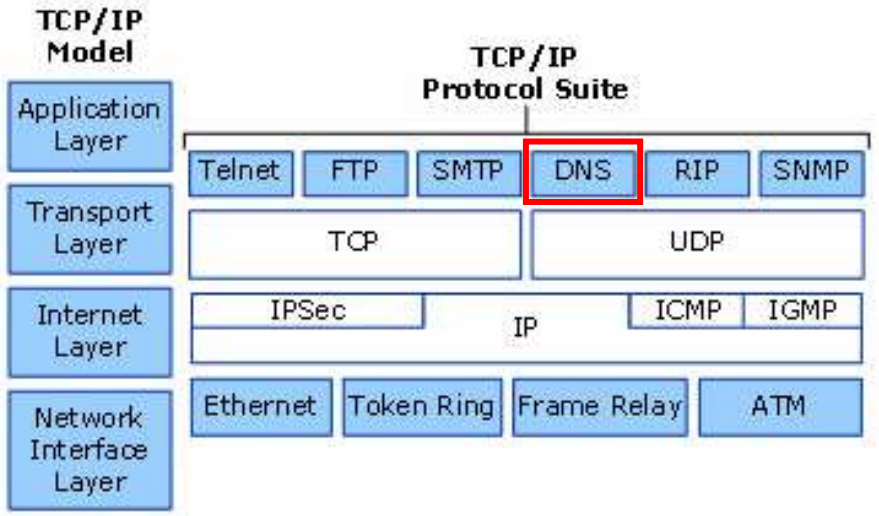
第9章 DNS安全

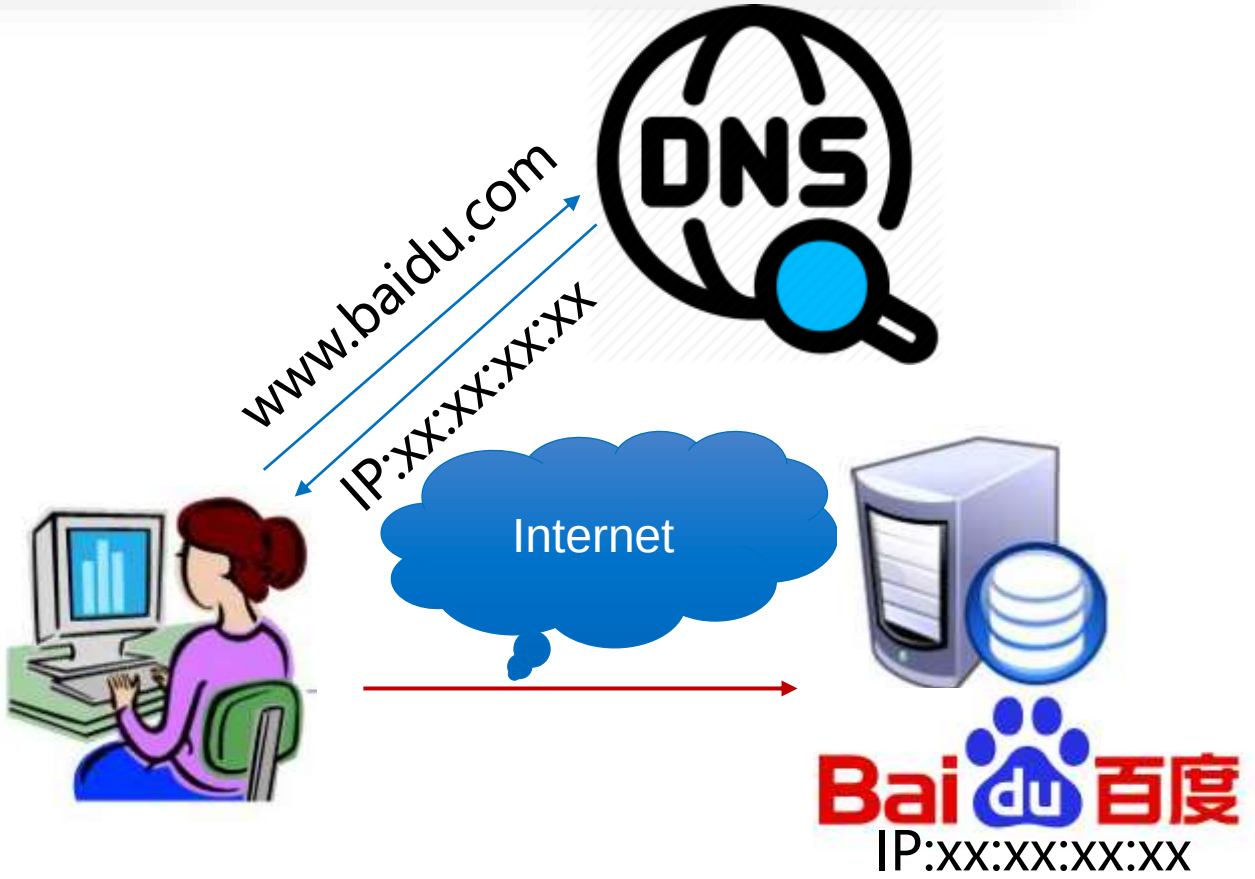
李琦
清华大学网研院

域名系统--互联网关键基础设施

域名系统（DNS）位于协议栈应用层，为互联网提供核心服务，包括web页面访问，收发邮件等互联网应用通过DNS查询IP地址后获取资源，已成为互联网关键基础设施

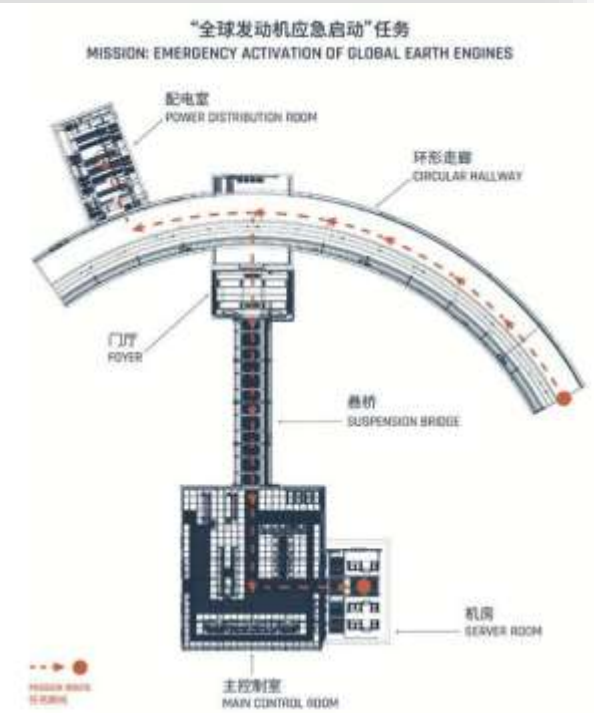


DNS位于协议栈应用层

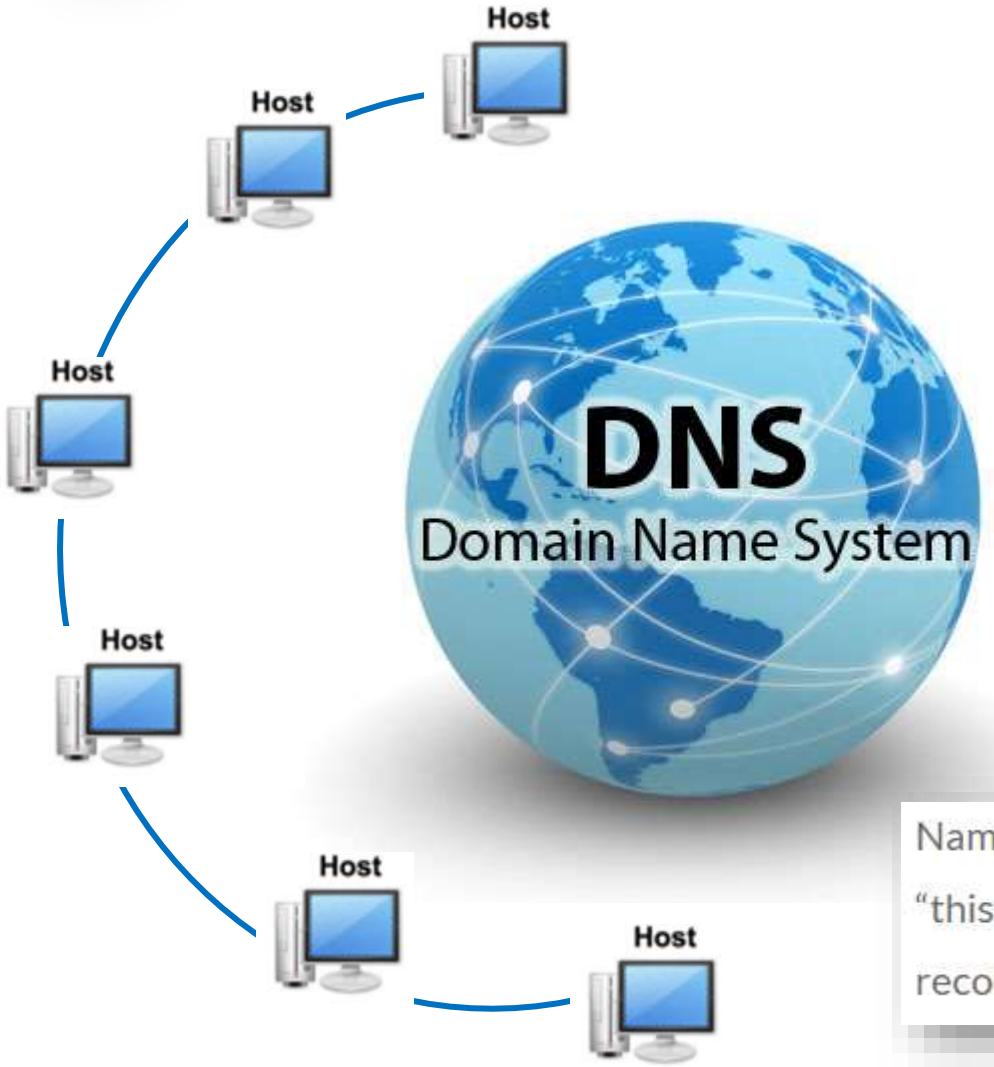


域名系统--互联网关键基础设施

域名系统（DNS）位于协议栈应用层，为互联网提供核心服务，包括web页面访问，收发邮件等互联网应用通过DNS查询IP地址后获取资源，已成为互联网关键基础设施



域名系统—超大规模分布式数据库



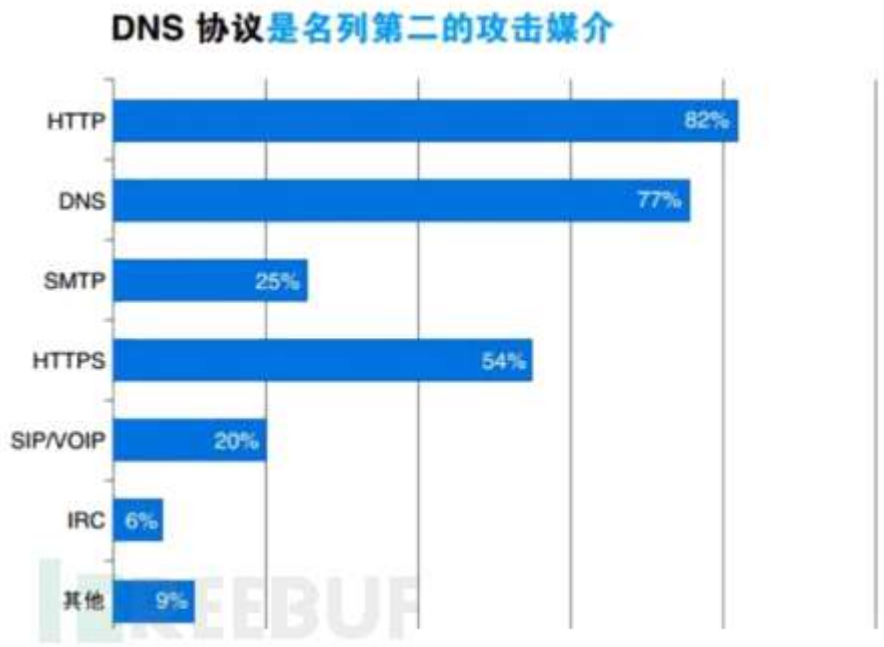
- 作为一个分布式数据库，域名系统使得任意联网计算机能够通过域名访问互联网
- 灵活的扩展性和优异的解析性能，能够持续高效地支持数亿规模的域名解析

Nameservers store DNS records which are the actual file that says “this domain” maps to “this IP address”. So is there a room somewhere that has all the nameservers and DNS records for every site on the Internet? No... that would be ridiculous.

难以采用集中式数据库

域名系统面临的安全威胁

互联网渐进式演进发展模式决定了域名系统中大量安全威胁将长期存在，不可能通过重新设计的途径解决



人民网 people.cn

人民网 >> 财经

国家顶级域名遭攻击 系统安全受关注

李雁争

2013年08月26日07:15　来源：上海证券报　手机看新闻

打印　网摘　纠错　商城　分享　推荐　人民微博　字号

原标题：国家顶级域名遭攻击 系统安全受关注

25日凌晨，部分.CN域名出现大面积瘫痪。业内人士认为，尽管目前服务正在逐步恢复，但是这一故障给我国的域名系统安全敲响了警钟。预计今后一段时间，域名系统安全保障工作力度会加大。

域名系统安全对抗不断升级

- 缓存污染、DDoS攻击等DNS安全威胁层出不穷，显示了全球DNS安全的脆弱性
- 与此同时，各类安全方案也致力于提升DNS安全性，互联网厂商如果有足够的针对自身信息系统的安全预案，就足以应对全面而复杂的威胁

- 美国国家安全局(NSA) 发布企业加密域名系统协议指南
- NSA指出，加密DNS请求的支持对于确保本地隐私和完整性保护至关重要

搜狐 | 新闻 体育 汽车 房产 旅游 教育 时尚 科技 财经

信息安全D1net

353 文章　12万 总阅读

查看TA的文章>

美国国家安全局发布企业加密DNS应用指南

2021-01-20 15:12

美国国家安全局(NSA)上周三发布了有关企业采用加密域名系统(DNS)协议(特别是基于HTTPS的DNS)指南。

DNS负责将URL中包含的域名转换为IP地址，但由于以明
成为一种流行的攻击媒介。

讨论

畅所欲言

你觉得 域名系统如果停止服务，互联网还能正常运行吗?

