

# The Pitfall of Threat Intelligence Whitelisting: Specter Botnet is 'taking over' Top Legit DNS Domains By Using CloudDNS Service

[blog.netlab.360.com/the-pitfall-of-threat-intelligence-whitelisting-specter-botnet-is-taking-over-top-legit-dns-domains-by-using-cloudns-service/](https://blog.netlab.360.com/the-pitfall-of-threat-intelligence-whitelisting-specter-botnet-is-taking-over-top-legit-dns-domains-by-using-cloudns-service/)

Hui Wang

November 18, 2021

18 November 2021 / [DNS](#)

## Abstract

In order to reduce the possible impact of false positives, it is pretty common practice for security industry to whitelist the top Alexa domains such as [www.google.com](#) , [www.apple.com](#) , [www.qq.com](#) , [www.alipay.com](#) .

And we have seen various machine learning detection models that bypass data when they sees these popular Internet business domains.

The security war between the white and black never ends, white hats want to see “black” in the data, while hackers always try to blend in and appear “white”. In the follow article, we will see an interesting case which shows that the white we see is not necessarily white.

Our BotMon tracking system recently highlighted that the Specter botnet family started to use two domains [api.github.com](#) and [www.ibm.com](#) as C2 domains for its control communicate, while everyone knows for sure there is just no way for these FQDN to be malicious. The hacker utilized a pretty bizarre feature from one public DNS provider CloudDNS to make this all possible.

Doing this will definitely bring troubles to IoC based treat intelligence security, as the C2 are absolutely white.

## Origins

We first disclosed the **Specter botnet** back in September last year ([September 2020](#)). The botnet is a remote control trojan (RAT) for Linux platforms with flexible configuration and highly modular/plugin-based, It is consist of three major modules: Dropper, Loader, and Plugin, with the main functions determined by the Loader & Plugin, and this botnet has always been active since our disclosure.

In September this year, our BotMon's C2 auto-extraction system alerted us that there was an update of Specter's sample and the auto-extracted C2 was [api.github.com](#) on its port [80](#) .

There is no need to explain what api.github.com is, although we have seen many malware using github.com before, they pretty much all just use its web service to download their own malicious programs.

But here how can Specter use api.github.com as its C2 communication node and pass control traffic back and forth between github and its bots? Was github hacked, or was it a bug in our own C2 auto-extraction module?

We took a close look at the sample (md5: `2aec3f06abd677f5f129ddb55d2cde67`), and saw that the Specter update focused on the structure of the C2 configuration file, the C2 config in the previous versions could be located by searching the `SpecterCF` string. The new version eliminated this. The following is the decrypted C2 configuration file of this sample, the green part is the C2: `api.github.com:80` mentioned above.

```

00000000: 44 FD 0D 9F 80 CF 49 23 9F D5 D9 6D AF 59 FE D9 D? ?@I#???m?Y??
00000010: 5C B4 CB DD 38 26 40 6A 8D E6 AB 66 50 D1 B2 92 \???8&@j???fP??
00000020: 70 61 75 65 36 7A 63 33 00 01 00 00 0E 00 00 paue6z3    0
00000030: 00 61 70 69 2E 67 69 74 68 75 62 2E 63 6F 6D 02 api.github.com
00000040: 00 00 00 38 30 9E E9 9F 55 1E 01 3B 6C E1 B7 B6 80???U  1???
00000050: D9 A9 B4 CE B9 00 00 00 00 08 07 00 00 01 00 00 ??????  0
00000060: 00 01 00 00 00 01 00 00 00 01 00 00 00 01 00 00  0
00000070: 00 01 00 00 00 01 00 00 00 01 00 00 00 0A 00 14  0
00000080: 00 0A 00 14 00  0

```

C2,PORT
DNS IP

The data in red is the new from this update, and what is it? After parsing in small-end format, we found that they are the following 4 IP addresses, belonging to the DNS Hosting provider CloudDNS

```

85.159.233.158
108.59.1.30
217.182.183.225
185.206.180.169

