



ISP Blog

03
Apr
2018

CloudFront Hijacking

By: Matt Westfall

I recently spent some time exploring the issue of CloudFront domain hijacking. This is not a new issue but I think it has gone mostly unnoticed for a few reasons:

- CloudFront's default behavior is not intuitive. Some standard DNS configurations can mislead users into thinking that their vulnerable domains are configured correctly.
- In the past year, [misconfigured S3 buckets](#) have been everyone's priority. Other AWS security issues have played second banana.

SEARCH THE BLOG



SUBSCRIBE



By RSS or

SUBMIT

- Because a misconfigured domain presents an obvious error message, one would assume there is no “low-hanging fruit” for attackers.

It follows that the extent of this vulnerability may not be obvious to those who’ve casually read other research on this topic. There are a couple [reports on HackerOne](#) but I believe this issue is still relatively unexplored, especially considering the severity. When subdomains from a high-trust domain are hijacked, they can be set up as a watering hole for delivering malware that has a high likelihood of bypassing filtering mechanisms.

What caught me off-guard was the prevalence of exploitable domains. Almost 2,000 domains were parked and turned over to Amazon during this exercise to prevent them from being exploited. This came as a result of simply finding the right targets and scripting the testing process.

Background

CloudFront is a Content Delivery Network (CDN) provided by Amazon Web Services (AWS). CloudFront users create “distributions” that serve content from specific sources (an S3 bucket, for example).

Each CloudFront distribution has a unique endpoint for users to point their DNS records to (ex. `d111111abcdef8.cloudfront.net`). All of the domains using a specific distribution need to be listed in the “Alternate Domain Names (CNAMEs)” field in the options for that distribution.

When a CloudFront endpoint receives a request, it does NOT automatically serve content from the corresponding distribution. Instead, CloudFront uses the `HOST` header of the request to determine which distribution to use. This means two things:

- If the `HOST` header does not match a domain in the “Alternate Domain Names (CNAMEs)” field of the intended distribution, the request will fail.
- Any other CloudFront distribution that contains the specific domain in the `HOST` header will receive the request and respond to it normally.

This is what allows the domains to be hijacked. There are many cases where a CloudFront user fails to list all the necessary domains that might be received in the `HOST` header.

Example

- The domain `test.disloops.com` is a CNAME record that points to `disloops.com`

POPULAR POSTS

CloudFront Hijacking

1,496 views

Lateral Movement with PSEXEC

188 views

10 Steps to Successful Privileged Access Management – Part 1

83 views

10 Steps to Successful Privileged Access Management – Part 2

68 views

10 Steps to Successful Privileged Access Management – Part 3

60 views

Privilege Escalation via Group Policy Preferences (GPP)

52 views

Improving Provisioning in VMware with Ansible

42 views

The AWS Shared Responsibility Model: Part 1 – Security in the Cloud

30 views

Voter Privacy vs. the Security of the Electronic Voting System

21 views

The AWS Shared Security Model – Part II: A Step Towards FedRAMP Compliance

15 views

CATEGORIES

Access Control

Application Security

Architecture and Engineering

Authentication

Awareness and Training

Blacklisting

Breach

Business Continuity

Certification and Accreditation

Cloud

- The `disloops.com` domain is set up to use a CloudFront distribution.
- Because `test.disloops.com` was not added to the “Alternate Domain Names (CNAMEs)” field for the distribution, requests to `test.disloops.com` will fail.
- Another user can create a CloudFront distribution and add `test.disloops.com` to the “Alternate Domain Names (CNAMEs)” field to hijack the domain.

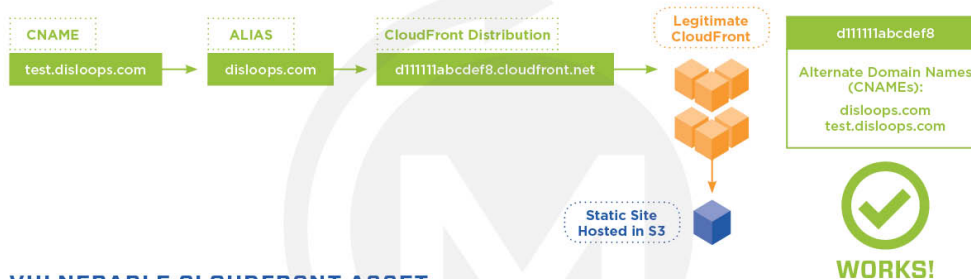
This means that the unique endpoint that CloudFront binds to a single distribution is *effectively meaningless*. A request to one specific CloudFront subdomain is not limited to the distribution it is associated with. See below (click to enlarge):

