-scale test and see which level you are comfortable with, and adjust over time.

Between the two approaches of flow-based network monitoring mechanisms, there is obviously no right or wrong solution. Sometimes you need to go with the technology that is already part of your network; other times it is worth exploring new technologies. Generally, we prefer the sFlow technology over NetFlow because of scalability and broader vendor support.

NOTE

FastNetMon Project

One of the open source projects we participate and contribute to is FastNetMon. It has both an open source community and a commercial paid edition. The project aims to use flow exports to quickly detect DDoS attacks and automatically trigger mitigation techniques.

Network Mirrors and Deep Packet Inspection

The technologies we have mentioned so far mainly covered up to Layer 4 in the OSI model. They are suitable for monitoring and detecting activities at scale in a macro-level for your infrastructure. Whenever we see a segment in a movie or TV show depicting a Network Operations Center (NOC), or a real-world NOC for that matter, macro-level monitoring is the type of output that is rightfully projected on the giant screen while the engineers look busy doing some analyzation of the data.

While SNMP and flow data can give you a great place to start, they sometimes sacrifice the details in favor of scale: SNMP, by nature, is not meant to dissect beyond the basics of the packet payload, and we already discussed the sampling nature of flow-based detection. Imagine a slow-and-low attack on your HTTP web server like the one that we mentioned in Chapter 1. In order to detect the specifics of the attack, we need to actually look at the contents of the packets instead of relying on just the header. This is typically done by placing a network mirror that indentifies a source port on a network device, makes a copy of the transmitted packet, and transmits out of the mirror port.

As illustrated in Figures 2-5 and 2-6, in many instances the only way to be 100 percent positive of the attack behavior is to look at the packets in detail. In both cases, we are able to see the payload of the packet. In the case of NTP amplification, we are able to see the NTP Monlist IP addresses that we can use for mitigation.

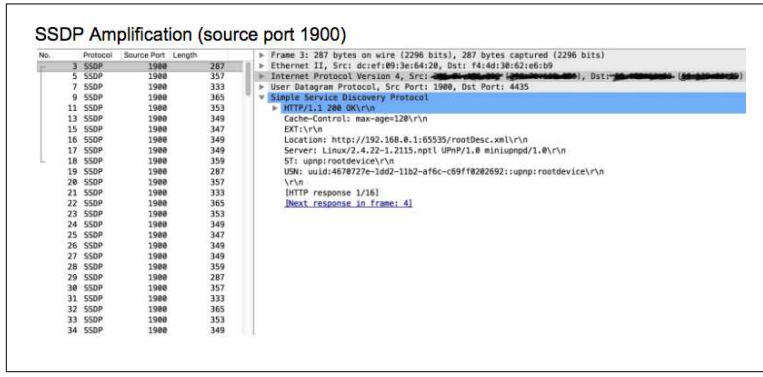

Figure 2-5. SSDP amplification packet

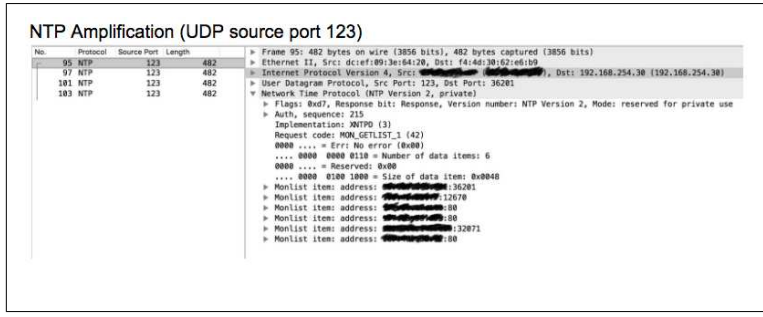

Figure 2-6. NTP amplification packet

While simple network mirrors are easy to construct, they are difficult to replicate in scale. With the advance of software defined networking (SDN), big data, machine learning, and cloud, we are seeing an increase of technologies that combine the three fields into an attractive DDoS detection mechanism:

- SDN, in the form of OpenFlow protocol (Figure 2-7), can offer two advantages over the traditional network in terms of monitoring and detection:
  - More precise matching of packets: as much as 15-tuple criteria of matching.
  - Once matched, the controller provides the mechanism to replicate traffic flow on demand without impacting the original flow.
- Big data technology provides a way to store and index data for efficient information gathering.
- Machine learning allows for an automatic self-learning cycle of the DDoS training set.
- Public and hybrid cloud provides a lower bar of entry for utilizing SDN, big data, and machine learning.

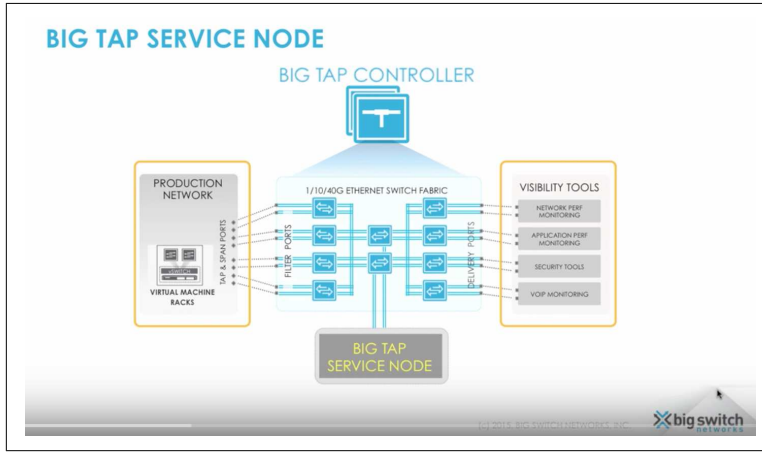

Figure 2-7. OpenFlow controller-based network monitoring (source: http://bit.ly/2FzuDp7)

It is worth pointing out that the technologies we have mentioned can be decoupled and used independently of each other. Another example of real-time packet inspection is shown in Figure 2-8.

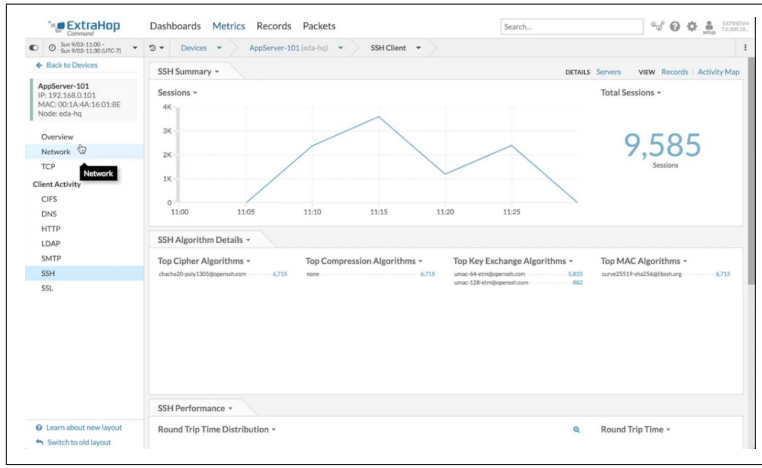

Figure 2-8. Packet inspection and reporting (source: http://bit.ly/2DStzjP)

With the rise of Bring Your Own Device (BYOD), we have users of the technology bringing their own device while utilizing the services, such as email, provided by the company. We have also seen a growing trend of host-based monitoring and detection in the marketplace both in commercial and open source projects. While they are great for detecting a breach of security, such as social engineering and compromised data breach, they are not as relevant for DDoS attacks. They can provide value in specific use cases when the agent is installed on a host that is under attack and we need to isolate the attacker and pattern. But in general, they are more useful in detecting other types of security breaches than DDoS detection.

Anomalies and Frequency-Based Detections

We are still in the early stage of machine learning, but it is already showing great promise in making detection of DDoS attacks easier.

If we take a step back and review the steps we normally take in detecting a DDoS attack, they typically include:

- Baseline our normal traffic usage, such as interface utilization level, requests per second, etc. This baselining needs to take into

account the normal fluctuation over the course of a day, quarter, and year.
- Detect any deviation from our defined normal usage. For example, in the SNMP section, we see a burst of traffic that is five times our normal usage.
- Further examination to see if the event was caused by a known event, such as an e-commerce site during a Black Friday sale, or if it was caused by DDoS attacks.
- If not caused by a known event, we will start to collect information and match against the well-known pattern of attacks, and decide mitigation action.
- Document the event for future reference and knowledge.

Many of the steps can be replaced by computers with machine learning capabilities. In fact, the computer is much better suited for the job because it can identify "needle in the haystack" types of anomalies much better than a human can. Elasticsearch is an open source technology that supports scalable, near-real-time search technology. Along with its sister projects Logstash and Kibana, sometimes referred to as the ELK stack, it is a great example of how machine learning can drastically help with DDoS detection.

We will use the following workflow as an illustration of the example:

1. Collect NetFlow, SNMP, and log information via Logstash input.
2. Normalize and augment data via Logstash filters and databases.
3. Output data to Elasticsearch for indexing.
4. Use machine learning x-pack to create a model baseline of data set, identify anomalies from baseline, and correlate influencers as the cause of outliers.

The example in Figure 2-9 shows a continuation of baselining traffic data.

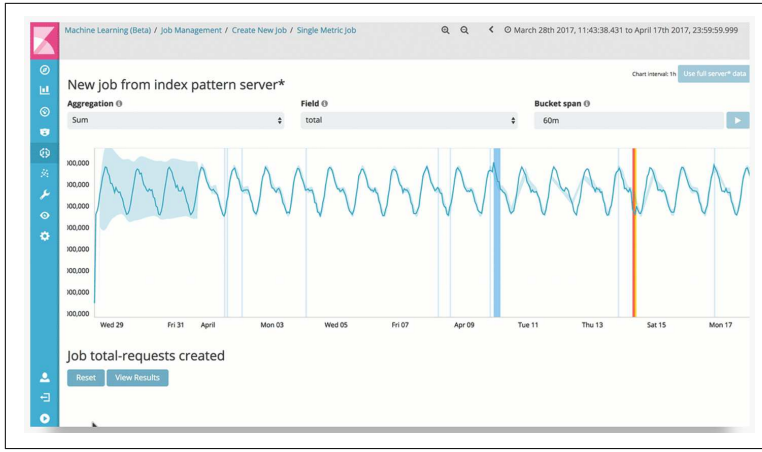

Figure 2-9. Modeling of data (source: http://bit.ly/2GDJuAu)

Once the baseline is determined, Figure 2-10 shows that an outlier can be identified.

Figure 2-10. Outlier identification (source: http://bit.ly/2GDJuAu)

A correlation of event to outcome can be guessed, as shown in Figure 2-11.

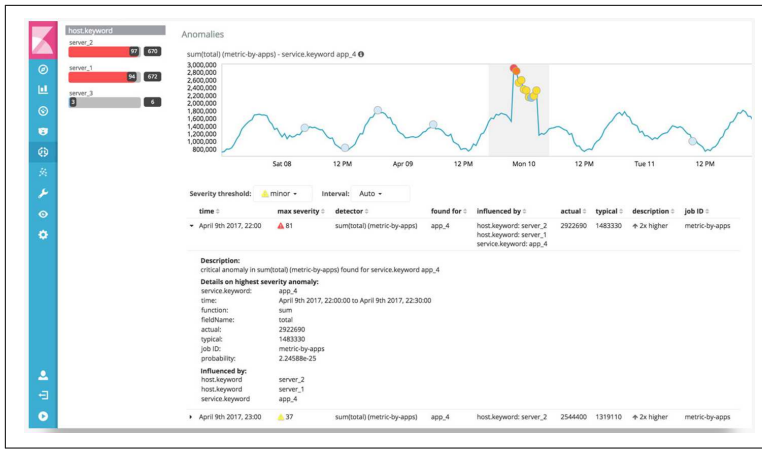

Figure 2-11. Influence of outlier (source: http://bit.ly/2GDJuAu)

The biggest gain from the workflow is a continuous baselining of traffic. Keep in mind that the first time an outlier event happens, even as a known event, it will generate an alert. A good example would be during the year-end holiday season when sales volume is expectedly higher than normal. If this is the first year the model is being built, a false positive alert will be generated. However, as time goes on, the model will become more accurate.

