Protecting Websites from Attack with Secure Delivery Networks

David Gillman, New College of Florida and Akamai Technologies
Yin Lin, VMware
Bruce Maggs, Duke University and Akamai Technologies
Ramesh K. Sitaraman, University of Massachusetts Amherst and Akamai Technologies

Secure delivery networks can help prevent or mitigate the most common attacks against mission-critical websites. A case study from a leading provider of content delivery services illustrates one such network's operation and effectiveness.

The Web has become an indispensable medium for conducting business, performing financial transactions, accessing news and entertainment, playing games, and interacting with government and other types of services. We have come to expect the websites that facilitate these activities to be available at all times and to perform well.

However, threats against such sites have never been greater. As Akamai Technologies, a leading provider of content delivery services, reports in its *State of the Internet* for the third quarter of 2014,[1] distributed denial-of-service (DDoS) attacks against its customers are increasing in terms of both the bandwidth and the number of requests generated by the attackers. From 2009 to 2014, the size of the largest attack, in gigabits per second, grew year by year from 48 to 68 to 79 to 82 to 190 to 321. At the same time, the number of packets per second in the largest attack grew from 29 million to 169 million. The number of DDoS attacks detected and mitigated has also more than doubled over the past two years, reaching 5,634 in 2014.

Beyond DDoS, online theft of data such as credit card numbers, personally identifiable information, business secrets, and login credentials has also grown rapidly. In fact, Web application attacks are the most common cause of data breaches today.[2] Not surprisingly, the financial cost of DDoS attacks and other forms of cybercrime is increasing rapidly. In a survey of 277 companies in 16 industry sectors,[3] the Ponemon Institute found that the average financial cost of cybercrime to survey participants was US$11.6 million in 2012. That cost is expected to grow rapidly in succeeding years as attacks increase in size, frequency, and sophistication.

Attacks impact every segment of the Internet ecosystem. In Q3 2014, those against Akamai customers spanned every segment of online services. The

0018-9162/15/$31.00 © 2015 IEEE

heaviest-hit industry sectors were gaming (33.67 percent), media and entertainment (23.65 percent), software and technology (19.44 percent), financial services (9.22 percent), and Internet and telecom (8.82 percent). Further, attacks originated in every corner of the globe, with the largest sources being the US (23.95 percent), China (20.07 percent), Brazil (17.6 percent), and Mexico (14.16 percent).

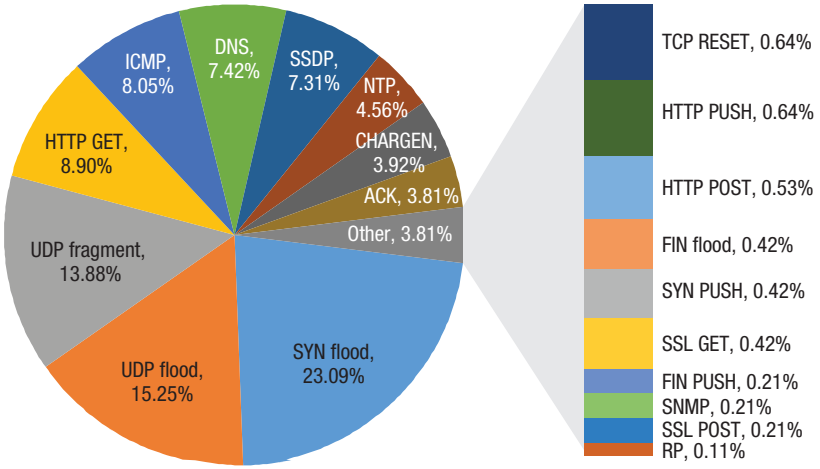

FIGURE 1. Classification of distributed denial-of-service (DDoS) attacks that occurred in Q3 2014 on websites hosted by Akamai Technologies. Attackers can exploit a variety of protocols to conduct DDoS attacks, especially volumetric and reflection/amplification attacks. CHARGEN: Character Generator Protocol; ICMP: Internet Control Message Protocol; NTP: Network Time Protocol; RP: Reserved Protocol; SNMP: Simple Network Management Protocol; SSDP: Simple Service Discovery Protocol; UDP: User Datagram Protocol.