本章的内容组织

第一节
DNS概述

- DNS的演进
- DNS域名结构及区域组织形式

熟悉DNS运行原理

第二节
DNS使用及解析过程

- DNS使用
- DNS解析过程

思考DNS安全问题

DNS在设计之初缺乏安全考虑

第三节
DNS攻击

- 缓存中毒攻击
- 恶意DNS服务器回复伪造
- 拒绝服务攻击

第四节
DNS攻击预防策略

- 基于密码技术
- 基于系统管理
- 新型架构设计

掌握DNS攻击技术

了解DNS防御策略

多种攻击技术需要特定的防御策略进行防护

第五节
典型案例分析

- Kaminsky攻击
- 恶意服务器回复伪造攻击
- 拒绝服务攻击

第一节 DNS概述

- DNS的演进
- DNS域名结构与区域组织形式

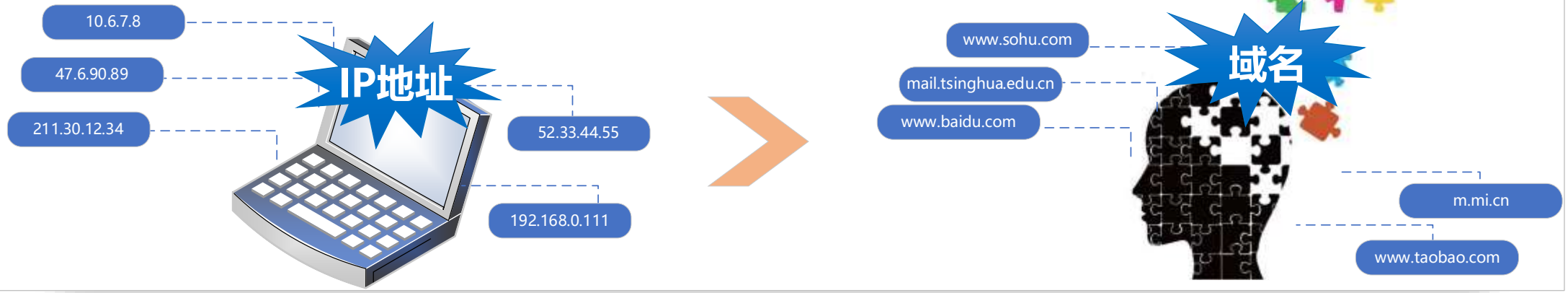
DNS的演进

网络空间两套命名体系：

用于路由寻址的IP地址和便于人类记忆的域名（Domain Name）

域名系统（DNS）功能：

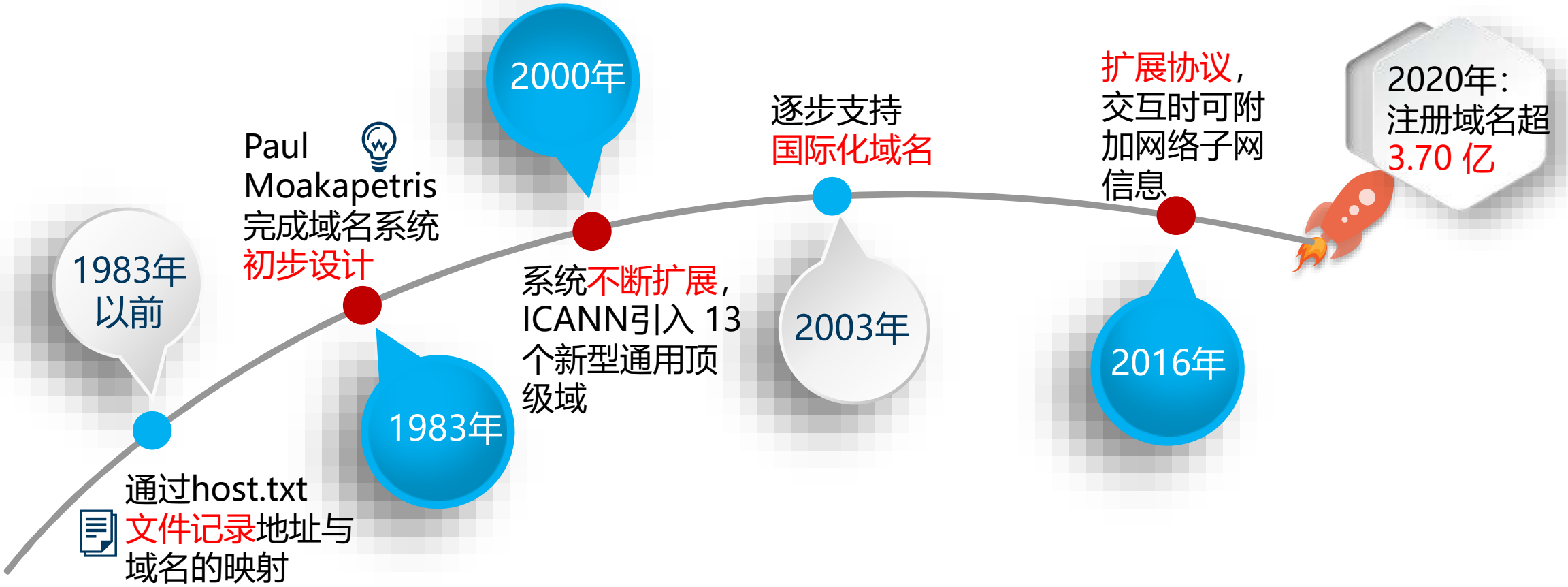
实现域名与IP间转换



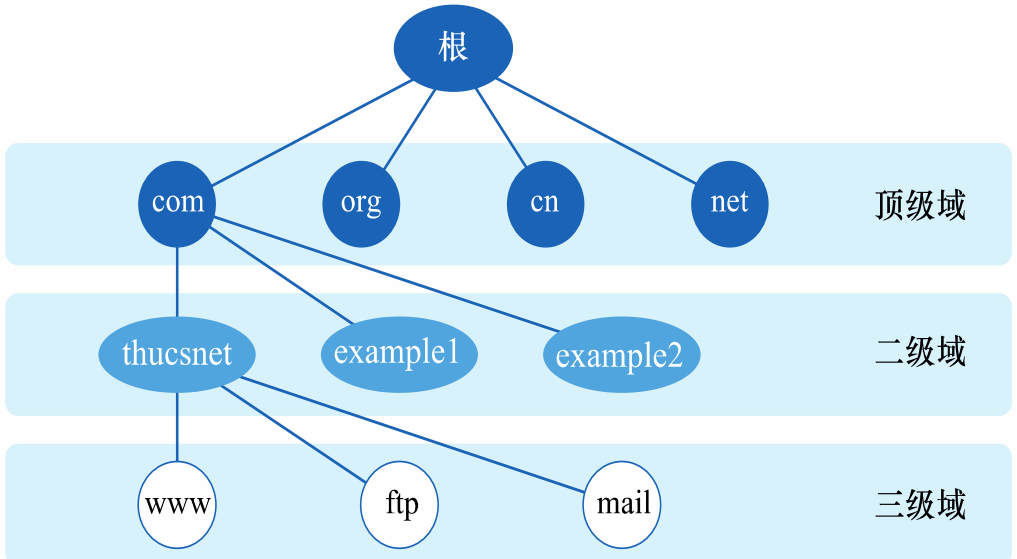
没有DNS，互联网就无法工作

DNS的演进

从host.txt文件到大型分布式系统，DNS用于地址与域名的映射



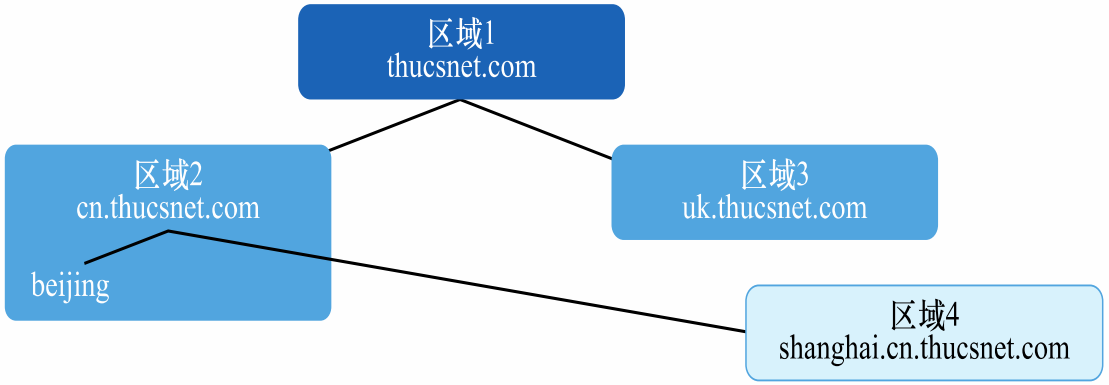
DNS域名结构



域名空间层次化授权结构

- 域名系统采用层次化树形结构
- 树形最顶层为根（Root），进一步划分顶级域（TLD），顶级域管理机构授权给二级域名（SLD）
- 层次化授权行为最多可迭代127 次

DNS区域组织形式



DNS区域组织示意图

example.org分配cn. example. org，uk. example. org两个子域，cn. example. org包含两个区域

每个DNS区域至少有一个权威域名服务器发布关于这个区域的信息

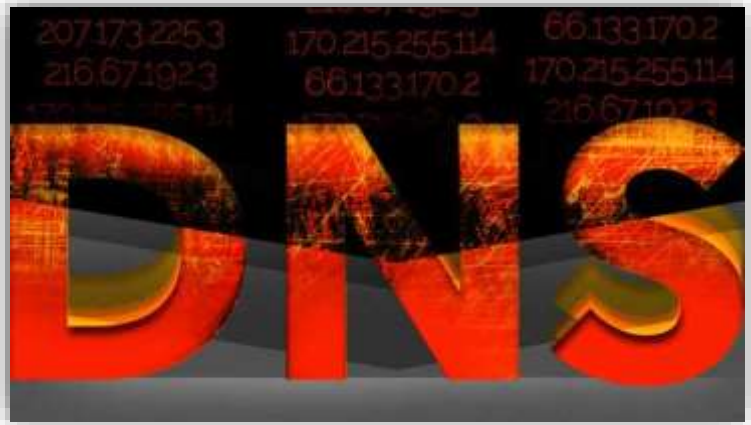