Another open source tool that has gained a lot of traction is Graylog. This is a more log-centric approach where you can centrally collect Syslog and event log messages and spot problems early.

Summary

In this chapter, we identified the various DDoS detection methods and mechanisms. We looked at SNMP and flow-based detection, as well as network mirrors and packet inspection. As we move into the world of machine learning, it is showing great promise in making DDoS detection easier and more autonomous.

In the next chapter, you will use the data we collected from the network and application and start to examine different types of mitigation and countermeasures against DDoS attacks.

CHAPTER 3

DDoS Mitigation and Countermeasures

We already know that the effects of a DDoS can be catastrophic for your service, business, and infrastructure. In the last chapter, we looked at various ways we can detect a potential or ongoing attack. In this chapter, we will explore ways to mitigate the attacks.

Even though we can detect the attack by macro or micro behavior, from our experiences, for mitigation, we need to dig into the low-level, nitty-gritty of the attack to devise a mitigation strategy. Like doctors who need to prescribe precise medicine based on the symptoms and predicted disease, the mitigation strategy needs to match the type of attack you are experiencing. A payload filter targeted to stop an HTTP GET flood, for example, will do no good to stop a TCP SYN flood.

Generally speaking, the DDoS attacks consist of the same type of exploit repeated over many times. For example, the TCP SYN Flood attack consists of one type of packet, TCP SYN, repeated from different sources arriving at your network over and over again. The challenge for mitigating the attack is in the volumetric and differentiation aspects of the attack. The mitigation consists of differentiating the legitimate request (in this case, TCP SYN) from the malicious sources, and doing so at an extremely high traffic rate.

Multivector Attacks

It is worth noting that we are seeing a rise of multivector attacks which combine multiple types of DDoS attacks. From a mitigation perspective, it is important to separate them out and mitigate them individually. The packets might arrive at your network edge simultaneously, but you need to treat them as if they are several separate attacks.

The options for DDoS mitigation are plentiful, and implementing the right solution against the exact attack at hand is key. We typically favor tools and features in the equipment common to all networks. Sometimes, however, you need higher performance, purpose-built DDoS mitigation systems. The value of these systems comes from their precision, visibility, learning, and deterministic performance.

The biggest question that you have to answer is how much collateral damage you are willing to take on. Like a lizard who discards its own tail in order to get away from a trap, when your entire network is down due to an ongoing DDoS attack, you might be willing to sacrifice part of your network in order to preserve other parts of your business. On the other hand, given the choice, the countermeasure should mitigate the attack with the least amount of impact.

Collateral Damage

It is often a hard pill to swallow, especially from a business perspective, to accept the fact that sometimes you need to make choices about which asset to protect while giving up other assets.

It is important to prioritize different sections of your business before the attack happens. For an e-commerce website, perhaps the search-and-order pipeline is driving your sales, and protecting the hosts responsible for that feature is more important than others. On the other hand, a nonprofit organization might place more emphasis on their landing page, which explains their mission statement.

The point is to prioritize as much as you can and get a consensus among your stakeholders within your organization.

To begin with, we should go through a few basics in the traffic flow of these common attacks such as floods, spoofing, and reflection. Having a good sense of the flow of traffic will then help us understand an appropriate deployment topology. We will discuss the general categories of mitigation techniques, including network and application mitigations. We will then apply the knowledge by diving into two of the most common DDoS attacks and their associated mitigations.

This is one of the most important chapters of this book. In a sense, we are all here to learn about how to stop DDoS attacks when they happen. Without further ado, let's look at the DDoS terms and traffic flow.

DDoS Terms and Traffic Flow

Before moving on, we should examine some of the most common types of DDoS traffic flow and terms. They will help us understand the more complex attacks covered in later sections. We briefly covered spoofing and reflection in Chapters 1 and 2; here we will review them in more depth, as well as introduce new concepts.

Traffic Flood

As we covered previously, traffic floods consist of attacks that consume resources such as bandwidth and packet processing capacity. If you imagine an internet connection as being a water pipe and the traffic being water inside of the pipe, the flood of traffic will be a momentary burst of water that fills up the whole water pipe.

One might ask the question, "Why not just get a bigger pipe?" It is true that the problem can be mitigated at this point in time by adding capacity, but that solution will not scale as attacks grow in size. Please also keep in mind that there is a monetary cost to adding this additional capacity. If this capacity is merely "attack insurance" then it is more challenging to justify.

Source Spoofing

While IP source spoofing is not an attack on its own, it is an important concept to understand. As explained by Wikipedia:

> In computer networking, *IP address spoofing* or *IP spoofing* is the creation of Internet Protocol (IP) packets with a false source IP

address, for the purpose of hiding the identity of the sender or impersonating another computing system.

An attacker can spoof the source address of the attack when connected to an ISP or a provider that allows this. "How can the ISPs be so careless and trusting to allow spoofed IPs?", you ask. Well, if you go back to the early days of the internet, it was a wide-area network connecting local academic and research networks that were mainly trustworthy. Therefore, the basic design of the IP protocol and infrastructure do not generally take into consideration the fact that some malicious user can create fake source IP address for the purpose of attacking others.

Checking Spoof IP Address at the ISP Level

ISPs are increasingly checking for spoofed IP addresses in their network. However, from our experience, the majority of ISPs still do not do this. The issue is the overhead associated with doing this extra layer of checking, both in terms of hardware and staff resources for maintaining such configuration. Imagine a router trying to route packets as fast as it can; by checking only the destination IP instead of both source and destination IP, it can increase its packets-routed-per-second performance.

However, ISPs are increasingly finding out that by preventing spoof IPs they can save money in the long term by decreasing the number of DDoS attacks overall. One collective effort is the BCP38/RFC2827 for network ingress filtering.

It is important to point out that when sending traffic from spoofed addresses, the attackers have no intention of receiving a response. We can use this fact to our advantage when we try to identify spoofed IPs and mitigate against the attacks.

Reflection and Amplification

The mechanics of reflection relies on a system to source a reply from the reflection point to direct the response to the spoofed source. If you recall our favorite pizza shop example from Chapter 1 (Figure 1-6), our bad guy friend is pretending to be Mike (spoofed source) and calling the pizza shop (reflection point). If successful, Mike and the pizza shop are both victims, with Mike sustaining a

loss of productivity and the pizza shop potentially losing money and resources.

If we apply the same concept in the digital world, when we have a UDP-based service that does not validate source IP, coupled with an attacker's ability to spoof IP, they form the basis of a large-scale DDoS attack. We have already seen an example of the UDP reflection flow in Figure 1-7 back in Chapter 1.

In Figure 3-1, we can see that the majority of the flood attacks consist of amplification floods.

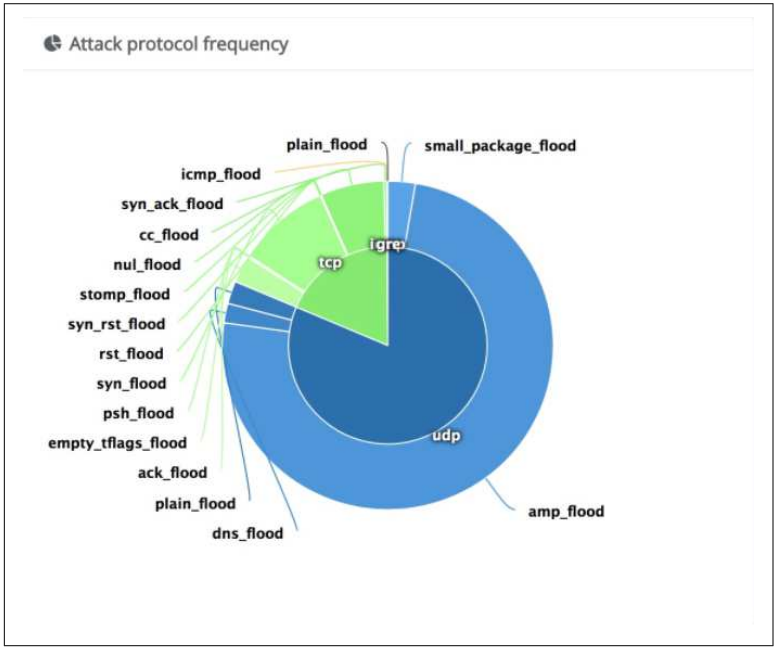

Figure 3-1. Attack protocol frequency (source: http://ddosmon.net/insight)

Digging into amplification as a method, we can break down the popularity of different protocols used. Each has its own qualities that make it more enticing to the attacker. In Figure 3-2, the graphic gives us an idea of the popularities of amplification protocols as of this writing. DNS makes up half of the amplification attack traffic seen on the internet, with NTP, CLDAP, Chargen, and SSDP following close behind.

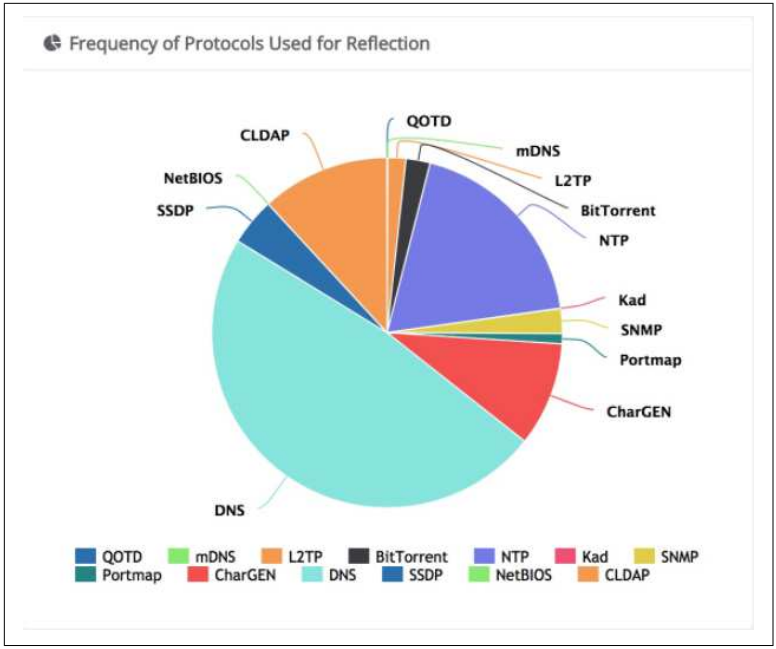

Figure 3-2. Frequency of protocols used for reflection (source: http://ddosmon.net/insight)

This amplification and reflection would be particularly damaging when a small request produces a large response. There are lots of well-documented attacks using this reflection and amplification method with DNS, NTP, and SNMP.

DDoS Mitigation Topology

Now that we understand a bit about the DDoS traffic flow and terminology, we can assess our general approach toward mitigation. The first mitigation approach we can take is to build our own mitigation solution. With this approach, we take full control of evaluating the equipment, setting it up, and making sure it does its job.

Of course, in today's world of cloud computing and anything-as-a-service, another approach is to utilize the various cloud-based DDoS mitigation providers. This will generally be pay-as-you-go at the expense of lesser control and potentially higher-cost per mitigation. If you do decide to take this approach and outsource your DDoS mitigation to a cloud-based provider, it is worthwhile to use this

chapter and understand the mitigation technology. After all, the cloud-based mitigator uses similar technologies to perform mitigation. They just rent the capacity out when the customers need them.

For both on-premise and cloud approaches, or a hybrid approach, you should examine existing topology and pressure points where mitigation can be applied. In this chapter, we will focus on mitigation and discuss in detail cloud-based DDoS mitigation in Chapter 4.

