Maneuvering Around Clouds: Bypassing Cloud-based Security Providers

Thomas Vissers[‡], Tom Van Goethem[‡], Wouter Joosen[‡], Nick Nikiforakis[†]
[‡]iMinds-Distrinet, KU Leuven, 3001 Leuven, Belgium
firstname.lastname@cs.kuleuven.be

[†]Department of Computer Science, Stony Brook University
nick@cs.stonybrook.edu

ABSTRACT

The increase of Distributed Denial-of-Service (DDoS) attacks in volume, frequency, and complexity, combined with the constant required alertness for mitigating web application threats, has caused many website owners to turn to Cloud-based Security Providers (CBSPs) to protect their infrastructure. These solutions typically involve the rerouting of traffic from the original website through the CBSP's network, where malicious traffic can be detected and absorbed before it ever reaches the servers of the protected website. The most popular Cloud-based Security Providers do not require the purchase of dedicated traffic-rerouting hardware, but rely solely on changing the DNS settings of a domain name to reroute a website's traffic through their security infrastructure. Consequently, this rerouting mechanism can be completely circumvented by directly attacking the website's hosting IP address. Therefore, it is crucial for the security and availability of these websites that their real IP address remains hidden from potential attackers.

In this paper, we discuss existing, as well as novel "origin-exposing" attack vectors which attackers can leverage to discover the IP address of the server where a website protected by a CBSP is hosted. To assess the impact of the discussed origin-exposing vectors on the security of CBSP-protected websites, we consolidate all vectors into CLOUDPIERCER, an automated origin-exposing tool, which we then use to conduct the first large-scale analysis of the effectiveness of the origin-exposing vectors. Our results show that the problem is severe: 71.5% of the 17,877 CBSP-protected websites that we tested, expose their real IP address through at least one of the evaluated vectors. The results of our study categorically demonstrate that a comprehensive adoption of CBSPs is harder than just changing DNS records. Our findings can steer CBSPs and site administrators towards effective countermeasures, such as proactively scanning for origin exposure and using appropriate network configurations that can greatly reduce the threat.

CCS'15, October 12–16, 2015, Denver, Colorado, USA.
© 2015 ACM. ISBN 978-1-4503-3832-5/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810103.2813633.

Categories and Subject Descriptors

C.2.0 [**Computer-communication Networks**]: [Security and protection]; K.6.5 [**Security and Protection**]: [Unauthorized access]

Keywords

Cloud-based security; DDoS attacks; Web attacks

1. INTRODUCTION

Although Distributed Denial-of-Service (DDoS) attacks have threatened the availability of online services for years, attacks are rapidly increasing in volume, complexity and frequency. Early 2014, the Network Time Protocol (NTP) was exploited in order to conduct amplification attacks [45] of previously unseen magnitudes, leading to multiple record-breaking volumetric attacks that reached up to 500 Gbps [35, 52]. Unfortunately, these powerful attacks are no longer exceptional cases. For instance, in 2014, there were four times as many attacks that crossed the 100 Gbps barrier as compared to 2013 [4]. Consequently, these massive attacks are now regarded as "the new normal" [23], an observation further confirmed by the frequent news reports of high-profile websites and web applications that become victims of such attacks [15].

Aside from advancing in strength and complexity, DDoS attacks are becoming increasingly accessible to the general public. The main cause is the rising popularity of websites offering DDoS attacks as a service, which enable non-technical users to launch DDoS attacks with the click of a button. These services, often called *booters* or *stressers*, allow their customers to orchestrate powerful DDoS attacks for just a few dollars through convenient, and user-friendly web interfaces [20].

To cope with the elevated risk and increased difficulty in fending off large DDoS attacks, several companies engineered highly capable, globally distributed networks that are able to deal with with DDoS traffic and malicious web requests. The resulting cloud-based defense infrastructure is then shared among the companies' customers. It is safe to assume that not all customers will be suffering from a large DDoS attack simultaneously, and thus companies can dedicate enough bandwidth and processing power to clients that are, at any given point, under attack.

Since several of these Cloud-based Security Providers (CBSPs) solely rely on changing the DNS settings of a domain name to reroute a website's traffic through their security infrastructure, the rerouting mechanism can be, in principle, completely circumvented by directly attacking the website's hosting IP address. Therefore, it is crucial for the security and availability of these websites that their real IP address, referred to as the *origin*, remains hidden from potential attackers. Past reports have claimed that the origin of CBSP customers can potentially be acquired through various methods, such as querying historical DNS data for a domain, and searching for subdomains that directly resolve to a server's real IP address [34]. Although these origin-exposing attack vectors have been known since 2013, the global extent of this issue has not yet been evaluated.

In this paper, we assess the magnitude of this problem on a large scale, i.e., we evaluate the number of protected domains whose CBSP-based protection can be bypassed. First, we discuss eight existing as well as novel vectors that have the potential to expose the underlying IP address of a CBSP-protected web server. Next, we consolidate these vectors into CloudPiercer, an automated origin-exposing tool. We deploy CloudPiercer and conduct the first large-scale experiment where we evaluate 17,877 domains that are protected by five different CBSPs. CloudPiercer uses a novel verification method to ensure that an IP address retrieved by the vectors is indeed the real origin of a website. After this verification step, we find that over 70% of protected domains expose their real IP address and, as a consequence, can be attacked directly, rendering the cloud-based protection service useless. Furthermore, we elaborate on the impact and prevalence of each exposing vector and discuss the feasibility of remediating the problem.

The main contributions of this paper are the following:

- We provide a comprehensive overview of novel and previously known origin-exposing vectors that allow attackers to bypass CBSPs.
- We report on the first large-scale measurement of this crucial security issue and conclude that the majority of CBSP clients are at risk, while providing insights into which vectors are most widespread.
- We discuss the difficulties of mitigating origin exposure, while suggesting several effective countermeasures that can vastly remediate the problem.

2. BACKGROUND

As Distributed Denial-of-Service (DDoS) attacks are becoming increasingly powerful, it becomes infeasible for websites to protect their own infrastructure. Even advanced, on-site, defense systems are rendered useless when the amount of traffic exceeds the processing capabilities of upstream devices or simply saturates the entire network connection. Furthermore, with the constant evolution of web application threats, there is also a need for increasing resources to fend off breaches. As a result, website owners turn to Cloud-based Security Providers (CBSPs) to protect their infrastructure. These companies reroute traffic from the original website through their network where malicious traffic is filtered before it ever reaches the network of their customer.

2.1 Modus Operandi of CBSPs

CBSPs act as reverse proxies for the web servers they are protecting. They inspect incoming traffic for various clients simultaneously, by routing it through their own distributed infrastructure. These cloud-based infrastructures, often referred to as *scrubbing centers*, act as highly-available traffic filters that are capable of absorbing extremely large volumetric DDoS attacks. Furthermore, they often integrate Web Application Firewalls (WAFs) to filter out malicious web application traffic, such as application-layer DDoS attempts, SQL injections and XSS attacks.

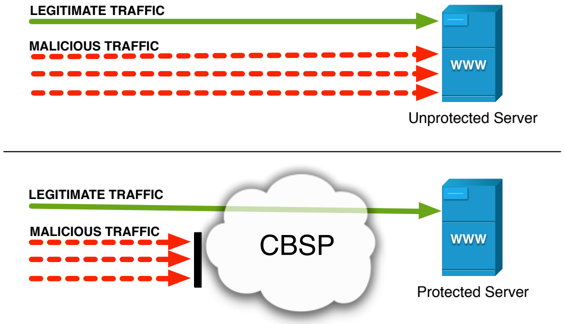

Figure 1: An unprotected server receives malicious traffic, potentially breaching the web server or denying service to the legitimate traffic (upper). Malicious traffic heading towards the protected server is absorbed by the CBSP, only allowing legitimate traffic to pass through (lower).