THE ATTACK LANDSCAPE

DDoS attacks employ a wide range of mechanisms, as Figure 1 shows.[1] To give a better sense of how such attacks work, we describe two popular types: volumetric attacks, also known as floods, and reflection/amplification attacks. In addition to DDoS attacks, we describe attempts to steal data using Web application exploits. Our goal here is not to be comprehensive but only to provide a flavor of common current attack modes.

Volumetric attacks

A flood attempts to overwhelm some component of the platform hosting the website by sending fake requests to the site. Imposing a very large demand for the platform's resources can degrade or even completely deny service to legitimate users.

SYN floods are the most common, constituting nearly one-quarter of all reported DDoS attacks on websites hosted by Akamai in Q3 2014.[1] A SYN flood works as follows. To establish a TCP connection with a server, a client sends a packet with the SYN flag set. To acknowledge receipt of the SYN packet, the server sends back a packet that has both the SYN and ACK flags set. The client then completes the "three-way handshake" by sending a packet with the ACK flag set, thus establishing the TCP connection. In a SYN flood, the attacker, acting as a client (or clients), sends a large number of SYN packets to a webserver, but never responds to the server's SYN-ACK packets with ACK packets. After sending a SYN-ACK response, the server waits for an ACK packet from the client, which never arrives. The large number of "half-open" TCP connections, where the server is waiting for an ACK packet, tie up memory on the server, leaving too little to serve legitimate users. The flood could also exhaust bandwidth resources of network components en route to the server.

UDP floods are the second most common, accounting for 15 percent of the DDoS attacks against Akamai customers in Q3 2014.[1] The User Datagram Protocol is connectionless and thus does not require an initial handshake between the client and server. To prevent firewalls from filtering these packets, attackers often use spoofed IP source addresses for the packets in the flood so they appear to originate from multiple legitimate sources. Attackers can also randomize the port to which UDP packets are sent to subvert port-filtering firewalls.

Besides network-layer protocols such as the Internet Control Message Protocol (ICMP), and transport-layer protocols such as the Transmission Control Protocol (TCP) and UDP, attackers often exploit application-layer protocols. The most common application-layer attack is a *DNS flood*, in which the attacker generates numerous Domain Name System requests and, typically, directs them at the authoritative name servers of the target website. When the name servers' resources are exhausted, legitimate users cannot receive valid DNS responses. Unable to resolve the website's domain name, the name servers deny service to legitimate users.

Large-scale volumetric attacks often present multiple types of floods simultaneously. For instance, the largest attack campaign measured by Akamai used both SYN and UDP floods, and generated 321 Gbps of bandwidth and 72 million packets per second at peak. Such attacks typically employ botnets of personal computers and servers infected with malware. A more recent trend is for attackers to take over other sorts of devices commonly deployed in small enterprises or at home using ARM-based DDoS binaries.[1] Such devices include cable modems, mobile devices,

embedded devices, and home electronics such as printers. These are often easier targets because they are managed by individuals or small organizations without the benefit of sound security training and procedures.

Reflection/amplification attacks

Volumetric attacks are symmetric in the sense that the resources expended by the attacker are comparable to those wasted by the target. Hence, to overwhelm a target, an attacker must have more resources than the target. While attackers may not be greatly concerned about expending the resources of a compromised computer in a botnet, they nevertheless often employ a technique called *reflection* to amplify their attacks. A distributed reflected DoS (DRDoS) attack could enable a perpetrator to direct more than an order of magnitude more traffic at a target than the attacker directly generates. This amplification introduces an asymmetry between the attacker and the target by allowing the attacker to create more attack volume with fewer resources.

A common DRDoS attack seen by Akamai[4] works as follows. The attacker sends numerous name resolution requests to recursive name servers around the world—often open resolvers that any client can use. The attacker does not place its own IP address in the source address field of its request packets but instead uses the target's IP address. Hence, the resolving name servers send their responses back to the target instead of the attacker. This technique is called reflection because the attack traffic originates from third-party name servers rather than directly from the attacker. Reflection makes attack traffic more difficult to identify and filter because the traffic seems to come from legitimate sources. The DNS protocol also provides the attacker with an amplification mechanism. A 64-byte DNS request packet sent by the attacker could result in over 3 kilobytes sent to the target. The increase in the attack traffic by a factor of 50 or so is the amplification factor. Attackers often choose DNS requests that provide the largest responses—for example, by using the query type ANY, which returns all record types for the domain name as a part of the response.

