Anatomy of a DDOS Attack



This is a true story of a client who recently came under DDoS attack, and successfully defeated it. It's a good example of a modern DDoS, and it illustrates many helpful points about how to properly react when (not if) your organization becomes the target of a DDoS.

Increasing Frequency and Scale

DDoS attacks are becoming very common. According to a recent report from Arbor Networks, 21 percent of data center operators now experience more than 50 attacks per month. This is up from only eight percent last year.

There are two reasons for this:

- Attack resources have become abundant and cheap. For example, the devastating assault on Dyn in October 2016 reportedly cost the attacker a mere \$7,500.
- Little technical knowledge is required. Malware-as-a-service is now thriving in multiple illicit marketplaces. Malware authors compete to offer tools that are easier to use, with more features and better automation.

As a result, the web is experiencing larger and larger attacks. In 2016, massive DDoS assaults shattered previous records, with the Dyn attack reaching an extraordinary 1.2 Tbps.

Also, attackers have many possible ways to mount a DDoS assault. There are multiple ways to exhaust the victim's resources. The attack can be aimed at the network, as in a UDP flood, or at services, such as a SYN flood or HTTP flood. Nor does the attacker even need a lot of bandwidth to mount an attack; application-layer attacks can be effective with little actual traffic sent.

Of course, all of this is opaque to the victim. Usually, DDoS targets know only two things: they are being attacked, and they think there's nothing they can do about it.

But that second idea is incorrect, as we shall see.

A Desperate Victim

When the phone rang on the Reblaze hotline, the voice on the other end was desperate and confused. He didn't know what to say, except: "We are under attack—what can we do?"

We started to ask him some questions about his situation. As often occurs, he didn't know what type of attack he was facing. He didn't know where it was coming from, or what its volume was.

All he knew was that he was under attack, and all his security assets were failing to protect his website. His DDoS protection, IPS/WAF, and servers were not functioning. Worse, his security solution providers were offering him no assistance. (In some cases, he couldn't even reach the appropriate people.)

The only possible "solution" to this crisis was found in the ransom note he received:



(At the time, twenty Bitcoin represented about \$20,000.)

Of course, paying ransom demands is always a bad idea. First, it encourages (and provides funding for) the attackers to continue their criminal activities. Second, as many victims have found, often the attackers don't keep their promises, and they don't cease the attack even after they receive payment. (If they were honorable people, they wouldn't be cybercriminals.)

Third, it's addressing the symptom rather than the problem. Even if a particular attack gets halted, the vulnerability that it exploited remains. And the victim will be liable to being successfully assaulted again and again, by any of the countless attackers now active on the web.

So, the victim knew it was unacceptable to pay the demanded ransom. But that's about all he knew about his situation.

The Initial Assessment

As we prepared to go to battle against the attackers, this is the battlefield we saw.

- We did not know the attack type.
- We did not know the attack volume.
- We did not know the attack origin.
- We did not know the nature of the website/application.

All we knew was that there was a ransom note.

In the past, this situation would be close to hopeless. Today, with modern cloud web security, it wasn't a problem.

We immediately built a unique, dedicated cloud infrastructure for this client on Google Cloud Platform. This took only a few minutes. (Most of this time was spent talking with him. Thanks to our automated setup processes, creating the infrastructure itself takes roughly one minute.)

We were now ready for traffic. But suddenly, the client reported that the attack had finished, and so he no longer needed our help.

We knew from past experience that the attack had *not* finished. We explained that this was probably was just a tool test by the attacker. Sure enough, twenty minutes later, the client called back. The attack was on again.

Following our instructions, the client changed his DNS records to point toward our newly created HTTP/S load balancer. (To deploy Reblaze's web security, there is no hardware or software to install. A simple DNS change is all that's required.)

As the DNS change propagated, a high rate of traffic began to flood into our system. Since we had had no initial information about the scale of the attack, we had set the system to automatically adjust. The autoscaler kicked in, and we quickly jumped from a handful of servers to tens of servers.

Backend services						
1443 Endpoint protocol: HT1	TPS Named port	https Tim	eout: seconds Health c	eck: rbz-autoscale-44	3 Session affinity: None	Cloud CDN: enable
Advanced configura	ations					
Advanced configuration of the second seco	Zone	Healthy	Autoscaling	Baland	ing mode	Capacity

At this point, the HTTP/S load balancer was using 42 servers.

