REBLAZE TECHNOLOGIES LTD.

Anatomy of a DDoS Attack

Reblaze

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.

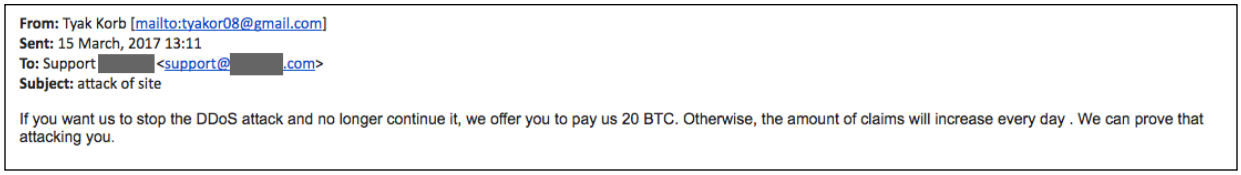
A Desperate Victim

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:


From: Tyak Korb [mailto:tyakor08@gmail.com]
Sent: 15 March, 2017 13:11
To: Support [redacted] <support@[redacted].com>
Subject: attack of site

If you want us to stop the DDoS attack and no longer continue it, we offer you to pay us 20 BTC. Otherwise, the amount of claims will increase every day . We can prove that attacking you.

(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.

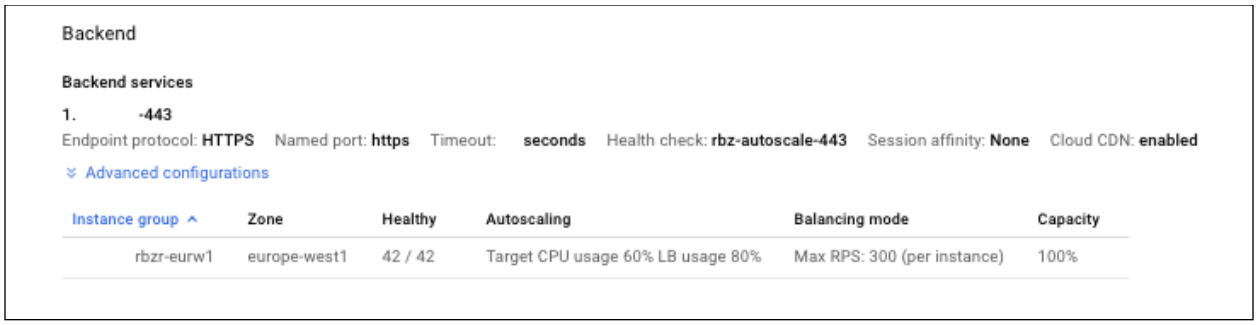
We immediately built a unique, dedicated cloud infrastructure for this client on Google Cloud Platform. This took only a few minutes. (Most of this time was spent talking with him. Thanks to our automated setup processes, creating the infrastructure itself takes roughly one minute.)

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.



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.

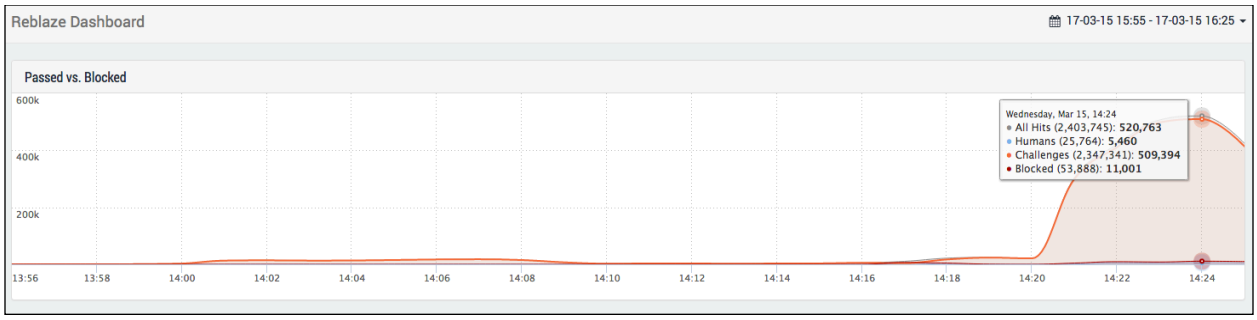
We noted that the nature of the attack was an HTTP flood. It was coming from hundreds of servers across the globe, with each server generating around 40 RPS (requests per second).