- 一个域名没有被分割成子域或包含了子域全部数据，则域名和区域相同
- 一个域名被分割成子域，每个子域有自己的区域时，域名和区域具有不同的意义

DNS区域组织形式

权威域名服务器

- 每个DNS区域的权威域名服务器，发布关于该区域的信息，并响应DNS查询请求
- 权威域名服务器可以配置主从服务器，主服务器存储所有区域记录的记录，而从服务器使用自动更新机制维护主记录的副本

权威域名服务器



DNS区域组织形式

DNS根服务器（ROOT）

List of Root Servers

HOSTNAME	IP ADDRESSES	MANAGER
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 (ISI)
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

根服务器管理互联网主目录，13台IPV4（名为“A”至“M”），1个主根服务器由美国运营，12个辅根服务器，9个由美国运营，英国和瑞典、日本各1个，每个根服务器通过anycast方式在多个国家和地区部署

https://www.internic.net/domain/root.zone

DNS区域组织形式

“雪人计划”

- 2015年6月，我国下一代互联网工程中心领衔发布运营IPv6根服务器的“雪人计划”
- 2016年在美国、日本等16个国家架设25台IPv6根服务器，形成13台原有根加25台IPv6根（3台主根）的新格局。中国部署其中4台（1台主根和3台辅根服务器）

“雪人计划”IPv6根服务器全球分布情况

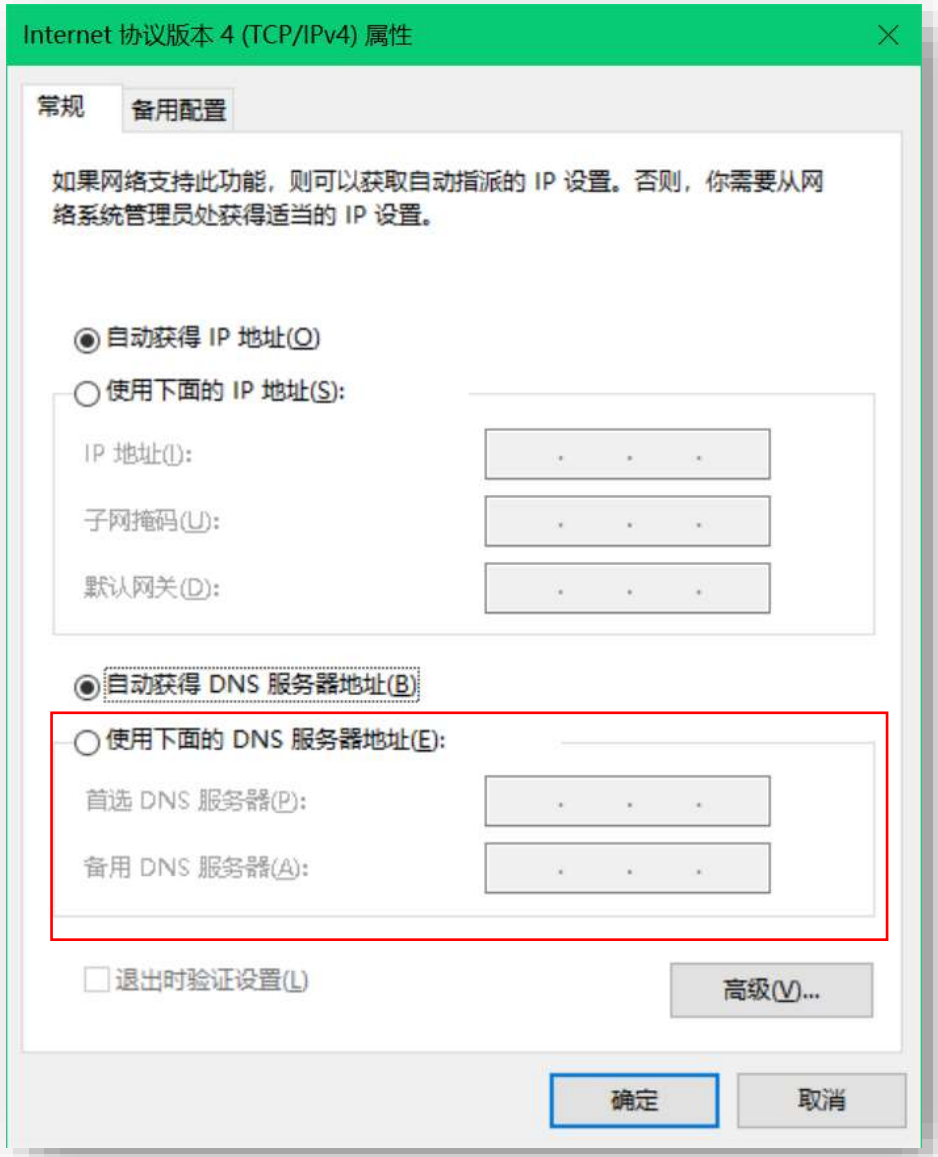
国家	主根服务器	辅根服务器	国家	主根服务器	辅根服务器
中国	1	3	西班牙	0	1
美国	1	2	奥地利	0	1
日本	1	0	智利	0	1
印度	0	3	南非	0	1
法国	0	3	澳大利亚	0	1
德国	0	2	瑞士	0	1
俄罗斯	0	1	荷兰	0	1
意大利	0	1			