Reactive Versus Proactive Always-On Mitigation

The first driver for your mitigation topology is to determine if you want a reactive or proactive always-on approach. The main trade-off is the infrastructure cost. Your network and system's first job is to accomplish your business goals, whether that is to attract business leads, communicate your mission, or to sell items online. In order to accomplish that, your infrastructure needs to run as lean as possible. If you were to place an always-on mitigation solution by watching the network traffic at all times, this would translate to additional overhead and cost. Since this device is the gatekeeper, it needs to be provisioned with as much capacity as your network, with the same availability goals; otherwise, it will become a bottleneck.

In a proactive always-on mitigation, your mitigation device might be a purpose-built device that watches over all the traffic. This device or layers of devices will need to be placed in between the external and internal network to prevent external threats.

On the other hand, if you were to choose a reactive mitigation, you might be able to scale out the solution better. You can use routing protocols to divert traffic toward the mitigation layer only when needed. Since the mitigation devices are only used when needed, you only need capacity that can sustain the attack volume and not necessarily your overall network capacity.

The downside of a reactive mitigation compared to a proactive model might be the added complexity and additional delay introduced. The added complexity of routing requires additional planning, configuration, and knowledgeable staff. The delay occurs between the time we detect an attack and the time it takes for routing to kick in and onramp the traffic toward the mitigation device. This delay is also realized as service impacting if the attack is large enough to overwhelm the target resources.

Potential Points of Attack Mitigation

Regardless of whether the mitigation approach is proactive or reactive, it makes sense to place the DDoS mitigation devices as close to the attack source as possible. You do not need to carry all the "dirty" traffic across your network just to drop them deep within your network, where other scale problems could come to light, making it a waste of network resources. Therefore it is beneficial to drop all the traffic you need to drop as early as possible. In a typical network where the attacks are coming in from the internet, the mitigation should be as close to the edge as possible.

DDoS Mitigation Outside Your Network

With so many services hosted outside of traditional network boundaries, such as in the cloud, it is a good idea to define the term "network" as befits your network. For example, the edge border might include the virtual router device you deployed in your cloud virtual private network.

In Figure 3-3, we are placing a dedicated, purpose-built, reactive mitigation system between the internet edge and the core portion of the network. When you need to mitigate, the traffic is redirected toward the mitigation systems, and clean traffic is passed back to the network.

If you utilize cloud-based mitigation, the redirection will almost always take place outside your network border. The traffic is redirected via DNS or BGP before they ever reach your border device, and clean traffic is returned back to your premise via physical or virtual tunnel links. In this case, the mitigation point is outside your organization.

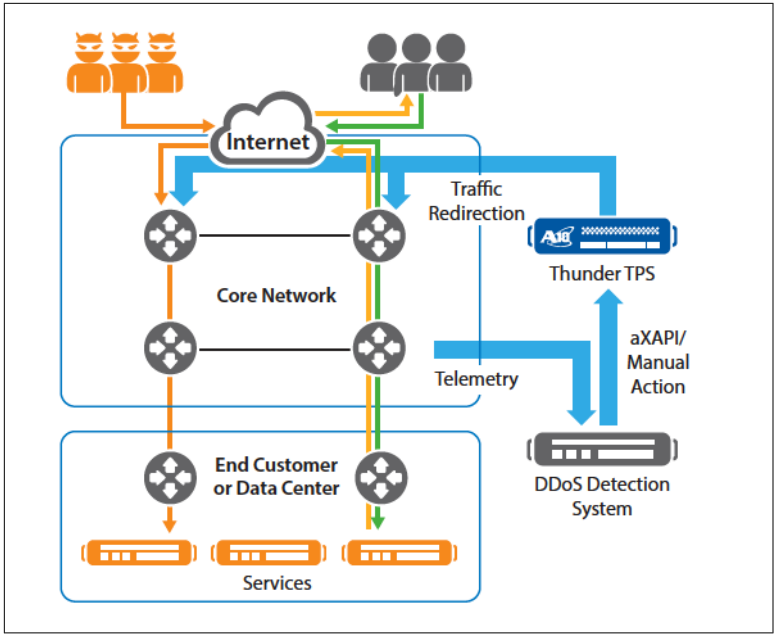

Figure 3-3. Asymmetric reactive mitigation (source: http://bit.ly/2DW99GD)

It is worth noting that the actual mitigation technique could be a feature on your existing devices such as IPTables or an Access List instead of dedicated scrubbing hardware. If facing a choice of where to turn on mitigation, the same general approach should be taken. The mitigation feature should be enabled as close to the internet edge as possible.

Effective DDoS attack mitigation involves more than just one control and data plane tool. In most infrastructure, there are tools already at your disposal that you should take advantage of. In other cases, a dedicated mitigation device should be deployed. The key is to find the right tool to take care of the right attack. Let's explore some common tools in the following sections.

Network-Level Mitigation Tools

Your network consists of routers and switches that connect your hosts together. Switches and routers can be extremely powerful systems to deal with volumetric attacks. Mitigation at the network layer, compared to higher layers such as the application or presenta-

tion layer, is faster. Mitigating an application is done in software and therefore performance takes a hit. These dataplane units generally employ forwarding application-specific integrated circuit (ASIC) with connected ternary content-addressable memory (TCAM) that are capable of filtering at line rate.

Common protocol anomalies are built into some network device forwarding ASICs. These per-packet anomalies include IP protocol version or invalid TCP flags. The good news is these features are built in and you do not need to do anything to enable them; the bad news is you might need to check with the equipment vendor on which anomalies are being checked.

However, the line-rate switching benefit of the ASICs with TCAM is also one of the downsides when it comes to DDoS mitigation. Anything done completely in hardware is often not particularly flexible and difficult to get telemetry from. Also, switches and routers do not track protocol state. This makes mitigating anything that requires state, such as session exhaustion, impossible.

At the network level, we can configure access lists to pass, drop, and in some cases rate-limit traffic. In most cases, your routers and switches already have access lists as a feature to be used for dropping traffic if needed. You want to pay close attention to the resource limitation on your network equipment, such as the number of access lists it can handle. The limitation is not as straightforward as a hard set of numbers when you take into account IPv6 versus IPv4 access lists, extended versus standard access lists, as well as the various TCP flags that you can configure your access lists to inspect.

A common method for dealing with DDoS attacks is through methods employed in IP routing. As standard IP routing is destination-based, we can use routing to route the packets toward a destination of null, the routing equivalent of a black hole. If BGP is involved, we can use Remote Triggered Blackhole (RTBH) to remotely signal our upstream router to route the particular destination into a NULL route. As network vendors would tell you, they have no problem dropping as many packets as you would like (joking!).

NOTE

BGP FlowSpec

Standard BGP limits you to policy based on IP addresses alone. With BGP Flowspec defined in RFC 5575 we gain the ability to influence behavior based on a much broader set of criteria. We can match up more fields supported by BGP Flowspec (source and destination, IP protocol, source and destination port, ICMP code, and TCP Flags), as well as more dynamic actions such as drop or rate-limit.

Unfortunately, BGP Flowspec is not supported by all providers. You also need a device that is Flowspec-capable, which tends to be higher-priced, more feature-rich routers.

RTBH creates a large amount of collateral damage. In this case, you are trying to reduce the blast radius of the attack by blackholing the IP address that is being attacked. In theory, the rest of the IPs that you are using are not going to have to take the hit. In a way, the attacker wins in this scenario because you are giving up on the attacked IP. Along the same lines, you can also use IP routing to announce a prefix toward a device or interface that has uRPF enabled. This allows you to programmatically add source IPs to drop. Being more surgical with BGP requires Flowspec as we stated previously.

One of the latest trends in networking is the rise of software defined networking (SDN). One aspect of SDN, such as OpenFlow, is the separation of control and data plane. One of the benefits of the separation, as applicable to DDoS, is the ability to gain greater insights into OSI Model Layer 4 properties and to filter accordingly without losing the switching performance.

Session-Level Mitigation Tools

In many cases, attacks are meant to break a particular application or service by exhausting the resources that keep application-state information, such as the number of sessions. State exhaustion here refers to any attack that is attempting to occupy and hold open resources on your service. For example, a TCP stack may hold open resources after an SYN-ACK is sent in response to an attacker's SYN.

Session exhaustion can sometimes be used in conjunction with the flood attacks. In the previous example of the TCP SYN flood attack, it is effectively an attack that targets both packet processing and state in your infrastructure. It will achieve its goal of bringing down your resource if either method succeeds.

Let us take a look at the tools you can use to mitigate session-level attacks. The common ones are firewalls, application delivery controllers, hosts, and purpose-built mitigation devices. We will look at each of the tools as a standalone device, but keep in mind that they can and should be used in conjunction with each other in a tiered approach.

In most networks, stateful firewalls are typically used alongside routers and switches to inspect traffic that requires higher fidelity. The value of firewalls generally is their ability to maintain state and apply interesting policy down to the source/destination pair. Because of this sometimes complex set of policies, the firewall is normally constrained by packet-per-second and memory. Therefore, firewalls can mitigate DDoS attacks to a certain point, but when the limits are reached, they became the bottleneck in any DDoS mitigation strategy.

Some firewalls have higher hardware capacities, but the fact of the matter is that if you are learning every source IP, an attack that includes spoofing will almost always overload the firewall. With IoT-based botnets, we are seeing more stateful attacks that are based on real public IP connections, so tracking the sessions would be even more taxing for the firewalls.

Application delivery controller (ADC; sometimes known as load balancers) have both stateful and stateless features. However, to get the most value out of an ADC, the consumer would need to enable the stateful features and even add some OSI Layer 7 scripting. All of the additional features will reduce the performance of the ADC.

Of course, the most important and vulnerable link in this chain is your host. The host has the highest visibility in terms of application awareness, so they have the most knobs and granular control you can utilize. However, in a typical data center, you can sometimes have hundreds of thousands of hosts; managing and fine-tuning all of them could be a challenge.

Purpose Built DDoS Mitigation Devices

These systems are the most flexible but carry a higher price tag than their network device peers. They can provide both stateless and stateful functions. Since they are purpose-built, they are focused and effective in dealing with most of the common DDoS attacks we have seen in this book. However, the downside is they are yet-another-device that you have to manage and another skill set you have to staff for. Also, due to the high price tag, they are generally reserved for entities who can justify the dollars spent.

In the following sections, we will apply the theories we have learned so far in this chapter and apply them toward mitigating two of the most common DDoS attacks: combating the classic flood and combating the state exhaustion scenario.

Example 1: Combating the Classic Flood

By now, we have already learned that one of the most common DDoS attacks is to congest your network. This congestion can happen at your internet egress or at some other bottleneck in your network; if you recall, what matters most is the weakest link in your end-to-end connection from customer to the resources you are providing. The pre-mitigation step against these flooding scenarios, as with any other mitigation techniques, requires you to understand what your current capacities are. This can be your bandwidth capacity and packets-per-second capabilities. This information will be matched to the flood level you are observing, at which point you need to initiate the various mitigation tools you have. Let's dig deeper into the problem, understand the toolbox, and review a few potential solutions depending on your situation.

NOTE

Note About This Example

We will dive into more depth and detail in this example to illustrate our points with graphs and data points. We will go from attack formation to analyzing the data, and from there we will form our mitigation response.

Please feel free to skip to the next example if you feel you already have a firm grasp of the information at hand.

Imagine you are the potential DDoS attacker who wishes to flood your target, Rich's Bank's network pipe. What is the first question the would-be attacker wants to ask? Probably how big your network's internet connection is. How easy is it to find the answer to this question? Sometimes a simple traceroute from publicly available service is all it takes (Example 3-1):

Example 3-1. Traceroute from a public cloud provider to Rich's Bank

```
rg:~ ddosresearch$ traceroute richs-bank.com

traceroute to richsbank.com (10.20.30.40), 64 hops max,
52 byte packets

7  xe-3-1-1.cr0-protect.ip4.you.net (1.1.1.1)
120.828 ms  120.228 ms  124.684 ms
```

From the traceroute output, an experienced attacker would be able to safely guess that the internet connection is likely a Juniper router with a 10-gigabit Ethernet interface (xe-3-1-1). Furthermore, since this traceroute was completed from multiple public IP addresses at different cloud providers with the same output, the attacker can assume with high certainty that Rich's Bank is indeed behind a single 10-gigabit interface. The attacker now can have a goal to generate 11 gigabits of traffic per second toward the target and the legitimate requests will get dropped.

