KCon

纵横

顶级域攻击与接管指南

演讲者：ztz

> ABOUT ME

- ztz @360高级攻防实验室
- 专注攻防对抗研究、漏洞挖掘、武器开发
- // Google Hall of Fame
- // Golang
- // Ruby
- // Homebrew

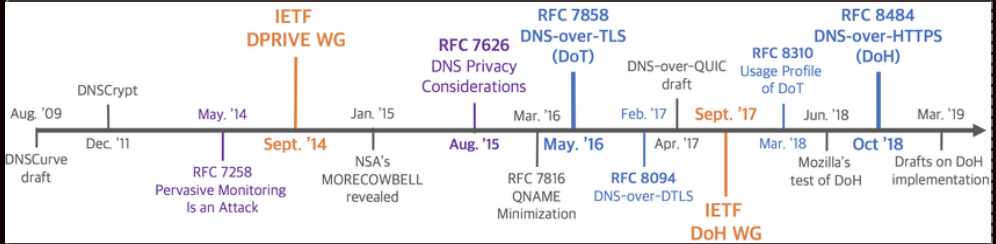
> 主要内容

- DNS Protocol 介绍
- 云 DNS 场景劫持探讨
- 顶级域 DNS 场景劫持探讨

> DNS 协议

- 诞生于 1983 年，不断更新到现在
- 实现主机与 IP 地址转化的“互联网电话簿”
- ICANN 负责域名系统的维护和运作

> DNS 协议


Aug. '09
DNSCurve draft
DNSCrypt
Dec. '11
May. '14
RFC 7258 Pervasive Monitoring Is an Attack
IETF DPRIVE WG
Sept. '14
Jan. '15
NSA's MORECOWBELL revealed
RFC 7626 DNS Privacy Considerations
Aug. '15
Mar. '16
RFC 7816 QNAME Minimization
RFC 7858 DNS-over-TLS (DoT)
May. '16
DNS-over-QUIC draft
Feb. '17
Apr. '17
RFC 8094 DNS-over-DTLS
Sept. '17
IETF DoH WG
RFC 8310 Usage Profile of DoT
Mar. '18
Mozilla's test of DoH
Jun. '18
RFC 8484 DNS-over-HTTPS (DoH)
Oct '18
Mar. '19
Drafts on DoH implementation

> 域名组成

sub.domain.tld.　根域

子域　权威域　顶级域

> 根域

- 根域是域名系统中最高级别解析域
- 根域服务器是 DNS 中最高级别的域名服务器，提供到顶级域服务器的映射
- 根域服务器地址硬编码在递归 DNS 服务器中

List of Root Servers

HOSTNAME	IP ADDRESSES	OPERATOR
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	Verisign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California, Information Sciences Institute
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	Verisign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

> 顶级域（TLD）

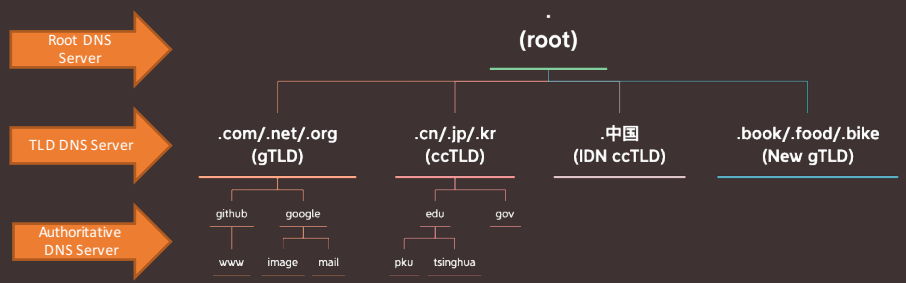
- 顶级域是互联网域名系统的等级中，位于根域空间的最高级域名，由 IANA 管理，分发托管商负责
- 托管商提供某一顶级域下所属域名解析服务
- https://data.iana.org/TLD/tlds-alpha-by-domain.txt

最初创立的通用顶级域	
通用顶级域	用途
.com	商业机构；**现无限制**
.edu	供教育机构使用
.gov	供美国政府的联邦、州及地方各级部门使用
.mil	供美国军事机构使用
.net	原供网络服务供应商使用；**现无限制**
.org	原供不属于其他通用顶级域类别的组织使用；**现无限制**
IANA – 完整的顶级域列表（页面存档备份，存于互联网档案馆）	