```

The new Specter sample send dns request to C2 using the following code snippet, which has the logic to craft the dns request packets and the ask the DNS IPs described above about the FQDN to finally get the C2 address.

```

v18 = compose_dns_packet((const char *)((v10[7] << 24) | (v10[6] << 16) | (v10[5] << 8) | v10[4]), &v12, &v13, 1u);
if ( v18 )
{
    v15 = random_proc(0, 5);
    for ( i = 0; i <= 4; ++i )
    {
        v3 = v15 + i;
        v14 = v3 % 5;
        v4 = v3 % 5 + 2;
        if ( (v10[4] * v4 + 3) << 24) | (v10[4] * v4 + 2) << 16) | (v10[4] * v4 + 1) << 8) | v10[4] * v4 )
        {
            v17 = dnsquery(
                (int)18,
                &v12,
                (v10[4] * (v14 + 2) + 3) << 24) | (v10[4] * (v14 + 2) + 2) << 16) | (v10[4] * (v14 + 2) + 1) << 8) | v10[4] * (v14 + 2)],
                0x35u,
                5,
                3u);
        }
    }
}

```

Readers can use the dig command below on their own and see the difference quite clearly by comparing their output.

```
dig api.github.com @8.8.8.8
```

```
dig api.github.com @85.159.233.158
```

At this point the fog clears and the C2 `api.github.com` used by Specter is actually a subdomain under ZONE `github.com` registered with the DNS Hosting provider CloudDNS. As long as the hacker uses the resolution server provided by CloudDNS, the resolution of `api.github.com` can be any IP the hacker picks.

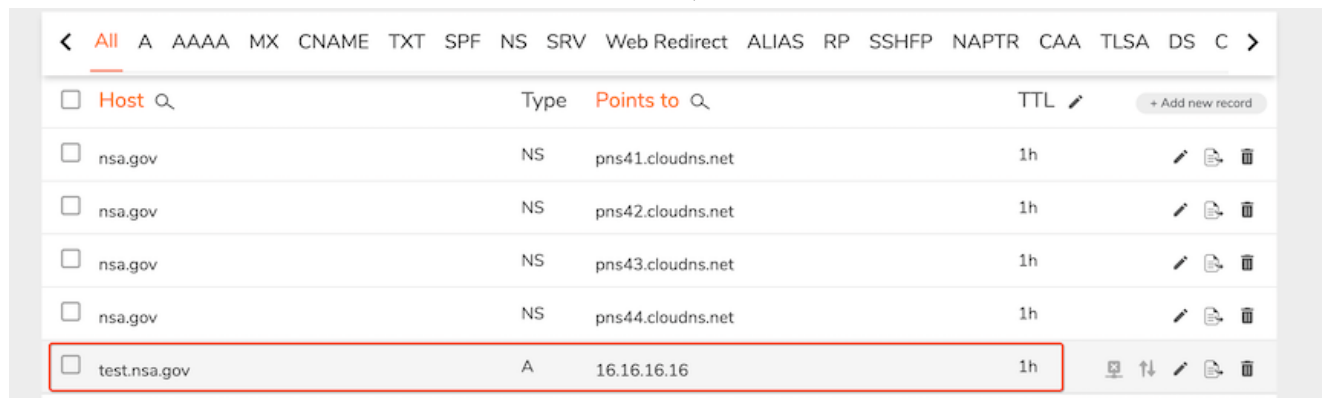
Github was not hacked, the Specter botnet operator did not enter the wrong C2 domin, our C2 auto-extraction was not buggy, but the white domain `api.github.com` did indeed become a working C2 domain for this botnet. And this totally legit domain can easily deceive malware analysts and is a great challenge for security tools based on black and white list rules.

## CloudDNS

Given the above exploitation process, let's explore CloudDNS a little bit more here.

CloudDNS is a global managed DNS service provider based in Europe, offering services including GeoDNS, Anycast DNS and DNS DDoS protection.

CloudDNS allows arbitrary registration of DNS Zones and the addition of sub-domain resolution. We registered(and later removed after test) a DNS Zone named `nsa.gov` , added a sub-level domain name `test` and resolved to `16.16.16.16` . CloudDNS assigned us 4 Name Servers to resolve this domain name, as shown below.



The screenshot shows a DNS management interface with a table of records. The table has columns for Host, Type, Points to, and TTL. The record for test.nsa.gov is highlighted with a red box.