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.

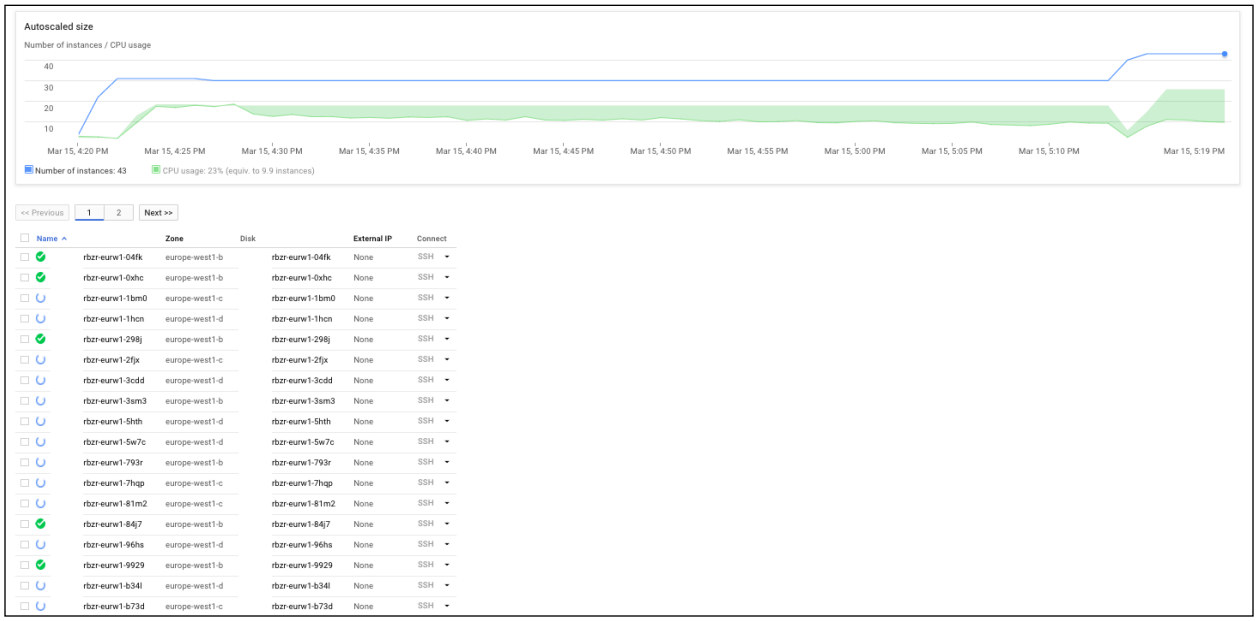
In this case, it quickly became clear that almost all the client's traffic was coming from bots. This graph from the Reblaze dashboard shows the early stage of the client's traffic. Of all the incoming requests, only about one percent were from legitimate human visitors.

"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.