The next step for the attacker would be to figure out how to generate more than 10 gigabits of traffic toward the target. If the attacker follows the traditional path of buying internet connections from service providers, say 11 x 1Gb connections, and launch the attack, the attack would not last very long or successful. Why? Because it would be pretty easy to block only 11 different sources for Rich's Bank, not to mention the attacker's ISP would be legally required to terminate his or her service in most parts of the world.

You might be quick to point out that the attacker can use the UDP amplification method that we discussed earlier—and you would be absolutely correct! Recall from earlier in the chapter, the amplification method is the most common technique used in flood attacks when the majority of the attack consists of DNS reflection.

In Figure 3-4, we dig a bit deeper into the amplification factor of the various protocols.

Protocol	Bandwidth Amplification Factor	Vulnerable Command
DNS	28 to 54	see: TA13-088A [4]
NTP	556.9	see: TA14-013A [5]
SNMPv2	6.3	GetBulk request
NetBIOS	3.8	Name resolution
SSDP	30.8	SEARCH request
CharGEN	358.8	Character generation request

Figure 3-4. Amplification bandwidth factor (source: http://bit.ly/2s4HFcD)

What you can see is DNS has a bandwidth amplification factor of up to 54x while NTP is 556x! This is quite a large difference. Would our attacker pick NTP amplification over DNS? Not necessarily. The complexity involved in identifying and dropping attacks under DNS amplification is relatively similar to any legitimate response to DNS queries. On the other hand, it is easy to block off-net NTP traffic. Our would-be attacker would use the DNS as the amplification and reflection attack of choice.

NOTE

The Discoveries of Amplification Points

You might be wondering how the attacker finds these amplifiers. Generally, they are discovered by scanning the internet and executing the exact query to test for amplification.

The simple workflow of the DNS amplification attack is as follows: the attacker identifies the attack target, spoofs the IP address of the target, and makes a DNS Request to the amplifier. The amplifier then responds with many times the packet from the requests to the attack target. This manifests as a large number of DNS responses from potentially millions of endpoints around the globe.

Analyzing the Attack

Let us take a look at a snapshot of a DNS amplification attack packet in Figure 3-5 as seen by the attack target for the purpose of device a mitigation strategy.

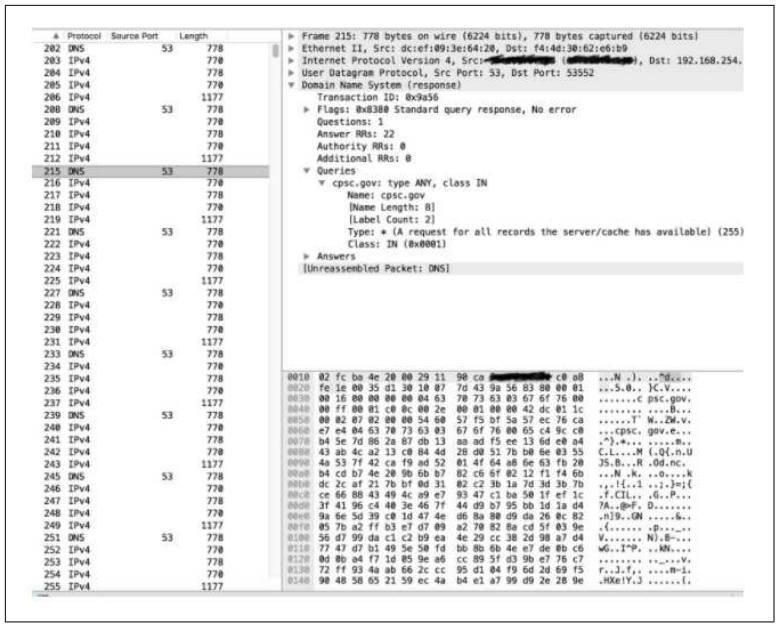

Figure 3-5. DNS amplification packet

A few key characteristics can be identified:

1. All of the packets in question are from UDP source port 53 (normal for DNS responses) with a trailing set of IP fragments.
2. All packets are large for UDP DNS (not necessarily bad).
3. The DNS responses are for an ANY query.
4. There are a very large number of answers.
5. Each response is for the same domain name. The domain name itself is unimportant. The attacker has found one that behaves to their liking and they use it as a packet generator.

Let us zoom out and look at the packet distribution in this "lag" attack that was sent in pulses with a variety of attacks in Figure 3-6. The green line (top line) in this case, represents the DNS responses with fully formed DNS headers. The blue line (second line) shows all of the trailing IP fragments. The red line (third line) is the attack target server, which is not listening on any destination ports. It is sending back ICMP port unreachable messages in 100 ms increments.

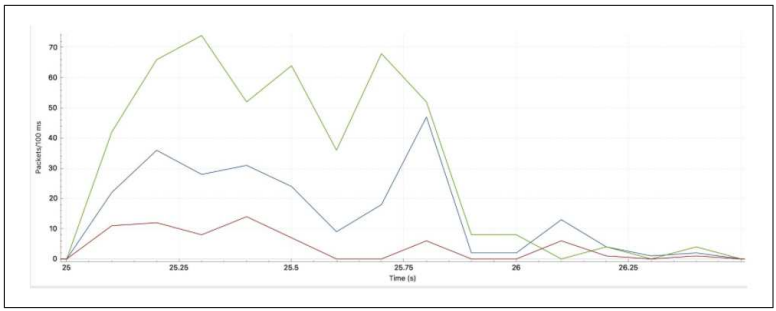

Figure 3-6. Packet distribution of attack

From the packet trace and packet distribution analysis, we can see that the DNS behavior is not what you would expect. If the DNS ANY response is large, the DNS server should have switched the conversation from UDP to TCP. To successfully carry out the attack, the whole transaction must be in UDP and use an open DNS resolver that supports extension mechanisms for DNS or EDNS0.

We are now ready to construct our mitigation strategy.

Mitigation Strategy

To successfully mitigate this attack, it is important to understand if it is your internet edge that is saturated or if it is a bottleneck that leads to a service that you are protecting. Protecting the internet edge in this scenario is challenging with on-premise techniques. If you have a 10-Gb per-second internet connection, and there is greater than 10-Gb per-second traffic destined to your company, there is little you can do with on-premise equipment.

Let us explore some of the options on hand:

1. We can try to reduce the dependency on off-net DNS transactions.

   This is probably easier said than done. The reason that this is imperative is that these are DNS responses, and most of our legitimate DNS responses will look similar to the attack aside from the fact that you may have responses from millions of previously unknown DNS servers
2. You can drop or rate-limit UDP source port 53 at the internet edge.

This is a reasonable measure especially if you do it during attack time as a reactive measure. To aid this effort, perhaps you can have an off-net DNS server that is explicitly allowed but rate limited. To drop UDP source port 53 traffic, you have a few options:

- Call your upstream provider and tell them to drop source port 53 during the attack.
- Make the list of source ports to drop more dynamic using BGP Flowspec, if your provider supports such feature, so the blocked list is only constructed during time of attack.

3. Blackhole

   You can instruct your upstream provider to blackhole or null route the attacked destination. This is probably the last option resort, as you are letting the attacker take out the host/device/service that is under attack; however, it may be more important to save the rest of the hosts in the network.

Let us take a look at a second example of combating the state exhaustion DDoS attacks.

Example 2: Combating State Exhaustion

Bandwidth and packet saturation isn't the only type of attack that can bring down your service. Even when network monitoring tools show everything in the green, you can be the victim of an attack that is focused on state tracking in your infrastructure. It's important to understand your entire path from the internet to service and back again to see where these sorts of problems can occur.

State exhaustion here refers to any attack that is attempting to occupy and hold open resources on your service. For example, a TCP stack may hold open resources after a SYN-ACK is sent in response to an attacker's SYN. This is a base premise of the SYN flood and why it is effectively both as an attack that targets packet processing as well as state exhaustion in your infrastructure. When the attacker is spoofing an entire internet worth of sources, you can see how any stateful system might run out of resources pretty quickly.

Hosts are not alone in their ability to fall victim to this. Firewalls, application delivery controllers, and other devices that employ some stateful inspection or translation have similar limitations. Unfortunately, a session exhaustion attack is not easy to mitigate reactively, especially if your goal is to maintain high uptime. Some preparation to hardening your service must happen. One such attack we will dig deeper into here is called Slowloris.

Attack Dynamics and Analysis

According to Wikipedia:

> Slowloris tries to keep many connections to the target web server open and hold them open as long as possible. It accomplishes this by opening connections to the target web server and sending a partial request. Periodically, it will send subsequent HTTP headers, adding to—but never completing—the request. Affected servers will keep these connections open, filling their maximum concurrent connection pool, eventually denying additional connection attempts from clients.

In Figure 3-7, we have dissected the packets coming in from a Slowloris attack.

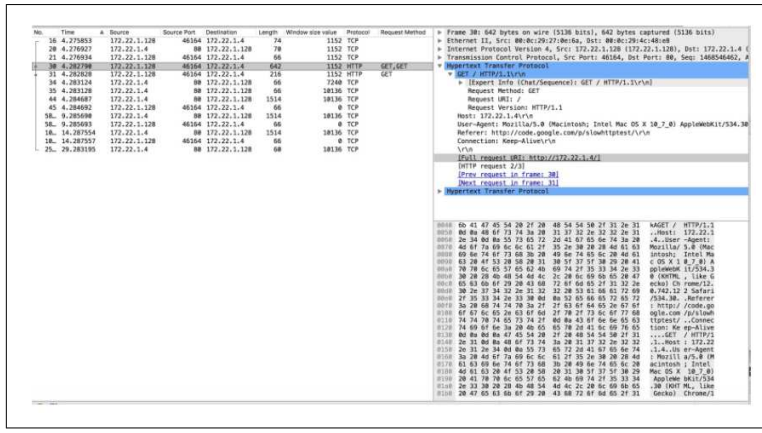

Figure 3-7. Slowloris packet capture

What you see here is a session generated by slowhttptest. It is important to notice a few properties of the attack:

- The length of the entire session is 30 seconds due to the RST sent by the server. The attacker could have kept the session open

for much longer if this didn't happen. This shows the intent of keeping sessions open for quite a while.

- The TCP window size of the attacker is quite small but not unreasonably small at first, especially considering windows advertised by mobile devices.
- The page that is being served by the target is just the Apache default page. However, the transmission is stretched across many packets at a slow interval due in part to the small window size.
- The attacker's advertised window hits 0 and stays there after the 4-seconds mark.

The effect in the test environment of one attacker versus one server is pretty telling. Notice the behavior of Apache after 20 seconds of slowhttptest traffic in Figure 3-8.

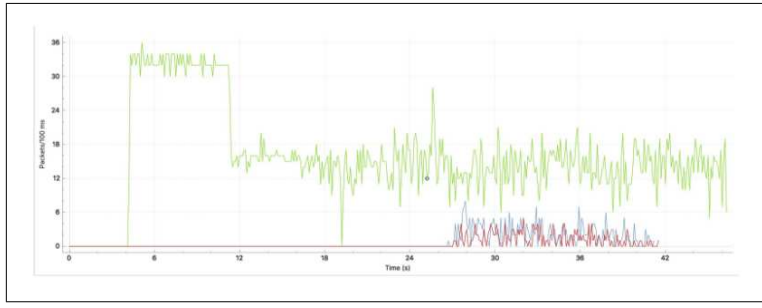

Figure 3-8. SlowLoris packets per second

The green line (top line) shows our attacker sending HTTP GETs to the server. The server reached its own threshold at approximately 12 seconds into the test. At approximately 27 seconds, we see the effects clearly as out-of-order packets (represented in red) increase and Apache sends a 504 code for as many GET requests as it can handle. Finally, Apache dies at 41 seconds while the tool continues to run.