Analyzing the Attack

With the client's traffic flowing through our system, we could now diagnose and analyze the attack.

We noted that the nature of the attack was an HTTP flood. It was coming from hundreds of servers across the globe, with each server generating around 40 RPS (requests per second).

On a per-server basis, this rate is fairly normal. Therefore, all the traffic had gone 'under the radar' for the client's security solutions, as it would have for most solutions available today. Because his solutions were not properly filtering the traffic, his website had been overwhelmed.

Not so once the traffic started flowing through the cloud we had deployed for him. Among other pioneering technologies, the Reblaze platform includes advanced human/bot recognition algorithms. User context, behavior, and multiple other factors are continually tracked and analyzed to determine the user's origin and intent. This enables the platform to quickly and accurately identify and mitigate bot traffic.

In this case, it quickly became clear that almost all the client's traffic was coming from bots. This graph from the Reblaze dashboard shows the early stage of the client's traffic. Of all the incoming requests, only about one percent were from legitimate human visitors.

"Blocked" requests are direct attacks that were blocked. "Challenges" are seemingly benign requests that are analyzed to determine their origin. "Humans" are legitimate human visitors.

Passed	vs. Blocked													
500k 100k												• Humans (25,	13,745): 520,763 764): 5,460 2,347,341): 509 ,3	94
200k														
3:56	13:58	14:00	14:02	14:04	14:06	14:08	14:10	14:12	14:14	14:16	14:18	14:20	14:22	14:24

With traffic flowing through our system, we now could understand what we were facing. Due to initial uncertainty, we had set the scaling too low. Now we could fine-tune our configuration.

This image from the Google cloud console shows the auto-scaling in action:

Autoscaled														
	stances / CPU usage													
40														
30														
20														
Mar 15,	4:20 PM Ma	ar 15, 4:25 PM	Mar 1	5, 4:30 PM I	Var 15, 4:35 PM	Mar 15,	4:40 PM	Mar 15, 4:45 Ph	Mar 15, 4:50 P	Mar 15, 4:55 PM	Mar 15, 5:00 PM	Mar 15, 5:05 PM	Mar 15, 5:10 PM	Mar 15, 5:19
E Number of instances: 43 ECPU urage: 23% (equity to 9.9 instances)														
<< Previous	1 2 Ne	xt >>												
Name ^		Zone	Disk		External IP	Connect								
- 9	rbzr-eurw1-04fk	europe-west1-b		rbzr-eurw1-04fk	None	SSH 👻								
- 0	rbzr-eurw1-0xhc	europe-west1-b		rbzr-eurw1-0xhc	None	SSH 👻								
□ U	rbzr-eurw1-1bm0	europe-west1-c		rbzr-eurw1-1bm0	None	SSH 👻								
□ ∪	rbzr-eurw1-1hcn	europe-west1-d		rbzr-eurw1-1hcn	None	SSH -								
- 4	rbzr-eurw1-298j	europe-west1-b		rbzr-eurw1-298j	None	SSH 👻								
	rbzr-eurw1-2fjx	europe-west1-c		rbzr-eurw1-2fjx	None	SSH •								
	rbzr-eurw1-3cdd	europe-west1-d		rbzr-eurw1-3cdd	None	SSH -								
□ U	rbzr-eurw1-3sm3	europe-west1-b		rbzr-eurw1-3sm3	None	SSH •								
□ U	rbzr-eurw1-5hth	europe-west1-d		rbzr-eurw1-5hth	None	SSH 👻								
□ U	rbzr-eurw1-5w7c	europe-west1-d		rbzr-eurw1-5w7c	None	SSH -								
□ ∪	rbzr-eurw1-793r	europe-west1-b		rbzr-eurw1-793r	None	SSH -								
- U	rbzr-eurw1-7hqp	europe-west1-c		rbzr-eurw1-7hqp	None	SSH -								
- U	rbzr-eurw1-81m2	europe-west1-c		rbzr-eurw1-81m2	None	SSH -								
	rbzr-eurw1-84j7	europe-west1-b		rbzr-eurw1-84j7	None	SSH -								
0	rbzr-eurw1-96hs	europe-west1-d		rbzr-eurw1-96hs	None	SSH -								
	rbzr-eurw1-9929	europe-west1-b		rbzr-eurw1-9929	None	SSH -								
	rbzr-eurw1-b34l	europe-west1-d		rbzr-eurw1-b34l	None	SSH -								
- U	rbzr-eurw1-b73d	europe-west1-c		rbzr-eurw1-b73d	None	SSH 👻								

