O'REILLY®

Compliments of ORACLE® Dyn

Modern Defense in Depth

An Integrated Approach to Better Web Application Security

Stephen Gates

Modern Defense in Depth

An Integrated Approach to Better Web Application Security

Stephen Gates

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

Modern Defense in Depth
by Stephen Gates

Printed in the United States of America.

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

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editors: Virginia Wilson and Nikki McDonald
Technical Reviewers: Allan Liska and Melissa Kelley
Production Editor: Christopher Faucher
Copyeditor: Octal Publishing, LLC
Proofreader: Matthew Burgoyne
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest

January 2019: First Edition

Revision History for the First Edition
2019-01-18: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781492050353* for release details.

This work is part of a collaboration between O'Reilly and Oracle Dyn. See our *statement of editorial independence*.

978-1-492-05033-9

[LSI]

Table of Contents

Preface

For decades, organizations have applied security strategies, technologies, and expertise designed to solve the cyberthreat issues they face daily. These issues include infections from advanced malware (including ransomware), exploitation of operating system and application vulnerabilities, attacker takeover of computers and devices, phishing of employees leading to advanced persistent threats, code injections and abuse of websites and applications, denial-of-service outages, financial fraud, data theft, and more. The list of successful campaigns resulting in losses is indeed lengthy.

Today's organizations are not, however, at fault. They are deploying the best security technologies available; they are implementing them in the recommended fashion; and, in most cases, they are following industry-accepted best practices. However, the increases in data breach figures in the past few years alone—affecting millions, if not billions of people worldwide—are staggering. Having been personally affected, like so many others due to some of the largest data breaches on record, motivated me to write this book and hopefully establish that a better way is possible.

Why This Book

It's been repeatedly demonstrated that the steps we are taking to protect ourselves and our organizations from cyberattackers must be inadequate. If not, why are attackers still so successful, and why are our organizations still being breached? We can all probably agree that something is simply missing in our fight against cybercrime. In this book, we are here to discover together what is missing, what's not working, and why. In addition, I make solid recommendations

about how to integrate the technologies so often found in our fight against cybercrime, of which any organization can take advantage, in order to make considerable improvements to solving this problem once and for all.

As a hands-on cybersecurity manager and practitioner with nearly two decades of experience deploying most of the very technologies covered in this book, I believe I discovered what might have been missing all along: the concept of *integration*. My goal in this book is to take you down the path of what I've experienced firsthand, demonstrate what our current approaches are like, highlight some of their deficiencies, and draw a parallel to a better approach to cybersecurity. I also provide solid guidelines on how we can work together to achieve something greater through making the present lines of defense in your organization operate as one cohesive unit. To meet your expectations concerning the concepts I am about to divulge, I thoroughly cover every concept while attempting to be as brief as possible.

Pertaining to the title of this book, the concept of *Defense in Depth* (DiD) has been around long before the inception of the internet. It has been widely recommended and, therefore, widely practiced in all sorts of different industries and organizations. In the context of cybersecurity, the current approach to DiD calls for *independent lines of defense* to be deployed between the internet and an organization's networks, internal users, publicly exposed web applications, and private data.

From my personal experience and own observations, the currently accepted approach to DiD is seriously lacking, and a new approach is desperately needed. This new approach is explored in depth in this book. What I aim to prove is that the concept of *integration*, modeled similarly to a modern military, is the missing element that is so desperately needed today—to thoroughly protect our organizations from cybercrime.

After ingesting the content found in this book, you'll learn how and where to apply modern DiD strategies to the security postures within your own organizations. Furthermore, I demonstrate that anyone can measurably improve the defensive stances for their organizations by applying integrated approaches similar to a modern military. By the end of this book, you should have a solid understanding of how the recommendations presented within it can

be implemented today, often with the current security technologies you already have in place.

The Audience for this Book

This book is designed to help those who are in the role of cybersecurity management, given that you are ultimately responsible for protecting your networks, your internal user communities, your public-facing web applications, and your data from the cyberthreats you face daily. This book is directed at chief security officers (CSOs), chief information security officers (CISOs), security directors, security managers, and other similar roles.

What You Will Learn

In this book, I highlight the security technologies that are currently deployed and how they're deployed, so you can recognize the shortcomings when presently trying to protect *internal users* and *public-facing web applications* from cyberattacks. I expose the deficiencies in the currently accepted definition of DiD within the context of cybersecurity to help you realize that a better model exists. Then, I demonstrate what's needed today to fully protect public-facing web applications so that you can learn how to best protect them within the context of the cloud. Following that, I help you understand the available options to fully integrate the security technologies deployed while exploring the pros and cons of in-house-versus-outsourced security operations centers (SOCs). And finally, I paint a picture of what steps you can take to "intelligently integrate" your security approaches in the context of automation and supervised machine learning.

CHAPTER 1

What's Not Working, and Why?

When you examine the context of defending your users and public-facing web applications deployed in your data centers, you need to understand what's not working, and why. We discuss the expense and complexity of available solutions, what attackers know and understand, the deficiencies seen in both user and web application protection, a major noise problem that exists, and, finally, why attackers are so successful.

Expense and Complexity of Solutions

For nearly two decades, organizations have taken the multivendor approach as suggested by industry experts, deploying independent lines of defense that operate autonomously in nearly every case. Unfortunately, most of these technologies are designed to solve only a single problem, and they are often found to be marginally deployed, which equates to expensive and ineffective solutions.

For example, to combat cyberthreats targeting *users* today, it has become a common practice to deploy independent lines of defense between users and the internet. These include next-generation firewalls, advanced intrusion prevention systems, network access control, and end-point malware protection. Data loss prevention systems, sandboxes, identity access and management systems, automated patching solutions, security information and event management solutions, and so on are often deployed around the periphery of the networks supporting the users' network connectivity.

In addition, many of the security technologies deployed require various skill levels to effectively deploy, tune, and manage, adding to their overall costs. The various technologies also come with their own support, maintenance, and renewal costs, in addition to end-of-support and end-of-life announcements.

For organizations to gain *measurable value* from the technologies they purchase and deploy, they must be able to implement the technologies to their fullest ability. In many cases, before the technologies are completely deployed, operators are pulled away from deployment and tuning activities to work on new or more critical projects. As a result, the "complete value" of the solution is never realized because it has been marginally deployed.

Finally, and often because of industry consolidation, even if an organization deploys a single vendor's solutions, the systems still do not communicate with one another, increasing cost and complexity overall.

Marginally Deployed Web Application Firewalls

The number of organizations that have invested in hardware-based web application firewalls (WAFs) to protect their public-facing web applications is enormous. However, many organizations have deployed their WAFs out-of-band, in monitor mode, or have never adequately tuned the WAF rules to their fullest capabilities.

Attackers Understand How Security Technologies Work

Today's cyberattackers fully understand the shortcomings in the security technologies that organizations deploy, as well as the way they are deployed. For example, attackers know that almost every network today is protected by a firewall. However, attackers still know how to gain access to internal networks quite effectively, right through firewalls.

Because attackers can't penetrate the firewalls from the outside, what do they do instead? They take advantage of unsuspecting computer users and *phish* (fool) them into taking some sort of action. The action on the user's behalf can be as simple as clicking a link or

opening an attachment in an email. In the case of clicking, the internal user starts the conversation from *within* the firewall, outward.

When the return traffic (as a result of the click) arrives on the external side of the firewall, it allows the traffic to seamlessly pass to the internal user. Normally, the return traffic carries a piece of malware, an exploit of a system or application vulnerability, or even worse—ransomware. As soon as the traffic arrives at the user's computer, it executes the malicious code and normally allows an attacker to gain a foothold into an organization.

Why Do Phishers Phish?

It's simple. Attackers completely understand how firewalls work. Because nearly every network is protected by firewalls that are very effective at blocking all incoming unsolicited traffic, how do attackers get around them? They don't get around them because there is no way to do that. Instead, they get the victim to do the work for them.