Other protocols used in reflection/amplification attacks include the Network Time Protocol (NTP) and the Simple Network Management Protocol (SNMP).[5]

Web application exploits

Although DDoS attacks attempt to bring down websites, other exploits aim to steal private and sensitive data by leveraging vulnerabilities in the Web application. The Open Web Application Security Project (OWASP; www.owasp.org) lists numerous exploits, two of which Akamai sees frequently: SQL injection attacks and cross-site scripting (XSS) attacks.

SQL injection attacks. These attacks "inject" a portion of an SQL query into a user response to modify an SQL query that is executed by the Web application's back-end database. If the exploit is successful, the attacker can read or modify private content from the database, resulting in theft or loss of data.

As a simple example, suppose that a website has an HTML form for entering a user ID. When the user enters the ID and submits the form, the website's database executes the SQL query `select * from infotable where userid =` ⟨`input`⟩. A malicious attacker could input "50 or 1 = 1," resulting in the following query being executed: `select * from infotable where userid = 50 or 1 = 1`. Since 1 = 1 evaluates to true, the `where` clause always evaluates to true. Thus, the query returns all records in `infotable`, potentially revealing private data to the attacker. If the table contains login credentials with administrative privileges, the attacker could potentially take over the entire website.

This attack is easily prevented by type-checking the inputs—that is, a valid input must contain a single numerical value for `userid`. SQL injection attacks, however, remain among the most popular for taking over or defacing websites. As a recent example, the Syrian Electronic Army, a "hacktivist" group sympathetic to President Bashar al-Assad, used an automated SQL injection tool to enter, compromise, and deface major media websites.[6]

XSS attacks. These involve an attacker introducing a malicious script into a dynamically generated webpage on a trusted website. When the victim's browser downloads the page and executes the script, the script might extract private information from the victim's computer, or perform actions in the guise of the victim, without the victim's awareness. In a "cookie-stealing" attack, the script relays the victim's cookies to the attacker who then uses them to access private information about the victim on the website.

Consider the simple illustration of an attacker who inserts a malicious script into the Comments box of a news article accessed by many users. The script might use a script tag to download an external JavaScript snippet written by and served by the attacker, to be executed by the user's browser: ⟨`script src = "http: //theftsRus.com/script.js"`⟩⟨`/script`⟩.

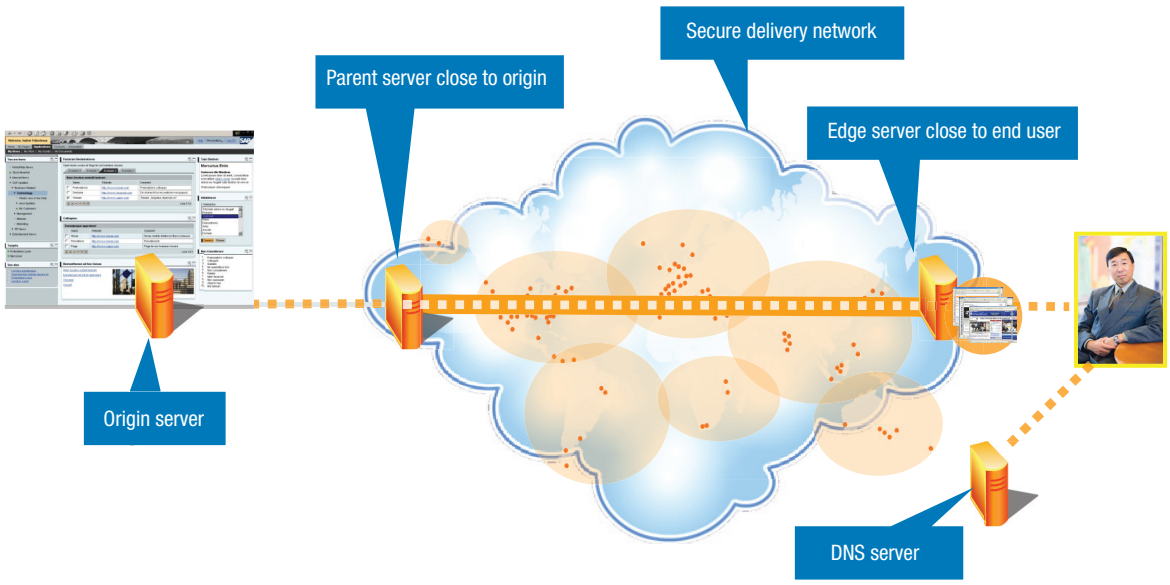

FIGURE 2. Akamai's secure delivery network employs four types of servers—DNS, edge, parent, and origin—to stop DDoS attacks and data theft.