Host	Type	Points to	TTL
nsa.gov	NS	pns41.cloudns.net	1h
nsa.gov	NS	pns42.cloudns.net	1h
nsa.gov	NS	pns43.cloudns.net	1h
nsa.gov	NS	pns44.cloudns.net	1h
test.nsa.gov	A	16.16.16.16	1h

Once created successfully, the Name Servers assigned by the platform can be used to resolve the domain name we created.

```
~]$ dig test.nsa.gov @pns44.cloudns.net

; <<> DiG 9.10.6 <<> test.nsa.gov @pns44.cloudns.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7009
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;test.nsa.gov.                IN      A

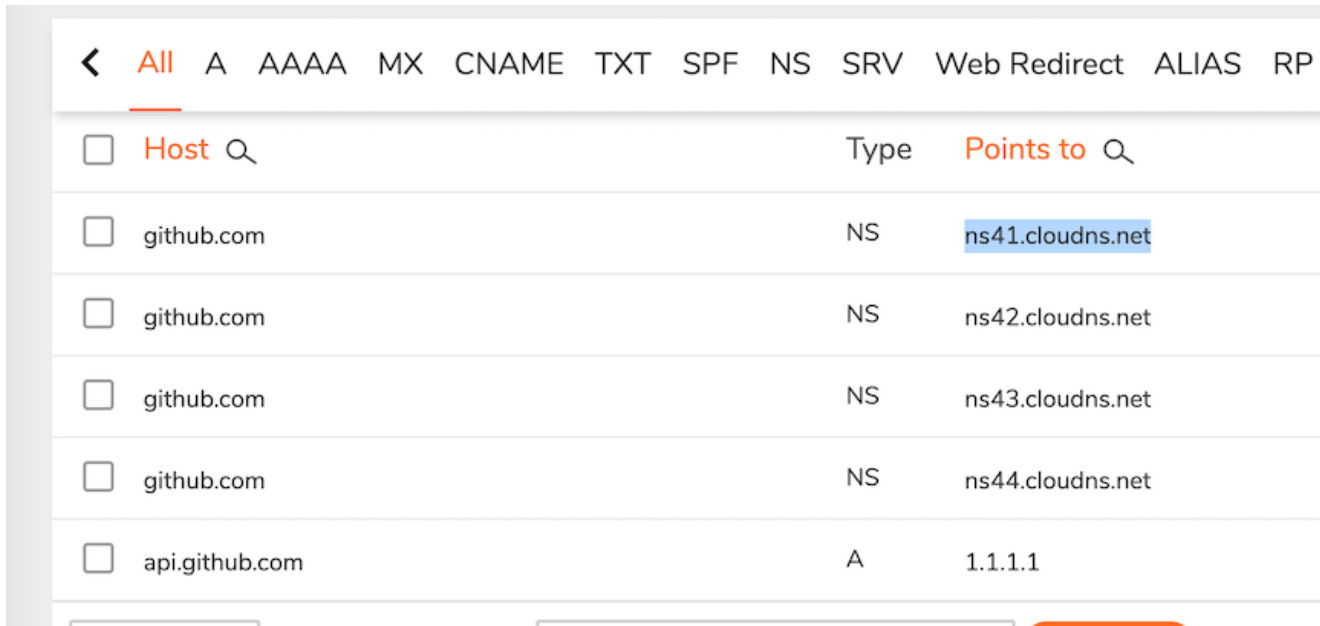
;; ANSWER SECTION:
test.nsa.gov.                3600    IN      A      16.16.16.16

;; AUTHORITY SECTION:
nsa.gov.                     3600    IN      NS     pns44.cloudns.net.
nsa.gov.                     3600    IN      NS     pns42.cloudns.net.
nsa.gov.                     3600    IN      NS     pns41.cloudns.net.
nsa.gov.                     3600    IN      NS     pns43.cloudns.net.
```

Theoretically, we can register any Zone on CloudDNS that is not registered or not restricted by CloudDNS, and the aforementioned Specter C2 [api.github.com](https://api.github.com) is a domain name generated in this way.