This Approach Is Not Adequately Protecting Internal Users

When observing the perimeter defenses (commonly called *border* or *edge defenses*) most organizations deploy today to protect their internal user community, we can see a common methodology. Most organizations deploy layer upon layer of *independent technologies* designed to stop various cyberattacks. These lines of defense are normally deployed in a serial fashion with one line of defense deployed behind another, or they are deployed out-of-band, operating in a monitoring fashion only.

Next-generation firewalls are normally deployed as the de facto "first line of defense" at the edge of a network. Using these firewalls, most organizations block all incoming traffic destined to their user community that originates from the internet. However, understanding how phishing works, firewalls are severely limited in their ability to block these attacks.

Looking further in, past edge firewalls, organizations normally deploy advanced network intrusion prevention systems as the next line of defense. These technologies are designed to block "known"

exploits of system and application vulnerabilities that often find their way past the border firewalls. However, unknown exploits that can infect users' computers often pass right through intrusion prevention systems quite easily.

When potentially suspect traffic makes it past next generation firewalls and advanced intrusion prevention systems, the traffic next finds its way to a sandbox technology deployed downstream, usually at the periphery of the internal network. Because most sandbox technologies are deployed out-of-band, they capture copies of network traffic destined to a user computer, ingest the traffic, and try to make some sense out of what the traffic entails.

How Sandboxes Work

When an internal user mistakenly clicks on a link or downloads executable code from the internet, sandbox technology captures a copy of the downloaded code and executes it in the same fashion as the user's computer would. The whole idea here is to execute the code within a sandbox container to observe the code's intention without allowing it to spread an infection elsewhere. Because the code has already found its way to the user's computer, this after-the-fact execution of the code serves to alert security personnel that an infection might have already taken place.

The next line of defense most organizations deploy is endpoint malware detection and protection software (antimalware) on the users' computers themselves. Understanding that there are millions of discrete variants of malware found on the internet today, these software-based solutions are limited in their ability to defend against every known malware strain, due to computer processing limitations and malware-signature storage.

Surrounding all the security technologies deployed, organizations often deploy peripheral solutions to address data loss prevention, network access control, identity and access management, automated patching, and a long list of other independent technologies designed to detect and/or block internal and external malicious activity.

Most would agree that there is an abundance of technologies and solutions that make up the various lines of defense deployed in most enterprises, just to protect internal users from attackers on the inter-

net. However, few, if any, of these technologies are integrated in any fashion whatsoever, and none them are aware of the other technologies deployed. I fully believe that this "independent lines of defense approach" is a major contributor to high number of successful attacker campaigns, and this commonly accepted methodology must be addressed given that it has been proven to not be adequate in many cases.

This Approach Is Not Adequately Protecting Internet-Facing Web Applications

Next, let's see how similar, nonintegrated cybersecurity technologies are most commonly deployed in today's data centers to defend the public-facing web applications hosted there. Here you will find parallel lines of defense that are nearly the same as the lines of defense found when protecting users from the internet. Again, the independent lines of defense are very apparent in organizations' data centers.

Today, many organizations that use data centers deploy their own authoritative Domain Name System (DNS) servers, web servers, and publicly exposed web applications in what is known as the *demilitarization zone* (DMZ) within corporate data centers. These DNS servers, web servers, and web applications are normally protected by firewalls as the first line of defense. However, most people don't realize that firewalls provide little, if any, protection for devices connected in a DMZ, because organizations must configure the border firewalls to allow all inbound traffic from the internet on TCP/UDP port 53 (DNS), TCP port 80 (HTTP), and TCP port 443 (HTTPS).

As a result, organizations still must do better than supposedly protecting their DNS, web servers, and applications with a DMZ. Organizations are next forced to deploy more independent lines of defense like advanced intrusion prevention systems, web application firewalls, bot management solutions, server-based malware protection, and so forth within the DMZ itself to protect the devices deployed there, as well.

This nearly replicates the lines of defense that are deployed to protect the internal user community and adds additional cost and complexity because most of these systems in the DMZs are designed to protect only applications, not user's computers. Another issue with

the aforementioned approach is that, again, none of the lines of defense are integrated and none of them are aware of any other line of defense deployed.

Noise, Noise, and Even More Noise

One of the most significant problems experienced today—because of all the independent security technologies deployed to protect users, DNS, web servers, and web applications that we just mentioned—is the massive number of event and alert logs that each solution generates. Today's security technologies are very noisy, and, in most cases, organizations are completely overwhelmed by the sheer number of logs that they are supposed to consume daily, not to mention the log and alert fatigue that security personnel experience when they observe those same logs and alerts over and over again.

As a result, security information and event management (SIEM) solutions are being deployed today to provide correlation of events, and not just log collection. SIEMs are often implemented with the hope that an organization will be able to effectively manage the massive number of log and alert entries generated by the overabundance of independent security solutions deployed in separate lines of defense. Significant numbers of analysts are in high demand today. They are needed to comb through the logs and alerts daily in the hope of finding something of interest that indicates a successful attack is taking place or had taken place in the past.

Integration Is What's Missing with This Approach

When an attacker breaches one of the independent lines of protection in this antiquated Defense in Depth approach (as mentioned in the previous two sections), the other layers are often completely incapable of detecting that one defensive layer was breached. This is primarily because these layers are completely unaware of one another, and they are not integrated except for aggregating logs to the associated SIEMs. The security technologies deployed have no concept of the upstream and downstream defenses and have no ability to make automated changes "on the fly" to one defensive layer versus another. This fact has continually allowed attackers to remain

resident in networks for long periods of time—often without detection.

Hacker Dwell Time

The time between infection and detection is often called *hacker dwell time*. Looking at nearly every data breach in the past few years, the victims have unequivocally stated that the attack and the associated loss of data originally took place months, if not years, before it was detected. In most cases, organizations detect a breach only after third parties began to observe and report on questionable activity indicating a data breach had taken place.

Another observation to note is that most historical breaches happened because of an attacker bypassing or defeating one line of defense. For example, after an attacker gains a foothold in an organization directly through the border firewall (normally by way of phishing attack), the attacker next begins to operate covertly within the internal network, looking like any other legitimate user. Attackers attempt to capture login credentials to critical systems or find ways of exploiting internal systems to get closer to the data they're looking to steal.

Suppose, for example, an intrusion prevention systems (IPS) deployed as an independent line of defense downstream of the firewall detects an exploit or piece of malware coming from the same IP address on the internet. Does the IPS make a call to the firewall instructing it to begin blocking the source IP address of the malicious traffic upstream? Today, the answer is no. There is no construct in place for these lines of defense to be integrated, and they have no ability to share internal threat intelligence and put it into action. Another example of the lack of integrated lines of defense is as follows:

If an internal user computer was just infected with ransomware and, by some chance, security personnel were made aware of the initial infection via a log or alert, can recursive DNS help eliminate the spread of the infection elsewhere in the network? Yes, it can. Because a recursive DNS server can block a user trying to access a specific domain, did the ransomware-related log or alert trigger an automated change to an organization's recursive DNS servers to

block the ransomware callback domain (which is part of the infection process)? The answer is likely no.

Remember, most ransomware strains must make an initial callback to an attacker for an encryption key exchange, or the attacker will never be able to decrypt the user files. When ransomware needs to perform this callback, we can use recursive DNS servers to stop it and subsequently help eliminate the ransomware from becoming an epidemic by spreading to other devices located elsewhere in the network. As we can see by these two examples, integration between the lines of defense is desperately needed.

Conclusion

In this chapter, I highlighted many of the technologies that are so often found in nearly all organizations today and how they're deployed. In fact, I have implemented many of the technologies mentioned so far in this book exactly in the same way as previously described.