As depicted in Figure 1, all traffic towards a CBSP-protected web server, often referred to as the *origin*, is redirected through cloud-based scrubbing centers. After inspection of the incoming requests, only "clean traffic" is forwarded to the web server, effectively stopping attacks before they even reach the customer's premises.

Rerouting mechanisms

Several different strategies exist to route a web server's traffic through the cloud-based infrastructure. For instance, a website administrator can either opt for an *always-on* or for an *on-demand* strategy. The former redirects all traffic through the scrubbing centers on a permanent basis. The latter only starts redirecting traffic when necessary. Usually, this requires customer-premises equipment (CPE), that locally monitors incoming traffic. In case an attack is detected, this device initiates the redirection mechanism.

When traffic-redirection is active, there are two mechanisms to reroute traffic through the scrubbing centers. The first option is *DNS rerouting*, where an administrator changes the DNS settings of his website's domain name so that it resolves to an IP address that belongs to the CBSP. Normally, when a visitor requests a webpage, e.g., from `example.com`, his computer will first make a request to a DNS server to discover the corresponding IP address. Next, the visitor's browser can request the page from `example.com`'s web server using the discovered IP address. In the case of CBSPs, the visitors of the protected domain will receive an IP address of the CBSP's scrubbing center from the DNS server. Hence, the visitor will direct his requests to the scrubbing center, which in turn will transparently forward the legitimate requests to the origin, i.e., the actual web server of `example.com`.

Alternatively, a technique called *BGP rerouting* can be adopted. When the entity managing the website controls an entire /24 IP block, it can withdraw the BGP announcements for that block from its own routers. At this point, the CBSP can initiate BGP announcements for that IP range from their own network. Consequently, all traffic destined for the web server's IP address will start flowing through the CBSP's scrubbing centers. Since BGP rerouting is only available to entities that manage entire IP blocks and are able to install dedicated hardware, DNS rerouting has become the cloud-based security alternative for the masses [7].

2.2 CDNs as CBSPs

At their core, Content Distribution Networks (CDNs) are globally deployed services that increase the performance of websites by bringing static web content closer to users. The network usually consists out of a large set of geographically-distributed cache servers. This allows a CDN to quickly serve cached content from a server that is near a particular user. This setup reduces response times, load, and bandwidth of a website's main web server.

Similar to CBSPs, a CDN intercepts requests to a web server, which enables it to inspect incoming requests and selectively decide whether to serve cached content or forward the request to the web server for a dynamically generated response. Therefore, traffic towards the web server has to be constantly redirected through the CDN. To achieve this, CDNs either opt for URL rewriting or DNS rerouting [28]. Considering that a CDN's infrastructure is inherently capable of inspecting requests to leverage intelligent caching techniques, they are ideally placed to provide cloud-based security as well. Since traffic is already being redirected through their CDN, scrubbing centers and WAFs can be conveniently chained in the infrastructure. Moreover, in terms of volumetric DDoS attacks, a CDN is an ideal fit for mitigation strategies due to their geographically distributed and highly-available network. By using Anycast [1], servers spread across the globe can each process a small portion of the distributed attack, effectively making it feasible to absorb large amounts of malicious traffic.

As a result from this overlapping feature set, a significant share of CBSPs has emerged from CDN providers that started offering security services on top of their existing platform. Similarly, several security-focused companies that provided cloud-based services, have also started incorporating caching features to their infrastructure. Consequently, the line between CDNs and CBSPs is blurred. As such, the origin-exposing vectors that we discuss in Section 3 are applicable to CBSPs as well as to CDNs with security extensions.

3. POTENTIAL ORIGIN EXPOSURE

While CBSPs have become really popular because of their ability to stop real, large DDoS [38] and web application attacks, there are concerns about their DNS rerouting mechanisms. The concept of cloud-based security relies on keeping the underlying web server, the so-called origin, secret and inaccessible by direct traffic. However, in the case of DNS rerouting, this is achieved by hiding the origin's IP address and relying on redirection through the use of the website's domain name. Consequently, as illustrated in Figure 2, the website is *only* protected against traffic that uses the *domain name* to initiate the connection. So, in principle, if attackers are able to discover the real IP address of the origin, they can target traffic to the web server directly, thereby circumventing all security mechanisms present in the CBSP's network.

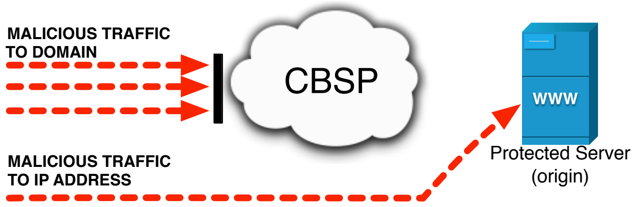

Figure 2: In the case of DNS rerouting, only traffic that uses the domain name is diverted through the CBSP's network. Traffic that uses the IP address of the protected server can reach the web server directly.

We refer to this security concern as the risk of *origin exposure*. This issue, which is specific to DNS rerouting, has been raised several times before [31, 34], and has, in the past, received some attention by the press, followed by several reactions from the security companies [27,48,53]. Many different potential vulnerabilities exist that might expose a CBSP-protected website's origin. We refer to these potential vulnerabilities as *origin-exposing vectors*. In the remainder of this section, we discuss eight origin-exposing vectors, of wich four have been reported previously, as well as four novel vectors, namely Temporary DNS exposure, SSL Certificates and specific instances of Sensitive Files and Outbound Connection Triggering. All vectors combined later form the basis of our automated scanning tool, CLOUDPIERCER.

3.1 IP History

When setting up cloud-based security, website administrators are required to change the DNS settings for their domain. From that point on, the origin's IP address is no longer listed in the DNS records of the domain name. As already mentioned in earlier sections, this secrecy is crucial for preventing origin exposure. However, if the origin is still assigned the same IP address as before the adoption of a CBSP, the server can be exposed through historical knowledge of the domain and its corresponding IP address.

Several companies specialize in harvesting data about domain names by continually tracking their DNS configuration. This allows them to build a vast database of historical DNS records, mainly used for domain marketing research, which can also be leveraged to track down an origin's IP address.

Accessing these databases is almost effortless and some of these services even offer a small number of free queries. However, these databases do not cover all existing domains as some TLDs do not share their zone files, making it harder to discover and monitor some domains. At the same time, domains that are not indexed in these databases are certainly not guaranteed to be safe from IP history vulnerabilities. For instance, if an attacker has been targeting a particular victim for a prolonged period, he could have manually gathered information about the domain and its origin before it was protected by the CBSP.

Because of the multitude of parties that could be collecting historical information about websites and their IP addresses, several CBSPs recommend administrators to assign a new IP address to their web server after migrating their DNS records to the CBSP [48].

3.2 Subdomains

Since the CBSP acts as a reverse proxy for multiple clients simultaneously, it relies on information available in HTTP requests to distinguish between requests intended for different clients. More specifically, by looking at the domain listed in the HTTP `Host` header, the CBSP can correctly forward incoming traffic to the intended origin. An unfortunate side-effects is that protocols that do not contain host information, such as FTP and SSH, cannot be properly handled by the CBSPs' proxies and are thus, by default, broken.

There are two ways around this problem: first, instead of using the domain name, an administrator can directly specify the origin's IP address when working with non-web protocols. This, however, lacks the flexibility of a domain-name-based solution since the IP address must be either hardcoded in scripts and program profiles, or remembered by a website's administrator.

Alternatively, administrators can create a specific subdomain, such as `origin.example.com`, that directly resolves to the origin's IP address. This provides a convenient tool for non-web protocols to bypass the CBSP and establish a direct connection with the origin. However, since this workaround effectively creates a direct path to a website's origin, it is a potential backdoor that, if discovered, can be abused by attackers. In the absence of misconfigured DNS servers allowing unauthenticated Zone Transfers, subdomains are not directly visible when querying the DNS records of the main domain name. An attacker can, however, perform a dictionary attack by trying to guess valid subdomains, using dictionaries of words popularly used in subdomains.