[Compliance](#)
[Configuration Management](#)
[Continuous Monitoring](#)
[Cyber Security](#)
[Disaster Recovery](#)
[Encryption](#)
[Engineering and Architecture](#)
[FedRAMP](#)
[FISMA](#)
[Full Disk Encryption](#)
[Honeypot](#)
[Information Leakage](#)
[Insider Threat](#)
[Mobile](#)
[National Security](#)
[Open Source](#)
[Operations](#)
[Organized Crime](#)
[Passwords](#)
[Pen Test](#)
[Physical Security](#)
[Policy and Procedure](#)
[Privacy](#)
[Program Management](#)
[Qualitative Analysis](#)
[Quantitative Analysis](#)
[Risk Assessment](#)
[Risk Management](#)
[Security Appliance](#)
[Security Monitoring Infrastructure](#)
[Security Operations Center](#)
[SOC](#)
[Social Engineering](#)
[Trusted Platform Module](#)
[Vulnerability Assessment](#)
[Vulnerability Management](#)
[White Paper](#)

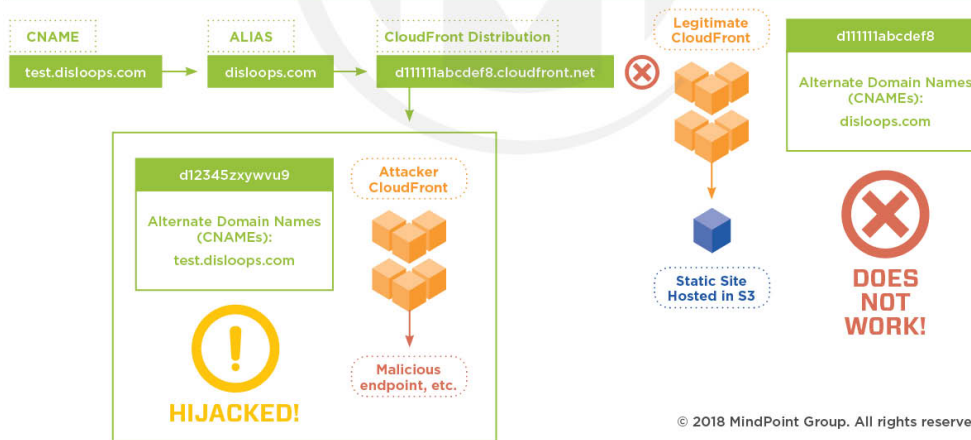
NO CDN



CORRECT CLOUDFRONT USAGE



VULNERABLE CLOUDFRONT ASSET



© 2018 MindPoint Group. All rights reserved.

Figure 1: A typical scenario in which a CloudFront configuration leaves a domain vulnerable to hijacking.

Research

Without going into a ton of detail, I wanted to script the process of finding vulnerable domains. I created a Python script called *CloudFront* that:

- Accepts a list of domains
- Runs the domains through [dnsrecon](#) to find more domains
- Selects the domains that actually point to [CloudFront IP space](#)
- Tests them for configuration issues
- (Optionally) adds them to a new CloudFront distribution

The script itself is here: <https://github.com/MindPointGroup/cloudfrunt>

```
$ python cloudfrunt.py -o cloudfrunt.com.s3-website-us-east-1.amazonaws.com -i S3-cloudfrunt -l list.txt
CloudFrunt v1.0.3

[+] Enumerating DNS entries for google.com
[-] No issues found for google.com

[+] Enumerating DNS entries for disloops.com
[+] Found CloudFront domain --> cdn.disloops.com
[+] Found CloudFront domain --> test.disloops.com
[-] Potentially misconfigured CloudFront domains:
[#] --> test.disloops.com
[+] Created new CloudFront distribution EXBC12DE3F45G
[+] Added test.disloops.com to CloudFront distribution EXBC12DE3F45G
```

The next step was to create a list of promising targets. Ultimately, the best solution was just to scrape [Robtex](#) for a premium report on every IP in the CloudFront address space.

That search yielded 90,500 unique domains attached to about a million IP addresses. So I created an EC2 instance to test from, split up the list and ran CloudFrunt against it in parallel.