What I have discovered is that these independent lines of defense commonly found in most organizations are not working in concert, have no awareness of one another, and are not sharing internal threat intelligence or acting on it. Because these technologies are doing nothing more than operating as lone citadels (towers), as highlighted in Figure 1-1, it's no wonder that attackers are so successful.

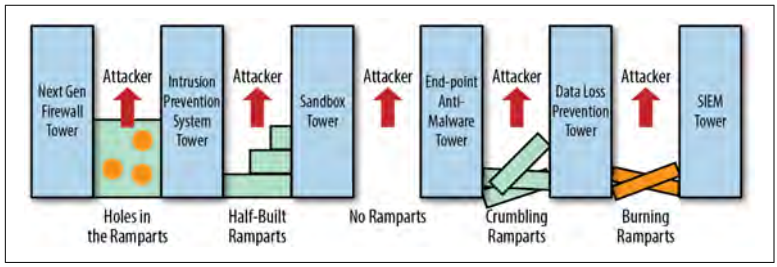

Figure 1-1. Lack of proper integration leads to technologies operating as lone citadels.

Figure 1-1 emphasizes the gaps that often occur due to the lack of integration in our current security approaches. Understanding that all of these technologies are operating independently, attackers consistently find ways of slipping through the "openings" that seem to

exist between them. Integration of our defenses is the key to better security overall.

What I have also discovered is that regardless of the security technologies deployed, when there is no integration, no automation, no internal intelligence sharing, and no symmetrical actions being taken between the lines of defense, attackers will continue to achieve success at the cost of the victim. Simply put, there must be a better way.

The need to look elsewhere for a more effective model is warranted. My next suggestion is to make a comparison to today's modern military. So, let's take a look at how that model operates.

CHAPTER 2

Learning from Military Defense

In comparison to a modern military, the previous examples of protecting users and web applications have little, if any, similarity to the way a modern defense in depth (DiD) approach works in the context of warfare. For example, as one line of defense is attacked in the military, the other lines of defense downstream are adjusted by way of the internal threat intelligence gained to adequately shore up all defenses. There is a complete synergy that exists in the military lines of defense. Next, we look at the conventional definition of DiD as well as explore how a modern military operates in the context of *integrated lines of defense*.

Military Usage of Defense in Depth

DiD is a conventional military defense tactic that is being practiced today across many different industries. Traditionally, DiD provided a means of *slowing down* an attack against a target by using *independent layers* of protection, often called "lines of defense." The standard, widely accepted definition is that DiD argues against using a single line of defense because the likelihood of failure is usually quite high. DiD accepts the notion that when one defensive line fails, another line will take its place and ensure that risks are kept to tolerable levels.

The main deficiency in the current DiD definition is that it calls for "independent lines of the defense," which does not convey how a modern military operates. Today, lines of *communication and intelligence* overlay the independent lines of defense found in the military

that brings "modern awareness" to the battlefield. This sharing of intelligence produces *integration* of the lines of defense. When one line is under assault, intelligence about the enemy's tactics, techniques, and procedures (including weaponry) are collected and communicated to all other lines of defense. This intelligence is used by the other lines to shore up their own defenses and allows time for adjustments to be made where needed.

Figure 2-1 presents an example of how "integrated lines of defense" are obtained in a modern military.

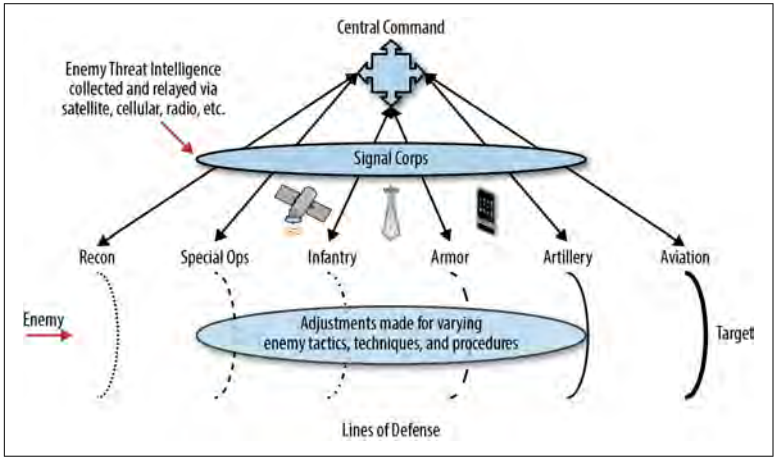

Figure 2-1. The military approach to integrated lines of defense.

Examples of the possible lines of military defense that are very common today might include Recon (reconnaissance), Special Ops, Infantry, Armor, Artillery, and Aviation. Although these lines of defense appear to be quite independent of one another, they are actually integrated in a very cohesive fashion.

The integration comes in the form of enemy threat intelligence that is relayed to Central Command via satellite, cellular, or radio communications by the work of the Signal Corps, as shown in the figure. After the threat intelligence is ingested by Central Command, it is then put into action by relaying this intelligence back to the appropriate lines of defense, as depicted in the figure, that can consume the intelligence and put it into its proper action.

In a modern military operation, when one line of defense is under assault, the other lines of defense become acutely aware of the line that's under attack due to information sharing from Central Com-

mand. Depending on the type of action being taken by the enemy, adjustments are made to the other lines to support the line that's being affected. For example, Central Command might call for an altered special operation, adjustment to infantry defenses, movements to armor (tank) units, an artillery display maneuver, or an expanded aviation reconnaissance mission.

An interesting parallel can be drawn between how a military utilizes this modern DiD strategy and how cybersecurity could do the same.

Cybersecurity Usage of DiD

Internal intelligence sharing produces an integration that is often lacking from most organization's cybersecurity DiD strategy. This must change. If an organization observes a covert attack against a public-facing web application that concealed its way through several preceding lines of defense, it makes complete sense to initiate an adjustment on the fly to block the source of that attack upstream by way of the internal intelligence sharing. However, and in most cases, there is no construct in place to permit the sharing of internal threat intelligence across the various lines of defense.

In addition, there often tends to be an overlap in the various technologies that encompass the lines of defense, with no clear delineation between where a defensive line begins and where it ends. Therefore, a deep understanding of where security technologies are deployed, how they operate, how they block attacks and attackers, what they do best, where they're lacking, and how they can be integrated is badly needed.

Today, integrated DiD strategies must account for many attack vectors, a broadening attack surface, increases in threat actors, limitations of security technology, and the shortage of skilled personnel. Clearly, the *independent* lines of defense used so often in the past must move toward *integrated* lines of defense of the future for effective protection and thorough risk management.

The way in which we can perform this is by putting in place a construct whereby internal threat intelligence gained from one line of defense is shared among all other defenses within an organization. In the case of protecting users and the array of technologies found there, sharing of intelligence is imperative to integrating the technologies together. Similarly, in the case of protecting web applications,

these same concepts do apply. The end result will be a cohesive and integrated defensive strategy that's much more adept at blocking attacks than the standalone technologies so often deployed.

Conclusion

What I have tried to demonstrate in this chapter is that a better model exists, and the modern military is the model we need to replicate in our fight against cybercrime. Next, let's take a look at the lines of defense currently available to protect cloud-based web application deployments. When implementing web applications in a cloud environment, all these lines of defense are imperative because they all perform slightly different functions.

CHAPTER 3

Cloud-Based Lines of Defense for Web Application Security

In this chapter, you learn what lines of defense I highly recommend to fully protect web applications deployed in cloud environments. This discussion begins with the *very outside edge* of the cloud, which is where traffic enters the cloud environment from the internet. You can think of this as a boundary between the internet and the cloud resources deployed downstream. This discussion ends with the *very inside edge*, which you can think of as the very last line of defense before a web application is actually accessed by a user or attacker on the internet. All of the technologies discussed in this section make up the lines of defense in what I call the *modern cloud edge*.