Let us devise a mitigation strategy appropriate for the session exhaustion attack.

Mitigation Strategy

We will focus on using the host as a mitigation strategy. In our scenario, Apache has a few features that can be tweaked to help with this type of attack.

The mod_reqtimeout Apache module, shown in Example 3-2, has a few options that give you insight into how it may help with the attack.

Example 3-2. mod_reqtimeout (source: http://bit.ly/2GBistv)

```
<IfModule mod_reqtimeout.c>

  RequestReadTimeout header=20-40,MinRate=500 body=20-40,MinRate=500

</IfModule>
```

This configuration instructs Apache to wait for 20 seconds to complete the HTTP header transfer. If the client maintains sending the header at 500-bytes per second, then the server will wait for up to 40 seconds for the header to complete.

The purpose of modQOS is to add some constraints around server resources and the types of clients that you wish to serve. A few configuration options of note that are helpful for this attack follow:

QS_SrvMaxConnPerIP
: Slowloris does not require a large number of sources to be effective. In fact, it is one of the selling points of the attack that an attacker could take down a server with a low-powered PC. This feature places a ceiling on the number of connections established per source IP address.

QS_SrvMinDataRate
: Slowloris achieves its goal primarily by launching many slow connections and trying to maintain them for as long as possible. This option allows the server to drop a connection based on a minimum speed

You can find out more information on mod-qos at *http://mod-qos.sourceforge.net/*.

Emulate DDoS Attacks for Better Response

Needless to say, it is extremely stressful when you are under a DDoS attack. You are racing against the clock to determine the type of attack and the right mitigation approach, and to carry out the implementation. These steps are often done under the nervous eyes and breath of the service owner and business managers.

To avoid any surprises during the actual attack, you should use attack emulation as we have done in this chapter with slowhttptest. You can also hire other legitimate stress testers to emulate DDoS attacks and do a dry run for your mitigation strategy. This will prepare the staff so they can practice running the standard operating procedure during peacetime.

Hping3 is a high-level tool that can carry out a variety of penetration testing, including small-scale DDoS attacks. You can find more Hping3 examples on the Hping3 website.

Scapy is another open source tool written in Python that can craft packets from the ground up. As you have total control over your packet header and payload, you are able to do a lot of fuzzing with Scapy. You can find more Scapy examples on our GitHub repository.

Summary

We covered a lot of ground in this chapter. We started out by looking at the general mindset and approach for DDoS mitigation, defining the terms and traffic flow of some of the common DDoS attacks, and DDoS mitigation topologies.

We also looked at the network- and application-level mitigation techniques that we can use, before combining what we learned and studying how to combat classic flood and state exhaustion attacks.

In the next chapter, you will learn to evaluate different cloud-based mitigation vendors.

CHAPTER 4

Evaluating Cloud-Based Mitigation Vendors

We live in the world where cloud computing, essentially rented computing capacity, is commonplace. Vendors such as Amazon Web Services (AWS) and Microsoft Azure allow you to utilize their computing power without building your own. Among the broad umbrella of cloud computing services, there are subcategories such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

In this chapter, we will focus on DDoS mitigation vendors who fall under the SaaS model, where they offer their software as a service, often charging a license fee to start and a metered usage fee when you use their services. As active DDoS practitioners, we are familiar with current vendor brands and offerings in the marketplace today. However, we want to focus on the technologies and features instead of any particular vendor brands or their "secret sauce."

In this chapter, we want to answer the question of whether to build your own on-premise DDoS solution, buy the service from a cloud-based provider, or do both. By understanding the pros and cons of using a cloud-based DDoS mitigation provider, you can start to reflect back to your own network and conclude with your own answer to the build-versus-buy question.

Focus on On-Premise Versus Cloud

In previous chapters, we discussed the DDoS detection and mitigation techniques available once you have the traffic data. This chapter will place the focus on the infrastructure differences between building an on-premise solution versus outsourcing to a cloud-based provider. Once the traffic has been shifted from your network to cloud provider, the techniques used to detect the particular type of attack will be similar.

We will dig deeper into the operational model and techniques of a cloud DDoS scrubbing center. The chapter will conclude with an evaluation checklist derived from the topics we covered.

Why Use Cloud-Based DDoS Mitigation?

The advantages of using a cloud-based DDoS mitigation solution are very similar to the reasons you would use a cloud-based solution for your infrastructure. Some of the advantages, such as lowered cost and faster time-to-build, are self-evident. However, there are additional advantages that are specific to cloud-based DDoS mitigation solutions, such as real-time updated attack patterns. Let's take a look at the main advantages of having a cloud-based DDoS mitigation solution.

Overall Cost Savings

Let's face it—like most infrastructure components, building an effective DDoS mitigation solution can be complicated and entail upfront investment. The investment can be in the form of time, money, and knowledge. After all, not only does the company need to buy hardware, but they also need to train staff to operate the hardware, set up the protection parameter, and constantly update and adjust to the state of their infrastructure.

Two of the often overlooked costs of setting up an on-premise DDoS mitigation solution are the average cost per mitigation and the cost of solution upkeep. Not all companies are targeted the same way by attackers. Generally speaking, the bigger the company, the more attacks and attack varieties they receive. But that is not always the case—for example, a small online radio station advocating oppo-

sition views to a dictatorship country elsewhere can often be a target of state-sponsored DDoS attacks. It stands to reason that the more you experience attacks, the less money you spend per attack from your investment. The cost savings of build versus rent should be compared against the level and sophistication of the DDoS attacks that you receive.

The cost of upkeep is also another hidden cost that may be overlooked. For example, if your on-premise solution uses a blacklist of IP addresses, there is a certain upkeep cost of keeping the list up-to-date. With the size of attacks getting bigger by the month, hardware often needs to be refreshed and adjusted to accommodate bigger attacks.

All of the costs mentioned here mostly exist regardless of whether it is on-premise or in the cloud. However, the cloud mitigation provider can often aggregate the demands from several customers, which results in lower cost per customer. The customer does not need to keep an always-on solution and is charged based on usage. This might speak well for companies that favor variable operation cost instead of upfront fixed cost.

Proven Operating Procedure and Knowledge

As discussed, not all companies are DDoS attacked at the same frequency or size. If the attacker is motivated by ransom money, it makes sense to attack a more established company. On the other hand, smaller companies often do not experience DDoS attacks unless triggered by an event. A disgruntled former employee who decides to spend a few dollar to attack his or her former employer is not unheard of, but will likely catch the on-call engineer off guard due to its rarity.

An IT department might still want to be prepared for so-called Black Swan events but cannot afford to invest in the ever-changing landscape of long-tailed events. The cloud provider, in this case, can give the customer a proven standard operating procedure (SOP) for each of these events and provide a guiding hand. This is especially true for enterprises that need to support a variety of technologies but on a smaller scale. They simply do not have the manpower or resources to go deep into a particular technology vertical.

More Network Visibility and Fewer Bottlenecks

The internet is a giant, enormous web of connected computers and networks. The inter- and intra-network connections vary in size: some interconnections are big whereas others are small, ranging from tens to hundreds of gigabits per second in the core to kilobits per second on the edge. As a network administrator, you will have control over your own network but the other networks on the internet are outside of your control.

The DDoS attacks that exhaust network bandwidths take advantage of the fact that enterprise or small service providers have to be connected to other networks but often have only a limited amount of exit points. How well you defend against these type of attacks depends on whether you can stop the attack closest to the source and how well you can balance your incoming traffic.

For example, a common practice for large-scale networks is to have a presence at strategic locations called internet exchange points. Even though these network administrators have no control over other networks, they are able to have enough exposure at these exchange points to make better judgments about the source of the attacks and therefore keep the network resources available. The cloud-based mitigation providers are usually better positioned to be at such internet exchange points than typical enterprise and smaller network operators.

The network footprint is such an important factor in DDoS mitigation that the cloud-based providers often list their presence at the various exchange points. Another way to think of it is: if your public exposure surface is big, then it takes more attacks to cover all of the surfaces.

Dedicated Staff and Better Reaction Time

Because the core competency of the cloud-based mitigation provider is indeed DDoS mitigation, they can dedicate more resources to the effort. As many business owners can tell you, one of the most expensive resources is human capital. A well-trained, experienced engineer who is well versed in different DDoS attack pattern is worth their weight in gold. The cloud-based mitigation provider can often afford to have many experienced engineers in-house.

The providers can oftentimes provide dedicated teams to monitor the internet DDoS weather by investigating darknet activities, and closely monitoring newly disclosed vulnerabilities and any network anomalies. These actions can lead to faster reaction times when the DDoS happens and eventually less time to restore services.

It cannot be stressed enough that the biggest value cloud providers bring to the table is the aggregation ability to spread the cost amongst different entities. This directly results in lower fixed and variable cost per attack, better operational maturity, quicker responses, and more experience in defending against DDoS attacks.

When Not to Use Cloud-Based DDoS Mitigation

There are many advantages to using a cloud-based DDoS mitigation provider, as we have seen in the previous section. In this section, we will look at the other side of the coin and examine some of the reasons you might not want to utilize a cloud-based solution.

Many of the differences boil down to a rent-versus-own comparison. When you rent something, typically you pay a fee for the right to use the item for a limited period of time, with limited control over the item. However, when you own the item, you are able to have full control and modification rights. You are able to modify and tweak the solution to be 100% compatible with the rest of your infrastructure.

Let's look at the reasons you would not want to use a cloud-based DDoS solution in more detail.

Control

The biggest reason to build your own on-premise DDoS mitigation solution might come down to control. If you have the solution on premise, you have total control as opposed to handing over control to the vendor. The control is both in terms of giving up some of the control over your own network as well as lack of control over the vendor's mitigation strategy. Much like using a cloud provider for your compute and storage needs, if you utilize cloud-based DDoS mitigation, you are extending your network to an outside vendor. You will need to be comfortable with the amount of control you are giving up based on the type of mitigation technique.

Trusting Your Vendor

It is worth stressing one more time that, if choosing to go with a cloud provider, your service availability is in their hands. This point is sometimes missed in the minds of many because this is not seen until a breach event.

In a reactive mode, there are generally two types of traffic redirection techniques utilized by cloud-based DDoS mitigation to direct traffic to them: DNS redirection and BGP network advertisement. Both of these require giving the cloud-based provider the rights to redirect traffic from your own premise to theirs so they can scrub the traffic clean by filtering out malicious traffic.

The DNS redirection schema requires a change of DNS mapping from your original IP address to one that is owned by the cloud provider. This is less intrusive but could take some time while the DNS change is being propagated throughout the internet.

DNS Propogation

As many of you already know, DNS change is slow to propagate and relies on your end user's setting. Cloud-based DDoS providers that use DNS often drop the TTL for the protected domain to one second, so they can make changes and have them updated instantly.

However, a lot of ISPs won't honor a one-second TTL on their recursive DNS servers, and most organizations rely on their ISPs' recursive DNS servers. So, even though the protection is active and enabled, a large chunk of the internet might not be going to the right place.

The BGP network advertisement change is immediate and requires the cloud provider to advertise your block on your behalf. For example, if your company owns 1.1.1.0/24 block and you advertise it to the upstream provider, the cloud provider will now advertise the same block. Due to the nature of internet routing and common practice, you need to have a registered public IP block that is bigger than /24 (254 public IP address) in order to utilize this method.

Other Network Restrictions

There are generally more restrictions on BGP redirection than DNS change. Some providers require separate public IP blocks for establishing tunnels, and most of them recommend modifying an upstream access list to only allow tunnel traffic inbound during mitigation. Please check carefully during the evaluation stage to make sure you are comfortable with what they are recommending.

Once the network is redirected to the cloud provider, the inbound traffic will be scrubbed and the clean traffic will be passed back. Using parenting as an analogy, having your network traffic traverse through someone else's network is almost like having your kids sleep over at other people's house. There is always an uneasy feeling about it regardless of how much you trust the other party.