SECURE DELIVERY NETWORKS

Protecting websites is challenging not only because of the variety of possible attacks but also because of the Internet's open architecture, which is based on mutual trust. Although this has been a key to the Internet's success, it is also its single biggest security flaw. "On the Internet," goes the adage, "nobody knows you're a dog."[7] The Internet does not provide a standard means for authenticating website user identity, enabling attackers to operate under numerous untraceable aliases that make it difficult to differentiate them from normal users.

Another challenge is navigating the tradeoff between website security and the user experience. Most techniques that increase security also degrade performance; screening every user that visits a site to prevent attacks might slow down the site enough to make it unusable, inadvertently reaching the same end state desired by the attacker.

With few exceptions, individual content and service providers cannot afford to single-handedly defend themselves against DDoS attacks and data theft. With such attacks now reaching hundreds of gigabits per second, attack traffic directed at a website might be several orders of magnitude larger in volume than the site's normal daily peak. Overprovisioning the website in anticipation of such an attack is often prohibitively expensive.

To protect its customers' mission-critical websites, Akamai has built a *secure delivery network*. An enhancement of the content and application delivery networks[8,9] that host and deliver most major websites today, the secure delivery network consists of four architectural components that each constitute a potential target of attack:

- *DNS servers*. Thousands of DNS servers distributed around the world resolve the URLs requested by users into the IP addresses of the edge servers that deliver the requested content or service.
- *Edge servers*. About 150,000 distributed edge servers cache and deliver content (whole or partial websites) to users and forward requests for uncached content to the origin, potentially via an overlay network of parent servers. These servers act as a distributed firewall to stop DDoS attacks and data theft at the network edge.
- *Overlay network*. Several thousand parent servers route communication between the origin and the edge servers.
- *Origin*. Operated by the content or service provider, the origin consists of servers that run proprietary applications and databases containing confidential data such as user logins, personally identifiable information, and proprietary business data.

Figure 2 shows how these components interact as a user accesses a website hosted on the secure delivery network.

Akamai's secure delivery network routinely serves tens of terabits per second of traffic and can be used as a shield to absorb attack traffic. In September 2014, for example, traffic on Akamai's network averaged over 15 Tbps, with regular sustained spikes of over 5 Tbps in the traffic of individual customers. By contrast, the largest DDoS attacks seen to date are a few hundred Gbps, a factor of 20 smaller than spikes of customer traffic.

PROTECTING THE DNS SYSTEM

A website hosted on Akamai's secure delivery network typically grants authority for its domain names to Akamai's DNS

system. When an attacker sends a flood of DNS requests to bring down the website, they are received by Akamai's system of name servers. Processing a large volume of requests can exhaust a name server's resources, making it unable to respond to new requests and resulting in DoS. Also, the sheer volume of incoming packets could exceed the capacity of the link connecting a name server to the Internet. The router behind the link will then fail to forward some packets, and the requests might never reach the name servers, also resulting in DoS.

The secure delivery network protects the DNS system in two ways.

The first is to deploy numerous name servers in many locations. The DNS system must respond quickly to requests from all corners of the globe, even when under attack. The availability of redundant DNS capacity around the world and the use of IP anycast routing[10] make it harder for an attacker to overwhelm the name servers even in isolated parts of the world. Switches at each location further balance traffic to the name servers behind them by hashing the source port and other parameters, balancing the load at the level of individual recursive name servers.

Akamai's secure delivery network also filters out potentially malicious DNS requests at the earliest possible instance. DNS request packets have a fixed size; any larger packet can thus be ignored. DNS requests can also be filtered based on the source IP address if, for example, the address generates an anomalous rate of requests. Further, each Akamai name server runs its own firewall in its Linux kernel that is configured using `iptables`. The kernel firewall can be configured dynamically to filter out attack traffic. For instance, as soon as the attack vector is known, a new configuration can be deployed in real time to filter out the attack traffic.