Defensive Line 1: Edge Routers

Edge routers can often act as the first line of defense because they are fully capable of discarding unwanted traffic, given that they are processing it anyway. Organizations that either implement Border Gateway Protocol (BGP) FlowSpec on their own edge routers or work with cloud providers (who do the same to offload other downstream lines of defense) have discovered the best approach to defend against various attacks using this line of defense.

Although not always thought of in the terms of security (because edge routers are often managed by network teams and not security teams), as already mentioned, edge routers are fully capable of acting as the first line of defense to defend networks, websites, and

applications. Routers are fully capable of implementing access control lists (ACLs) that are designed to block certain classes of traffic as well as certain sources of traffic. Although providing traffic routing is their primary responsibility, routers can also become aware of the shortest paths, fastest paths, and alternative routes to any destination, while also providing redundancy and network resiliency. Integrating edge routers into all other defenses found downstream would serve to help protect these defenses themselves because edge routers are the first line.

When used to their fullest ability, routers can become a critical line of defense. Concerning network security and availability, the most common usage of BGP from a service provider or large enterprise perspective is to use BGP to redirect unwanted traffic to a discard interface commonly called a remotely triggered black hole (RTBH). Here, vast amounts of unwanted traffic can be discarded right at the edge of any network.

However, there is one drawback to using RTBH. This method of traffic discard normally focuses on the destination of the traffic and blocks all traffic intended for that destination. The real drawback here is that both good and unwanted traffic will be discarded, effectively taking the destination device or service completely offline.

Because of the limitations when using RTBH, BGP flow specification, or *FlowSpec*, was created as a better method of discarding unwanted traffic at the edge-router layer. FlowSpec allows specific Network Layer Reachability Information (NLRI) to be defined, which expresses additional information about traffic filters put in place and what traffic should be discarded at the routers.

Defensive Line 2: DDoS Defenses

Distributed Denial-of-Service (DDoS) attacks are the oldest known cyberthreat to internet availability. Being around for more than two decades, DDoS attacks are still the attack of choice for threat actors looking to extort money via a warning of a pending outage or to take an organization offline due a host of other motivations. Today's DDoS attacks are much more sophisticated, blending multiple attack vectors into a single barrage of outage-inducing traffic. The need for DDoS defenses to not only protect networks, but also protect DNS, websites, and applications, regardless of where they're located, is imperative to maintaining uptime.

Applying DDoS defenses as the next line of defense protects the entire cloud infrastructure against DDoS attacks. When attacks target Domain Name System (DNS) with volumetric or protocol-based attacks, DDoS defenses should be immediately engaged to protect DNS. When attacks begin targeting the networks that support websites and applications, similar defenses can be engaged to protect the network layer. Finally, when attacks against websites and applications are detected, Layer 7 (L7) DDoS mitigation points can be notified of the attacks and can be used to thwart them by utilizing L7 DDoS protection layers.

To defeat DDoS attacks, organizations must have defenses in place that address each attack at the proper protocol layer. Understanding that DDoS attacks come in many flavors (for example, attacks that take advantage of Layer 3, 4, and 7 vulnerabilities), defenses must be implemented at the appropriate layer. These defenses might include various detection and mitigation algorithms, ACLs, protocol whitelists, and IP blacklists.

Defensive Line 3: DNS

Early on in the internet, organizations first relied on their upstream internet service providers (ISPs) to effectively manage the DNS on their behalf. As organizations became more reliant on the internet for the very success of their online business models, they began hiring teams of specialists and brought DNS on premises. Measures to oversee equipment failures and circuit outages were adequately implemented; however, some organizations managing DNS completely on their own were finding it cost prohibitive and risky.

From a web application availability perspective, there is nothing more important than adequately addressing the critical nature of the internet's DNS architecture; an organization that wants to be found on the internet must have a bullet-proof DNS implementation. Implementing managed DNS as the next line of defense focuses on its capacity for ensuring the availability of web applications. Because DNS operates as the single directory service on the internet, without DNS, the internet would cease to provide the tremendous value it does today. Simply put, DNS drives availability, and any threats that would affect availability must be adequately addressed in this line of defense.

Because of the risk related to equipment failures, circuit outages, code misconfigurations, human-induced blunders, and the continuous cyberattacks on DNS, today's largest internet-dependent organizations (like social media companies that rely 100 percent on internet availability) realize that outsourcing their DNS to a managed DNS provider makes sense. Today, many organizations either outsource DNS completely, or they use a split view of DNS and outsource their authoritative DNS, while still keeping their recursive DNS on premises.

Today's cloud operators that also offer managed DNS via cloud infrastructures have, in my opinion, eliminated any possibility of experiencing DNS breakages and outages. They provide rapid change propagation, intuitive configuration consoles, zone management, and active failover coupled with zone scaling and vanity name servers. They also provide integrated traffic steering to enhance the online experience and ensure that an organization's users (customers, employees, and partners) reach the best digital asset while taking the optimal path.

Cloud-based, managed DNS providers are beginning to use automation to collect, analyze, and correlate key internet performance metrics from strategic viewpoints of the internet. Integrating billions of data points into their operations daily, they are capable of dynamically routing users to the most responsive sites and applications based on geography, internet conditions, and the organization's business models.

Because router-initiated defenses by way of ACLs and BGP FlowSpec are implemented as the first line of defense, and DDoS defenses are implemented as the second line of defense, managed DNS operating in the third line of defense fully protects against availability outages.

So far, the modern cloud edge includes:

- Edge routers
- DDoS defenses
- Managed DNS

As organizations adopt and move their public-facing web applications to the cloud, these three lines of defense must be moving

toward full integration, given that they all play a critical role in maintaining web application "availability."

First Three Layers of Protection

If attacks are being detected targeting the DNS layer, upstream routers in the first line of defense and upstream DDoS defenses in the second line of defense can be made aware of the attacks seen at the DNS layer. Automated protections can be implemented by these upstream lines of defense to protect the DNS layer.

Defensive Line 4: Reverse Proxies

Incorporating reverse proxies as the next line of defense is primarily due to the fact that these technologies are fully capable of providing considerable amounts of protection for downstream web applications. Using reverse proxies makes a great deal of sense because they are fully capable of providing a valuable fourth line of defense.

In the early 1990s, organizations began to realize that some layer of protection was needed between the internet and its earliest users. The original devices, which today we call "firewalls," originally operated as proxies. These proxies were used to provide some level of protection for users when they were perusing the internet. The proxies sent in requests to the internet on behalf of the user and provided a layer of segmentation between what is considered the "inside" of a network and what is considered the "outside." At some level, the concept of proxies within firewalls is still in use today.

Proxies are designed to protect users from the internet; the concept of "reverse proxies" is just the opposite. Reverse proxies are designed to protect the internet from users. When reverse proxies are deployed in-line in front of websites and applications, they not only hide the IP address of the actual websites and residing applications, they can protect them from users on the internet as well. We can embed many defensive mechanisms within reverse proxies.

For example, reverse proxies are fully capable of consuming threat intelligence in the form of threat feeds, which include whitelists and blacklists. ACLs and firewall-like policies are also supported. Reverse proxies can support functionality that enable them to apply traffic and request rate limits designed to limit the amount of traffic

that any given IP address can pass downstream, based upon time. Finally, reverse proxies can also protect websites and applications by enabling bot management and web application firewall plug-ins and components. Simply put, reverse proxies are a critical layer in the modern DiD approach for web applications.

Defensive Line 5: Bot Management

Positioning bot management as the next line of defense is justified because comprehensive, cloud-based bot management solutions integrated into the reverse proxies already exists. When traffic from a suspected malicious bot is received, there is no reason to allow this traffic to pass downstream. Therefore, bot management operates well as the next line of defense.

When incoming traffic to publicly exposed web applications adheres to the policy enablement and enforcement provided by the first four lines of defense, does this mean that the traffic is considered harmless? Absolutely not. Today, much of the traffic finding its way to an organization's web applications is not coming from innocuous human visitors. Instead, much of the traffic that organizations receive from the internet is coming from infected, consumer-based IoT devices—commonly called bots.