3.3 DNS records

Once a website is protected, the DNS `A` record of its domain name points to an IP address of the CBSP instead of directly to the origin. However, it is possible that traces of the origin are still present in other DNS records.

For instance, `MX` records reference the mail servers that are responsible for accepting email messages that are destined for mailboxes on a given domain. When only HTTP traffic is forwarded by the CBSP, SMTP needs to be able to establish a direct connection with the mail server. Therefore, the `MX` records should directly resolve to the mail server's IP address in order to keep email services operational. This can lead to origin exposure, especially when the mail server is listening on the same network interface as the origin's web server.

Another potentially problematic case are `TXT` records, often used for mechanisms such as the Sender Policy Framework (SPF) [21]. This framework aims to counter email address spoofing by validating the IP address of the sender against a list of approved IP addresses. The list of addresses from which emails may be sent, has to be placed in an `TXT` record of the domain [30]. Thus, if one wants the origin server to be able to send out emails using the SPF mechanism, they are forced to expose its IP address in the appropriate `TXT` record. Note that the solution to this problem is not obvious; an administrator has to choose to either abandon the Sender Policy Framework (thereby opening himself to email abuse), or accept that the protected web server cannot send verified emails.

The origin exposure, unfortunately, is not limited to `TXT` and `MX` records. Especially when a CBSP does not manage the DNS records of its customers' domains, exposure from other records may be overlooked by the customer. For instance, if the origin is accessible through IPv6, `AAAA` records are present. If the CBSP's setup instructions only cover the change of the `A` record of the domain, the `AAAA` record might be left unchanged, effectively keeping the origin exposed through its IPv6 address.

3.4 Temporary exposure

Administrators might temporarily pause the cloud-based security service, e.g., for maintenance or server migrations. During this time, the domain might temporarily resolve to its origin, effectively leading to a brief origin exposure. Temporary leaks can occur in many DNS record types, including `MX`, `CNAMEs`, and `TXT`. Attackers who are closely monitoring their victim might be able to witness a temporary exposure. Once the origin is known, the web server remains vulnerable even after the leak has disappeared, since the attacker can keep reusing the leaked IP address. The leak will only be remediated when the administrator decides to, yet again, change the IP address associated with the victim website.

3.5 SSL certificates

If administrators want to enable HTTPS for their website while under the protection of a CBSP, they can let the CBSP set up a certificate for their domain. This enables the CBSP to take care of securing the front-end connection between their own cloud infrastructure and a visitor. Alternatively, the administrator can hand over the private key of their origin's certificate to the CBSP. In this case, the CBSP can set up the front-end SSL connection with the website's own certificate. In order to secure the back-end connection between the CBSP and the origin, the origin must present a certificate. However, this certificate lists the domain name as the subject, and therefore identifies itself as the origin. In other words, if an attacker is able to scan all IP addresses and retrieve all SSL certificates, he can find the IP addresses of hosts with certificates that are associated with the domain he is trying to expose. Because of recent advancements in network scanners, performing such a massive scan has become quite feasible. For example, using ZMAP [14] and an appropriately fast network connection, allows an attacker to conduct a scan of the entire IPv4 address space on a single port in 45 minutes.

3.6 Sensitive files

Sensitive files located on the server form another vector through which a server's IP address can be exposed. For instance, files that were created during the development or configuration phase, in order to aid the administrator, can be used to expose a server's origin, especially when they show detailed information regarding the server. Furthermore, as already explained by Akamai [27], verbose error pages and log files can also disclose the origin that is meant to be kept secret. Usually, these types of files are meant to be removed or given proper access restrictions once a website goes into production, but presumably this is not always done correctly.

3.7 Origin in content

Instead of using a domain name to link to content, a webpage is free to use the IP address of the server directly. For example, a developer might use the IP address directly in the HTML of a page during an early development phase of the website. Although this is probably rather uncommon, it does form a potential origin-exposing vector. Furthermore, the IP address might be listed in the HTML as part of server-side software calculating web server statistics.

3.8 Outbound connections

Although a web server's incoming connections are rerouted through the CBSP's infrastructure, this is not the case for outbound connections. When a web server initiates an outgoing connection on its own accord, the CBSP is not used as a proxy. Consequently, the origin establishes a direct connection with an external host, effectively exposing its IP address to that particular host.

In order to exploit this phenomenon, an attacker can attempt to deliberately trigger the origin to initiate outgoing connections. Many possibilities exist in this regard and these are usually very specific to the applications running on the web server. Some examples include the possibility to upload a file via a URL, or link back mechanisms such as PingBack [25], which retrieve external webpages to verify whether a claimed link to their website is real or not.

4. LARGE-SCALE ANALYSIS

To assess the magnitude of the origin-exposure problem, we conduct a large-scale analysis in which we attempt to uncover the origin of CBSP-protected domains. First, we consolidate the eight origin-exposing vectors into one automated origin-exposing system called CLOUDPIERCER. Then, we assemble a list of clients from five CBSP companies by studying their DNS configurations and obtaining their adoption rate across the Alexa top 1 million websites. Starting from these client lists, we use CLOUDPIERCER to evaluate 17,877 long-term, CBSP-protected domains against origin exposure. In the final step of CLOUDPIERCER, all collected candidate IP addresses are validated with a novel verification method to assess whether each discovered IP address is indeed the one of a protected website. Using CLOUDPIERCER, we are not only able to measure the amount of bypassable domains but also to gauge which origin-exposing vectors are the most prevalent.

4.1 CBSP Providers

For our purposes, we are interested in analyzing various *always-on, DNS rerouting* CBSPs. As mentioned in Section 2.2, several CBSPs are CDNs that offer additional security services, and vice versa. Since it is not straightforward to externally distinguish between clients that only use the CDN capabilities from those who are specifically paying for a plan that includes security, we selected five well-known providers that have a *specific focus on security*, i.e., at least some form of cloud-based security is present by default in all of these provider's pricing plans. The selected providers are *CloudFlare, Incapsula, DOSarrest, Prolexic (PLXedge)* and *Sucuri (Cloud Proxy)*. We gather a list of clients from each provider, enabling us to study their necessary configurations and their adoption by popular websites.

Security Provider	DNS Configuration	Domains
CloudFlare	`NS`	35,552
Incapsula	`A, CNAME`	1,841
DOSarrest	`A`	1,295
Prolexic	`A`	829
Sucuri Cloud Proxy	`A`	281

Table 1: Cloud-based security providers, along with the DNS records that are adjusted by their clients, and the number of protected domains that were found in the Alexa top 1 million.

Clients

In order to identify protected clients, we need to be aware of the different DNS configurations that are required by each of the CBSPs. To retrieve this information, we first attempted to subscribe to each company, and took note of the set up process. If we were not able to register, e.g., due to the absence of trial or free service tiers, we searched for publicly available instructions or retrieved representative configuration settings by manually finding other existing clients.

Generally, we found two different types of DNS configurations that are used to reroute a website's traffic: changing the `NS` records and changing the `A` records. Incapsula, DOSarrest, Sucuri and Prolexic instruct their clients to change their domain's `A` record to a specific IP address, that is under the CBSP's control. In some cases, the `CNAME` or `A` record of the `www` subdomain is configured as well.

When the `NS` records of the domain have to be changed, as it is the case with CloudFlare, the entire DNS records of the domain name become actively managed by the CBSP's name servers. Consequently, all DNS records of the domain and its subdomains fall under their direct authority. This enables the CBSP to provide all the necessary DNS records in order for rerouting to take place. However, the configuration of additional custom records, such as the `MX` records to identify the domain's mail server, has to be managed through the CBSP's own custom interface where these additional records need to be added by the client.