And this screenshot (from a short while later) shows how load balancing was handling the traffic:

Despite the amount of traffic being scrubbed by the Reblaze platform, the client's servers still were not operating correctly. Much of the attack traffic was aimed directly at his network, bypassing our cloud deployment.

Details Monitorin	outining							
Backend	443 -					1 hour 6 hours	12 hours 1 day 2 days 4 days 7 day	/s 14 days 30
RPS for -44	3 by Instance Group							
	3 by Instance Group	~						
14,000								
10,000								
8,000								
6,000								
4,000								
2,000								
	Mar 15, 4:25 PM M	Aar 15, 4:30 PM Mar 15, 4:35 PM	Mar 15, 4:40 PM Mar 15, 4:45 PM	Mar 15, 4:50 PM Mar 15, 4:55 PM	Mar 15, 5:00 PM	Mar 15, 5:05 PM	Mar 15, 5:10 PM	Mar 15, 5:20 P
_								
Total inbound traffic		eurw1 (europe-west1-b): 1,754.6 📕	-rbzr-eurw1 (europe-west1-c): 1,713 🔳	-rbzr-eurw1 (europe-west1-d): 2,101.7				
Served from car Frontend Location (Total inbound traffic Asia 1,105.67 RPS North America 843.53 RPS			-tzreuwi (auspewesti-c): 1,713 🔳	-three-work (europe-west1-d): 2,101.7				
Frontend Location (Total inbound traffic Asia 1,105.67 RPS			-threewel (europe-westi-c): 1,713 🔳	-tbzr-euror1 (europe-west1-d): 2,101.7 10 of 10 instances healthy Backend Soc errors: 0.03 RPS			CPU Utilization: Rate: 1,754.6 RPS	
Frontend Location (Total inbound traffic Asia 1,105.67 RPS		Backend	-tzz euwi (europe west - c): 1,713 🔳	10 of 10 instances healthy				

So we asked him to change his servers' external IP addresses. He did so, and dropped the old ones. Additionally, he closed access to all traffic except that being passed in from Reblaze IPs.

Immediately, his site was restored to normal, up and running as if nothing was going on. Legitimate human traffic passes immediately through the Reblaze platform with near-zero latency, so his normal visitors had (and continue to have) full access to the site. However, all hostile traffic is blocked.

An Attack Revealed

An hour after his traffic started to flow through Reblaze, this is what the client's dashboard revealed.

Reblaze was in report-only mode for the first few minutes. In this mode, the platform allows all traffic to pass through, while showing what it would have filtered had it been in blocking mode. This mode is commonly used by new clients to fine-tune their deployments before going live.

Reblaze provides full transparency into the traffic flowing through it. Analysis that used to be extremely difficult is now very simple. Just by looking at the statistics, we could understand everything about this attack.



We knew what domains were being attacked:

Applications	Countries	Sources) Targets	Signatures	→ Referers	Extensions	Brow		Companies
Top Applications									reload charts
	Hits	Blocked	Humans	Bots	Challenges	Cloud	Anon.	Blockage	D/L
.com	30,255,24	40 27,310,36	2 27,515	30,227,725	2,941,108	95	14,809	99.99%	65.0 G
en.	646,00	02 605,42	0 4,081	641,921	36,637	85	142	99.39%	907.0 N
ru.	1,69	97 2	6 554	1,143	1,138	2	0	68.59%	9.0 N
de.	65	59 1	2 291	368	357	16	11	55.99%	3.0 N
ar	16	56	0 136	30	19	3	123	11.45%	1.0 N
fr	10	34	3 106	28	20	13	27	17.16%	753.0
www.com	ł	50	6 17	33	23	9	2	58.00%	212.0

We knew the originating countries of the attacking bots:

We knew the top user-agents participating in the attack:

The top referrer gave it up that the attack was made to a direct link:

The target URLs came out very clearly, showing that the attack was aimed toward "/".