Although there is a tremendous need for good bots to visit, catalog, and store information about an organization's websites and applications, there is no reason to allow unwanted visitors in the form of bad bots to orchestrate malicious interactions with these sites and applications. Most malicious bots probe, prod, and peruse sites and applications looking for unintended vulnerabilities, taking advantage of them wherever possible. Other bots continually attempt to commit fraud, consume resources, and perform a host of other unwanted activities. To put it succinctly, if organizations have no oversight of their malicious bot problem by way of a bot management line of defense, it's only a matter of time before impact can be expected.

Is there any reason to allow traffic derived from these bots to ever find its way to the lower layers of the DiD approach or even to the exposed websites and applications? Absolutely not. The best place to defeat malicious bots is at the fifth defensive line. This defensive line's sole intention is to detect and eliminate malicious bot traffic. But how is that done best today?

Because most bots do not use the same browsers that human-run computers do when visiting websites and applications, this is a great way to distinguish bots from humans. Having a line of defense that can issue various bot challenges to detect and defeat bots is critical to lessening the damages that they can cause. As a matter of fact, most bots use what are known as "headless browsers" running from command-line interfaces. In most cases, today's bots are not running JavaScript within these browsers in the same fashion as human visitors would—and this is the key to distinguishing malicious bots from human visitors.

By having technology that forces all visitors to take a "test" (commonly called a challenge) in the form of JavaScript challenges, human interaction challenges, device fingerprint challenges, and even CAPTCHA challenges, organizations can decrease the amount of bot-induced traffic reaching their sites and applications. Most of these challenges (except CAPTCHA) are completely hidden to the human visitor, and they can eliminate the probing, prodding, and scanning that bots perform for attack reconnaissance purposes or for other fraudulent activities.

Devices that pass the challenges are allowed entry to the lower lines of defense, whereas bots that fail these challenges are blocked from any interaction with downstream websites and applications. However, all visitors are continuously monitored for changes in their overall behavior, and when they deviate from what is considered the norm, additional challenges can be invoked by a concept called *edge scripting*. This concept is used to execute additional challenges that devices will need to successfully engage before their traffic is passed downstream.

In addition, if the same sources of bot-induced traffic begin to increase the frequency of attempts to gain entry, blacklists of unwanted IP addresses can be generated, and these lists can be implemented by one, if not all the upstream lines of defense. Integrating the intelligence gained at this line of defense with all previous lines of defense makes the most sense.

Defensive Line 6: Web Application Firewalls

None of the other previous lines of defense have the ability to discern a legitimate web request from a malicious one, because the difference between the two is extremely small. Web Application

Firewalls (WAFs) are among the few technologies specifically designed to eliminate malicious web requests from attackers. This is because they have a better framework in place to understand the underlying web application. Believing that the other lines of defense can detect and block a malicious web request is imprudent, and the case for must-have WAF technology as the next line of defense is nearly irrefutable.

After incoming traffic originating from malicious bots has been removed from the traffic streams (by way of the bot management solution), the next line of defense comes into play. Cloud-based WAFs are imperative to identify attacks targeting web applications. Operating differently from network firewalls, WAFs are tasked with blocking attacks that take advantage of known vulnerabilities in commonly used web applications, in addition to attacks targeting poor coding practices. Operating as a sort of plug-in within the reverse proxies themselves, WAFs are a critical layer of defense to protect public-facing web applications.

Because WAFs have knowledge only of traffic that traverses TCP ports 80 and 443, all other incoming traffic is most often simply discarded by the other upstream lines of defense. Traffic on ports 80 and 443 is inspected by the WAFs and compared against a long list of rules that most often dictate what an incoming web request should look like. There are many malicious examples of web requests that can inject code into web applications, allow an attacker to gain privileged access, or manipulate applications to expose sensitive data, so incoming requests need to be deeply scrutinized.

This is performed by comparing all incoming web requests against a long list of rules commonly bundled into rulesets. Today, most WAF vendors have implemented the OWASP ModSecurity Core Rule Set (CRS), which contains generic attack detection rules for use with ModSecurity or compatible WAFs. The whole point of having WAF technology deployed is to eliminate malicious web requests that easily pass through all previous lines of defense.

Some WAF vendors have limited or no ability to create "custom rules" outside of the CRS, whereas other WAF vendors completely support customization. The ability to write custom rules allows for more flexibility and provides surgical detection and mitigation of very specific web requests and their associated attacks. When

researching and evaluating WAF technologies, ensure that they can support custom rules in an easy-to-implement fashion.

One of the greatest values of WAFs is for use in *virtual patching*. Because WAFs sit upstream of web applications and are normally deployed in an inline reverse proxy approach, they are a great place to put protections in place for known, yet unpatched, vulnerabilities. For example, if a vulnerability in a commonly used web application was announced by a vendor, and no patch (fix) was yet available, operators of a customizable WAF could put defenses in place by writing specific rules designed to block known exploits of a certain vulnerability. This is an excellent example of using a WAF to provide virtual patching when a vendor-supplied vulnerability patch simply does not yet exist.

Therefore, not only do WAFs protect applications and the integrity and confidentiality of data normally sitting downstream, they also play an important role in protecting the availability of applications, as well. When WAFs are tightly integrated into the upstream reverse proxies and bot management lines of defense, L7 DDoS attacks are easily detected and mitigated. Because most L7 DDoS attacks originate from bots running scripts, reverse proxies can limit the amount of incoming traffic from any device, bot management can issue challenges to detect bots and drop their traffic, and customized WAF rules can identify request patterns as an indicator of attack. Integration of these layers plays an important role in defeating all L7 DDoS attacks.

The true key to realizing the value that WAFs provide is derived from applying the appropriate rules in the proper places. Blindly applying every rule to every web application downstream often induces large numbers of false positives. Unfortunately, and in the case of false positives, many organizations either run their antiquated hardware-based WAFs in some sort of passive out-of-band mode or they set many, if not all, rules into detect-only mode when deployed inline or in cloud environments. Regrettably, this does nothing more than create a great deal of noise and a false sense of security. The recommendation here is to implement WAFs to their fullest ability and put as many rules as possible into block mode, without affecting legitimate traffic.

Beyond WAF rules and rulesets, the concept of daily application scans and vulnerability tests is highly recommended. The objective

here is to identify vulnerabilities in the implementations of web application code and use the information to provide virtual patching by way of customized WAF rules. This buys time and provides protection while code developers are fixing any issues found in their *application code implementations*, by way of the scans and vulnerability tests. In addition, some open source and commercially available application scanning and testing tools can provide detailed information regarding the results and *make recommendations* for how to shore up the defenses the WAFs are currently providing.

One challenge here is that vulnerability management teams that usually run vulnerability scans often are not the same people responsible for patching vulnerable systems or even creating rules to implement virtual patching via a WAF. My recommendation here is that there needs to be a clear line of communication and responsibility, on both ends, for prioritizing and patching systems. However, in most organizations this can be resolved.

Concerning WAF rules, do not take "detected" rules lightly. Typically, rules that fire repetitively are indications of a continuous attempt by attackers to exploit an application or, even worse, to gain access to the downstream data. However, WAF rules that continuously trigger can cause operator and analyst alert-fatigue. Often these rules are either turned off or the alerts are ignored, which can result in increased risk for organizations.

When organizations see repetitive questionable activity coming from a certain source IP address, my recommendation is to implement dynamic blacklists of the repeat offenders and block their traffic by way of the upstream lines of defense.

Finally, given that most web applications access highly critical and private data downstream, building an impenetrable moat around an organization's data is highly recommended. Today's cloud-based WAFs play a critical role in protecting applications and data, especially when organizations are moving away from brick and mortar data centers and toward the cloud.