第二节 DNS使用及解析过程

- DNS使用
- DNS解析过程

DNS使用

设置DNS服务器

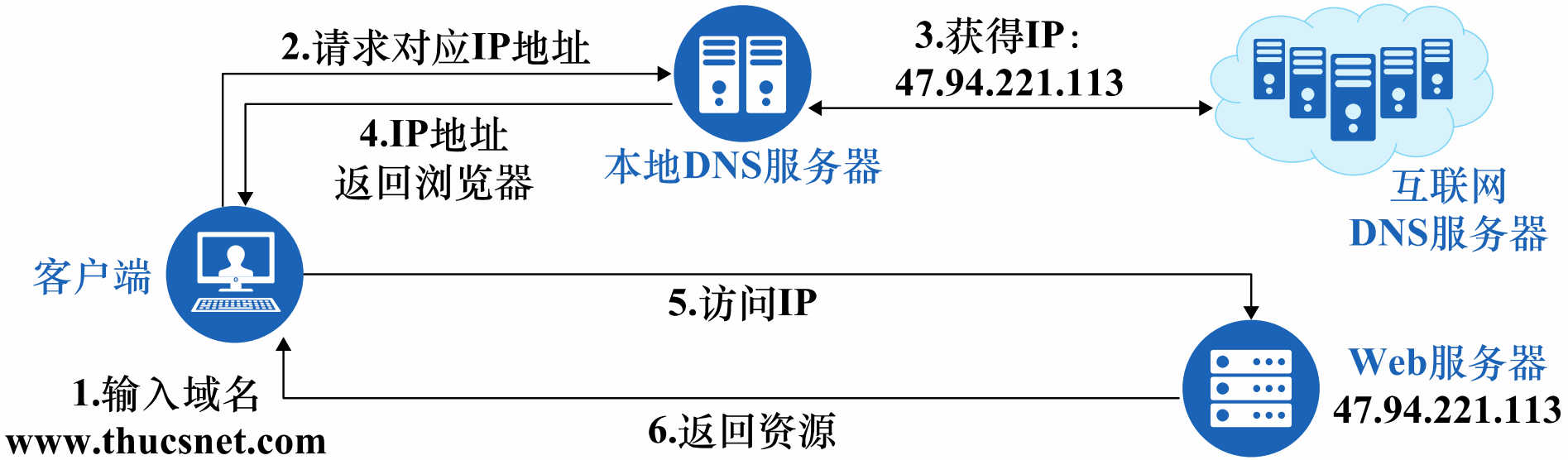


Windows中设置及查看DNS服务器

```
# Dynamic resolv.conf(5) file for glibc resolver(3) gen
erated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WIL
L BE OVERWRITTEN
nameserver 127.0.1.1
```

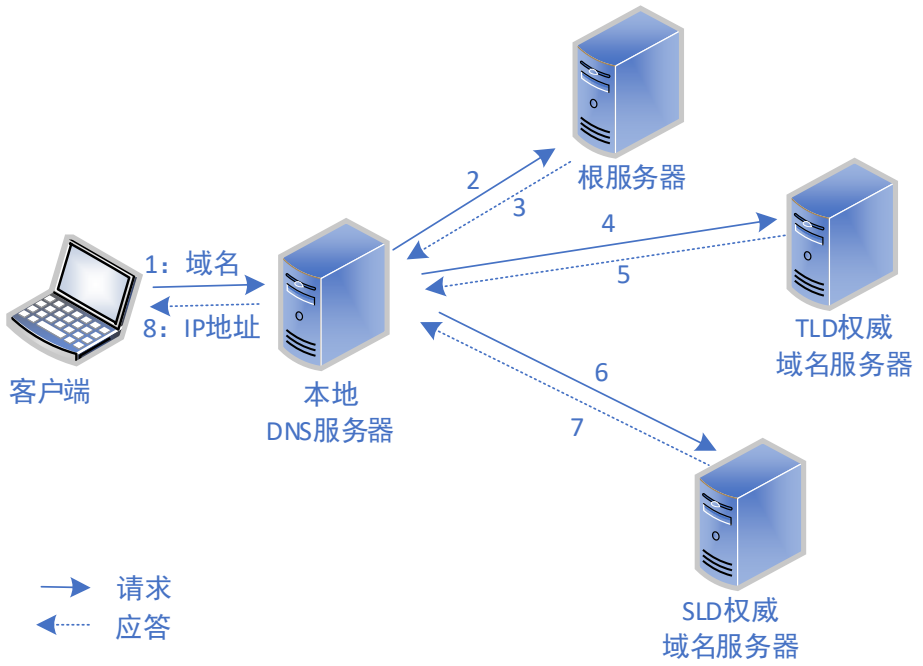
Ubutun直接修改/etc/resolv.conf

DNS使用



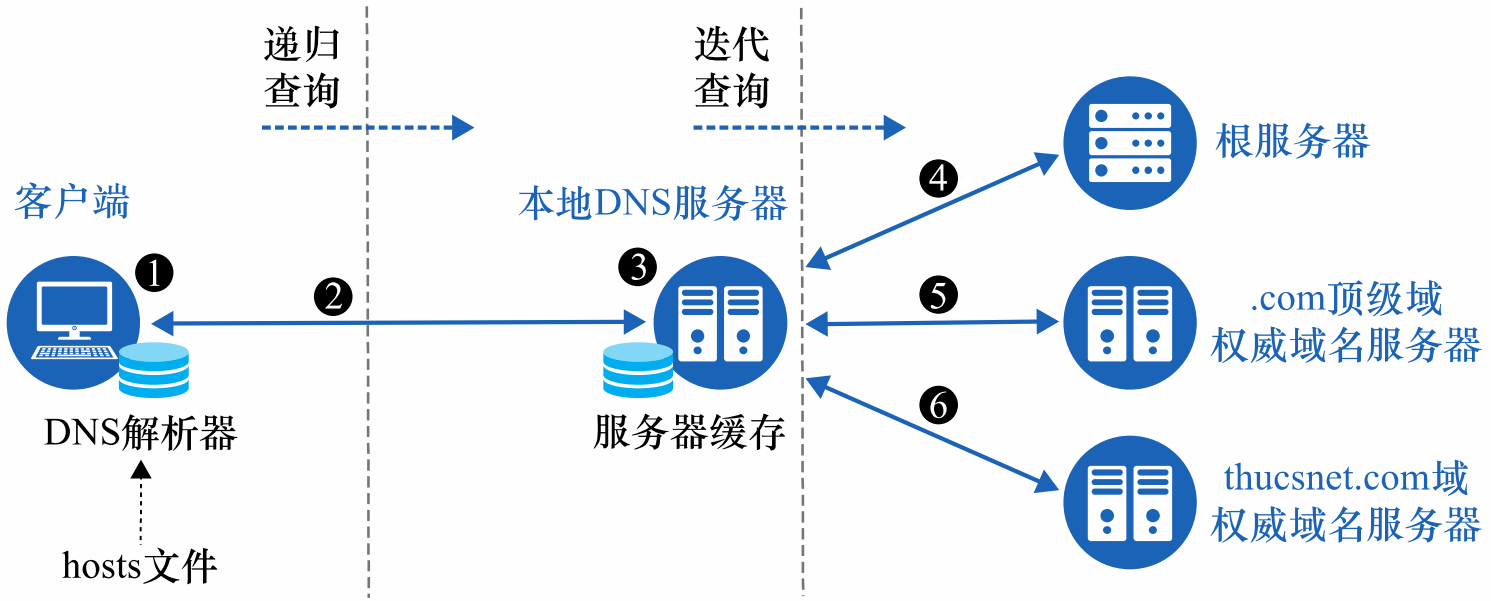
1. 浏览器输入域名
2. 请求到达DNS解析程序
3. 查询获得IP地址
4. 返回IP至Web浏览器
5. 访问IP地址
6. 浏览器显示该页面

DNS请求过程-完整流程



- 本地DNS服务器（LDNS）进行查询，权威域名服务器按照域名空间层次化结构自上而下应答
- 权威域名服务器接收请求，若所查询域名在其区域文件中则返回结果，否则返回其子域授权信息（步骤3、5、7）
- 查询方进行迭代查询，直至获得最终的解析结果并向客户端返回（8）

DNS请求过程-实际流程



1.客户端：查询本机缓存及hosts文件

2.向LDNS发送请求

3.LDNS查找缓存，有结果就返回

4-6. LDNS查找根服务器、顶级域、thucsnet.com，获取结果

7.本机获取IP地址

并不是每一次域名解析都完成整个查询流程

DNS迭代查询过程交互信息格式

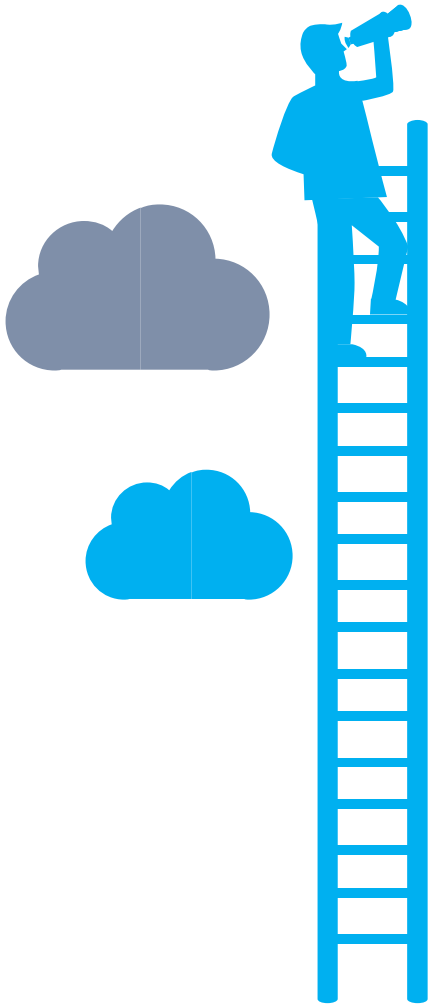
常见资源记录类型及其含义

资源记录类型	含义
AA	域名对应主机的IPv4地址
AAAAAAAA	域名对应主机的IPv6地址
MXMX	域名对应邮件服务器地址
NSNS	域名对应权威服务器名称
CNAMECNAME	域名指向的别名记录

- 域名解析应答，每个数据条目称为一个资源记录（Resource Record, RR）
- 资源记录的数据格式为标准五元组：域名、类别、类型、生存时间（TTL）与数据

DNS迭代查询过程交互信息格式

DNS应答包括问题、回复、授权和附加部分



问题区（Question Section）

查询内容，如www.thucsnet.com. IN A表示查www.baidu.com的A记录

应答区（Answer Section）

A/ CNAME/ NS等记录查询的答案

授权区（Authority Section）

通知本地DNS服务器问谁会更接近答案

附加区（Additional Section）

存放附加记录，如给出权威NS记录相应的A记录

DNS迭代查询过程交互信息格式

发送“dig @a.root-servers.net www. thucsnet.com”命令，根服务器在授权部分提供.com区域权威域名服务器，对应地址在附加部分

```
;; QUESTION SECTION:
;thucsnet.com.                  IN      A

;; AUTHORITY SECTION:
com.                    172800  IN      NS      h.gtld-servers.net.
com.                    172800  IN      NS      d.gtld-servers.net.
com.                    172800  IN      NS      f.gtld-servers.net.
com.                    172800  IN      NS      b.gtld-servers.net.

;; ADDITIONAL SECTION:
h.gtld-servers.net.     172800  IN      A       192.54.112.30
d.gtld-servers.net.     172800  IN      A       192.31.80.30
f.gtld-servers.net.     172800  IN      A       192.35.51.30
b.gtld-servers.net.     172800  IN      A       192.33.14.30
```