Adoption

To assess the adoption of CBSPs, we analyze the top 1 million most popular websites, according to Alexa [3]. By retrieving each domain's DNS records and comparing them with the collected CBSP configurations, we can straightforwardly compile a list of the most popular CBSP-protected domains. Table 1 lists the number of clients found for each company, along with the type of DNS configuration that was used for identification.

When we evaluate the adoption of cloud-based security, we find that 4% of the web's most popular 1 million websites are protected by one of the five companies under analysis. Moreover, cloud-based security services appear to be a more prominent solution amongst the more popular websites, since, if we restrict our search to the top 10K websites, the CBSP adoption increases to 9%. To further quantify the relationship between CBSP adoption and website popularity, Figure 3 shows the distribution of each company's client list across rankings. Four out of five companies have a significantly higher portion of domains in the top 100K segment, further strengthening the correlation between CBSP adoption and website popularity. More specifically, Cloud-

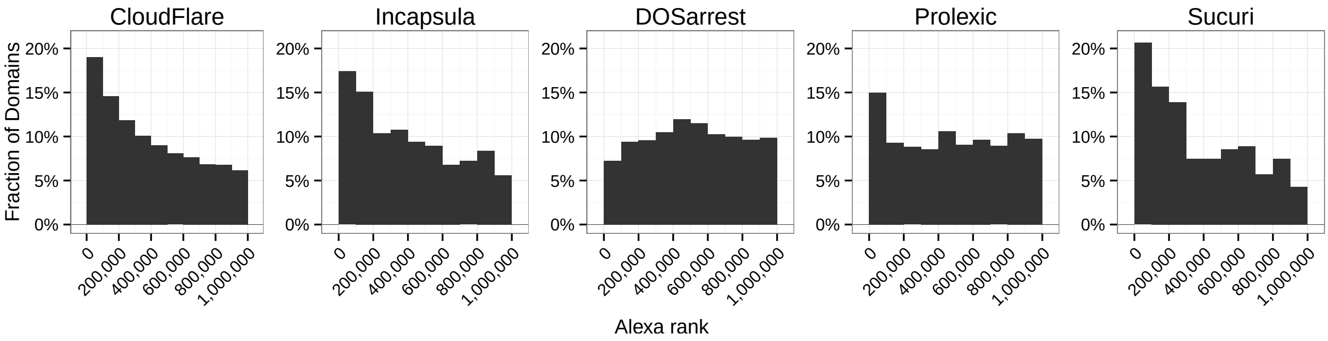

Figure 3: The portion of domains protected by each company, across segments of the Alexa top 1 million. For example, about 15% of the domains protected by Incapsula are situated between rank 100,000 and 200,000.

Flare, Incapsula and Sucuri have visibly less clients coming from the lower parts of the Alexa ranking. DOSarrest and Prolexic do not show this kind of correlation. However, we found that both companies have large domain parking services as one of their customers. These parking services are responsible for a large number of relatively unpopular undeveloped domains that are placed under protection by the CBSP thereby affecting the ranking distribution of the clients of these CBSPs.

Evaluated domains

For our large-scale analysis, we subjected the entire list of clients in the Alexa top 1 million of Incapsula, DOSarrest, Prolexic and Sucuri, as input to CLOUDPIERCER. Because of the disproportional popularity of CloudFlare, we decided to test a random sample of only half of their clients. This sample is small enough to allow us to conduct our experiments in a reasonable amount of time and large enough so that any conclusions can be safely generalized to the population of CloudFlare's clients. In addition, we limited the experiment to domains that remained customers of a CBSP during, at least, our 6-month monitoring period. Through this filtering process, we aim to remove negative bias, by excluding customers that were simply trying-out a CBSP and were, perhaps, not interested enough to take all necessary precautionary steps and eliminate origin-exposure vectors.

Overall, this process resulted in a final list of 17,877 domains, which we refer to as the *evaluation set.*

4.2 Origin Verification

To determine whether a discovered IP address is the actual origin of a CBSP-protected website, we evaluate whether we can retrieve the website's landing page using that IP address. First, we ensure that the IP address is a valid candidate by verifying that it does not belong to an address block owned by a CBSP. Then, our verification starts by visiting the website through its CBSP-protected domain name to retrieve the URL of the landing page. For example, when issuing a regular HTTP request to http://example.com, the browser might be redirected to a landing page with a different scheme, host and path, such as https://blog.example.com/about_me.html. Next, we use PhantomJS [18], an instrumented browser, to initiate an HTTP request to the candidate IP address, incorporating the previously extracted scheme, host and path of the landing page. If the candidate IP address is the actual origin of the website, this HTTP request should return the same webpage as the request using the domain name, as both requests are identical from the web server's perspective.

Determining, however, whether two webpages are identical is not as straightforward as executing a simple string comparison. For instance, when loaded twice, the same page can result in different HTML as dynamically generated content may be included in the website's response, such as, rotating articles and advertisements. In addition, timestamps and random values embedded in a webpage can also alter the resulting HTML. Moreover, several CBSPs inject content into the displayed page, such as, analytics scripts, which will not be present in a direct response from the origin.

To account for this natural variability, we designed a more intelligent and robust HTML comparison technique. Instead of comparing strings, we examine the structure of the DOM (Document Object Model). We parse both HTML strings with LXML and BeautifulSoup [43] into a tree representation of the nodes in the DOM. Next, we determine the difference between the two trees by calculating the Zhang-Shasha's tree edit distance [55], which counts the number of edit operations (insertions, deletions and substitutions of nodes) to get from one tree to the other. Furthermore, we extended the implementation [17] by incorporating normalization which is necessary to meaningfully compare the measured distances between tree-pairs of different sizes. Normalization is achieved by dividing the calculated edit distance by the sum of the tree sizes. We refer to this metric as the Normalized DOM-edit Distance (NDD).

Prior to our large-scale analysis, we measured the inter-page and intra-page NDD distributions from a random set of domains from the Alexa top 1 million, enabling us to calculate an optimal threshold. Additionally, we evaluated the effect of a more coarse-grained tree comparison by pruning the DOM trees to a certain maximum nesting depth. We measured the NDD between 3,500 pairs of different website's landing pages, as well as between 3,500 pairs of the same landing page loaded twice. Furthermore, we conducted this test for different tree pruning levels. Afterwards, we used this data to choose an optimal threshold that is used to decide whether two different HTML documents are, in fact, the same webpage. When evaluating thresholds, we focussed primarily on limiting false positives (two different webpages that are falsely marked as identical). At the end of this process, we found that a threshold of 0.18, at a maximum nesting depth of 5 levels, results in zero false positives and only 0.36% false negatives.

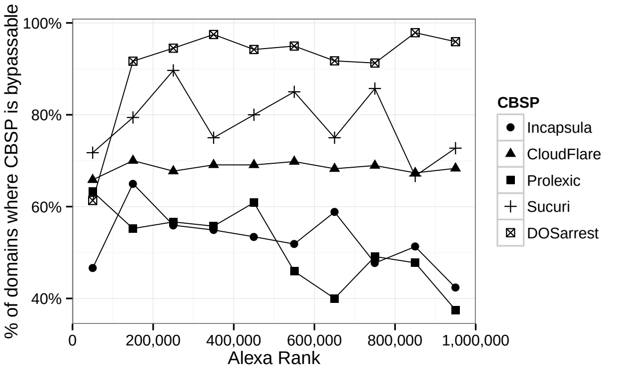

Figure 4: For each 100K-rank segment in the Alexa top 1 million, the percentage of domains where the CBSP is bypassable.

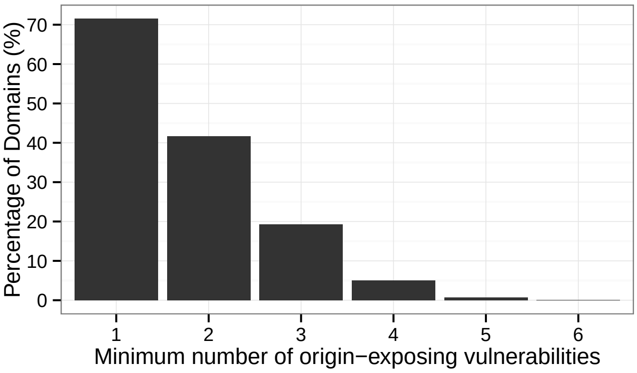

Figure 5: The percentage of domains that is susceptible to one or more origin-exposing vectors.