Defensive Line 7: API Defenses

The next line of defense that organizations need to consider is for protecting their publicly exposed application programming interfaces (APIs). Today, organizations use APIs to support their mobile

apps, mobile users, and web-based partners, and APIs are becoming a major security risk that are often overlooked. Attackers understand that APIs can be manipulated to expose sensitive data, are vulnerable to man-in-the-middle (MITM) attacks, and can certainly be affected by denial-of-service (DoS) attacks, as well. Defending publicly exposed APIs as the next line of defense is becoming increasingly important.

Most people don't realize that the growth and usage of APIs on the internet is soaring higher than ever before. APIs are being used by all sorts of organizations to increase their ability to provide goods and services on the internet, while streamlining their operations and application usages by end-users and partners alike. Having methods of protecting these API from all sorts of malicious activities is becoming imperative, given that attackers have determined that poorly protected APIs are a new attack target. Like browser-based web applications, APIs can be used to expose a glut of previously unknown vulnerabilities.

Having defenses in place to protect APIs makes a great deal of sense today. From upstream reverse proxies limiting the amount of traffic any API server can receive, and bot detection and mitigation by way of implementing *security token challenges* to validate legitimate calls, to WAFs applying relative rules to API traffic, increasing levels of protection can be achieved. Because adequately protecting APIs requires the functionality of the other upstream lines of defense, having all lines of defense at hand will allow organizations to safely increase their usage of APIs well into the future.

Defensive Line 8: Caching

The final line of defense is most often thought of in the context of website speed and consistent performance—from a visitor perspective. These days, having responsive sites and applications are essential due to visitors' expectations. In the past, visitors were more lenient with slow response times, sluggish screen repaints, time-consuming downloads, and so forth. But today, visitors are generally unforgiving if there is a delay in accessing the information they desire. Therefore, caching is an important final line of defense.

Caching not only improves site responsiveness, it provides a line of defense to protect the downstream origin servers from a host of different assaults. Attackers understand that site and application

latency caused by DoS attacks can be disastrous for organizations, so having a caching line of defense can not only protect cloud-based applications feeding static content, but also protect downstream origins providing dynamic content. Although caching is not always thought of as a line of defense, it can play an important role.

One reason for this is that, at times, the upstream lines of defense might not immediately block all unwanted traffic, which is primarily due to the time from detection to "engaging" mitigation. For example, when suspected malicious bots are being challenged by the bot manager implementation in defensive line 5, small amounts of the bot traffic might leak through before mitigation is engaged. When caching is enabled, the bots in many cases are repetitively gaining access to only content that is being cached. This protects origin servers while mitigation is being engaged upstream.

Example of Caching's Value

Recently, the website of a well-known internet service provider came under an application layer (L7) DDoS attack. Because the attack was initiated by malicious bots, the upstream bot manager line of defense they had deployed was capable of mitigating the vast majority of the DDoS attack. There was a minor amount of attack leakage that occurred due to the time from detection to attack mitigation. However, the leakage was fully defeated by the caching line of defense, and the origin servers did not see the L7 DDoS attack at all. The provider's website was fully protected, and no impact was experienced.

Conclusion

In this chapter, we looked closely at the various lines of defense that are desperately needed to protect public-facing web applications deployed in cloud environments. Because each defensive line provides different functionality, and they block attacks at different layers of the overall protocol stack (that is, the OSI Model), if one defensive line is missing, not operational, or not functioning optimally, the entire defensive posture can be severely affected.

Therefore, I recommend that when your organization begins searching for a cloud provider to host your public-facing web applications,

you ensure that the provider has all of the aforementioned defensive lines in place.

Next, we discuss how all of the technologies in the lines of defense just covered can be integrated to achieve a modern DiD approach.

CHAPTER 4

How to Achieve the Integrated Approach

In this chapter, we cover the concept of *cloud edge* and *cloud core* and what technologies reside within each. Because you will hear these terms often when working with cloud environments, it makes sense to cover these terms first. The purpose of this discussion surrounding *edge* and *core* is to understand that you cannot effectively protect the core without adequately protecting the edge first. But how can organizations achieve integration with all the defensive lines previously discussed in Chapter 3? And what are the pros and cons of on-premises security operation centers (SOCs) versus outsourced SOCs? Let's take a look at the terms "cloud edge" and "cloud core" first, before moving on to the discuss how to achieve integration. Then, we end with a comparison of SOC approaches.

Cloud Edge and Cloud Core

When looking from the perspective of a visitor (or attacker) who wants to gain access to your public-facing web applications, the traffic first arrives at what we again call the cloud edge. Today, there are cloud providers that have built their cloud edge from the ground up, implementing all the security technologies listed here:

- Edge routers
- DDoS defenses
- Managed DNS

- Reverse proxies
- Bot management
- Web application firewalls
- API defenses
- Caching

Conversely, in cloud environments, you will often hear the term *cloud core*. The cloud core is where the web applications reside. Inside the core, you will often find compute, storage, connectivity, and, of course, databases containing private and highly valuable data. Also, you will often find other security-related technologies that perform encryption, access control, key management, and so forth that are more often thought of in the context of the core because that is where they most often reside.

Integrate Like a Modern Military

The modern military uses the concept of *integration* in all its defenses by way of capturing and communicating internal threat intelligence gained about the tactics, techniques, and procedures of their adversaries. This intelligence is shared across each of the preceding lines of defense as well as to the lines that follow. What is achieved here is that the lines of defense begin to work in unison, in an integrated fashion, providing synergy and cooperation between all lines of defense. The aim of integrating the lines of defense is to address the shortcomings of the original "definition," which calls for "independent" lines of defense.

Cybersecurity lines of defense must be aware of each other, much like a modern military, in order to achieve a modern Defense in Depth (DiD) approach to web application security. All lines of defense must be fully capable of sharing internal threat intelligence bidirectionally between all other lines. In addition, where one line simply does not have the ability to block something malicious, another line must be engaged that can perform the required action. Next, let's discuss how integration is achieved in cybersecurity today.

How Integration Is Achieved Today

I know of only two ways organizations can integrate the lines of defense outlined in Chapter 3: either through a single user interface

or through human expertise. Let's take a look at how these two solutions work, including the advantages and challenges of each so that you can figure out which is best for your organization.

Method One

The first method to obtain integration between the lines, is obtained by integrating the user interfaces (UIs) that provide access to all lines of defense. In most organizations, every technology in each line of defense comes with its own UI. This results in many different operating requirements, expertise, and expense. Most organizations today operate with dozens of UIs in their organizations.

On the other hand, there are some promising steps being made in the cloud. For example, some cloud-based web application security vendors offer a fully integrated UI, from which all defensive lines can be accessed, monitored, controlled, configured, and supported —all from a single screen. An integrated UI is one of the first steps that should take place in a modern DiD approach to better web application security.

Although integrating the UIs of the deployed security technologies is an advantage to the overall technology management, and it can give you the impression that the lines of defense are actually fully integrated "under the hood." Unfortunately, that's not always the case. The following is an example of what I mean by this:

Organizations often receive tactical threat intelligence from external sources in the form of threat feeds, and an integrated UI can be used to help push those threat feeds to the various lines of defense. However, one major challenge organizations face is that this is nearly always a manual process, and it does not always address the collection and sharing of internally gained threat intelligence similar to the modern military. Also, it does not address automating configuration changes on one line of defense from the intelligence gained from another line. Let's look at different approaches.

Method Two

The next level of integration being achieved today is by way of human expertise. This concept currently holds a great deal of promise. This is beginning to be performed in various organizations. For example, many of today's cloud-based web application security providers who offer the highest levels of security-as-a-

service (SECaaS) are integrating their security technologies through integrated UIs as well as by human expertise. They're integrating the aforementioned lines of defense with multiple security operation centers, operating 24/7 and fully staffed with highly competent security and networking experts. These experts are tasked with operating like a *Central Command in the Military*, integrating the lines of defense by way of proficiently utilizing *automation*, *scripting*, and *API techniques*.