DNS迭代查询过程交互信息格式

LDNS通过“dig @f.gtld-servers.net www.thucsnet.com” 得到thucsnet.com区域的权威域名服务器及地址

```
;; AUTHORITY SECTION:
thucsnet.com.          172800  IN      NS      dns2.hichina.com.
thucsnet.com.          172800  IN      NS      dns1.hichina.com.

;; ADDITIONAL SECTION:
dns2.hichina.com.      172800  IN      A       106.11.141.114
dns2.hichina.com.      172800  IN      AAAA    2400:3200:2000:21::1
dns1.hichina.com.      172800  IN      A       106.11.141.113
```

DNS迭代查询过程交互信息格式

LDNS通过“dig @dns1.hichina.com www.thucsnet.com”得到IP地址

```
;; QUESTION SECTION:
;www.thucsnet.com.              IN      A

;; ANSWER SECTION:
www.thucsnet.com.       600     IN      A       47.94.221.113
```

DNS反向查询

- 使用dig -x IP，DNS解析器通过迭代查询发送请求，使用IP地址获得相关域名或主机名
- 如对地址8.8.8.8发起查询，则ANSWER SECTION得到地址对应的域名

```
$ dig -x 8.8.8.8

;; QUESTION SECTION:
;8.8.8.8.in-addr.arpa.          IN      PTR

;; ANSWER SECTION:
8.8.8.8.in-addr.arpa.   5       IN      PTR     dns.google.
```

第三节 DNS攻击

- 缓存中毒攻击
- 恶意DNS服务器回复伪造
- 拒绝服务攻击

DNS安全问题的原因及特征

DNS安全问题的本质原因?

1.客观上协议设计的不完美

2.主观上基于利益驱动，攻击者不断挖掘漏洞

DNS攻击有哪些共有特征?

1.针对明文传输和无身份认证的实体进行欺骗性攻击

2.寻找并突破域名间复杂依赖关系，实现对域名服务器攻击

3.针对防护措施不足的服务器发起拒绝服务攻击

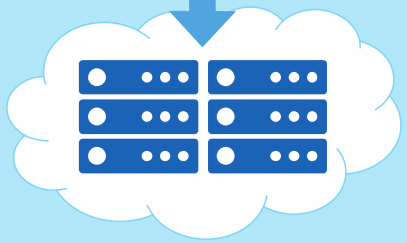
DNS攻击面

用户主机

（1）修改配置文件，导致DNS请求得到错误结果
（2）用户主机发送DNS请求时，攻击者发送欺骗回复
（3）安装流氓软件重定向DNS请求

本地
DNS服务器

本地DNS服务器向互联网DNS服务器发送请求时，攻击者发起攻击



互联网
DNS服务器

（1）用户接受来自恶意DNS服务器的虚假信息
（2）DNS服务器本身遭受拒绝服务等攻击

操控本地用户机

修改/etc/resolv.conf

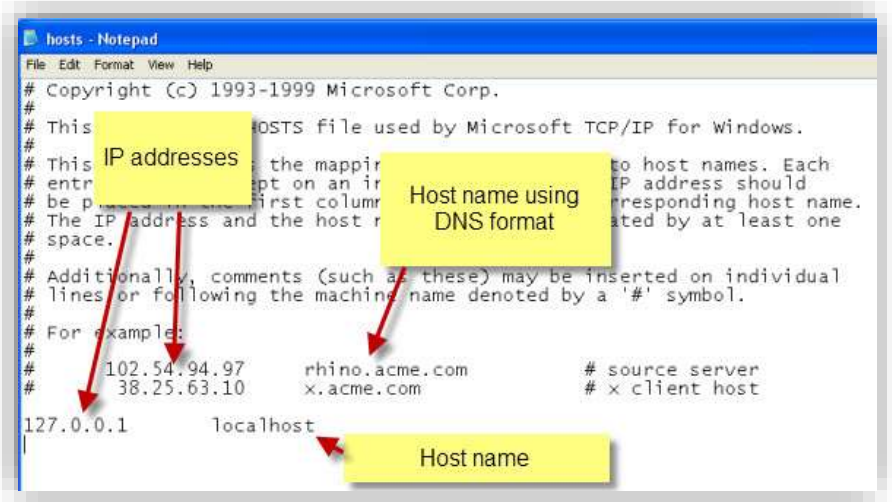
将主机DNS服务器修改成攻击者控制的服务器，攻击者回复恶意内容

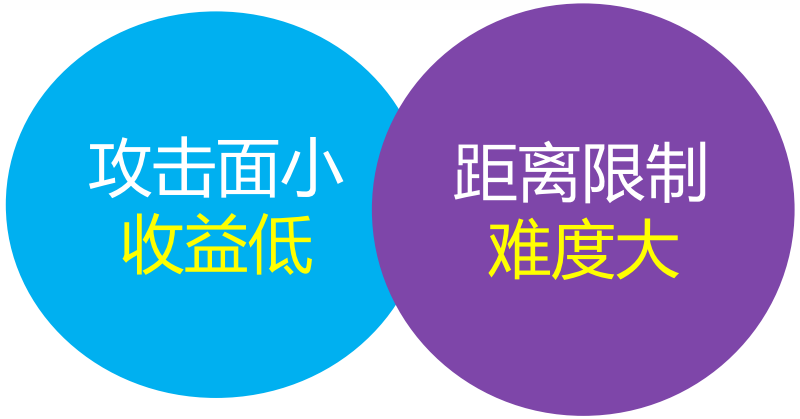
修改/etc/hosts

直接修改地址与域名映射关系，如将“www.taobao.com”修改为攻击者控制的IP地址

嗅探回复

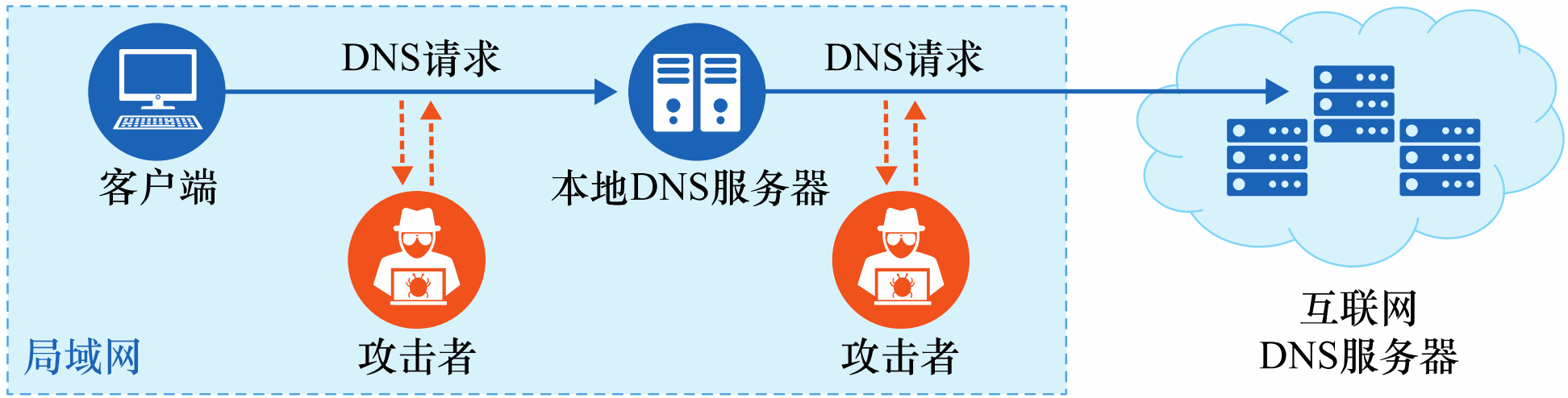
监听主机发出的DNS查询请求，注入恶意回复





本地缓存中毒攻击

- 伪造DNS回复，攻击者需要知道请求中的一些参数，如UDP源端口号、请求交易ID、请求问题等
- 由于UDP包没有加密，攻击者可以在局域网直接捕获并解析请求

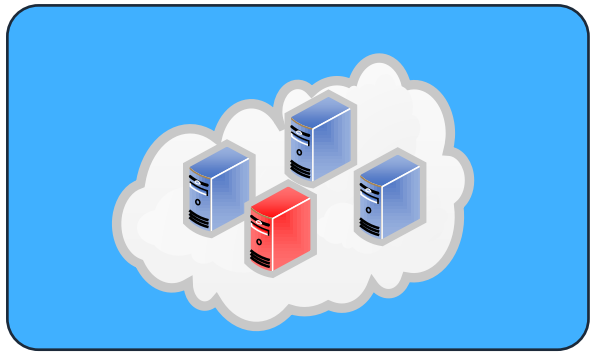


本地DNS缓存中毒示意图

本地缓存中毒攻击

只针对回复部分伪造，影响面：一个主机名

在伪造回复中，把主机名映射到攻击地址，告诉本地DNS服务器域名对应地址是攻击者的计算机



针对授权部分攻击，影响面：整个域

将ns.attack.com放在授权部分，查询目标域内任何一个主机名时，本地DNS服务器把请求发给ns.attack.com

本地缓存中毒攻击

由于能够嗅探到请求数据包，无需猜测端口号等信息，可采用scapy组装伪造数据包

```
from scapy.all import *

def spoof_ns(pkt):
  if(DNS in pkt ):
    IPpacket = IP(dst=pkt[IP].src,src=pkt[IP].dst)
    UDPpacket = UDP(dport=pkt[UDP].sport,sport=53)
    Ans = DNSRR(rrname=pkt[DNS].qd.qname,type='A',rdata='108.109.9.66',
ttl=172800)
    DNSpacket = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0,qdcount=1,
    qr=1,ancount=1,nscount=0, an=Ans)
    spoofpacket=IPpacket/UDPpacket/DNSpacket
    send(spoofpacket)
    spoofpacket.show()

    pkt=sniff(filter='udp and (src host 10.0.10.5  and dst port 53)',prn= spoof_ns)
```

本地缓存中毒攻击

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1367
;; flags: qr rd ra; QUERY: 1, ANWSER: 1, AUTHORITY: 0,
ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
; www.example.org.                    IN          A