DEFENDING AT THE EDGE: WEB APPLICATION FIREWALL

To access a website hosted by Akamai, a user first connects to one of the edge servers. The edge servers cooperate to implement a distributed Web application firewall (WAF). The WAF runs a security module on each edge server that classifies every Web request as safe or unsafe. The edge server serves all requests deemed safe; it either drops an unsafe request or serves it after raising an alert flag. The security module can further analyze and classify alert flags to detect abnormal request patterns that could be symptomatic of an attack.

Rules and policies

The WAF security module applies several rules to every request. Each rule specifies a condition that the HTTP request might satisfy, usually based on the contents of request headers and message body as well as the sender's IP address. The condition may be evidence of a valid request, such as the presence of the required header fields, or a security threat, such as the presence of signatures of a well-known DDoS attack tool. In either case a rule is said to be violated when there is evidence of a security threat, either by the condition's presence or absence.

A single rule violation is rarely persuasive evidence one way or another, but is considered an anomaly. The security module applies sets of rules to each HTTP request to detect complex combinations of anomalies that fit specific attack profiles—for example, one combination might indicate an SQL injection attack, while another might indicate XSS. The module commonly employs *anomaly scoring*, which associates each rule with a tuple consisting of a numerical anomaly score and an attack category that the rule is attempting to detect. Given a set of violated rules, the module obtains a total score by summing up the scores for each attack category. If the total score in a category exceeds a specified threshold, then the module takes action such as denying the request.

As a simple example of anomaly scoring, a rule that looks for the string 1 = 1 in the query string of the requested URL could add +3 to the anomaly score in the SQL injection attack category. If the cumulative anomaly score in the SQL injection category across all rule violations exceeds, say, +20, then the request would be categorized as a threat.

When the security module categorizes a request as a threat, it either denies the request or raises an alert and serves it normally. Given the danger of denying service to valid traffic in reaction to false-positive categorization, it is customary to err on the side of caution, scoring new rules conservatively and introducing new rule sets in a staged fashion so the violations initially only produce alerts without denying requests.

Rule types

The WAF security module applies five types of rules, which differ in the evidence they use and the threats they detect.

Network-layer controls. These rules block or allow requests based on the sender's IP address. The edge servers have access to a comprehensive database of geographic

AS THE EDGE SERVERS FORM A DEFENSIVE PERIMETER AROUND THE NETWORK, ONLY TRUSTED SERVERS INTERACT WITH THE ORIGIN SERVERS.

and network information about each IP address. Using this database, the security module can implement geographic and network-based blocking of requests—for example, all requests from a specific ISP in a specific country. This can ameliorate the impact of volumetric attacks, provided the attack origin is not too diverse and can be determined. Network-layer controls, however, can be ineffective or even counterproductive if the source IP addresses of the attack requests are spoofed.

Adaptive rate controls. The security module can classify clients according to their request pattern and can filter or rate-limit requests based on this classification. To track clients accurately, it is often necessary to use additional information about the request such as the user agent, cookie, and session key, in addition to the source IP address, since multiple clients behind a proxy all share the same IP address. Fine-grained tracking of this sort can detect and filter out attackers behind a proxy but leave bona fide traffic from the same proxy unaffected. A typical example of a rate-control rule will rate limit or block all clients with the same IP address, user agent, and session ID that exceed a particular request rate threshold. To implement the rate controls, the security module has access to a database that tracks client requests with the associated cookie, session ID, and other data.

Application-layer controls. The security module can inspect the header and body of HTTP requests to protect against Web application exploits such as SQL injection and XSS attacks.

Client reputation rules. Attackers often disappear, only to reappear later to attack the same website or a different one. Akamai's secure delivery network has a large amount of historical information across multiple websites about individual clients, as identified by their IP address, software characteristics, and device fingerprints. The WAF security module uses this information to classify clients' behavior with the goal of detecting and stopping attacks earlier. To achieve that goal, the module maintains a profile of each client in a number of different contexts, and assigns the client a reputation score in each context. The module modifies the reputation score in real time based on actions performed by the client. For instance, if the client successfully completes a CAPTCHA, it is unlikely (though possible) that the client is a bot. A content or application provider can dynamically set thresholds on reputation scores to allow or deny access by the client.