An Approach Similar to the Modern Military

Figure 4-1 presents a comparison that highlights how similar an integrated DiD approach to better web application security is to an actual modern military, which operates under the same precepts, especially concerning integration.

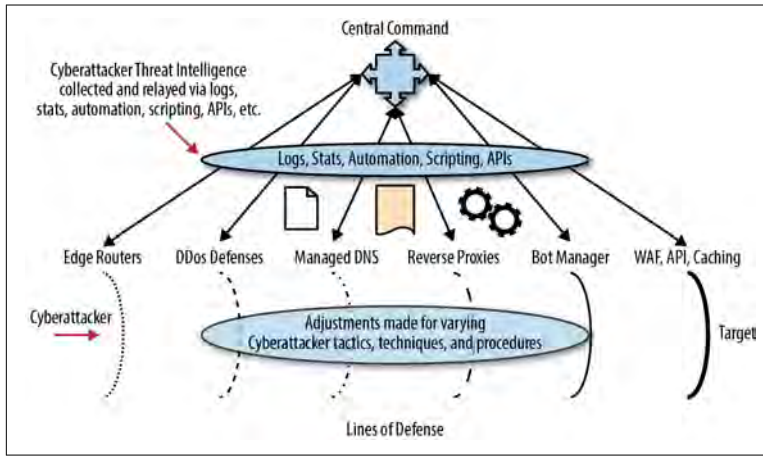

Figure 4-1. Lines of integrated defenses in the cloud

The figure highlights the integration needed to gain better web application security. Reading from left to right, the lines of defense near the bottom are very apparent. At the top, the SOC, acting like a military Central Command, receives logging and alert information from the various technologies and then uses this information to disseminate the *adjustments needed* in an automated fashion to the appropriate lines of defense via automation, scripting, and APIs. This demonstrates the true power of integration, as all lines of defense begin to act as one cohesive defensive force, similar to a modern military approach.

Security Products Support Management APIs

Nearly every security technology on the market today supports application programming interfaces (APIs). In this case, these APIs are not designed to be used like ecommerce or social media APIs. Instead, they support the gathering of information from the security technologies deployed in the form of logs, events, alerts, and even traps. The other usage for the APIs is automating security technology configurations with the ability to make changes "on the fly," often using automation. This is where the tremendous value of scripting and APIs comes in to play.

The Importance of Synergy

The synergy of automation, scripting, and APIs is one of the most vital talents required for SOC teams today. When people hear the term "APIs," they immediately think of application programmers because they are commonly involved with utilizing today's APIs. However, in this case, APIs are an extremely powerful tool for security experts who have mastered scripting techniques. When organizations are searching for security experts to be added to their SOC teams, finding those who have extremely high levels of understanding in relation to automation, scripting, and APIs is highly recommended. Let's take a look how automation, scripting, and APIs operate within the context of a SOC.

When a log (an alert or event, among other things) is generated by one of the lines of defense, this log is received at a centralized logging system located somewhere in the SOC. At that point, there are two approaches that can be taken. One is to have humans acknowledge the log, figure out what the log means, and then determine whether the log can be acted upon with regard to the other lines of defense. However, a more modern approach would be to receive the log and then automate the calling of a preconfigured script that takes some sort of action on one or more lines of defense, by way of making automated changes through the technologies' APIs.

Common Example

In the case of latter lines of defense, if one of these lines generates a log or alert pertaining to a repeat offender, a script can be called to

set up a blocking function at a preceding line of defense quite easily by making a simple change via an API. The concept of scripting is quite powerful due to the ability to write a script one time and then repeatedly call that script to convert a log or alert into an *action* with very little, if any, human interaction. To help to explain this better, let's observe the following scenario.

For instance, let's say the Web Application Firewall (WAF) line of defense detects a steady stream of dissimilar web requests that all appear to be malicious, repeatedly coming from the same source IP address (source). The source is not violating any access control list (ACL) rules on the upstream routers, and the source is not participating in a DDoS attack. The source is not attacking the DNS, and it is performing the required TCP three-way handshake with the upstream reverse proxy. The source has a browser with JavaScript enabled and passes all bot challenges, yet the WAF confirms that the source (likely being controlled by an attacker) is trying its best to break into the web application downstream. Can you defeat this activity upstream? Absolutely.

The best way to block this activity is to automate the calling of a script based upon the attacker source IP address, port, protocol, and behavior and then make a change to all preceding lines of defense via their APIs to block the source for a short amount of time. If the offending source eventually stops the unwanted behavior, another script can be called to remove the block and allow that source through as long as it continues to exhibit good behavior. No one would want to block the source IP address indefinitely due to the potential for IP address spoofing, which is very common. In this case, a short-term block is all that is needed.

Although in the early stages of an SOC, much of this is being performed via human intervention. As the SOC team and its support approaches mature, much of this activity can be fully automated. This is the true power being wielded in the hands of today's advanced SOC personnel.

Value of Intelligence

Beyond the usage of scripts and automation performed by the SOC team, the value of tactical and strategic threat intelligence can be realized. The intelligence gained by "internal means" can be put into action automatically, making it "actionable" threat intelligence. This

actionable concept also includes putting threat intelligence gained from external sources into combat, as well. What is achieved is sharing of intelligence across all lines of defense, from the entire edge to the core, and it can eliminate independent lines of defense once and for all.

Comparing On-Premises SOCs and Outsourced SOCs

Many enterprises today have invested heavily in their own on-premises SOCs, which is a great step in the right direction. These on-premises SOCs include a great deal of logging technology (security information and event management [SIEM]) most often manned by expert security analysts. The advantages of the on-premises SOC equates to measurable improvements in detecting and mitigating attacks, which results in better security.

However, there are a number of challenges facing the on-premises SOC solution: there is a shortage of available analysts and security experts (which is affecting the cybersecurity industry overall), small organizations often cannot afford the salaries these experts are paid, and SOC expert retention rates are poor because their opportunities for career advancement abound. And there is one drawback to this solution: because the on-premises SOC is working to defend a single organization, their view of the worldwide cyberthreat landscape is somewhat limited to the attacks targeting their own networks, users, and web applications, and so it can be difficult for them to obtain and quantify the broad picture.

On-Premises SOCs are Making a Significant Difference

Many on-premises SOC teams are making great advances in protecting their organizations against cybercrime and shortening the time from device "infection" to attacker activity "detection," especially if they are moving toward full integration of the lines of defense in their organizations, and when SOC personnel are competent in automation, scripting, and APIs.

In comparison, one of the benefits that an outsourced SOC offers is the value of the crowd-sourced knowledge gained from the many different customers they support daily. Today's cloud-based provid-

ers gain and share information across their entire customer base concerning internet routing conditions, the current state of DNS worldwide, global DDoS-related outages, latest and greatest botnets and their infected hosts, new attacker tactics, techniques, and procedures, latest vulnerability information, and more.

Advantage of Outsourced SOCs

Suppose, for instance, that one of the customers being managed by an outsourced SOC is experiencing a new attack vector, a previously unseen source of attack, or some trend or another. The intelligence gained regarding attackers' tactics, techniques, and procedures from that customer alone can be shared, in an automated fashion, via scripting and APIs, to shore up the defenses for every other customer. This has tremendous value because it nearly eliminates the concept of "every man for himself."

Many agree that there is currently a skills gap in the cybersecurity industry overall. This gap can be improved through collective human oversight by way of outsourced SOC teams managing the security postures of multiple customers simultaneously. This is the whole point of SECaaS, whereby human-based resources are shared among the masses. When automation, scripting, and API usages are in force, the few can quickly and completely support the many.

