CloudFlare for Command and Control

CloudFlare has a free service that protects your website against DDoS attacks, crawling, brute-force, and generic web application attacks. That's all great, but it also offers quick content delivery through its fast network, URL rewrite, caching rules, firewall rules, user-agent blocking, analytics, and even SSL certificates issued by CloudFlare!

For an adversary simulation shop looking to use traditional HTTP/S infrastructure for initial C2, this could prove to be a useful option. It adds a lot of features that can be gotten from Apache rules and such but through a web GUI - making it insanely easy to install.

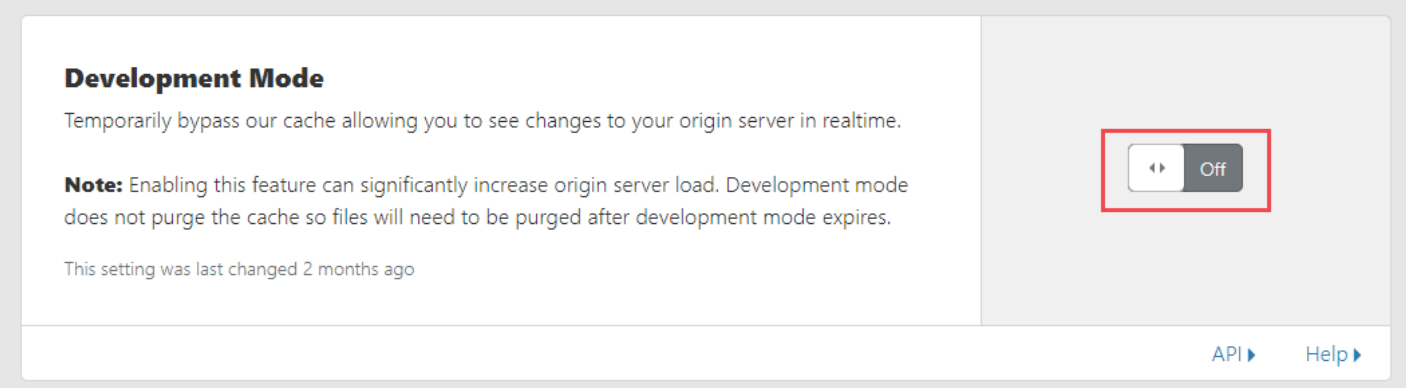
Setting up CloudFlare

It's relatively simple to set up CloudFlare, you sign-up and insert your domain name. CloudFlare will then request that you set your nameservers to theirs, doing so will give them access over the DNS for your domain. I'd like to think CloudFlare is relatively trust-worthy, given that we're doing this for legitimate purposes.

your domain by CloudFlare - fantastic.

Lessons learnt

CloudFlare by default loves to cache items. When setting up CloudFlare for C2, you must put the instance into Development mode, as shown below:



This ensures that your requests will always provide fresh responses.

Other useful options

Under `Crypto`, you can also set to always use HTTPS to on. This will force HTTPS to always be in-use and so that end-users cannot connect via. HTTP.

Under *firewall* you can specify firewall rules based on Country, ASN, or even IP ranges. This is especially useful if you want to white-list a certain country so that you don't receive a shell for a country you're not targeting. It helps with scope control. You can also combine this functionality with Jason Lang's list of popular security company IP ranges to prevent their security appliances from connecting to your infrastructure. This helps to prevent Sandboxes from analysing the payload in some scenarios, or even obtaining a stage, or decryption key.

User-agent blocking is another useful feature that can be deployed to force known incident user-agents such as `curl` and `wget` to a CloudFlare block page.

CF-IPCountry is header that's included in the CF request to your origin. If you can view this header, it can be used to quickly identify what country the request is coming from.

Other options to be wary of

E-mail address obfuscation under *scrape shield* might affect your C2 channel if your response has any e-mail addresses in the body. The e-mail addresses are essentially replaced with an image representation to prevent automatic scraping tools.

Additional opsec and overall security

For those who don't want to bother clicking on another link:

```
sudo apt-get install ipset
ipset create cf hash:net
for x in $(curl https://www.cloudflare.com/ips-v4); do ipset add cf $x; done
iptables -A INPUT -m set --match-set cf src -p tcp -m multiport --dports http,htt
```

Once again, you're trusting CloudFlare to not get hacked, or for them to not stick in a command injection at that `curl` URL specified.

I wrote this command that does some validation that might work better (untested):

```
curl https://www.cloudflare.com/ips-v4 | sort -u | grep -e '^\([0-9]\{1,3\}\.\){
| sed -e 's/^/sudo iptables -I INPUT -s /g' | sed -e 's/$/ -j ACCEPT/g' | bash

sudo iptables -I INPUT -s 0.0.0.0/0 -j DROP
```

Conclusion

wasting your time trying.

CloudFlare helps to extrapolate a lot of the configuration file changes required when using Apache mod re-write as well as the task of issuing certificates.

Previous
Vultr Domain Hijacking

Next - Red Team
Command and Control

Last updated 7 months ago

WAS THIS PAGE HELPFUL?