> 权威域

- 域名注册商面向市场提供权威域购买和注册服务
- 权威域由实际购买者负责管理和维护

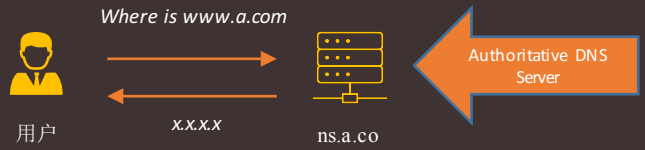
> 域名组成


Root DNS Server
TLD DNS Server
Authoritative DNS Server
.
(root)
.com/.net/.org (gTLD)
.cn/.jp/.kr (ccTLD)
.中国 (IDN ccTLD)
.book/.food/.bike (New gTLD)
github
google
edu
gov
www
image
mail
pku
tsinghua

> 注册域名时实际在注册什么

> 注册域名时实际在注册什么

> 注册域名时实际在注册什么



> 权威域 DNS 到底在哪里

- 域名注册商通常提供权威域 DNS 解析服务，域名购买后默认使用域名注册商 DNS 托管解析
- 单独 DNS 解析服务提供商：Cloudflare、各类云解析 DNS

> 云解析 DNS 管理权限交接

- 权限交接发生在更换 DNS 托管商时
- 老 DNS 托管商使用 EPP 协议请求顶级域 Zone file 更新，修改该域名 NS 指向到新 DNS 托管商
- 顶级域 Zone file 更新，权威域 DNS 解析服务器完成指向更新
- 云解析 DNS 验证 NS 指向成功，用户获得控制台管理权