However, there may be one important drawback when outsourcing your SOC, and it has to do with privacy. Most organizations do not want to share the fact that they are under attack with other organizations for a host of different reasons, which is understandable. Today, especially in the light of the EU's General Data Protection Regulation (GDPR) and other like regulations, privacy is a major concern and can never be taken lightly. My advice if you are considering an outsourced SOC is to make sure the provider shares only the source of attacks with others and keeps the target identities private.

Conclusion

In this chapter, we covered two methods of integration to empower you to do the same, similarly to the way a modern military operates. We discussed the importance of the synergy that you can obtain by providing examples of how my recommendations can be imple-

mented. Finally, we looked at the tremendous value of actionable intelligence and ended with a discussion about the benefits and challenges of different SOC approaches to help you decide what's best for your organization moving forward.

CHAPTER 5

The Future of Defense in Depth

Today, we observe attackers who control vast numbers of infected machines (bots) conscripted into their botnets. These bots are integrated with the attacker and are often fully aware of one another. Thus, we now are seeing automation and even machine learning being used by attackers themselves in their fight against us. Can organizations take advantage of machine learning as well? The answer is yes.

In this final chapter, I explain how security operation centers (SOCs) can use supervised machine learning (SML) in the fight against attackers. I've included a checklist that will help your organization prepare for the future of SML and outlined steps that you can take to achieve success.

What the Future Holds

As more organizations move their web applications to one of the many cloud environments operating today, the entire industry will need to shift more toward integrated lines of defense, grouping technologies together based upon where they operate in the protocol stack and where it makes the most sense. For instance, there are already cloud security-as-a-service (SECaaS) offerings available today whereby the independent lines have already been eliminated through singular user interfaces (UIs), human-based oversight, automation, scripting, and the usage of application programming interfaces (APIs).

In the very near future, as learning-enabled machines observe the operations of SOC personnel and when these machines begin to perceive repetitive actions performed by the SOC, these very same machines will be capable of learning from their human counterparts and begin to perform the very same steps. This will require human control over the machines, the aforementioned SML.

For example, when a log is received from the various lines of defense into the SOC, a learning-enabled machine will be able to detect that an attack is taking place and act immediately, on its own. This might include calling and executing the appropriate script to change one or many configurations on the various lines of defense via APIs and put nearly immediate protections into place.

This activity will not eliminate SOC personnel. Instead, it will give them the automated and advanced weaponry needed to defend against today's dynamic threats. This view of the future is not based upon conjecture; rather, we can already observe it in some mature, cloud-based SOC environments.

The SOC personnel of the future will spend most of their time managing the SML process, and their focus will be on creating foolproof feedback loops to ensure that the machines do not inadvertently make a mistake on their own. The industry is getting very close to realizing the full potential of SML in the context of the modern Defense in Depth (DiD) approach.

To take advantage of the future integrated lines of defense found in your own organizations and your cloud implementations, there are a few concrete things you need to be doing now if you want to be ready for (and be part of) this vision. This includes employing learning-enabled machines to provide complete oversight that will lead to the full and automated integration of the lines of defense your organization uses daily. To prepare, you'll need to do the following:

1. Acknowledge that SML, automation, scripting, and APIs are the way of the future.
2. Define the various lines of defense in your own organization and fully understand how they operate, where they operate, what they do best, and the deficiencies of each one.

3. Fully understand the detection and mitigation capabilities of each line of defense in the context of what they're capable of detecting and mitigating within the overall protocol stack.
4. Determine whether the technologies you've implemented today fully support configuration and monitoring capabilities via APIs. If not, seriously consider replacing them.
5. Begin to develop and train your internal staff on the concepts of automation, scripting, and APIs within the context of making configuration changes "on the fly" to the various security technologies deployed.
6. Begin to attract and hire SOC and network operations center (NOC) personnel that fully understand automation, scripting, and configuration changes via security technology management APIs.
7. Set up test-bed and simulation environments to mimic your own circumstances and use these to experiment and learn how best to take advantage of automation, scripting, and APIs.
8. Search for vendors who agree with the approaches found in this book and who can provide recommendations on how to integrate the various lines of defense in your organization.
9. Invest in SML training, technologies, and approaches and set aside budget for research and development to create your own machine learning tools on-premises.
10. Thoroughly scrutinize vendors that say they already have artificial intelligence (AI) in place today, considering that true AI is quite a few years away from being a reality within the context of information security and our current lines of defense.
11. Do not attempt to oversell the promise of AI into your organizations just yet. Instead, focus on SML, automation, scripting, APIs, and integration because this is where the *measurable gains* in cyberdefense will be obtained first.

Now, let's examine my prediction concerning the use of good bots within your own lines of defense. The concept of good bots is nothing new in the light of Googlebot, Bingbot, Yahoo Bot, and other "good bots" that provide a valuable service in the way the internet operates today.

Using Good Bots to Your Advantage

It is possible to create internally commissioned *good bots* that can run as daemons in the background on the current and future technologies that comprise the various lines of defense. You could use these good bots across the infrastructure that provides device and technology management for one line of defense to learn more about what the other lines of defense do, what they are capable of, or under which current conditions they're operating.

Because most technologies have physical, logical, and other limitations, and when these limitations are close to being exceeded, most technologies will send an alert in the form of an SNMP trap or Syslog message in the hope of alerting their technology operators that a stressed condition exists—and might be increasing. When operators do not take an action that will alleviate the reason for the stressed condition, all sorts of negative repercussions can be experienced. All technologies have their limitations.

For example, if a moderately sized distributed denial-of-service (DDoS) attack is effectively being blocked by an edge router's access control lists (ACLs), but the router's processing limits are about to be exceeded, a good bot perusing the lines of defense could become aware of the situation, alert SOC personnel to act, or even initiate a change on its own to take evasive action. This action could include removing the ACL blocking the attack on the router, letting the attack leak through, and then blocking the attack with the downstream DDoS defenses, instead. Given that the DDoS defenses are the very next line of defense that the traffic will encounter, it can easily be blocked due to resources that this line still has available. This concept can be compared to a fallback maneuver performed by a military whose line of defense is about to be overrun.

In this case, all lines of defense can be made aware of any processing limit that's about to be exceeded and can automatically offload (fallback) attack traffic to some other line to ensure that no latency or outage is incurred due to overconsumption of available resources. When thinking about the usage of SML, automation, scripting, and APIs, nearly any idea can be conceivable.

In Conclusion

As you've seen, *integrated lines of defense* are desperately needed—both within the enterprise and in the cloud. Because the entire concept of cybersecurity is all about providing availability, confidentiality, and integrity to systems, applications, and the data we hold essential, integration today is a must-have to be successful in the modern cyber battlefield. The way a military battlefield operates is a great example of how to integrate our defenses today, whether you're using an on-premises or outsourced SOC.

Cloud providers today must move toward full integration of their lines of defense, or their customers will likely experience similar data breaches as seen throughout the internet. If that happens, it could seriously slow down or even halt cloud adoption overall. Modern, cutting-edge SECaaS and infrastructure-as-a-service (IaaS) providers already are bringing some of these concepts to reality.

It's only a matter of time before all organizations (enterprise, SMB, government, education, healthcare, finance, and so forth) will begin to move in this same direction and implement methods to integrate their own defenses and make measurable improvements to their security postures. The same concepts and approaches discussed throughout this book can be applied to the independent and standalone security technologies so often found in these organizations today.

Finally, following the recommendations herein, not only can organizations experience the *fastest time to attack detection*, they can also achieve the *most effective attack mitigation*, likely for the *lowest possible cost*. Simply put, this is where the industry is headed today.

About the Author

Stephen Gates brings more than 25 years of computer networking and information security experience to his Edge Security Evangelist & SME role at Oracle Dyn. He helps service providers, hosting providers, CDNs, and enterprises solve their DDoS and web application security problems. He has an extensive background in the deployment and implementation of on-premises and next-generation cloud security solutions.

He is a published author with a Master of Science in Information Security and Technology Management and is in demand as a security thought leader and presenter at multiple industry events.