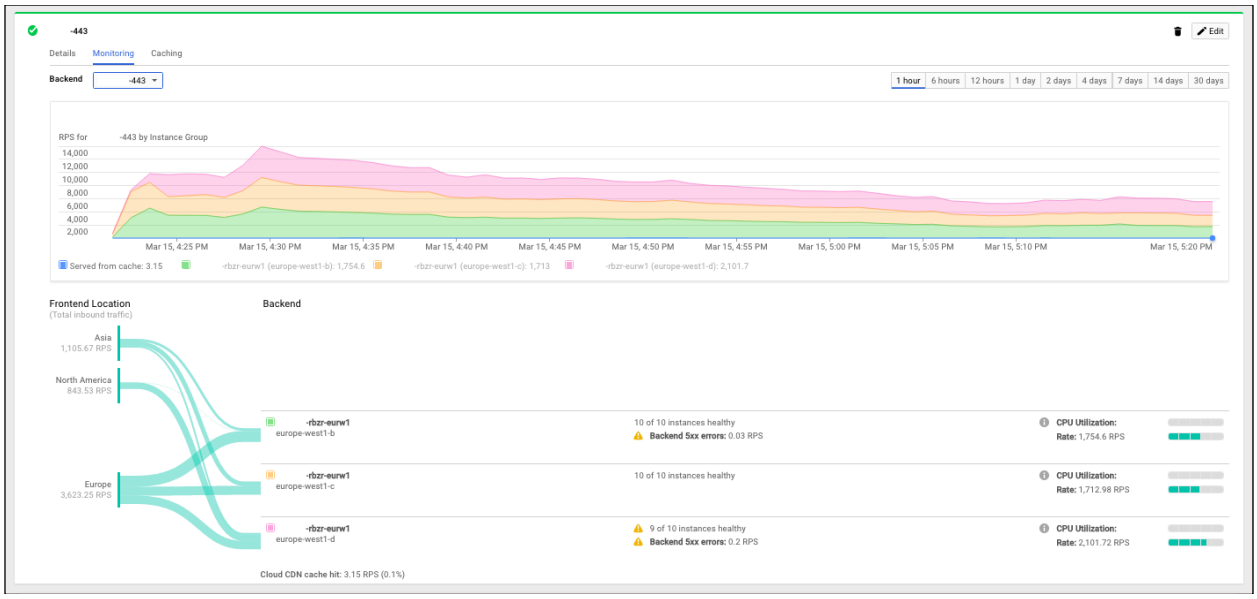
With traffic flowing through our system, we now could understand what we were facing. Due to initial uncertainty, we had set the scaling too low. Now we could fine-tune our configuration.

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



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.



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.

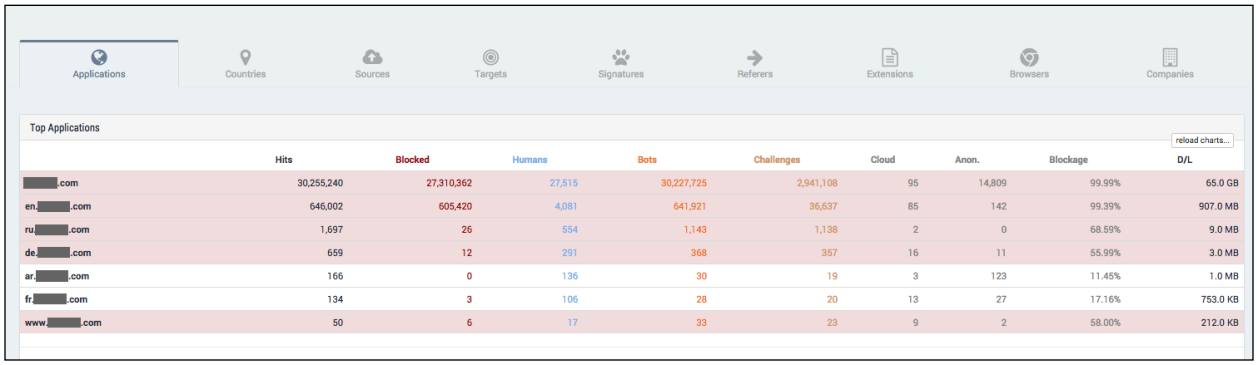
An Attack Revealed

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:

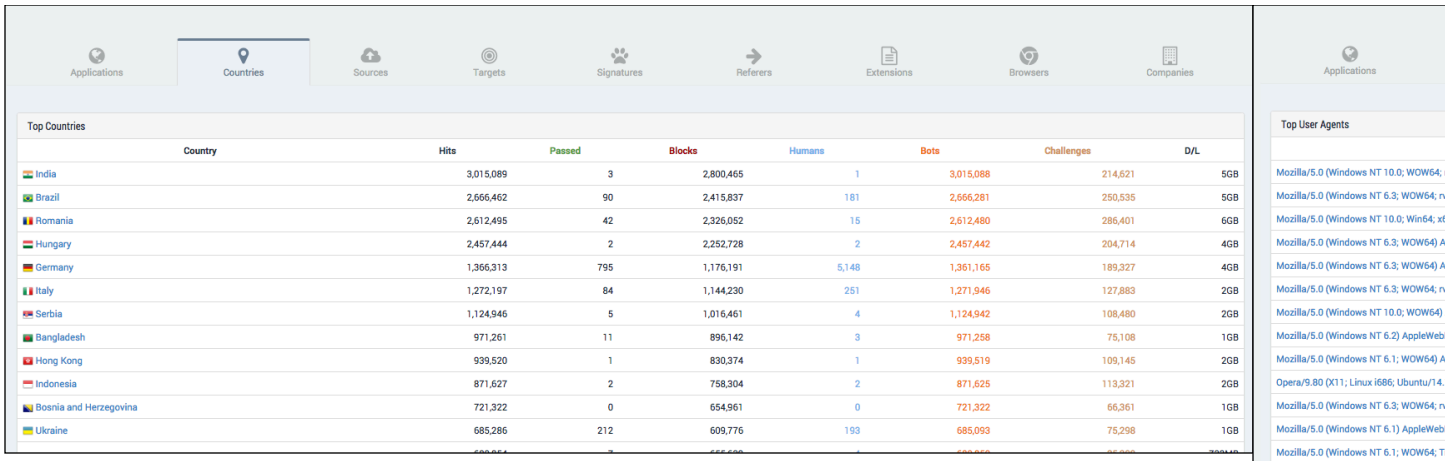


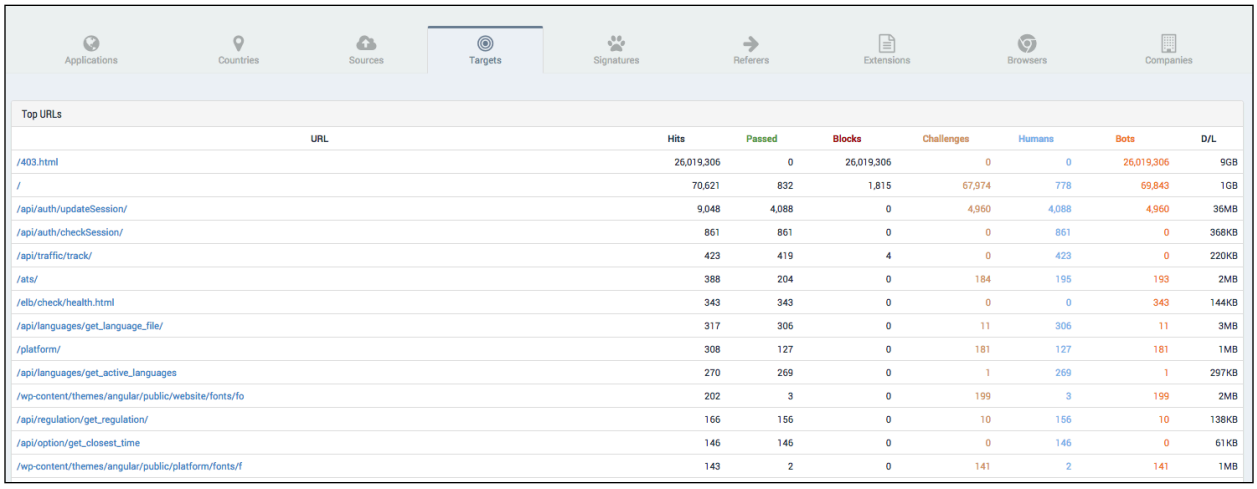
We knew the originating countries of the attacking bots:

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 “/”.





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 info@reblaze.com 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: www.Reblaze.com. 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.