4.3 Ethical Considerations

To realistically assess the magnitude of the origin-exposing problem in the wild, one cannot avoid scanning real online websites and web applications. During our analysis and the development of Cloudpiercer, we took all appropriate steps to ensure that neither the origins, nor the CBSPs, were negatively impacted by our measurements. In addition, we only used publicly available webpages and data from publicly available sources.

Since the evaluated domain names are a subset of the most visited websites in the world, their infrastructure is capable of processing an abundant amount of requests on a daily basis. Nevertheless, we took several steps to minimize the impact of our analysis. For instance, the number of contacted PingBack endpoints was limited to three per domain, although, often, many more were present. Furthermore, web requests and DNS queries were adequately spaced in time in order to minimize impact on servers. Overall, we believe that the effects of our experiment on each individual site were minimal and we are confident that for the majority of websites, the extra traffic generated by our requests was just part of the expected traffic variability.

4.4 Results

All 17,877 domains in the evaluation set were subjected, by Cloudpiercer, to each of the eight origin-exposing vectors. Afterwards, Cloudpiercer used the origin verification algorithm to determine which IP addresses were the actual websites' hosting IP addresses. These results allow us to measure, both the origin-exposing power of each attack vector, as well as the overall risk of the origin being exposed. We manually inspected a sample of 250 exposed origins and saw that there were no false-positive verifications.

Overall, we found that 71.5% of protected domains is bypassable by combining the effect of all origin-exposing vectors. Table 2 lists the success-rate of each individual vector for the client domains of the different CBSPs. Subdomains and IP history are clearly the major vulnerabilities, both compromising the origin of more than 40% of domains. Figure 4 sheds light on the differences in the bypass-ratio between highly-ranked and less popular domains. We observe that for four out of five companies, domains in the top 100K are less susceptible to being exposed. A possible explanation is that higher ranked websites are more security conscious and more concerned with preventing origin exposure. A similar phenomenon was also observed in [51], where the top 100K-ranked websites were found to adopt significantly more web security mechanisms. Conversely, the risk of being exposed through SSL certificates is up to 3.6 times higher in that same top segment, presumably due to a higher SSL adoption-rate amongst these security conscious websites. Except for that first segment, there are no clear global trends across the remaining lower ranks.

As shown in Figure 5, 42% of domains are susceptible to exposure by more than one origin-exposing vector. More specifically, 19% of domains need to patch at least three origin-exposing vectors before they are safe. These numbers indicate that the problem is prevalent as well as multifaceted.

In the following paragraphs we discuss the results in more detail by examining the specifics of each origin-exposing vector.

Subdomain exposure

Overall, the most prevalent attack vector is the existence of origin-exposing subdomains. The feasibility of an attacker finding origin-exposing subdomains was tested by trying a list 5000 possible subdomains, provided by DNS Recon [36] on each domain in the evaluation set. If an entry existed for one or more of the tested subdomains, we verified whether the IP address to which it resolved was the origin. Our results indicate that CloudFlare's and Sucuri's customers are particularly vulnerable, with respectively 48.9% and 51.5% of domains disclosing their real IP address through subdomains.

When we take a closer look at which specific subdomains are responsible for the exposure, we find that the `ftp` subdomain is the most dominant problem, with 3,952 out of 17,877 domains having this "backdoor." This result implies a strong desire by website administrators to be able to use an FTP client that is able to connect to the server through a subdomain. Other subdomains that frequently reveal the origin are often related to email services, such as `mail` (3,203 domains), `webmail` (1,662) and `smtp` (258). Furthermore, a large number of exposing subdomains is related to cPanel, a hosting control panel that provides a web interface to help administrators configure their websites [10]. The discovered, origin-exposing subdomains are: `cpanel` (1,456 domains), `webdisk` (1,645) and `whm` (1,359). These subdomains are

Provider	CloudFlare	Incapsula	DOSarrest	Prolexic	Sucuri	All Providers
IP History	37.0%	36.4%	88.8%	40.4%	66.7%	40.5%
Subdomains	48.9%	31.7%	3.3%	7.3%	51.5%	43.4%
DNS records	32.6%	11.2%	0.9%	1.2%	29.0%	27.9%
Temporary DNS	4.1%	0.8%	0.6%	2.0%	0.9%	3.6%
SSL Certificates	9.4%	10.7%	2.5%	6.7%	17.3%	9.1%
Sensitive files	6.4%	1.5%	0.4%	0.2%	8.2%	5.4%
Origin in content	1.2%	0.4%	-	0.9%	2.2%	1.0%
PingBack (OC)	8.2%	2.2%	0.3%	-	3.9%	6.9%
RefBack (OC)	0.5%	0.1%	-	0.3%	-	0.5%
All Combined	72.5%	53.8%	92.0%	52.0%	77.9%	71.5%

Table 2: The percentage of domains that can be bypassed using each origin-exposing vector, for each cloud-based security provider's customers.

tied to particular services and interfaces incorporated into cPanel. Although these are HTTP services, they have to be accessed through non-standard ports which are often inaccessible when standard firewall policies are used. Therefore, cPanel creates these "proxy subdomains" that are directly linked to a specific port on the origin [9]. Despite the effort of some CBSPs to support some typical ports used by these control panels [37], these origin-exposing proxy subdomains are still frequently used.

The second-most dominating origin-exposing subdomain, namely `direct` (3,583 domains), is rather specific to CloudFlare customers. This subdomain was, in the past, given as an example when a user first configures his domain on CloudFlare's web interface [8]. Apparently, a large number of these clients used the company's instructions to the letter and thus kept this example subdomain bypass to link directly to their origin.

Interestingly, DOSarrest and Prolexic customers are less prone to subdomain exposure, with only 3.3% and 7.3% of exposed domains respectively. This is most likely due to the fact that each of their customers receives a dedicated IP address. This one-to-one mapping between an IP address of the CBSP and an IP address of the origin enables the CBSP to simply forward certain ports to the correct origin, without requiring any additional information to identify the intended host.

IP history exposure

To assess the number of domains that are still accessible through a previously documented IP address, we used two domain tracking services, DomainTools [12] and MyIP.ms [33]. We queried these databases for every domain in our evaluation set and all historical IP addresses were marked as candidates.

After the verification of the collected, "historical" IP addresses, it is evident that exposure though historical data is severe. Across all providers, we find that 40.5% of the domains are vulnerable to being exposed by consulting IP History databases. Furthermore, the issue is prevalent with all five provider's customers. This implies that, in general, CBSP's customers often fail to configure a new IP address after setting up their cloud-based security service. This, in turn, indicates that customers are either unaware of this attack vector, or are neglecting the CBSP's recommendation to change their IP address, possibly because of operational or infrastructural barriers. Regarding DOSarrest and Prolexic, it should be noted that the misconfiguration of a single client is greatly influencing the global number of IP history bypasses. Namely, 92% of DOSarrest's domains that were vulnerable to IP history exposure, were caused by domains that belonged to a single domain parking service. For Prolexic, a similar parking service is responsible for 86% of their historically exposed subdomains.

DNS record exposure

DNS records are arguably the most trivial and practical attack vector that we studied. To assess whether they reveal a domain's origin, we simply retrieved all records for each domain at a single point in time. From each record we extracted all IP addresses and marked them as candidates for the origin. Additionally, if a domain was present in the DNS record, we resolved the domain and marked the resulting IP address as a candidate.