This lack of control can somewhat be easier to accept if you are already using the third party for other services, such as CDN, and the DDoS mitigation is another service that is added on top of it. If you are in this camp, making plans for vendor diversity would be a good idea.

Customization

Another area that makes cloud-based mitigation unfavorable is the lack of customization you can extend for your company. Because your company's business model is different than other businesses, your traffic pattern is often different. A gaming company's network traffic is very different than, say, a web hosting company's traffic. For example, if your network's traffic is only going to be small packets with 64 bytes of payload, an effective DDoS mitigation strategy might be to drop any traffic with a payload larger than 64 bytes.

Customization options when you utilize cloud-based mitigation are limited. The providers build the solution that appeals to the majority of the customers they intend to serve but neglect long-tail requirements. This is one of the necessary trade-offs for demand aggregation. There are certain knobs and switches you can leverage but the scope of change is within the control of the vendor. You are purchasing a predefined service that is suitable for 80% of the customers; if you happen to be the 20% need some customized work, the chance of the vendor catering to your needs is slim to none.

Vendor Lock-In

Earlier in the section, we covered the traffic redirection technologies that required a close collaboration between the user and the cloud-mitigation vendor. In an always-on scenario, the setup is even more tightly integrated. There are marketing materials that may lead people to believe that because the vendor exists in the cloud and the customer pays for the usage fee only, they can be free to switch vendors if need be. However, we would argue that if done correctly the setup is so integrated that the DDoS vendor lock-in is sometimes even more so than an on-premise solution. You might be able to switch between AWS and Azure when it comes to launching virtual machines in the cloud, but imagine needing to change your internet peering, upstream access list, or DNS authoritative pointing—these are not something you can switch at a moment's notice.

This vendor lock-in might be even more of a consideration since the field is somewhat new and full of start-up companies. What if the vendor is bought or become insolvent? What is the cost of alternative and ramp-up time if you were to switch? The potential cost of switching might make a cloud-based DDoS solution less desirable than originally thought.

Security Boundaries

Companies that need top security clearance might not have an option to use cloud-based providers due to security boundary concerns. In the United States, there are regulations regarding traffic that deals with consumer privacy, healthcare records, and government entities. In certain parts of Europe, such as Germany, data sovereignty is required where data that originated in one country cannot leave the country.

Business Certifications

If your company operates with business certifications such as the ISO 9000 family of quality assurance certifications, you need to make sure your cloud provider is in line with the necessary qualifications.

If your company operates in one of the vertical markets or countries with laws regarding traffic patterns, you will need to pay close attention when considering whether to use a cloud-based mitigator. This

is not to say a cloud mitigation provider cannot operate within the guidelines of your requirements, but be aware that not all cloud-based mitigation providers can conform to the security boundaries needs.

Cloud-Based DDoS Mitigation Methods

So far in this chapter, we have looked at the reasons to use and not to use cloud-based DDoS mitigation vendors. We covered some of the basic operations as they support our reasoning. In this section, we will take a more in-depth look at the operations of cloud-based mitigation vendors.

The cloud-based DDoS mitigation vendor life cycle consists of detection, mitigation, and reporting—some of which can be a hybrid model, such as integrating some of the reporting into your on-site tools. As more companies migrate services toward the cloud, the DDoS mitigation strategy should increasingly adopt the hybrid model. The cycle is a continuum, consisting of the triggered event, traffic reporting, evaluation, and then feedback to create better future detection and mitigation.

DDoS Detection Mechanism in the Cloud

The DDoS detection mechanism in the cloud is not much different than the on-premise detection mechanism. The two most important mechanisms for DDoS detection are NetFlow and packet traces. The exception would be that for NetFlow, the export destination could be at an external public IP. Exporting NetFlow data might be outside the comfort level of some customers, in which case the vendor might provide the customer with a vendor-controlled virtual machine as the NetFlow collection, and manage the virtual machine jointly with the customer. The general process is shown in Figure 4-1. This setup also applies to log collection for the cloud-based vendor.

Packet traces remain an important part of DDoS detection, especially for application-level attacks. This is often done by placing agents able to perform packet capture at strategic locations inside of the customer's network. One thing that is different from the cloud-based solution when doing packet capture is that oftentimes due to the risk of exposing customer data, only header data is sent over to

the cloud-based vendor or traffic is differentiated to remove risk before being forwarded on.

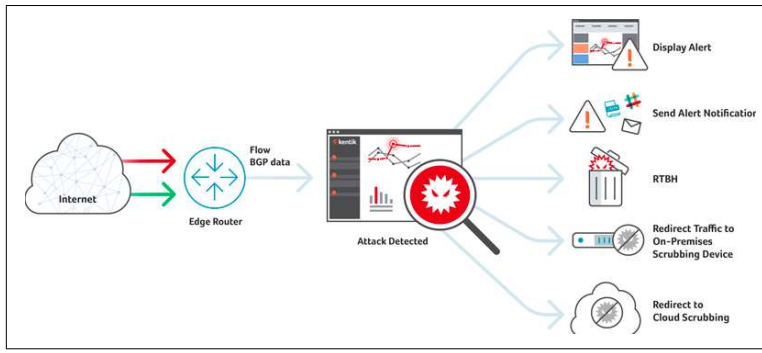

Figure 4-1. Cloud-based DDoS detection (source: https://www.kentik.com/ddos-detection/)

There is a growing trend of real-time analysis of data. Since the cloud provider typically aggregates data feeds from various customers, coupled with big data analysis, they can arguably detect reoccurring or reused DDoS attacks better than the on-premise solution since the customer only has a limited amount of data.

DDoS Mitigation Mechanism in the Cloud

The DDoS mitigation mechanism in the cloud requires careful consideration and upfront work before the DDoS event happens. Compared to an on-premise setup where you have complete control, in the case of an external scrubbing center, the traffic shift and management at each point needs to be mapped out. Also worth pointing out is that the traffic shift typically does not take place right away and may not shift due to external factors. For example, in the BGP advertisement model where the cloud provider advertises your public block with higher preference, you can use typical BGP attributes to influence the decision making, but there are always ways for other parties to override your "suggestions." In other words, traffic shift in BGP is more of an art than science.

The first cloud-based mitigation method is an *always-on solution*. The most common always-on deployment architecture is for the customer to couple the service with the Content Delivery Network (CDN). If you already use a particular vendor for CDN to distribute your content, they are already your gatekeeper with firsthand infor-

mation about the traffic. It could make sense to do an extra layer of analysis to drop any suspicious traffic. In Example 4-1, we can see an example of Cisco.com pointed to an Akamai (one of the major CDN providers in the world) edge network.

The trade-off in this model is that you are putting a lot of trust that the CDN network will be up 100% of the time. You are also giving up the visibility of customer traffic since you are one layer removed from them. Since this is an always-on model, you are likely to pay for an always-on upkeep fee as well as a traffic-based fee when the DDoS attack happens. Also, keep in mind that this mitigation method only involves traversing through hostnames via DNS; if the attack is directed toward IP address, this mitigation does not take effect.

Example 4-1. Cisco.com CNAME points to Akamai Edge

```
$ dig www.cisco.com
<skip>
;; QUESTION SECTION:
;www.cisco.com.      IN A

;; ANSWER SECTION:
www.cisco.com.    3406
IN CNAME www.cisco.com.akadns.net.
www.cisco.com.akadns.net. 300
IN CNAME wwwds.cisco.com.edgekey.net.
wwwds.cisco.com.edgekey.net. 18911
IN CNAME wwwds.cisco.com.edgekey.net.globalredir.akadns.net.
wwwds.cisco.com.edgekey.net.globalredir.akadns.net.
3406 IN CNAME e2867.dsca.akamaiedge.net.
e2867.dsca.akamaiedge.net. 20 IN A 23.41.176.89
```

If you prefer to stay away from an always-on solution, you can choose to redirect your traffic in a reactive mode after a DDoS attack occurs. In this case, you can utilize a *DNS change or redirection* from your own server to the cloud-based scrubbing center. The dirty traffic will be dropped and clean traffic will be sent back to your premises. This path from cloud vendor back to your premises can be a physical link, but more likely will be a virtual tunnel from the provider to your equipment. There are three key items that need extra attention:

- Where does the authoritative DNS record reside? This is the party responsible for making the DNS record change.

- The path of the clean traffic path needs to be carefully planned out and it is strongly recommended that it be established and tested prior to the actual DDoS event. In the case of the virtual tunnel, the customer needs to make sure in the case of traffic congestion due to a DDoS attack, the traffic can still make its way back to your premises.
- The tunnel endpoint should not be allowed to be targeted externally.

Another way to redirect traffic without a DNS change would be a BGP advertisement change. In this case, the public IP block with your resource is advertised by the cloud provider on your behalf upon detection of an attack. The operation itself is pretty straightforward; however, the devil is always in the details. Two key items that you should make sure in this scenario are:

- The cloud mitigation provider is well established and well peered in the various internet exchange points. A good resource for checking is *www.peeringdb.com*.
- The cloud provider has set up agreements with its peers for them to advertise your IP block. To prevent BGP hijacking, service providers now typically implement access checking to make sure the advertisements they receive are from legitimate owners. Since the cloud provider is advertising on your behalf, you would need to authorize the cloud provider to advertise on your behalf. In Figure 4-2, the looking glass provided by NTT America can be a tool used to see how the BGP prefix is viewed inside of the NTT America network.

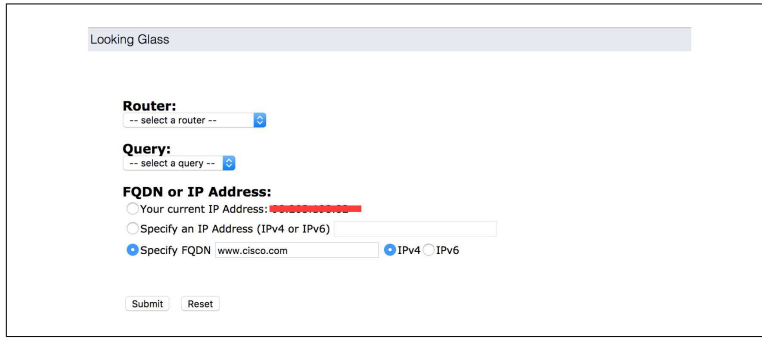

Figure 4-2. NTT America BGP looking glass (source: http://bit.ly/2EBDt8Z)

DDoS Event Reporting

Since you do not have visibility into the cloud provider's devices and network, a solid reporting and feedback loop is even more important in the setup. The rule of thumb is: the more reporting the better, but there is no right or wrong answer on how much reporting is needed.

At the very least, we believe a near real-time report of the start of the event, the anomaly detected, and end of the event, as well as various network statistics such as packets-per-second and bandwidth utilization, are required. A more useful and improved reporting mechanism would be a uniformed reporting mechanism from the cloud provider that can be managed via API so the customer can ingest and analyze the data automatically.

You need to also understand that traffic is being handled as perceived by the provider and realized by internal tooling in an overlay fashion. In Figure 4-3, we can see an example of alert reporting from one of the cloud-based DDoS detection providers.

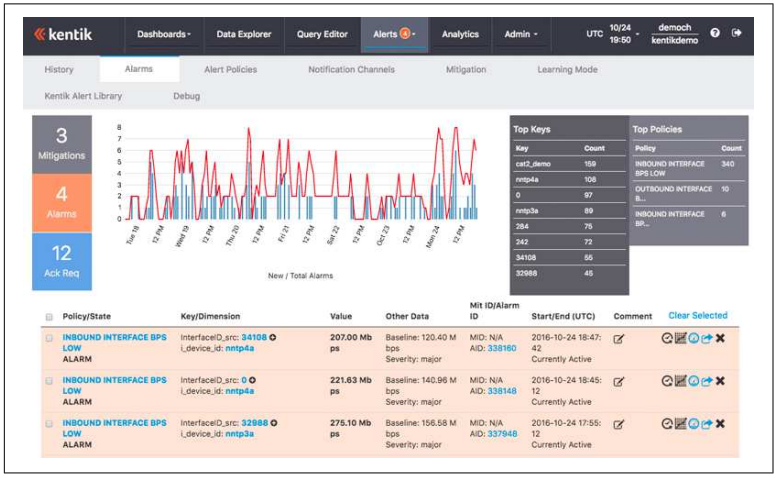

Figure 4-3. Kentik reporting (source: https://www.kentik.com/ddos-detection/)

Hybrid Model

You can often mix and match many of the techniques above. Just as the attacks have gotten multimetric, mitigation solutions have gotten more sophisticated as well. For example, you can utilize the

always-on model for a general scrubbing for the big volumetric attacks, then use an on-premise equipment for application-level attack detection. In the case where the on-premise equipment runs out of capacity, as a third option, you can redirect your traffic to the cloud mitigation provider.