Not only that, but CloudDNS also has a "mysterious logic" in determining whether a domain is "registered" or not. As mentioned earlier, [github.com](https://github.com) was already registered on CloudDNS by the Specter gang, but when we tried to re-register the github.com Zone, we were able to

do so, just with a different batch of NSs than the Specter gang, as shown here.



The screenshot shows a DNS lookup interface with a navigation bar at the top containing tabs for 'All', 'A', 'AAAA', 'MX', 'CNAME', 'TXT', 'SPF', 'NS', 'SRV', 'Web Redirect', 'ALIAS', and 'RP'. The 'All' tab is selected. Below the navigation bar is a table with columns for 'Host', 'Type', and 'Points to'. The table contains the following records:

Host	Type	Points to
github.com	NS	ns41.cloudns.net
github.com	NS	ns42.cloudns.net
github.com	NS	ns43.cloudns.net
github.com	NS	ns44.cloudns.net
api.github.com	A	1.1.1.1

So, CloudDNS supports creating same zones as long as they are on their different NS Servers, this is pretty bizarre behavior.

In fact, based on our test, not only CloudDNS, but also some other DNS hosting providers, have similar "vulnerabilities" in the verification of hosted domains, this is not the topic to be covered in this article though.

## Explore CloudDNS Random Registration ZONES

Based on our own Passive DNS data, we selected the TOP 1M popular second-level domains and did some serious tests. We wanted to find out how many SLDs of domains in the existing DNS system were registered with CloudDNS as new Zones and how many of them could be malicious.

The results of the probe showed that there were approximately 300 second-level domains that were registered in bad faith. Some of the maliciously registered SLDs in CloudDNS are as follows.

akadns.net  
onedrive.com  
plivo.com  
safe.com  
consalud.cl  
godaddysites.com  
shopee.com  
jsdelivr.net  
afraid.org  
rumahweb.com  
mydomain.com  
crypto.com  
eq.edu.au  
adnxs.com  
webcindario.com  
web.com  
lamborghini.com  
manager-magazin.de  
toto.com  
migalhas.com.br  
googleadservices.com  
example.com  
dlink.com  
whitehouse.gov  
domain.com  
googlesyndication.com  
fb.com  
payeer.com  
ya.ru  
mql5.com  
aaa.com  
hola.com  
wukong.com  
mihanblog.com  
wpengine.com  
jumia.ma  
protonmail.com  
tasnimnews.com  
nintendo.com  
tabnak.ir  
lichess.org  
digitalocean.com  
asriran.com  
amazon.com.br  
akamaized.net  
yjc.ir  
office.net  
4399.com  
opera.com  
wp.com  
yting.com  
avast.com  
cloudflare.com  
playstation.com  
hespress.com

leagueoflegends.com  
wixsite.com  
skype.com  
googlevideo.com  
wp.pl  
wix.com  
samsung.com  
doubleclick.net  
weebly.com  
udemy.com  
speedtest.net  
godaddy.com  
zoom.us  
espn.com  
spotify.com  
amazonaws.com  
adobe.com  
wordpress.com  
apple.com  
msn.com  
github.com  
office.com  
alipay.com  
netflix.com  
360.cn  
amazon.com  
qq.com

In addition, we also selected the popular TOP 1M FQDNs across to check against ClouDNS, and the results showed that there are over **300** FQDNs that can generate non-normal resolution in ClouDNS, and after clean up, we found that at least **192** FQDNs are maliciously registered.

## Summary

---

We have yet to see other malicious actors using this technique on a large scale, however, this is an important reminder for us that there are cases of malicious behavior being carried out under the cover of apparently normal network behavior.

## Contact us

---

Readers are always welcomed to reach us on [Twitter](#) or email us to netlab at 360 dot cn.

## IOC

---

### Sample MD5

---

0ffa01708fd0c67c78e9055b8839d24d  
162c245378b2e21bdab6ef35dfaad6b1  
2aec3f06abd677f5f129ddb55d2cde67

45.141.70.5

www.ibm.com @pns101.cloudns.net

api.github.com @ns103.cloudns.net

 [360 Netlab Blog - Network Security Research Lab at 360](#)

—  
The Pitfall of Threat Intelligence Whitelisting: Specter Botnet is 'taking over' Top Legit DNS Domains By Using CloudDNS Service