;; ANSWER SECTION:
www.example.org.          172800      IN          A      108.109.10.66

;; Query time: 19 msec
;; SERVER: 10.0.10.5#53(10.0.10.5)
;; WHEN: Fri Jan 22 12:27:52 EST 2021
```

查询用户从本地DNS服务器得到恶意回复

远程缓存中毒攻击

远程缓存中毒攻击的难点

由于不能嗅探DNS请求，很难获取两个数据：
一是UDP16bit的头部端口号
二是DNS头部的16bit交易ID

远程攻击者猜测准确的概率为1/2^32，成功概率极低

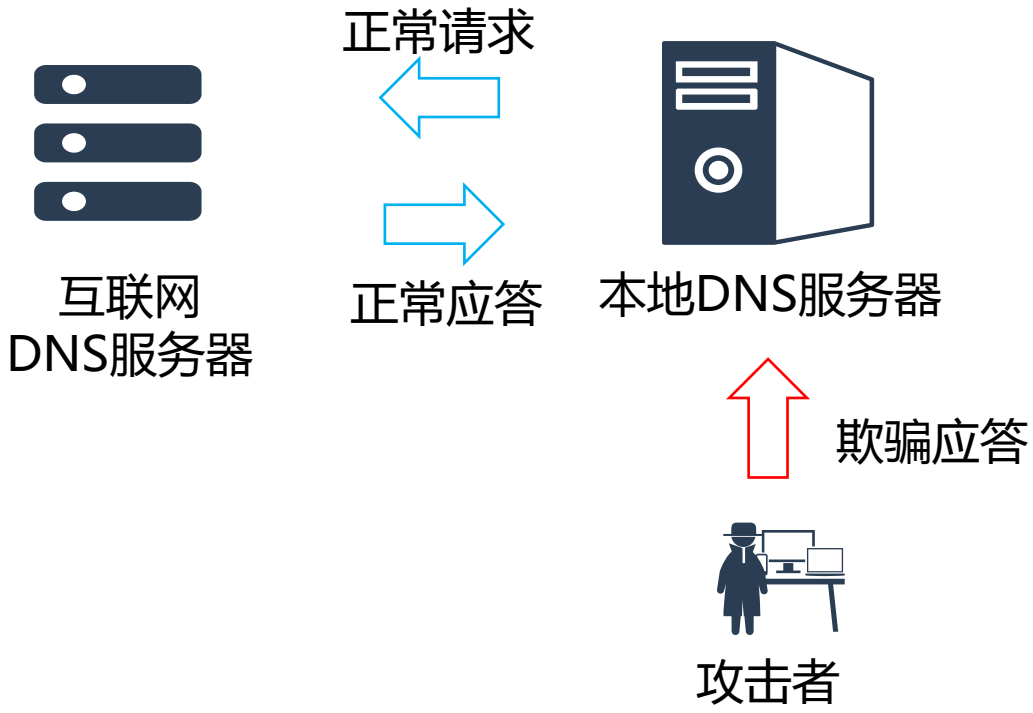
远程缓存中毒攻击

实现攻击的前提：本地DNS服务器发起域名查询请求

01 触发DNS服务器发送DNS请求

02 发送欺骗应答

03 使缓存失效



远程缓存中毒攻击

如何伪造一个被受害者接收的应答包?

接受欺骗应答面临的约束

01 缓存时限约束

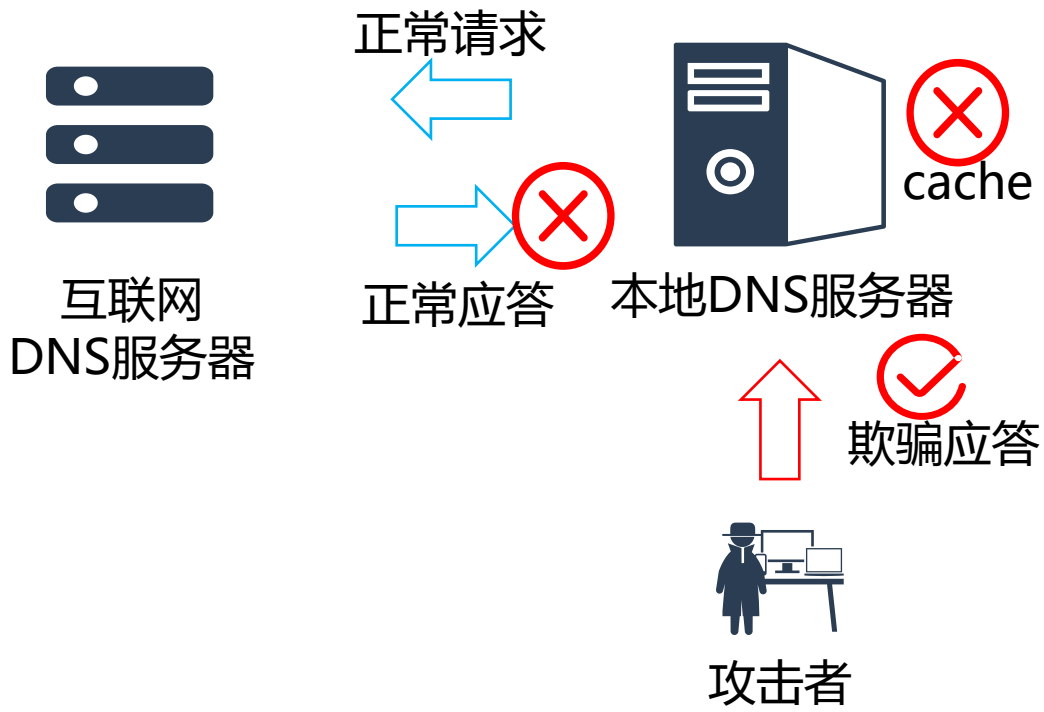
域名不能在缓存中

02 应答包匹配约束

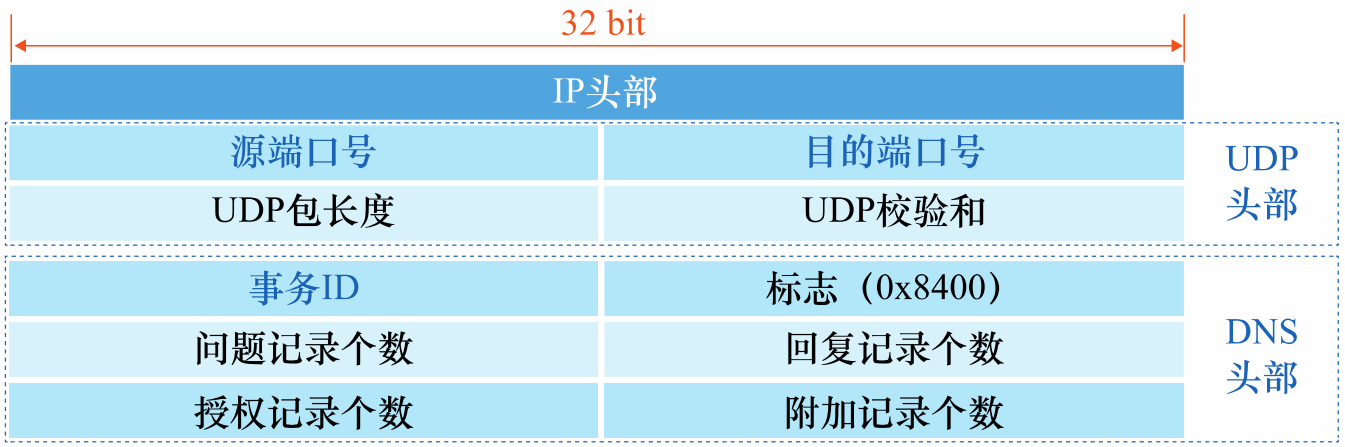
端口号、交易ID匹配

03 时间约束

伪造包比正常DNS应答快



远程缓存中毒攻击—构建回复包头



DNS回复包头格式

- 回复包有5个字段必须必须与请求包匹配，包括源IP地址、目的IP地址、源端口号、目的端口号、交易ID
- 如所攻击的域有多个域名服务器，则随机选择一个服务器发送回复包，或同时发送多个回复包

远程缓存中毒攻击—包头参数猜测

请求包源IP地址
DNS服务器地址

请求包目的端口号
DNS 端口号，53

已知

请求包的目的IP地址
权威域名服务器（可能有多个）

请求包的源端口号
16bit，随机猜测

交易ID
16bit，随机猜测

未知

远程缓存中毒攻击—构建回复包负载

问题记录

名字	记录类型	DNS类
abc.example.com	“A” Record 0x0001	Internet 0x0001

回复记录

名字	记录类型	DNS类	存活时间	数据长度	数据：IP地址
abc.example.com	“A” Record 0x0001	Internet 0x0001	0x00001000	0x0004	1.2.3.4

授权记录

名字	记录类型	DNS类	存活时间	数据长度	数据：域名服务器
abc.example.com	“NS” Record 0x0002	Internet 0x0001	0x00001000	0x0011	ns.attacker.com

DNS回复负载格式

- 问题记录和回复记录的名字应该与请求包相匹配，否则记录不会被接受
- 授权记录，域名必须和回复记录中的域名相关联，否则授权记录会被忽视

远程缓存中毒攻击—侧信道攻击

利用数据包分片重组技术注入虚假分片

利用 UDP 数据包分片重组技术，攻击者提前向递归解析服务器注入虚假 DNS 响应分片，利用数据包重组过程篡改重组后的域名解析结果，实现缓存中毒攻击

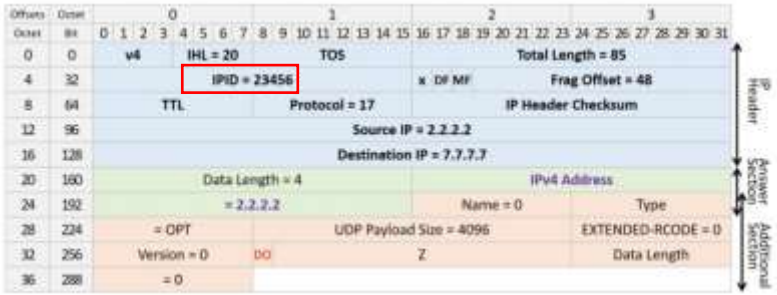


Fig. 1 ICMP fragmentation needed packet from attacker at 6.6.6.6 to nameserver at 2.2.2.2 indicating an MTU of 100 bytes for resolver at 7.7.7.7

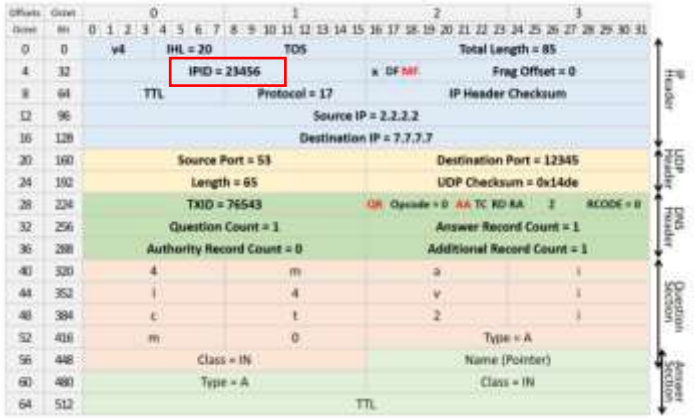


Fig. 2 First fragment sent by the nameserver at 2.2.2.2 to the DNS resolver at 7.7.7.7, assuming MTU of 68 bytes

1. Brandt M, et al. Domain validation++ for mitm-resilient PKI [C] CCS 2018: 2060-2076.