Despite its simplicity, we find that a significant number of domains is exposed by this vector. More specifically, the origin of CloudFlare-protected domains is exposed by DNS records in 32.6% of the cases. For Sucuri and Incapsula, this is 29.0% and 11.2% respectively. Most of these domains are leaking their IP address through their `MX` record (4,390 domains), followed by `TXT` records (1,134), where SPF is often the reason, as described in Section 3.3. The frequent exposure through these two records suggests that web servers that send and receive email are responsible for a substantial fraction of the discovered origin exposure. Interestingly, we also found 16 domains that were exposed through their `A` records. In these cases, both the origin and the CBSP's IP address were present in the domain's `A` records. We speculate that in this situation, the client has created an additional `A` record for the CBSP's IP address, while forgetting to remove the existing record that pointed to the origin. For the domains under the protection of CloudFlare, the DNS records are managed by the CBSP. Therefore, we excluded CloudFlare customers that were exposed through their `A` record, as this indicates that the administrator has deliberately paused the CBSP rerouting through CloudFlare's web interface.

Temporary exposure

To determine whether origins were temporarily exposed due to an interruption of the cloud-based security service or due to a transient leak in another DNS record, we repeatedly retrieved, on a daily basis, all DNS records of protected domains for a period of 10 weeks (Sucuri and Prolexic) or 16 weeks (CloudFlare, Incapsula and DOSarrest). We excluded the domains that were already exposed by the one-time DNS records retrieval, described in the previous paragraph. This allows us to isolate the domains that only temporarily exposed their origin. The number of temporal exposures is considerable. We discover that more than 3% of domains transiently revealed their origin through their DNS records during a 10 or 16-week period. The vast majority of them were exposed through their `A` record, indicating a brief disabling of the protection system.

SSL certificate exposure

In order to find IP addresses hosting SSL certificates associated with the domains in the evaluation set, we made use of the publicly available data of Rapid7's Project Sonar [42]. This project uses ZMAP [14] to periodically conduct scans of the entire IPv4 address range in search for, among other things, SSL certificates. We used their certificate data [41] and extracted all IP addresses that presented certificates related to the domains in the evaluation set. According to Durumeric et al. [13], 129,695 of the domains in the Alexa top 1 million (13%) possess browser-trusted certificates. This appears to be in line with the 9% of origins that we discovered by looking for IP addresses that presented a certificate for those domains. If the origin desires to secure the back-end connection (the one between the CBSP and the origin) with HTTPS, a certificate for its domain has to be presented by the origin. Paradoxically, this, in turn, makes the entire set up less secure by introducing the risk of origin exposure.

Sensitive files

We limit our search of sensitive files to the so-called *phpinfo* files. These files execute the PHP function `phpinfo()` [49], which outputs a large amount of data regarding the server, the execution environment, PHP compilation options, etc. This function is particularly interesting because it dynamically retrieves all this data each time it is called. Furthermore, it usually displays the server's IP address in the `SERVER_ADDR` field.

We attempted to find files that execute this function by trying several obvious file names, namely `phpinfo.php`, `info.php`, `test.php` and `phpMyAdmin/phpinfo.php`. Overall, we found that 5.4% of domains had at least one of these files accessible and exposed their origin in this manner. Presumably, the files are a remainder of the development setup, which the developers forgot to remove.

Origin in content

For the vectors involving analyzing the HTML content of pages, we crawled each domain in the evaluation set. First, we queried Bing for each domain using `site:example.com` to retrieve an initial seed of 50 webpages. Starting from this seed, we crawled additional pages by visiting internal links, up to a maximum of 500 pages per domain. On average we retrieved 328 pages per domain in the evaluation set.

To detect whether the origin was present on the website's pages, we searched the HTML source code of every crawled page for the presence of IP addresses. We found only a small number of domains (1%) that included the real IP address of their web server in one of their pages, making it one of the least effective origin-exposing vectors.

Outbound connections

Since triggering outbound connections is closely tied to the applications that run on any given web server, it is near impossible to get a comprehensive measurement of the associated risk. In order to get an impression of what is possible, we chose to conduct two experiments on potentially widespread mechanisms. The first one revolved around the Pingback mechanism, which is mostly found on WordPress [54], the most wide-spread blogging software [40]. The second experiment focussed on the verification of the HTTP referrer header, which is being used, e.g., by RefBack [47], to discover incoming links.

Pingback exposure. Pingback is a protocol that allows website owners to get notified when one of their pages or articles is mentioned on another website. When a server receives a notification, Pingback should automatically visit the other website to verify whether it actually contains a valid hyperlink. This verification procedure can be leveraged to trigger an outbound connection from the origin. For the *Origin in content* vector, every domain in the evaluation set was crawled. During this process, we simultaneously searched for Pingback enabled webpages. Next, we made an XML-RPC request to the Pingback endpoints, in which we included a URL of a page on our server that contained a unique token for each domain. As a result, we could extract candidate origin IP addresses by monitoring the incoming requests on our server and recording which IP addresses accessed which domain-specific, tokenized URLs.

Essentially, Pingbacks allow a third party to force a web server to issue a request to an arbitrary host. In the past, this had lead to the creation of entire WordPress botnets, which were abused to conduct DDoS attacks on websites [5]. As a consequence, awareness about Pingback abuse was increased, encouraging many security companies and administrators to take steps towards preventing it from happening again [32, 46]. During our analysis, we often found that websites and CBSPs were actively blocking our Pingback requests, or refrained from initiating any outbound connection to our server. However, we were still able to confirm that 6.9% of protected domains expose their origin's IP address through the Pingback mechanism.

Referrer verification exposure. In order to test exposure through referrer verification, we set the HTTP Referrer header to a tokenized URL during the entire domain crawling process. Similar to our Pingback approach, we monitored whether there were any connections made to our unique URLs, potentially by a web application of the origin that wanted to inspect the referrer page that had lead a user to the origin's website. Our results indicate that this vector poses only a minor risk for origin exposure. Only 0.5% of domains were exposed by making an outbound connection from their origin to the referrer of a visitor on their website. Our server was, however, contacted by a plethora of other IP addresses which mostly belonged to web spiders, such as, Googlebot [16] and Proximic [39].

5. DISCUSSION & COUNTERMEASURES

Our findings categorically demonstrate that a comprehensive adoption of CBSPs is harder than just changing DNS records. Multiple origin-exposing vectors are highly prevalent and they generally involve different underlying causes, making the problem complex and multifaceted. Additionally, the results of our large-scale analysis are lower bounds. In the wild, an attacker can go to a greater extent to discover the origin of a particular targeted victim. For instance, if an attacker has found an IP address associated with a website through one of the origin-exposing vectors, he could scan the entire IP address block to which it belongs in further search for the origin. This can be effective when a victim has requested a new, "clean" IP address, but that address is possibly close to the previous one, since it is distributed by the same ISP. Similarly, when associated servers, such as mail servers, are discovered through subdomains or DNS records, it is a reasonable assumption that the origin is located at a nearby address. Furthermore, attackers can manually analyze the website to trigger outbound connections, search for specific configuration files, test for more subdomains, and perform much more intrusive tests than those included in Cloudpiercer.

Ultimately, unlike us, an attacker is not necessarily bound to origin verification. As noted in [34], an attacker can deduce the location of the origin by starting a DDoS attack on one or more plausible IP addresses and observing the effect it has on the CBSP-protected website. If the origin is taken down by this attack, the CBSP will display either a static cached copy of the offline website, or a 404-like error message.

Countermeasures

Complete mitigation of origin exposure is hard, as administrators are required to fully understand the potential risks and comprehensively address all vulnerabilities in order to fully prevent an attacker from circumventing the CBSP. However, a tool similar to Cloudpiercer could be deployed by CBSPs to proactively scan their client's domains for exposed origins, creating awareness and helping administrators fix specific vulnerabilities.