OWASP core rules. The WAF supports the extensive core rules set developed by OWASP. The rules set detects various types of Web attacks including HTTP protocol violations, bots, and Web application exploits.

SECURING THE ORIGIN

The origin is the authoritative source for content and applications that Akamai's secure delivery network–hosted websites deliver to users. The content or application provider operates the origin infrastructure that must be protected from attack. As the edge servers form a defensive perimeter around the network by stopping requests from attackers before they can reach the origin, only the network's trusted servers interact with the origin servers. In fact, the locations and IP addresses of the origin servers need never be revealed to the outside world. Even if the attacker somehow discerns the origin servers' IP addresses, the origin can still apply access control lists to filter out any requests that do not originate from the secure delivery network's servers.

The edge servers cache a significant portion of the website's content, and as a result only a small fraction of requests are fetched from the origin. Even for webpages that are dynamic and uncacheable, as with banking applications, a significant portion of the page—including Cascading Style Sheets, JavaScript, and image files—is still cacheable. For a typical website hosted by Akamai, the amount of traffic served by the edge servers could be 25 times larger than that served by origin servers and parent servers. Hence, an attacker who requests content that is at least partially cacheable will be less likely to overwhelm the origin infrastructure.

A *cache-busting attack* attempts to overcome the caching defense by forcing a large volume of attack requests to be sent back to the origin.[11] Consider the following simple example. Some websites ignore query strings in URLs, while others customize their content based on query strings. The attacker chooses a target website that ignores query strings in its URLs, then makes a large number of requests for a large object on that website. The attacker makes it appear as if each of those requests is for a distinct object by appending a randomly generated query string. If the edge servers are not correctly configured, they would treat each of those requests as if it was for a distinct object. Since the object cannot be found in cache, the edge server will fetch it from

the origin, effectively transferring the entire attack to the origin. A correctly configured edge server, however, will use only part of a URL as a key for its object cache. When the query strings are not part of the keys, the edge servers foil the attack by serving the requested object from cache without fetching it from the origin.

CASE STUDY: OPERATION ABABIL

Modern attacks can use many techniques simultaneously and may persist for months. Here we describe the impact of, response to, and lessons learned from one series of attacks in 2012–2013 on Akamai's customers dubbed Operation Ababil.

Attack phases

Beginning in September 2012, a hacktivist group identifying itself as the Qassam Cyber Fighters (QCF) carried out a four-phase attack on the websites of multiple financial institutions using the BroBot botnet, a large collection of compromised WordPress and Joomla content management systems and virtual private servers with a great supply of bandwidth.

- *Phase 1*. From mid-September through mid-October 2012, QCF attacked one or two different bank websites each day with cache-busting attacks and floods of large UDP packets intended to saturate the DNS servers of the banks and their ISPs.
- *Phase 2*. From mid-December 2012 to late January 2013, QCF used short, widely separated, high-volume bursts of traffic to defeat rate-limiting controls and to probe for vulnerabilities in several bank websites. The attackers appended random query strings and query values to requests for cache-busting and to evade firewall filtering.
- *Phase 3*. From late February to mid-May 2013, QCF carried out a wave of attacks on multiple financial institutions that focused on application-layer (layer 7) protocols.
- *Phase 4*. For several hours in late July and again in mid-August 2013, the attackers, employing more complex obfuscation, discovered and exploited more content management system vulnerabilities and used fake plug-ins to infect multiple files in the websites of a few institutions.

> MODERN ATTACKS CAN USE MANY TECHNIQUES SIMULTANEOUSLY AND MAY PERSIST FOR MONTHS.

Impact and response

A Phase 1 attack aimed at blocking resolution of a customer's domain name on one day in September 2012 created a spike in DNS request traffic that lasted for about 4.5 hours. Traffic during the peak hour averaged over 23 Gbps, nearly 10,000 times normal for that domain name. The attack packets were missing valid DNS headers and stuffed with an abnormally large payload of around 1,400 bytes. The packets were directed toward UDP port 53. The attack included a SYN flood against TCP port 53, which is also used to receive DNS requests.

During Phase 2 of Operation Ababil, one target website's access rate was 9,000 pages per second, 90 times the normal rate. Over 90 percent of the attackers' requests were denied by firewall rules and not forwarded to the origin, saving it from a significant surge of traffic.