> 云解析 DNS 接管风险

- 先验证，后管理

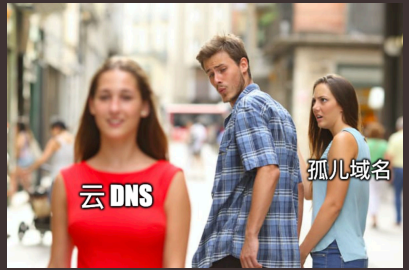
> 云解析 DNS 接管风险

- 孤儿域名：已绑定域名被解绑，NS 指向未修改

> 云解析 DNS 接管风险

- 孤儿域名接管：攻击者在控制台添加孤儿域名，完成域名接管

> 云解析 DNS 接管风险



> 云解析 DNS 接管风险

- 孤儿域名接管获得域名控制权
- 云 DNS 解析服务器域名通常托管于该云 DNS 中

孤儿域名 + 云 DNS = 云 DNS 域名接管

> 云解析 DNS 接管风险

- 举例说明
- 某云 DNS 解析服务器 ns1.xxcloud.cn
- 大量用户域名通过 ns1.xxcloud.com 托管

> 云解析 DNS 接管风险

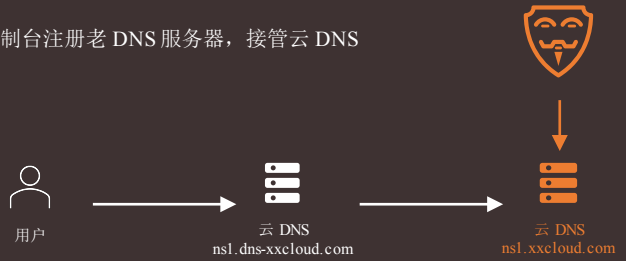
- 某云进行架构升级，在老 DNS 和用户之间加入一层新 DNS

> 云解析 DNS 接管风险

- 因缘巧合，此时老 DNS 服务器成为孤儿域名

> 云解析 DNS 接管风险

- 攻击者控制台注册老 DNS 服务器，接管云 DNS

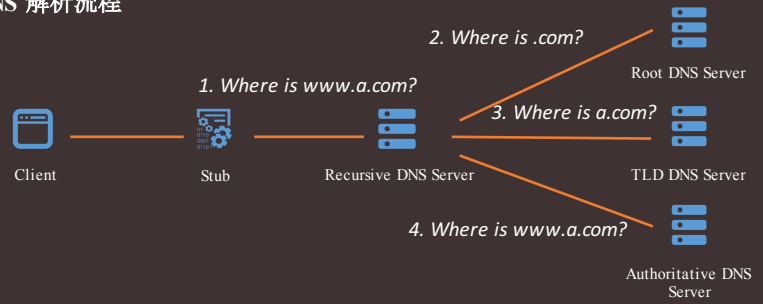


> 云解析 DNS 接管风险

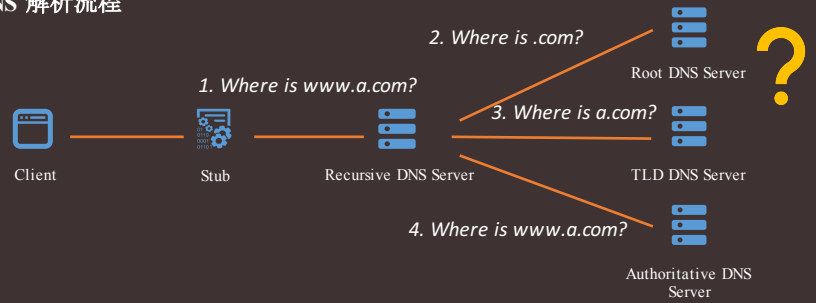
> 顶级域 DNS 接管风险

- 同样的道理，顶级域 DNS 是否存在类似问题？

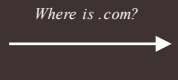
> DNS 解析流程


2. Where is .com?
Root DNS Server
1. Where is www.a.com?
3. Where is a.com?
Client
Stub
Recursive DNS Server
TLD DNS Server
4. Where is www.a.com?
Authoritative DNS Server

> DNS 解析流程


2. Where is .com?
Root DNS Server
1. Where is www.a.com?
3. Where is a.com?
Client
Stub
Recursive DNS Server
TLD DNS Server
4. Where is www.a.com?
Authoritative DNS Server

> DNS 解析流程



Root DNS Server

Root Zone file

```
com. 86400 IN NS m.gtld-servers.net.
com. 86400 IN NS c.gtld-servers.net.
com. 86400 IN NS e.gtld-servers.net.
com. 86400 IN NS l.gtld-servers.net.
```

> 顶级域接管

- 国外安全研究员 @IAmMandatory
- The .io Error – Taking Control of All .io Domains With a Targeted Registration - @IAmMandatory
- 接管过期的 .io 顶级域 NS 服务器
- 目前顶级域 NS 是否存在该问题？@fate0

```
bash-3.2$ dig NS io. @k.root-servers.net.

; <<>> DiG 9.8.3-P1 <<>> NS io. @k.root-servers.net.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19611
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 7, ADDITIONAL: 12
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;io.                IN  NS

;; AUTHORITY SECTION:
io.          172800  IN  NS  ns-a1.io.
io.          172800  IN  NS  ns-a2.io.
io.          172800  IN  NS  ns-a3.io.
io.          172800  IN  NS  ns-a4.io.
io.          172800  IN  NS  a0.nic.io.
io.          172800  IN  NS  b0.nic.io.
io.          172800  IN  NS  c0.nic.io.
```

> 顶级域接管

```
neo@oracle:~/projects/tlns$
neo@oracle:~/projects/tlns$
neo@oracle:~/projects/tlns$ sudo massdns -o J -r ./resolvers.txt -t NS ./tlds-alpha-by-domain.txt > tld_ns.txt
```

```
Concurrency: 10000
Processed queries: 1495
Received packets: 1709
Progress: 100.00% (00 h 00 min 08 sec / 00 h 00 min 08 sec)
Current incoming rate: 0 pps, average: 213 pps
Current success rate: 0 pps, average: 186 pps
Finished total: 1494, success: 1494 (100.00%)
Mismatched domains: 195 (11.41%), IDs: 0 (0.00%)
Failures: 0: 32.60%, 1: 36.88%, 2: 19.28%, 3: 7.03%, 4: 3.21%, 5: 0.60%, 6: 0
%, 18: 0.00%, 19: 0.00%, 20: 0.00%, 21: 0.00%, 22: 0.00%, 23: 0.00%, 24: 0.00
%, 36: 0.00%, 37: 0.00%, 38: 0.00%, 39: 0.00%, 40: 0.00%, 41: 0.00%, 42: 0.00
Response:  | Success:              | Total:
OK:        |      1494 (100.00%)   |      1688 ( 98.77%)
NXDOMAIN:  |         0 (  0.00%)   |         0 (  0.00%)
SERVFAIL:  |         0 (  0.00%)   |        21 (  1.23%)
REFUSED:   |         0 (  0.00%)   |         0 (  0.00%)
FORMERR:   |         0 (  0.00%)   |         0 (  0.00%)
```

> 顶级域接管

```
neo@oracle:~/projects/tlns$ cat tld_ns.txt | jq -r .data.answers[]?.data
a2.nic.ABARTH.
c0.nic.ABARTH.
a0.nic.ABARTH.
b0.nic.ABARTH.
m.dns.nic.ACO.
a.dns.nic.ACO.
n.dns.nic.ACO.
v2n0.nic.ACTOR.
v0n1.nic.ACTOR.
v2n1.nic.ACTOR.
v0n3.nic.ACTOR.
v0n2.nic.ACTOR.
v0n0.nic.ACTOR.
gtld.beta.aridns.net.au.
gtld.alpha.aridns.net.au.
gtld.delta.aridns.net.au.
gtld.gamma.aridns.net.au.
ad.cctld.authdns.ripe.net.
```