Applications	Countries	Sources) Targets	Signatures	Referers		Insions	Browsers	Companies	Applications
Top Countries										Top User Agents
	Country		Hits	Passed	Blocks	Humans	Bots	Challenges	D/L	
India 🗠			3,015,089	3	2,800,465	1	3,015,088	214,621	5GB	Mozilla/5.0 (Windows NT
💌 Brazil			2,666,462	90	2,415,837	181	2,666,281	250,535	5GB	Mozilla/5.0 (Windows NT
II Romania			2,612,495	42	2,326,052	15	2,612,480	286,401	6GB	Mozilla/5.0 (Windows NT
Hungary			2,457,444	2	2,252,728	2	2,457,442	204,714	4GB	Mozilla/5.0 (Windows NT
Germany			1,366,313	795	1,176,191	5,148	1,361,165	189,327	4GB	Mozilla/5.0 (Windows NT
11 Italy			1,272,197	84	1,144,230	251	1,271,946	127,883	2GB	Mozilla/5.0 (Windows NT
🚥 Serbia			1,124,946	5	1,016,461	4	1,124,942	108,480	2GB	Mozilla/5.0 (Windows NT
Bangladesh			971,261	11	896,142	3	971,258	75,108	1GB	Mozilla/5.0 (Windows NT
Hong Kong			939,520	1	830,374	1	939,519	109,145	2GB	Mozilla/5.0 (Windows NT
🚍 Indonesia			871,627	2	758,304	2	871,625	113,321	2GB	Opera/9.80 (X11; Linux i6
Bosnia and Herzegovina			721,322	0	654,961	0	721,322	66,361	1GB	Mozilla/5.0 (Windows NT
Ukraine			685,286	212	609,776	193	685,093	75,298	1GB	Mozilla/5.0 (Windows NT
			600.054	7	655,600		600.050	00.000	200110	Mozilla/5.0 (Windows NT
										Mozilla/5.0 (Mindows NT

Mozilla/5.0 (Windows NT 6.2) AppleWeb
Mozilla/5.0 (Windows NT 6.1; WOW64)
Opera/9.80 (X11; Linux i686; Ubuntu/14
Mozilla/5.0 (Windows NT 6.3; WOW64; r
Mozilla/5.0 (Windows NT 6.1) AppleWeb
Mozilla/5.0 (Windows NT 6.1; WOW64; 1
Mozilla/5.0 (Windows NT 6.3; WOW64)
Mozilla/4.0 (compatible; MSIE 7.0; Wind
Mozilla/5.0 (Windows NT 6.3; rv:36.0) G
Mozilla/5.0 (Windows NT 6.1; WOW64)

Mozilla/5.0 (Windows NT 10.0; WOW6 Mozilla/5.0 (Windows NT 10.0; Win64 Mozilla/5.0 (Windows NT 10.0; WOW6 Mozilla/5.0 (Windows NT 10.0; WOW6

Applications	Countries	Sources) Targets	Signatures		Referers	Extensions		Browsers	Companie	83
Top URLs											
		URL			Hits	Passed	Blocks	Challenges	Humans	Bots	D/L
/403.html					26,019,306	0	26,019,306	0	0	26,019,306	9GB
1					70,621	832	1,815	67,974	778	69,843	1GB
/api/auth/updateSession/					9,048	4,088	0	4,960	4,088	4,960	36MB
/api/auth/checkSession/					861	861	0	0	861	0	368KB
/api/traffic/track/					423	419	4	0	423	0	220KB
/ats/					388	204	0	184	195	193	2MB
/elb/check/health.html					343	343	0	0	0	343	144KB
/api/languages/get_language_file/					317	306	0	11	306	11	ЗМВ
/platform/					308	127	0	181	127	181	1MB
/api/languages/get_active_languages					270	269	0	1	269	1	297KB
/wp-content/themes/angular/public/we	bsite/fonts/fo				202	3	0	199	3	199	2MB
/api/regulation/get_regulation/					166	156	0	10	156	10	138KB
/api/option/get_closest_time					146	146	0	0	146	0	61 KB
/wp-content/themes/angular/public/pla	atform/fonts/f				143	2	0	141	2	141	1MB

The traffic logs showed that the primary target was "/". As shown here by the Dashboard, Reblaze returned a 403 Forbidden error to these requests.