A hybrid model is what we would recommend if you are able to do so. Unfortunately, in a hybrid model, you are essentially building two sets of mitigation solutions and all of the items we mentioned in the chapter are applicable to you. The bright side is that you will hopefully enjoy the benefits of both on-premise and cloud-based solutions.

Summary

In this chapter, we looked at the reasons why one would or would not use cloud-based DDoS mitigation providers, as well as the methods of utilizing cloud-based DDoS providers. Let us summarize by putting the items on a checklist:

Cloud-Based Mitigation Provider Checklist

Pre-Flight Questions:

1. When investigating a new technology, does your company usually build or buy the solutions?
2. Is your company already using a CDN provider? Do they offer DDoS protection service?
3. Is your company familiar with cloud providers such as Amazon AWS and Microsoft Azure?
4. How often does your company experience DDoS attacks?
5. How many resources can your company dedicate to DDoS mitigation?
6. Does your company currently have public IP address blocks larger than a /24?
7. How is DNS handled in your infrastructure? What about BGP, if any?

Vendor Selection:

1. Does the vendor have a proven track record? How long have they been in the business of DDoS mitigation?
2. Does the vendor provide both an on-premise and cloud-based solution? How about a hybrid model?
3. How much control does the vendor give you when it comes to mitigation techniques?
4. What is the reporting mechanism the vendor provides? Do they offer integration with your own tools?
5. What is the current footprint of the vendor, peering exchange location, and how many upstream and downstream peers do they have?
6. What kind of business and technical certification, if any, does the vendor possess?
7. Does the vendor charge upfront fees or just ongoing usage fees?
8. What is the ease of migration out of this vendor if the solution does not work out down the line?

Needless to say, the checklist is not a one-time process. The landscape of attack and mitigation is always shifting, so you should revisit your DDoS mitigation strategy every few months to make sure the setup still fits your needs. For example, you might initially choose to deploy a strategy involved cloud-based mitigation, but as your company grows you can decide to build your own on-premise mitigation.

CHAPTER 5

DDoS Focused Threat Intelligence

Threat intelligence has received a lot of attention lately. In today's world, almost all companies rely on digitized information. Show us a company that does not have valuable assets in digital form and we will show you a company that is not competitive in its own market. Digital assets are easy to move around and store, but also easy to be stolen and compromised. *Security threat intelligence* is a term that describes the collection of data that might be a threat to your valuable digital assets.

According to Gartner, the definition of threat intelligence is as follows:

> Threat Intelligence is evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard.

If applied to the context of DDoS threat intelligence, we can conclude that the results should be data-driven, evidence-based, and include analysis of data about existing or emerging DDoS threats and actionable responses.

In this chapter, we will discuss the collection of data that will reveal potential DDoS security threats and show you ways to store and analyze the data. From there, we can derive response that can help you prevent and defend against future DDoS attacks.

NOTE

DDoS Focused Threat Intelligence

Threat intelligence is a much written-about subject, but the relevance of the topic to DDoS is a bit fuzzy. In this chapter, we want to focus on how to collect relevant data that can be applied in DDoS mitigation.

This is a very exciting chapter for us, as we feel this is a way to turn the table on the attackers. So far in the book, we have been in a reactive mode where we are at the receiving end of DDoS attacks. Anybody who has played competitive sports knows if you only play defense, the best result you can hope for is a 0 to 0 draw; it is only when you start to play offense that you can score points and win the game. In discussing threat intelligence, we are going on the offense to actively collect data, set traps for the bad guys, and try to give ourselves advance warnings.

Let's get started!

IP Blocklists

The first task is to understand the difference between a known bad source IP and source IP addresses which can be a potential attacker. This is an important distinction as you will handle them differently with the policy that you create. These known bad source IP addresses should be the basis for your blacklist. "Known bad" implies that there has been some level of vetting from the security community.

By building your own threat intelligence based IP blocklist you can reduce the size of your attack surface while keeping known bad endpoints from stealing your data. Reducing the number of IP addresses that you need to check increases the effectiveness of both your DDoS detection and mitigation systems. The IP blocklist should include the BOGON IP address ranges that have not been allocated or allocated only for private use, as mentioned in the mitigation chapter. If you see sources from the BOGON list coming to your network, there is a good chance that they are spoofed IPs.

Know Your Customer Source

It would make your life easier as a DDoS protector to know the approximate source of your customer in advance. The geolocation IP correlation is never perfect, but it provides a good baseline. For example, it would be a red flag to see a sudden surge of IP sources from Russia if the majority of your customer base is in North America and Western Europe.

A free GeoIP database is the MaxMind database.

In Figure 5-1, we captured the incoming packets' source IP addresses at our edge while conducted a UDP flood attack using a DDoS for Hire system. In this particular attack, the BOGON list made up around 15% of the source IPs.

IPv4 · 10125

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
0.0.205.175	1	582	0	0	1	582
0.15.164.247	1	582	0	0	1	582
0.19.65.82	1	582	0	0	1	582
0.21.201.28	1	582	0	0	1	582
0.27.165.75	1	582	0	0	1	582
0.31.248.203	1	582	0	0	1	582
0.33.86.63	1	582	0	0	1	582
0.34.162.145	1	582	0	0	1	582
0.43.82.21	1	582	0	0	1	582
0.52.68.83	1	582	0	0	1	582
0.57.94.200	1	582	0	0	1	582
0.81.34.138	1	582	0	0	1	582
0.83.34.185	1	582	0	0	1	582
0.101.118.106	1	582	0	0	1	582
0.110.198.171	1	582	0	0	1	582
0.122.27.171	1	582	0	0	1	582
0.123.129.80	1	582	0	0	1	582
0.142.130.0	1	582	0	0	1	582
0.143.218.193	1	582	0	0	1	582
0.158.51.96	1	582	0	0	1	582
0.162.185.93	1	582	0	0	1	582
0.168.3.213	1	582	0	0	1	582
0.183.191.149	1	582	0	0	1	582
0.183.203.112	1	582	0	0	1	582
0.185.183.103	1	582	0	0	1	582
0.187.216.1	1	582	0	0	1	582
0.191.98.227	1	582	0	0	1	582
0.197.124.28	1	582	0	0	1	582
0.197.225.38	1	582	0	0	1	582
0.204.15.101	1	582	0	0	1	582
0.209.240.158	1	582	0	0	1	582
0.223.29.113	1	582	0	0	1	582

Figure 5-1. Incoming packets at edge in the 0.0.0.0/8 range

Please keep in mind that this or any IP blocklist is dynamic and evolving. You can start with a prebuilt list, such as from Team Cymru, but the most effective blocklist is the one that you build for

your context. The work at Team Cymru is one example of community-based efforts for DDoS mitigation.

Community Supported Efforts

A typical engineer's day is full of interrupt-driven tasks; keeping the lights on is a full-time job. Wouldn't it be great if there were community-based efforts that could help with DDoS mitigation? Yes, apparently a lot of people have the same idea. In this section, we will look at some of the projects that can help us in DDoS mitigation.

IP Geolocation Providers

We have mentioned IP geolocation a few times so far in the book. Though an IP geolocation provider is not a direct security-related service, it is one of the most important tools that we can use to limit our exposure to provide physical location context to the information we have gathered through various channels. If your business does not serve a particular geography then why let the traffic into your network? This is one way to help reduce your attack surface even further. This will be explored in more detail later in the chapter. While not perfect, they continue to evolve and improve over time.

MaxMind is one such provider with free and paid geolocation databases using simple APIs (Figure 5-2).

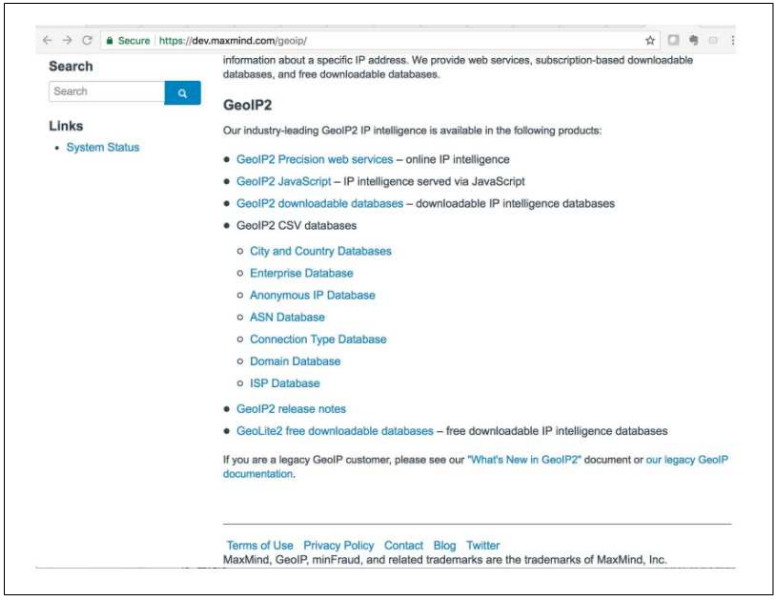

Figure 5-2. MaxMind GeoIP products (source: http://bit.ly/2BLheIa)

Purpose-Built Node Lists

As we have stressed in this chapter, knowing your customer sources and blocking unnecessary incoming traffic before it reaches your network border can be a very useful step in protection. Unlike the BOGON list we covered earlier, these lists need to be used with your domain knowledge in order to limit the collateral damage that you might experience if you accidentally block too broad of a scope. We picked out a few of the resources and provide brief reasons for why the list might be useful to you.

The United States State Department maintains a Trade Embargo List. This is an example of a business logic tie-in with technology. If your business is registered in the United States and unable to do business with countries on the list, why even allow the IP prefixes associated with the country inbound? It may make sense to reduce the size of your policy by blocking organizations themselves denoted by autonomous system number in network language. The IP-to-ASN list can be obtained from various sources, including MaxMind. Of course, your situation might be different; we use the US Trade Embargo List as an example to illustrate our point.

The *public cloud providers IP list*, such as IP addresses owned by Amazon AWS and Microsoft Azure, might be sources you can potentially add to your blocklist. For example, if you only operate an interactive online gaming site, how many of your users come from the public cloud? The answer is probably very few. Even if you utilize the public cloud for operational resources, you can always "explicit permit, implicit deny" when it comes to the IP space.

While *TOR exit nodes* are not generally involved in a volumetric DDoS attack, they are seen in a number of slow and low attacks. TOR or The Onion Router is an encrypted overlay used to hide the actual source address of the requester. This clearly does not mean they are doing bad things; however, these addresses are easily grouped for policy enforcement, such as session limiting.

There are also a few crowdsourced lists containing systems that have been accused of nefarious activities such as spamming, hijacks, housing malware, and others. The reputation of these networks and hosts is reduced. In Figure 5-3, we see a screenshot from NetLab 360 showing 24 hours of scanning behavior from their observation points.