On the morning of 5 March 2013, an Akamai customer faced one of the largest attack waves of Phase 3. The attack traffic peaked at 4 million requests per minute, 70 times the normal level of traffic for that site. The assault lasted around 20 minutes. Of the more than 2,000 BroBot agents that were identified, 80 percent were new IP addresses that had not participated in earlier campaigns. The attack focused on PDF files with random query parameters, marketing pages for new customers, and login pages of financial organizations' websites.

Alerted by the abnormal number of forward requests to the origin server, Akamai's secure delivery network immediately detected this attack and blacklisted 1,700 of the new bot IP addresses without operator involvement. Various WAF rules blocked most of the remaining attack traffic. Consequently, the traffic forwarded to the origin server remained below 1 percent of the link capacity; the origin's availability stayed at 100 percent throughout the attack, and its application performance was normal (response time around 2.5 seconds).

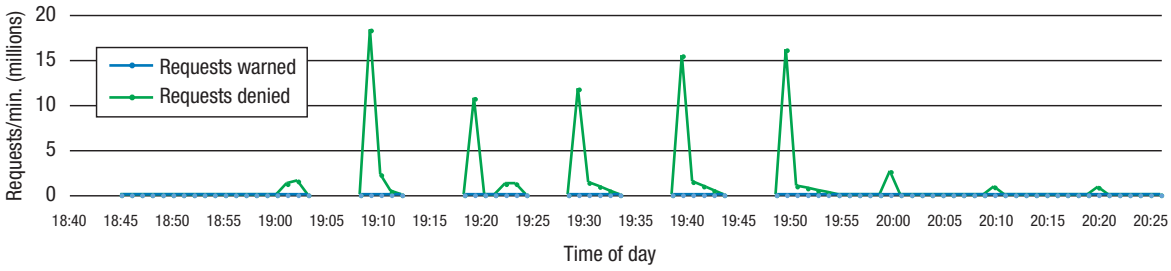

FIGURE 3. Observed high-speed probes on 31 July 2013 during Phase 4 attack of Operation Ababil. As the graph shows, almost all of the requests in these probes were denied.

On 31 July 2013, the network interdicted another DDoS attempt during Phase 4 of Operation Ababil. This attack was slightly more intensive than that of 5 March, with traffic volume exceeding 4.4 million requests per minute and involving more than 3,000 bots. Targeting many marketing webpages such as "details.do" as well as the DNS infrastructure, it immediately triggered an alert resulting in the harvesting and blocking of several new bot IP addresses. The early defensive action averted any perceivable impact on the availability and performance of the website and origin servers: average response time remained at 2 seconds before, during, and after the attack.

In this attack, QCF tested websites' vulnerability using short bursts of high-speed probes, as Figure 3 shows. If a website faltered, QCF returned later with a full-scale attack; if the site was resilient, they moved on to probe other sites. Akamai's secure delivery network successfully detected and blocked such probes.

Lessons learned

Operation Ababil provided several invaluable lessons for defending websites from attack.

First, the attack volumes were several orders of magnitude larger than the normal traffic for the websites targeted. This scale makes it uneconomical for individual content providers to build extra capacity as a buffer against anticipated attacks. A secure delivery network that is shared across numerous websites is a more affordable approach.

Second, during bursts, the attack traffic ramped up too quickly for reactive mitigation, underscoring the need for a proactive "always-on" website defense.

Third, as Operation Ababil evolved to focus on application-layer attacks, defenses at the lower network layers were no longer sufficient. It is important to defend at all levels of the network stack. Additionally, blacklisting known attackers' IP addresses is not enough—automatic rate control is required to defend against application-layer flooding attacks.

Finally, many of the QCF attacks featured cache-busting techniques, demonstrating the need to configure servers to exclude query strings from caching keys.

This article surveyed some common attacks on websites and techniques to mitigate those attacks. It is important to remember, however, that Web security is a cat-and-mouse game in which attacker and defender coevolve to counter each other's capabilities. For example, Akamai recently exposed a sophisticated brute-force attack on WordPress applications that went "under the radar" without triggering WAF rate-control rules that might automatically blacklist the source IP addresses.[12] The attacker simultaneously attacked nearly 500 WordPress sites such that each site received no more than a few brute-force attempts per hour from any given address, well below any reasonable rate-control limit. Detecting such an attack requires more complex WAF rules to better analyze traffic patterns across multiple sites. In fact, attack detection and mitigation will increasingly involve rapidly processing large amounts of data across various websites and time intervals. Although the vision of providing website security on a shared secure delivery platform is particularly well suited for such a sophisticated defense, many technological challenges lie ahead.