Statistics such as these are useful in many ways beyond their immediate context. They allow us to fine-tune our traffic scrubbing to ensure maximum accuracy and performance.

Also, Reblaze uses big data to continuously save all (anonymized) traffic data and statistics, for every Reblaze deployment worldwide—currently, there are petabytes of

data, representing many years' worth of traffic. Whenever an attack occurs, it's almost guaranteed that we've seen that attack pattern somewhere before. So our platform can know how the attack will unfold, and how to defeat it.

Moreover, Reblaze uses machine learning to continually improve its accuracy and performance. Even as new attack vectors arise on the web, the platform adapts and learns how to defeat them.

DDoS Attack Backfires: Targeted Site's Performance Improves

A few hours into the attack, the attacker's frustration was becoming visible. The DDoS traffic was now coming and going in waves, but was still being completely blocked by Reblaze.

Because none of the hostile traffic was reaching the targeted network, the client's site remained up and fully responsive to legitimate visitors.

In fact, the client found that since rerouting his traffic through Reblaze, his site had experienced an increase in performance. This was because our load balancer works with HTTP/2, and also the fact that Reblaze integrates fully with major CDN platforms. (This decreases server load, and allows our customers to serve more users with better performance.)

Therefore, by (unwittingly) causing his victim to become a Reblaze client, the attacker had caused our client to *increase* his site's performance, and serve his users better.

From this point, we continued to monitor the situation (which is very easy, made so by the Reblaze interface). However, once the platform is properly set up, it runs automatically and hands-free. So no further intervention was necessary.

We observed that the attacker continued his efforts for three more days. He even tried to amplify his efforts, with the attacks growing as large as 18,000 RPS. But it was all in vain.

Eventually, he gave up.

Lessons Learned

This example illustrates multiple points.

First, today's Internet makes it extremely easy for even an individual hacker to go after you. Attack resources are abundant, cheap, and easy to use. Anonymous payment methods like Bitcoin provide an attractive incentive for extortion attempts.

Second, many web security solutions are inadequate in this new threat environment. But you won't discover their flaws until it's too late.

Third, to be effective, a web security solution must provide multiple benefits. Among them are:

- Full transparency into your traffic. If you don't know what's going on within your site, you can't effectively control it.
- The ability to quickly deploy countermeasures and re-route your traffic through them. Only cloud solutions offer this ability.
- The ability to scale resources automatically, with as much bandwidth as needed to absorb even the largest attacks. Again, only cloud solutions offer this today.
- Sophisticated threat identification, including accurate bot detection, behavioral analysis, holistic user profiling, and more. Today's best practices include using big data and machine learning to increase accuracy and effectiveness automatically, wielding the computing capacity of the entire global cloud against would-be attackers.

Reblaze provides all the above benefits, and more. And it does so more cost-effectively than traditional solutions such as appliances.

To get a no-obligation demo of Reblaze in the U.S. or Canada, call **(408) 907-7712**. In the United Kingdom, call (0161) 711 0103 or (0845) 154 1110. International inquirers should call +972 (73) 252-7007. Or, send an email to <u>info@reblaze.com</u> for a prompt response.

About Reblaze

Reblaze is the comprehensive, cloud-based, robust protective shield for your web assets. Core technologies include: WAF/IPS, Multilayer DoS/DDoS protection (network, transport, and application), Anti-Scraping, High-level ACL, Advanced Human Detection and Bot Mitigation, Advanced Management Console, and Real-time Traffic Analysis. Added value services include: Layer 7 Load Balancing, Global Secured CDN, and a complete DNS solution. More information: <u>www.Reblaze.com</u>. Reblaze's clouds are fully compliant with SOC 1/SSAE 16/ISAE 3402, FISMA Moderate, PCI DSS, ISO 27001, FIPS 140-2, HIPAA, CSA (Cloud Security Alliance), and other standards and certifications.



Reblaze is a PCI-DSS Certified Level 1 Service Provider.