远程缓存中毒攻击—基于IPID的攻击

攻击成功的前提是受害者接受该IPID，即伪造的第二个分片与真实的第一个分片具有相同IPID值，攻击者可以采取不同的操作实现

Sequentially Incrementing	Per-Destination	Random
攻击者从域名服务器中采样IPID值及增长速率，然后计算可能的IPID	对每个目的地递增的IPID，攻击者采用算法预测可能的IPID	发送多个分片按概率命中，目前Windows版本支持100个，Linux版本64个分片
>60%	<40%	少量

远程缓存中毒攻击—基于交易ID的攻击

利用端口扫描先得到源端口号，再猜测交易ID

通过侧信道实现UDP端口扫描，将需要猜测的32位数据（$1/2^{32}$）进一步降低至16位（$1/2^{16}$），能够在几分钟内成功实现攻击

Figure 7: DNS Response Used to Overwrite Cache

Field	Value
Question	{nonce}.www.victim.com
Answer	
Authoritative	www.victim.com NS ns.attacker.com
Additional	

Table 3: Production Resolver Attack Results

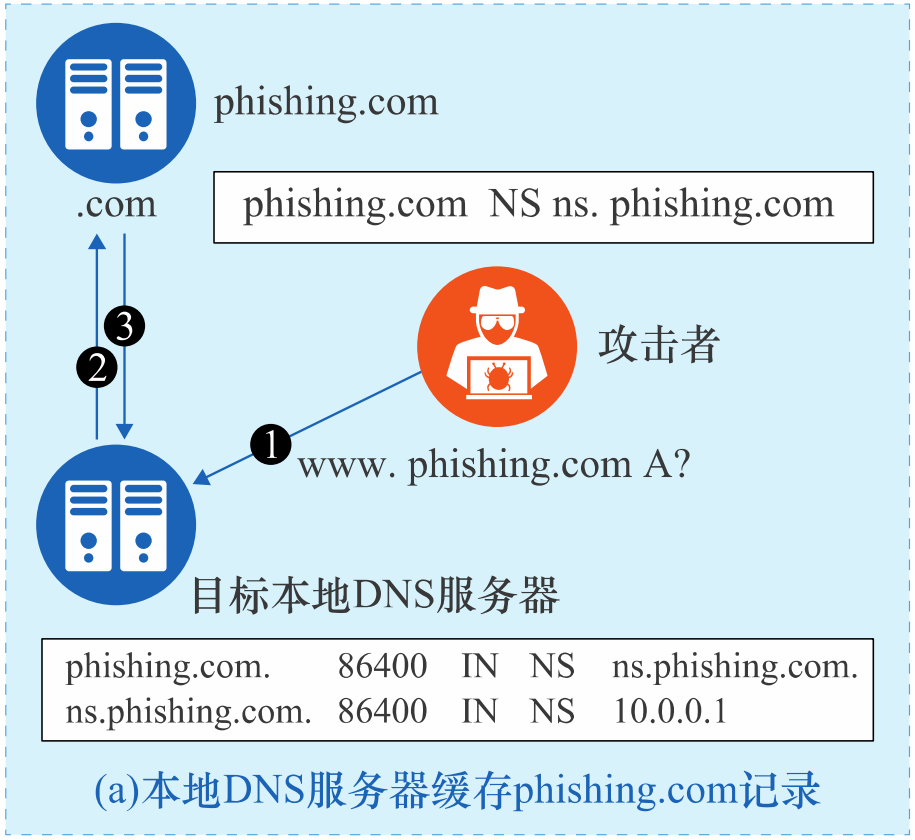
Exp.	RTT range	Probe loss	Name sever mute level	Average time taken	Success rate
Base(D)	0.2-1.2ms	~0%	80%	504s	20/20*
Base(M)	0.2-1.2ms	~0%	80%	410s	20/20*
Mute Lv.	0.2-1.2ms	~0%	75%	1341s	18/20*
Mute Lv.	0.2-1.2ms	~0%	66.7%	2196s	20/20#
Mute Lv.	0.2-1.2ms	~0%	50%	8985s	9/20#
Altered	37-43ms	0.20%	80%	930s	5/5*

*: 1-hour threshold. #: 3-hour threshold. D: Day. M: Midnight

1. Keyu Man, et al. DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels [C] CCS 2020: 1337-1350

远程缓存中毒攻击—幽灵域名

恶意域名被删除后，利用DNS漏洞继续存活



;; ANSWER SECTION

;; AUTHORITY SECTION
phishing.com. 86400 IN NS ns.phishing.com.

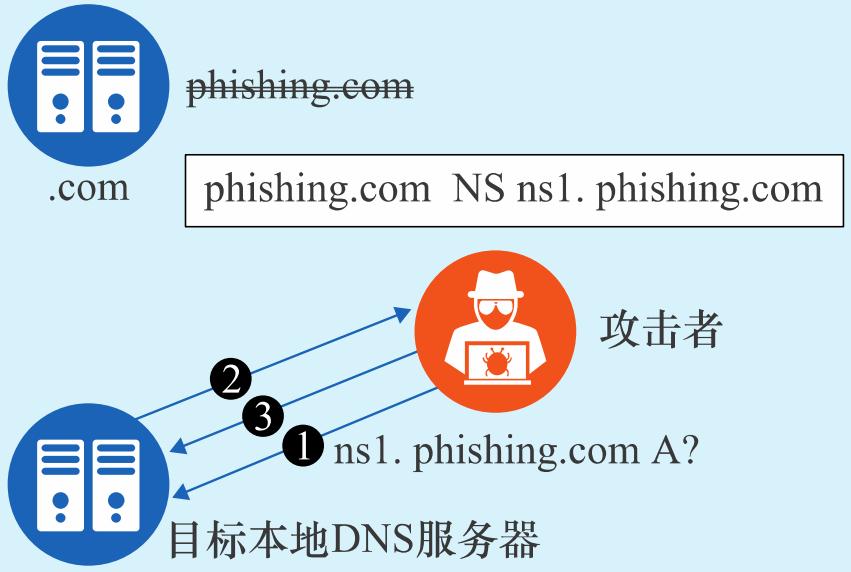
;; ADDITIONAL SECTION
ns.phishing.com. 86400 IN A 10.0.0.1

攻击者向目标DNS服务器查询一个phishing.com子域名，得到phishing.com的授权记录及A记录，TTL为86400，在43200秒后检测其为恶意域名，将其删除

1. Jian Jiang, et.al. Ghost Domain Names: Revoked Yet Still Resolvable[C] NDSS 2012:1-13

远程缓存中毒攻击—幽灵域名

恶意域名被删除后，利用DNS漏洞继续存活



(b)缓存过期前刷新缓存

;; ANSWER SECTION
ns1.phishing.com. 86400 IN A 10.0.0.1

;; AUTHORITY SECTION
phishing.com. 86400 IN NS ns1.phishing.com.

;; ADDITIONAL SECTION
ns1.phishing.com. 86400 IN A 10.0.0.1

攻击者在缓存过期前更换NS记录为ns1.phishing.com并向目标DNS服务器查询，触发TTL更新

来自恶意DNS服务器的污染--在附加部分伪造数据