ACKNOWLEDGMENTS

The authors thank David Fernandez and the InfoSec team at Akamai for providing statistics and feedback on an early draft of this article.

REFERENCES

1. D. Belson, ed., *State of the Internet: Q3 2014*, Akamai Technologies, 9 Jan. 2015; http://goo.gl/f0LGEc.
2. *2014 Data Breach Investigations Report*, Verizon Enterprise Solutions, 2014; www.verizonenterprise.com/DBIR/2014.
3. *2013 Cost of Data Breach: Global Analysis*, research report, Ponemon Institute, May 2013; www.ponemon.org/library/2013-cost-of-data-breach-global-analysis.
4. *An Analysis of DrDoS DNS Reflection Attacks: Part I of the DrDoS White Paper Series*, white paper, Prolexic, 2013; http://goo.gl/kOEV5B.
5. *An Analysis of DrDoS SNMP/NTP/CHARGEN Reflection Attacks: Part II of the DrDoS White Paper Series*, white paper, Prolexic, 2013; http://goo.gl/7r9Zq7.

ABOUT THE AUTHORS

DAVID GILLMAN is an assistant professor of computational science at New College of Florida and a member of the faculty research group at Akamai Technologies, where he formerly served as a senior architect. His research interests include networks, image processing, and clinical and health informatics. Gillman received a PhD in mathematics from MIT. He is a member of ACM and the IEEE Computer Society. Contact him at dgillman@ncf.edu.

YIN LIN is a researcher and software developer at VMware. His research interests include computer networks, distributed systems, and software-defined networking. Lin received a PhD in computer science from Duke University. Contact him at yinlin09@gmail.com.

BRUCE MAGGS is a professor of computer science at Duke University and vice president for research at Akamai Technologies. His research interests include computer networks, distributed systems, and computer security. Maggs received a PhD in computer science from MIT. He is a member of the ACM SIGCOMM Executive Committee and has served on the ACM Council. Contact him at bmm@cs.duke.edu.

RAMESH K. SITARAMAN is a professor of computer science at University of Massachusetts Amherst and chief consulting scientist at Akamai Technologies. His research spans all aspects of Internet-scale distributed systems, including algorithms, architectures, security, performance, energy efficiency, user behavior, and economics. As a principal architect at Akamai, he helped pioneer the first major content delivery networks that currently serve much of the world's Web content and online applications. Sitaraman received a PhD in computer science from Princeton University. He is a member of ACM and a senior member of IEEE. Contact him at ramesh@cs.umass.edu.

6. H. Noman, "The Emergence of Open and Organized Pro-Government Cyber Attacks in the Middle East: The Case of the Syrian Electronic Army," *Information Warfare Monitor*, 30 May 2011; www.infowar-monitor.net/2011/05/7349.
7. P. Steiner, "On the Internet, Nobody Knows You're a Dog," cartoon, *The New Yorker*, 5 Jul. 1993.
8. J. Dilley et al., "Globally Distributed Content Delivery," *IEEE Internet Computing*, vol. 6, no. 5, 2002, pp. 50–58.
9. E. Nygren, R.K. Sitaraman, and J. Sun, "The Akamai Network: A Platform for High-Performance Internet Applications," *ACM SIGOPS Operating System Rev.*, vol. 44, 2010, pp. 2–19.
10. J. Abley and K. Lindqvist, *Operation of Anycast Services*, IETF RFC 4786, Dec. 2006; www.ietf.org/rfc/rfc4786.txt.
11. S. Triukose, Z. Al-Qudah, and M. Rabinovich, "Content Delivery Networks: Protection or Threat?," *Proc. 14th European Symp. Research in Computer Security* (ESORICS 09), 2009, pp. 371–389.
12. O. Katz, "Cat and Mouse: Web Attacks Increasingly Sidestep WAF Protections," *The Security Ledger*, 31 Dec. 2014; https://securityledger.com/2014/12/cat-and-mouse-web-attacks-increasingly-sidestep-waf-protections.

See **www.computer.org/computer-multimedia** for multimedia content related to this article.