Apart from countering each origin-exposing vector, the logical first line of defense is a proper firewall configuration that blocks all connections except those originating from the CBSP. This will significantly complicate the life of an attacker who will not be able to tell whether an IP address is unreachable, or whether it, in fact, is the origin of a target website. Together with requesting a new IP address, this firewall configuration should be standard practice when cloud-based security is utilized. We can safely assume that the vast majority of customers are currently not adopting such a strategy, since, if they did, our origin verification method would have failed. It appears that administrators are either uninformed about the risks, or are deterred by the complications of properly whitelisting all IP addresses necessary to keep the website operational. We conducted a small-scale survey asking vulnerable websites about the missing firewall configurations and their CBSP-related security expectations but we, unfortunately, received no responses.

CBSPs could actively monitor whether their client's domain was assigned a fresh IP address, and whether the client's web server is blocking requests coming from outside of the CBSP's network. This information could then further be used to explicitly warn and motivate administrators to take the necessary measures to prevent exposure.

Another beneficial strategy for CBSPs is to assign a unique IP address to each customer, which is already the case with Prolexic and DOSarrest. As our results showed, this has a significant effect on the number of subdomains and DNS records exposures. If the necessary ports can be forwarded to the origin, there is no need to set up subdomains or `MX` records that connect directly to the origin's IP address. We expect that as the adoption of IPv6 expands, this defense mechanism will become increasingly more practical, even for very large CBSPs, such as CloudFlare.

Possibly, some larger websites that possess entire /24 IP blocks might be able to initiate BGP rerouting once the origin has been attacked directly. However, relying on this fallback scenario defeats the benefits of the always-on strategy and eliminates the protection against web application attacks.

6. RELATED WORK

To the best of our knowledge, our research is the first to review existing origin-exposing attack vectors for the bypassing of CBSPs, propose new ones, and systematically assess the magnitude of the exposure problem in the wild.

Over the years, a plethora of DDoS defense systems have been proposed. However, destination-based systems are usually rendered ineffective against large volumetric attacks that are able to saturate a site's uplink. Additionally, according to Huang et al. [19], systems that seek cooperation of many different parties usually face deployment issues. The authors argue that a lack of incentive prevents these cooperative systems from being widely deployed across the Internet's infrastructure. For instance, the profit of transit providers greatly depends on the amount of traffic they forward. Hence, these providers are cautious with implementing filtering systems that might negatively impact their business. In constrast, recent publications [11, 24] have documented the decline of the NTP DDoS attacks, impacted by a large-scale collaborative effort amongst ISPs, CERTs and academia.

A feasible non-collaborative solution for a victimized autonomous system (AS) was introduced in 2003 by Argawal et al. [2]. The concept is to reroute DDoS traffic through off-site cleaning centers that are dedicated to filtering and absorbing malicious attack traffic. The authors studied various network-layer techniques for diverting DDoS traffic to cleaning centers and, afterwards, redirecting the clean traffic to the protected web server. This work later became the inspiration for the patents of several popular DDoS mitigation services, such as Prolexic [29]. The use of rerouting techniques, such as BGP diversion and GRE tunnelling, resurfaced in Shield by Kline et al. [22]. In that paper, the authors focus on leveraging the off-site DDoS mitigation as an insurance model to solve the incentive problem. The authors also note that CDNs can be leveraged as DDoS defense systems in a similar fashion. In 2007, Lee et al. [26] already studied the inherent DDoS resilience of CDNs and proposed a novel scheme to further improve their robustness against attacks.

As CDNs further incorporated security features into their products, their business extended increasingly into cloud-based security providers. Thereupon, various studies evaluated these security components and several problems were uncovered. For instance, Liang et al. [28] analyzed how HTTPS was implemented within the context of CDNs. Inherently, a CDN is a man-in-the-middle (MITM) between the website and its visitors. This allows them to inspect incoming requests for the purpose of serving cached content and filtering out malicious requests. However, as HTTPS is intended for end-to-end encryption, this introduces various complications. In their study, the authors report on several implementation issues, including private key sharing, insecure back-end communication and numerous issues with invalid, stale and revoked certificates.

Another issue, discovered by Triukose et al. [50], allows CDNs to be abused to conduct a bandwidth amplification DDoS attack against their own customers. The vulnerability leveraged the fact that requests to CDN-enabled websites typically involve two decoupled TCP connections, with the CDN as a MITM. However, once the CDN forwards an attacker's request to the origin, the attacker can cleverly break off his own TCP connection with the CDN. Thereupon, the origin will waste bandwidth by sending a response to the CDN that will no longer be forwarded to the attacker.

Finally, in 2013, Nixon et al. [34] and McDonald [31] raised awareness of origin-exposing vectors that could enable attackers to bypass CBSPs and CDNs. We extend their work by proposing novel origin-exposing vectors and combining them into one automated origin-exposing tool with origin-verification capabilities, which we then deployed to conduct the first large-scale assessment of the issue. DOM-based similarities, which we leveraged for origin-verification, were previously used by [44] to detect phising attempts.

7. CONCLUSION

Cloud-based security is a popular solution to counter the increasing threat of DDoS and web application attacks. CBSPs that use proxying via DNS are adopted by at least 9% of the 10K most popular websites. Presumably, the trivial setup without infrastructural investments, combined with the benefit of an always-on protection service, attracts a large user base. The mechanism itself, however, suffers from a critical weakness. The entire mitigation service is completely dependent on the secrecy of the website's hosting IP address, the so-called origin. Moreover, several vulnerabilities are reported that have the potential to expose this origin.

In this paper, we discussed eight origin-exposing vectors, including various novel vulnerabilities. We consolidated all vectors into CLOUDPIERCER, an automated origin-exposing tool, which we then used to conduct the first large-scale analysis to measure the global risk of origin exposure. Our results demonstrate that the problem is severe: 71.5% of the 17,877 CBSP-protected websites that we tested, exposed their real IP address through at least one of the evaluated vectors.

Taking into account the severe consequences of an exposed origin and its prevalence amongst CBSP-protected websites, we opine that the problem is currently inadequately addressed. However, the findings of our research can be used both by CBSPs to encourage better practices regarding the adoption of their security infrastructure, as well as by administrators of CBSP-protected websites who can verify and remediate their own origin-exposing vulnerabilities. All five CBSPs have been notified of our findings prior to publication.

Finally, a silver lining of our findings is that a tool like CLOUDPIERCER can, in principle, be used by law enforcement. It is well known that miscreants use CBSPs to hide their real hosting location [6], making it harder to track and shut them down. Consequently, the discussed vectors and their reported effectiveness can be leveraged by the appropriate institutions to react quicker against malicious online activities.

8. AVAILABILITY

CLOUDPIERCER will be made available as a web service on https://distrinet.cs.kuleuven.be/software/cloudpiercer/, where users of CBSPs, after proving ownership of their websites, will be able to submit their URLs for scanning and get a detailed report on all the origin-exposing vectors that CLOUDPIERCER was able to find. We hope that the community will benefit from this service by allowing administrators to discover and eliminate vulnerabilities on their websites, before they are discovered and abused by attackers.

Acknowledgments

We thank the anonymous reviewers for their valuable comments, and Linode for providing us with virtual machines that made our large-scale experiments possible. For KU Leuven, this research is partially funded by the Research Fund KU Leuven, and by the EU FP7 project NESSoS. With the financial support from the Prevention of and Fight against Crime Programme of the European Union (B-CCENTRE). For Stony Brook University, this work was supported by the National Science Foundation (NSF) under grant CNS-1527086.

9. REFERENCES

[1] J. Abley and K. E. Lindqvist. Operation of anycast services. 2006.

[2] S. Agarwal, T. Dawson, and C. Tryfonas. Ddos mitigation via regional cleaning centers. Technical report.

[3] Alexa. Alexa - Actionable Analytics for the Web. http://www.alexa.com/, 2014.

[4] Arbor Networks. Worldwide Infrastructure Security Report. http://pages.arbornetworks.com/rs/arbor/images/WISR2014_EN2014.pdf, 2015.