Results

After a few days of allowing the script to run, I found that I had added almost 2,000 new domains to CloudFront distributions of my own. Each of them were automatically configured to point to the following page, which has undergone some revisions: [CloudFront Hijacking Demo](#)

It was immediately clear that this project had a bigger impact than anticipated. Among the affected domains were some owned by:

- Two distinct US Federal “dot gov” organizations
- Bloomberg Businessweek

- Commonwealth Bank of Australia
- Dow Jones
- Harvard University
- The League of Conservation Voters
- Red Cross
- Reuters
- University of Maryland
- ...and others

(Update 4/5/18: We're up to seven subdomains on three different US "dot gov" sites that have been parked and reported.)

The surprising effectiveness of this exercise meant that the reporting timeline was about to be expedited. I was in touch with an engineer from the AWS CloudFront team to discuss the findings about one week after the first tests began.

- Dec 29, 2017 – Initial investigation
- Jan 2, 2018 – Automated testing begins with increased capacity
- Jan 5, 2018 – Large number of domains have been found and squatted
 - Initial contact with AWS CloudFront engineers to discuss findings
 - Full incident report is created and submitted to AWS Security team
 - Federal domains are reported separately to US-CERT at [NCCIC](#)
 - AWS engineers are given access to the [CloudFront](#) scanning tool
- Jan 8, 2018 – Control of the vulnerable domains is transitioned to the AWS CloudFront team

Reporting and Remediation

We actually reported this issue to two different groups within AWS. Initially we reported directly to the CloudFront service team, but within twenty-four hours we had also reported the issue via the AWS public security channels.

From the outset, the CloudFront team worked quickly to take over hosting of the vulnerable domains. The domains were transferred to a CloudFront distribution under their control, which now points at a [landing page](#) of their own creation. As of writing this, many of the domains are still parked there.

A number of discussions with both the AWS Security team and the CloudFront engineers have followed. The CloudFront team accepts that the nuances of CloudFront's routing mechanism that lead to this condition leave room for improvement. However, AWS has deemed that this is not a vulnerability in the CloudFront service.

So after following the AWS disclosure process and working to protect the domains we identified, we are releasing this research and the open-source [CloudFruent](#) scanning tool itself on GitHub.

Conclusion

These issues are part of the growing pains of cloud adoption. For CloudFront in particular, most AWS customers with a single distribution can protect themselves by adding a wildcard domain (such as *.dis1oops.com) to the "Alternate Domain Names (CNAMEs)" field.

After this year's S3 bucket exposures, Amazon rolled out changes to the console and began alerting users with open buckets. They provided a similar service to users that uploaded their [API keys to GitHub](#) once the issue became pervasive. Depending on the impact of CloudFront domain hijacking, certain safeguards could appear in the near future.

(Update 4/4/18: As predicted, the CloudFront console is displaying a new popup when a user removes a CNAME from a CloudFront distribution warning them to keep their DNS records in sync with their CloudFront distribution.)

Please use [CloudFruent](#) to test your own organization for misconfigurations. It is simple to generate a list of domains in every Route 53 hosted zone to test against. Let me know if you have any issues or improvements to suggest!

CloudFruent open-source tool can be viewed/downloaded from the following GitHub repositories:

<https://github.com/MindPointGroup/cloudfruent>

<https://github.com/disloops/cloudfrunt>


Additional blog posts by Matt Westfall:

<https://disloops.com/>

 Follow @disloops 160 followers


 Follow @mindpointgroup 529 followers

About Latest Posts



Matt Westfall
Team Lead at [MindPoint Group](#)

Follow me



Categories: [Application Security](#), [Breach](#), [Cloud](#), [Configuration Management](#), [Cyber Security](#), [Pen Test](#) and tagged [Alternate Domain Names](#), [Amazon Web Services](#), [AWS](#), [CDN](#), [CloudFront](#), [CNAMEs](#), [Content Delivery Network](#), [Hijack](#), [S3](#)

Share:  Like 61   Tweet  Share

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

POST COMMENT

This site uses Akismet to reduce spam. [Learn how your data is processed.](#)

Contact Us

1330 Braddock Place, Suite 600
Alexandria, VA 22314
703.636.2033
info@mindpointgroup.com

Stay Connected



All Rights Reserved. ©2016 MindPoint Group
[Privacy Policy](#) | [Sitemap](#)