> 顶级域接管

```
neo@oracle:~/projects/tlns$ cat tld_ns.txt | jq -r .data.answers[]?.data | sort | uniq -i | wc -l
5671
```

> 顶级域接管

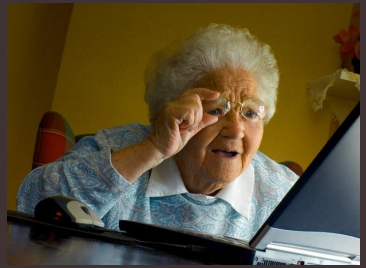
- 第一次尝试，寻找未注册或过期的顶级域 NS

```
neo@oracle:~/projects/tlns$ massdns -o J -r ./resolvers.txt -t A ./tld_ns.txt | grep NXDOMAIN
```

> 顶级域接管

```
00%, 36: 0.00%, 37: 0.00%, 38: 0.00%
Response:  | Success:                  |
OK:        |              [illegible]  |
NXDOMAIN:  |                3 (  0.05%)|
SERVFAIL:  |                0 (  0.00%)|
REFUSED:   |                0 (  0.00%)|
FORMERR:   |                0 (  0.00%)|
[illegible]
```

> 顶级域接管



> 顶级域接管

- 意料之中：域名均已注册，只是暂未设置 DNS 解析

> 顶级域接管

- NS 主从之间通过 AXFR 或 IXFR 域传送同步解析记录
- TLD 主从 NS 之间是否存在不一致现象？

```
c0.nic.io.
neo@oracle:~/projects/tlns$ dig ns io. @a2.nic.io. +short | sort
a0.nic.io.
a2.nic.io.
b0.nic.io.
c0.nic.io.
neo@oracle:~/projects/tlns$ dig ns io. @a0.nic.io. +short | sort
a0.nic.io.
a2.nic.io.
b0.nic.io.
c0.nic.io.
neo@oracle:~/projects/tlns$ dig ns io. @b0.nic.io. +short | sort
a0.nic.io.
a2.nic.io.
b0.nic.io.
c0.nic.io.
```

> 顶级域接管

- 开始 TLD 主从 NS 之间不一致现象研究
- 5k+ 条 TLD NS 记录，逐个分析 NS 之间的不一致现象

```
neo@oracle:~/projects/tlns$ cat check_ns_consistency.rb
#!/usr/bin/env ruby
require 'json'
require 'resolv'

require 'async/dns'


def check_domain_consistency(domain, nameservers)
  nameservers.each do |ns|
    Resolv::DNS.open(nameserver: [ns]) do |dns|
      ress = dns.getresources domain, Resolv::DNS::Resource::IN::NS
      nss = ress.map(&:name).map(&:to_s).map(&:downcase)
      if nss.sort != nameservers.sort
        puts "[+] ns diff on domain: #{domain}, ns: #{ns}"
      else
        puts "[-] no diff found in #{domain}, ns: #{ns}"
      end
    rescue StandardError => e
      puts "[!] error when query #{domain} on #{ns}: #{e}"
    end
  end
end


File.readlines('./tld_ns.txt').map { |line| JSON.parse(line) }.each do |record|
  domain = record['name']
  nameservers = record['data']['answers'].map { |answer| answer['data'].chomp('.').downcase }

  check_domain_consistency(domain, nameservers)
end
```

> 顶级域接管

- 一定数量 NS 失效：
- (46) ns-tld1.charlestonroadregistry.com
- (21) *.zdnscloud.com

```
~ dig ns FLY. +short
ns-tld5.charlestonroadregistry.com.
ns-tld2.charlestonroadregistry.com.
ns-tld4.charlestonroadregistry.com.
ns-tld1.charlestonroadregistry.com.
ns-tld3.charlestonroadregistry.com.
~
```

```
~ dig ns foo. +short
ns-tld1.charlestonroadregistry.com.
ns-tld5.charlestonroadregistry.com.
ns-tld3.charlestonroadregistry.com.
ns-tld4.charlestonroadregistry.com.
ns-tld2.charlestonroadregistry.com.
~
```

> 顶级域接管

- 部分 NS 存在解析不一致，**且不一致的解析记录可注册**
- **意味着存在接管风险**

> 总结

- DNS 协议族古老、臃肿庞大
- 老协议 + 新业务新场景 = 攻击面

感谢观看！

KCon 汇聚黑客的智慧

知道创宇 | KCon