[5] D. Cid. More Than 162,000 WordPress Sites Used for Distributed Denial of Service Attack. http://blog.sucuri.net/2014/03/more-than-162000-wordpress-sites-used-for-distributed-denial-of-service-attack.html, 2014.

[6] CloudFlare Watch. http://www.crimeflare.com/.

[7] CloudFlare. Cloudflare sees explosive growth in 2013. http://www.marketwired.com/press-release/cloudflare-sees-explosive-growth-2013-passes-15-million-customers-revenue-up-450-network-1862981.htm, 2013.

[8] CloudFlare, Inc. Sign up | CloudFlare | The web performance and security company. https://www.cloudflare.com/sign-up, 2015.

[9] cPanel. Tweak Settings - Domains. https://documentation.cpanel.net/display/ALD/Tweak+Settings+-+Domains#TweakSettings-Domains-Proxysubdomains, 2015.

[10] cPanel, Inc. cPanel and WHM. http://cpanel.net/, 2015.

[11] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir. Taming the 800 pound gorilla: The rise and decline of ntp ddos attacks. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 435–448. ACM, 2014.

[12] DomainTools, LLC. Domain Whois Lookup, Whois API and DNS Data Research - DomainTools. http://www.domaintools.com/, 2015.

[13] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman. Analysis of the https certificate ecosystem. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 291–304. ACM, 2013.

[14] Z. Durumeric, E. Wustrow, and J. A. Halderman. Zmap: Fast internet-wide scanning and its security applications. In *USENIX Security*, pages 605–620. Citeseer, 2013.

[15] K. Fiveash. PlayStation clambers back online days after DDoS attack PARALYSED network. http://www.theregister.co.uk/2014/12/27/playstation_clambers_back_online/, 2014.

[16] Google. Googlebot. https://support.google.com/webmasters/answer/182072?hl=en, 2015.

[17] T. Henderson and S. Johnson. Zhang-Shasha: Tree edit distance in Python. https://github.com/timtadh/zhang-shasha, 2014.

[18] A. Hidayat. PhantomJS - a headless WebKit scriptable with a JavaScript API. http://phantomjs.org, 2015.

[19] Y. Huang, X. Geng, and A. B. Whinston. Defeating ddos attacks by fixing the incentive chain. *ACM Transactions on Internet Technology (TOIT)*, 7(1):5, 2007.

[20] M. Karami and D. McCoy. Understanding the emerging threat of ddos-as-a-service. In *LEET*, 2013.

[21] Kitterman, Scott. Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. http://tools.ietf.org/html/rfc7208, 2014.

[22] E. Kline, A. Afanasyev, and P. Reiher. Shield: Dos filtering using traffic deflecting. In *Network Protocols (ICNP), 2011 19th IEEE International Conference on*, pages 37–42. IEEE, 2011.

[23] B. Krebs. The New Normal: 200-400 Gbps DDoS Attacks. http://krebsonsecurity.com/2014/02/the-new-normal-200-400-gbps-ddos-attacks/, 2014.

[24] M. Kührer, T. Hupperich, C. Rossow, and T. Holz. Exit from hell? reducing the impact of amplification ddos attacks. In *USENIX Security Symposium*, 2014.

[25] S. Langridge and I. Hickson. Pingback 1.0. http://www.hixie.ch/specs/pingback/pingback, 2002.

[26] K.-W. Lee, S. Chari, A. Shaikh, S. Sahu, and P.-C. Cheng. Improving the resilience of content distribution networks to large scale distributed denial of service attacks. *Computer Networks*, 51(10):2753–2770, 2007.

[27] D. Lewis. Bypassing Content Delivery Security. https://blogs.akamai.com/2013/08/bypassing-content-delivery-security.html, 2013.

[28] J. Liang, J. Jiang, H. Duan, K. Li, T. Wan, and J. Wu. When https meets cdn: A case of authentication in delegated service. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 67–82. IEEE, 2014.

[29] B. Lyon. Network overload detection and mitigation system and method, Jan. 13 2009. US Patent 7,478,429.

[30] K. Martens, J. Mehnle, and S. Kitterman. SPF Record Syntax. http://www.openspf.org/SPF_Record_Syntax, 2008.

[31] D. McDonald. The Pentesters Guide to Akamai. https://www.nccgroup.com/media/230388/the_pentesters_guide_to_akamai.pdf, 2013.

[32] Moore, Simon. WordPress Pingback Attacks and our WAF. https://blog.cloudflare.com/wordpress-pingback-attacks-and-our-waf/, 2014.

[33] MYIP.MS. My IP Address - Shows IPv4 and IPv6 | Blacklist IP Check - Hosting Info. http://myip.ms/, 2015.

[34] A. Nixon and C. Camejo. Ddos protection bypass techniques. *Black Hat USA*, 2013.

[35] P. Olson. The Largest Cyber Attack In History Has Been Hitting Hong Kong Sites. http://www.forbes.com/sites/parmyolson/2014/11/20/the-largest-cyber-attack-in-history-has-been-hitting-hong-kong-sites/, 2014.

[36] C. Perez. DNS Recon. https://github.com/darkoperator/dnsrecon, 2015.

[37] M. Prince. CloudFlare Now Supporting More Ports. https://blog.cloudflare.com/cloudflare-now-supporting-more-ports/, 2012.

[38] M. Prince. The DDoS That Almost Broke the Internet. https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet/, 2013.

[39] Proximic. Proximic Spider. http://www.proximic.com/spider.html, 2015.

[40] Q-Success. Market share trends for content management systems for websites. http://w3techs.com/technologies/history_overview/content_management, 2015.

[41] Rapid7 Labs. Internet-Wide Scan Data Repository. Project Sonar: IPv4 SSL Certificates. https://sonar.labs.rapid7.com/, 2015.

[42] Rapid7 Labs. Project Sonar. https://sonar.labs.rapid7.com/, 2015.

[43] L. Richardson. Beautiful Soup. http://www.crummy.com/software/BeautifulSoup/, 2013.

[44] A. P. Rosiello, E. Kirda, C. Kruegel, and F. Ferrandi. A layout-similarity-based approach for detecting phishing pages. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 454–463. IEEE, 2007.

[45] C. Rossow. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium*, February 2014.

[46] Stephenson, MaAnna. Disable XML-RPC in WordPress to Prevent DDoS Attack. http://www.blogaid.net/disable-xml-rpc-in-wordpress-to-prevent-ddos-attack, 2014.

[47] H. Story and A. Sambra. Friending on the Social Web. http://bblfish.net/tmp/2011/05/09/, 2011.

[48] N. Sullivan. DDoS Prevention: Protecting The Origin. https://blog.cloudflare.com/ddos-prevention-protecting-the-origin/, 2013.

[49] The PHP Group. PHP: phpinfo - Manual. http://php.net/manual/en/function.phpinfo.php, 2015.

[50] S. Triukose, Z. Al-Qudah, and M. Rabinovich. Content delivery networks: protection or threat? In *Computer Security–ESORICS 2009*, pages 371–389. Springer, 2009.

[51] T. Van Goethem, P. Chen, N. Nikiforakis, L. Desmet, and W. Joosen. Large-scale security analysis of the web: Challenges and findings. In *Trust and Trustworthy Computing*, volume 7, pages 110–125. Springer, 2014.

[52] S. J. Vaughan-Nichols. Worst DDoS attack of all time hits French site. http://www.zdnet.com/article/worst-ddos-attack-of-all-time-hits-french-site/, 2014.

[53] R. Westervelt. Cloud-Based DDoS Protection Is Easily Bypassed, Says Researcher. http://www.crn.com/news/security/240159295/cloud-based-ddos-protection-is-easily-bypassed-says-researcher.htm, 2013.

[54] WordPress.org. WordPress: Blog Tool, Publishing Platform, and CMS. http://wordpress.org, 2015.

[55] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262, 1989.