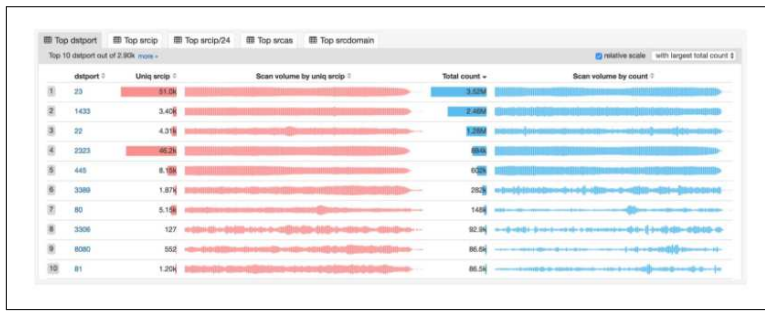

Figure 5-3. NetLab 360 network scan mon (source: http://bit.ly/2FCQgVN)

The information can be correlated back to the top source IP (Figure 5-4).

Figure 5-4. NetLab 360 network scan mon top SrcIP (source: http://bit.ly/2BJQ18I)

The *Spamhaus Drop List* contains a list of IP addresses and ASNs that are defunct and hijacked for criminal purposes. *EmergingThreats* contains a list of IPs of various botnet command and control centers, scanners, and unsolicited traffic.

Real bots are still widely used in DDoS attacks, now even more than ever. In the beginning, botnets were mostly constructed with infected hosts. Think of your coworker who opened an email attachment that was executed as a Trojan horse, and the program in turn opened up a backdoor for someone else to control the computer. This is an example of a more traditional bot. With the rise of internet connected everything, the making of botnets is evolving as well. A bot can now be an IP camera or temperature sensor that was purchase by somebody who never changed the default username and password.

A good example is the Mirai botnet which was used in a high profile DDoS attack that took down a large chunk of the internet in October 2016. It consisted of internet connected cameras, home routers, DVRs, and printers with default credentials used for their telnet ports. This was not an infection—just poor security policy and lack of attention by the vendors. The situation is further complicated by devices with public IPs reachable from the outside world.

In Figure 5-5, you can see an example of result of internet scanning of Mirai-infected hosts.

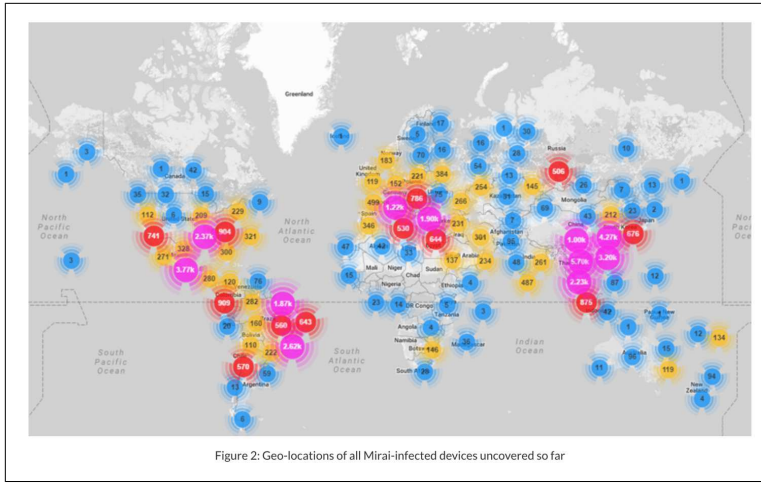

Figure 5-5. Geolocation of a sample of Mirai-infected devices (source: http://bit.ly/2BJzPo6)

One of the ways that security professionals create their own intelligence about botnets such as Mirai is through actively scanning the internet. While it is tempting to perform some of this scanning on your own it is best to rely on the professionals. Problems can arise such as ISP terms of service violations and reactive black listing of your IP address.

Knowing the IP addresses of infected hosts is a good step in narrowing down potential attack sources. Any list of IP addresses that belong to a botnet is dynamic due to many factors in the Internet. Adding this information to the drop list is not recommended for this reason, however it is very useful to keep as a list of potential suspects which you can apply policy to once an attack has been detected.

Honeypots

A honeypot, in the context of cybersecurity, is a way to attract potential hackers in order to understand them. Placed in various locations of the internet, honeypots can mask themselves as vulnerable hosts and trap potential hackers. We can then use the insights we gain from the interaction to combat and mitigate future DDoS attacks, or better yet, prevent future attacks altogether. If you do decide to deploy honeypots, be careful to separate your production

network from the honeypot network to decrease the risk of the dirty traffic bleeding into your production network.

Honeypots as Additional Signal Data

Please keep in mind that the data you collect from honeypots are not reliable and potentially contain false information. They can be used as additional signals after you clean them up and provide more structures to the original data. The challenge will always be to clean the data enough to make intelligent decisions about it.

Our primary use case for a honeypot, when DDoS-focused, is to gain a deeper understanding of the internet-scanning behavior by DDoS-capable botnets. They are also useful to detect the testing of your defenses by attackers before they attack. Since the honeypot can interact with client devices, we are able to see scanners looking for potential bots or source connection attempts, successful logins, and executing commands. Keep in mind that telling the difference between a security researcher and nefarious activity is not straightforward. Analyze this data further before adding to it to your blocklist.

An example of a honeypot project is the Cowrie Honeypot. The project was started by security researcher Michel Oosterhof and derived from the Kippo honeypot project. It is a medium-interaction SSH and Telnet honeypot designed to log brute force attacks and the shell interaction performed by the attacker. Cowrie is not perfect, as it has been fingerprinted but fortunately, from our research, most nefarious sources do not seem to care. Cowrie could be a logical place to start your honeypot effort before moving to more specialized honeypots targeting XML or HTTP specifically.

By combining and containerizing various honeypots, along with the ELK stack, the T-Pot Project provides an easy-to-install, all-in-one host that you can use on your premises to detect potential attacks and hackers. In Figure 5-6, we can glean a lot of information from simply putting a T-Pot host on a public VM, and look like a security superhero to management by making some very simple ELK modifications. Top source IPs, origin countries, destination ports, and application information accumulate over time, giving you insight into what attackers and researchers are looking for.

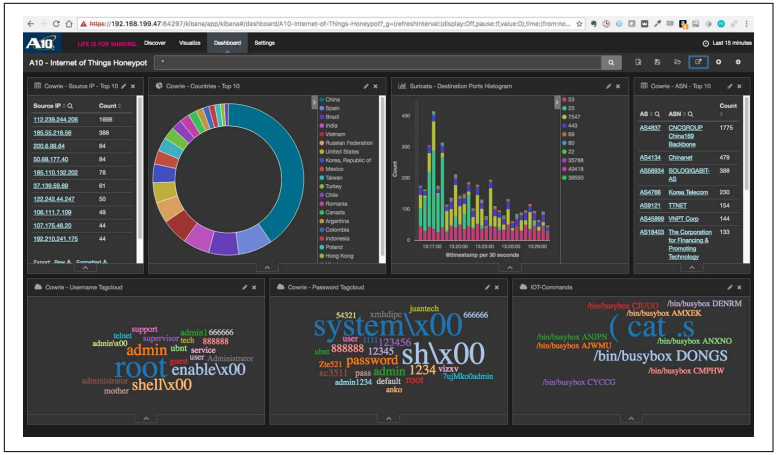

Figure 5-6. T-Pot Kibana output

In our opinion, honeypots are a great way to proactively collect data that you would otherwise have a hard time getting. It is a rich dataset that we do not see from normal logs. If you place the honeypot close to your datacenter edge, you can see exactly the potential threats to your digital assets.

DDoS-as-a-Service

When you have built up your DDoS mitigation shield, you need to stress test your own system by using the same tools that a potential attacker would use. Along with tools outlined earlier in the book, DDoS-as-a-Service can be a source to perform such test.

Check Your Local Laws and Regulations

As you can imagine, the DDoS-as-a-Service providers sometimes operate in a gray area. Be sure to check your local laws and regulations before you use their services.

You can spend time to become acquainted with a broad set of Booter systems. These might be the very systems that are being used to attack you from. Most will let you set up an account without giving them any payment information. We would recommend being as safe as possible with the addresses that you use to sign up for such a service. Use common sense in these cases: do not connect from your

corporate IP address or personal or corporate email addresses; instead, use a throwaway email address.

Once you are confirmed, login and look at the current running attacks and the time periods they have been running for. Also check out the types of attacks that they offer and the overall attack volume that they claim. As a bonus take note of the source IP addresses that the attack traffic comes from. If you are testing using an attack that is not spoofed then you are seeing the real IP address of an attacker and can add this to round off your threat intelligence system.

Unfortunately, as local laws vary wildly regarding DDoS-as-a-Service systems, it would not be responsible for us to recommend going further than a small-scale stress test.

Summary

In this chapter, you learned a number of ways to leverage community-based systems to construct a DDoS-focused threat intelligence system. This includes various IP lists that you can use for your own IP blacklist to block potential malicious sources. You also saw examples of tools such as Cowrie and T-Pot that can be used as honeypots for information gathering.

CHAPTER 6
Final Thoughts

> If you know the enemy and know yourself, you need not fear the result of a hundred battles.
>
> —Sun Tzu, *The Art of War*

We hope this journey has been as useful to you as it was fun for us to write!

The DDoS space is as interesting and challenging today as it was a few years ago when we first started to work in it. DDoS attacks have existed in some form since the beginning of the commercial web itself and the problem has gotten progressively worse. The simple yet effective nature of DDoS makes the subject more relatable to all new technologies, especially with the growth of cloud adoption and IoT. As you have read in this book, malicious users can come from all works of life using any new technology.

Fortunately for us, new and old technologies continue to combat and mitigate DDoS attacks. As more people become aware of the underlying nature of the attacks, we are able to integrate more of the mitigation technologies closer to the wire. We hope that by reading this book, you will be better able to recognize the malicious actors who might try to DDoS attack you and you can know your options better to defend against such attacks.

You are now part of the solution. Please join us in hopefully making DDoS attacks one day disappear from our vocabularies.

Acknowledgments

We would like to thank the O'Reilly team—including Acquisition Editor Courtney Allen, Developmental Editor Virginia Wilson, Production Editor Nick Adams, and the many others working behind the scenes—for giving us the opportunity to write this book and for the help provided along the way.

We would like to express our sincere appreciation for our technical reviewers Nick Payton, J. R. Mayberry, and Allan Liska for providing valuable feedback during the writing process. The book would not be the same without their input.

Finally, we would like to say a special thank you to Lee Chen, CEO of A10 Networks, for his confidence and support of this project.

Rich Groves

I'd like to thank my coauthor, Eric, for being such a relentless and hardworking author, engineer, and developer. This would have never happened without his expertise and hustle. I'd like to thank my wife, Laura, and my daughters, Morgan and Raewyn, for their patience and love during the writing of this book. Thanks also go to my employer, A10 Networks, and my boss, Raj, for being supportive and giving me the guidance to help me improve, and the room to do so.

Eric Chou

I would like to thank my wife, Joanna, for her constant support during the writing process and my two beautiful daughters for always inspiring me to become a better person.

About the Authors

Rich Groves is the Director of Research and Development at A10 Networks and the Principal Architect of its DDoS Mitigation Platform (A10 Thunder TPS). Large-scale security and network monitoring have been his major focus over the past 20 years. While at Microsoft, Rich created the Microsoft DEMon SDN powered network monitoring platform, which was later turned into a highly successful commercial product.

Rich was a core member of the Microsoft Digital Crimes Unit, where he took down and disrupted many large-scale botnets with systems that he created and ran for many years. Previous to Microsoft, Rich worked in high-level engineering roles at Time Warner Cable, Endace, America Online, and MCI. Rich currently lives in Honolulu, Hawaii, with his wife, two daughters, and three dogs.

Eric Chou is a seasoned technologist with over 17 years of experience. He has worked on and helped manage some of the largest networks in the industry while working at Amazon AWS and Microsoft Azure. He is passionate about network automation, Python, and helping companies build better security postures.

Eric is the author of *Mastering Python Networking* (Packt Publishing, 2017). Currently, Eric holds two patents in IP Telephony and is a Principal Engineer at A10 Networks with a focus on product research and development in the prevention and mitigation of DDoS attacks.