```
;; QUESTION SECTION:
; www.thucsnet.com.              IN      A

;; ANSWER SECTION:
www.thucsnet.com.       172800  IN      A       192.168.0.100

;; ADDITIONAL SECTION:
xyz.attack.com.         172800 IN       A       10.0.2.2
abc.thucsnet.com.       172800 IN       A       10.0.2.3
```

攻击手段

- 通过嗅探方式得到关于域名（如www. thucsnet.com）的DNS请求，发送伪造报文
- DNS服务器直接发送伪造报文

防御策略

- 本地DNS服务器接受回复字段内容，对于附加字段，如xyz.attack.com不在域内，会选择丢弃这些域外信息
- 域内信息，如abc.thucsnet.com的信息，本地DNS服务器可发出一个新的请求查询对应的真实IP地址

来自恶意DNS服务器的污染--在授权部分伪造数据

```
;; AUTHORITY SECTION:
attack.com.          172800    IN    NS    ns.attack.com.
thucsnet.com.        172800    IN    NS    ns.attack.com.
```

攻击手段

攻击者在授权部分放入NS记录，表明attack.com和thucsnet.com域的权威域名服务器均为ns.attack.com

防御策略

本地DNS服务器检查该记录，如只发送了attack.com域的请求，则不接受thucsnet.com对应的授权记录

授权记录　　LDNS检查　　拒绝伪造记录

来自恶意DNS服务器的污染--反向查找回复部分伪造

一些计算机从IP地址反向查找其域名，从而给予数据包相应权限，如希望example.com的数据包不能进入防火墙，而thucsnet.com的数据则可以

攻击手段	example.com的权威服务器在收到反向查询时回复其对应域名为“thucsnet.com”，欺骗查询者
防御策略	得到反向查询结果时，用这个结果做一次正向查询，并将查询得到的IP地址与原来的IP地址进行比较

拒绝服务攻击

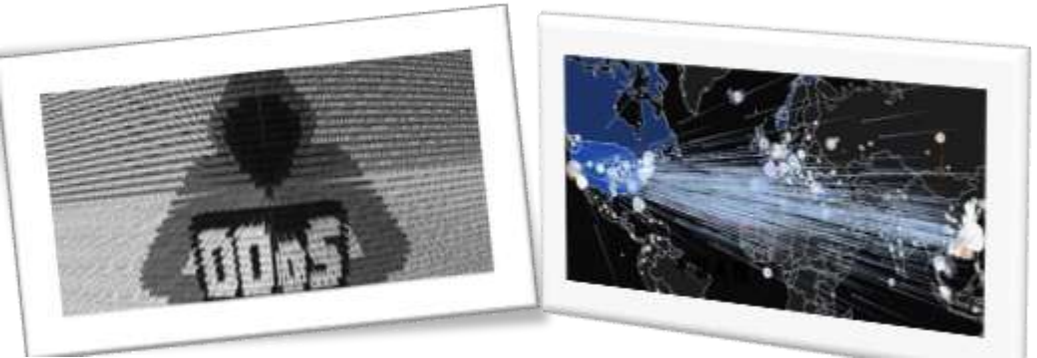
拒绝服务

对root和TLD服务器的拒绝服务攻击

如果攻击者可以成功攻破root区域的服务器，则整个互联网将会崩溃。但是由于root域名服务器基础设施采用分布式部署方式，很难被全部攻破

对特定域名服务器的拒绝服务攻击

2019年亚马逊的云计算部门AWS遭受了持续了大约八小时的DDoS攻击，AWS通过Shield Advanced提供了缓解，但无法完全阻止攻击



攻击根服务器——难以成功

DNS根服务器技术：IP Anycast + BGP

通过BGP在多个不同地点同时广播一个IP地址，BGP路由器从中选择（最近）一个作为路由

Threat Mitigation for the Root Server System

Root Server Operators
August 2019

Introduction

As of 02/28/2021 3:53 a.m., the root server system consists of 1370 instances operated by the 12 independent root server operators.

As of 02/28/2021 3:53 a.m., the root server system consists of 1370 instances operated by the 12 independent root server operators

攻击顶级服务器—巨大威胁

2013年8月25日凌晨，.CN域名凌晨出现大范围解析故障，导致大面积.CN域名无法解析，直到当日凌晨4点左右，CN根域名服务器解析才开始部分恢复

DDoS攻击背后的利益链条

大家可能会有疑问，看似普通的DDoS攻击其背后究竟隐藏着什么？一句话：为了利益。本次事故中攻击者使用的手法(譬如攻击一些“私服”的网站或主机) 并不罕见，且近些年有愈演愈烈的趋势。自国内的互联网事业兴起以来，国内有一些常年进行DDoS攻击的组织或个人，胁迫某些“私服”游戏的运营团队并收取 “保护费”，如果不合作便采取DDoS暴力攻击，使其无法正常运营。而这

中国遭到的DDoS攻击表明 TLD服务器也不安全

2013年08月28日 15:59:49 | 作者：胡杨编译 | 来源：网界网 | 查看本文手机版

本文手机版

摘要：上周末发生的让中国的部分互联网断网的DDoS攻击表明，全球各国域名的互联网实力有很大区别。

标签 顶级域名 CNNIC TLD服务器 DDoS攻击

【CNW.com.cn独家译稿】上周末发生的让中国的部分互联网断网的DDoS[注]攻击表明，全球各国域名的互联网实力有很大区别。

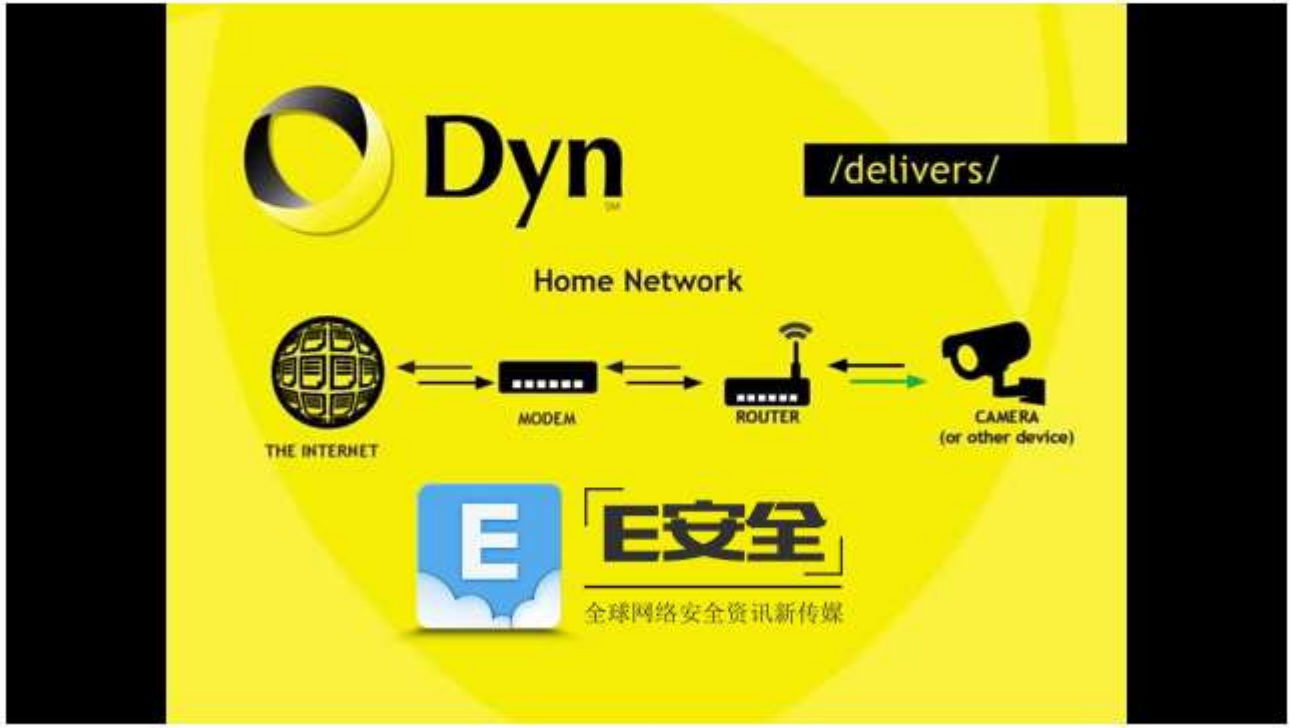
运行中国“.cn”顶级域名的服务器在美国东部时间星期日早上2点遭到了攻击。运行这个顶级域名的中国互联网信息中心(CNNIC)证实了这次攻击，并且向受到影响的用户道歉。

攻击特定域名服务器—影响深远

2016年，攻击者控制大量物联网设备发起DDoS攻击，造成CNN,BBC,PayPal等网站无法访问

美国Dyn公司声明：关于2016年10月21日的DDoS攻击

2016-10-23 19:06



第四节 DNS攻击预防策略

- 基于密码技术
- 基于系统管理
- 新型架构设计

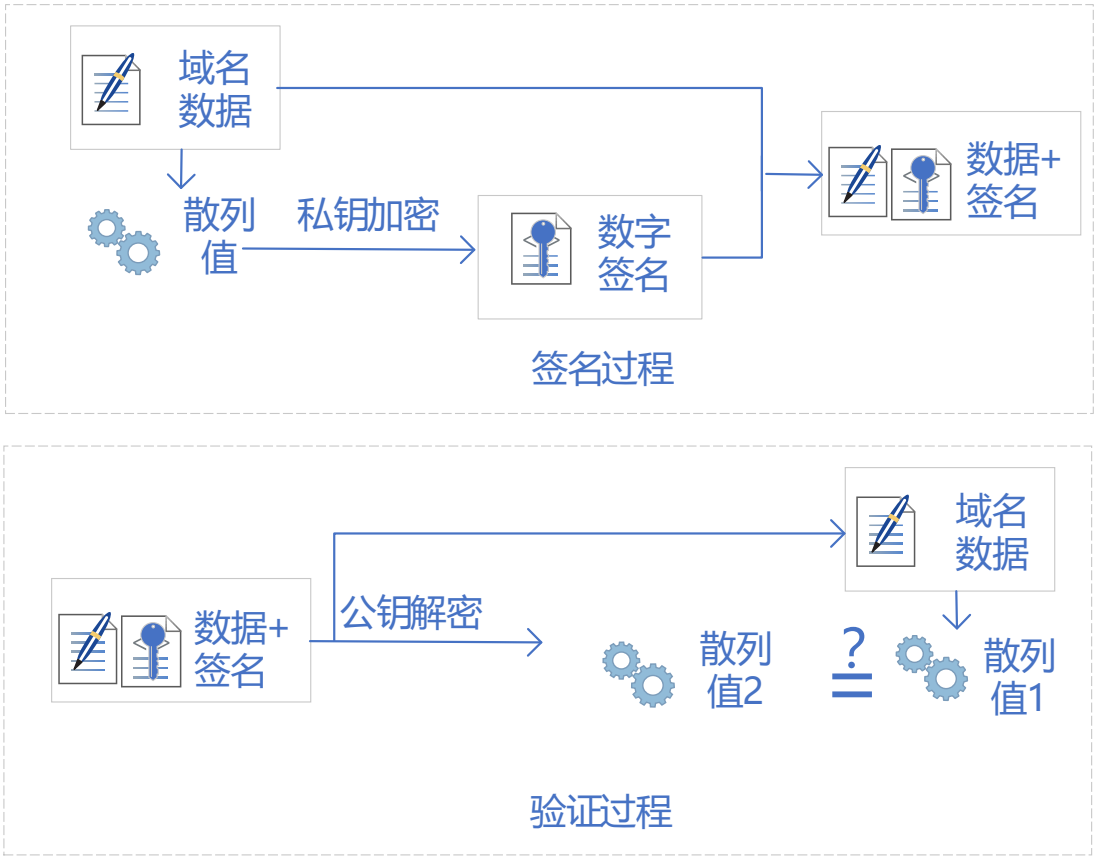
通过非对称加密验证身份--DNSSEC

DNSSEC引入公钥加密/认证体系，通过签名提供端到端数据真实性和完整性保护

部署DNSSEC的权威域名服务器对其区域文件中资源记录用私钥进行数字签名

接收方用公钥验证签名，判定域名解析结果是否在传输过程被篡改

通过非对称加密验证身份--DNSSEC



DNS签名及验证过程示意图

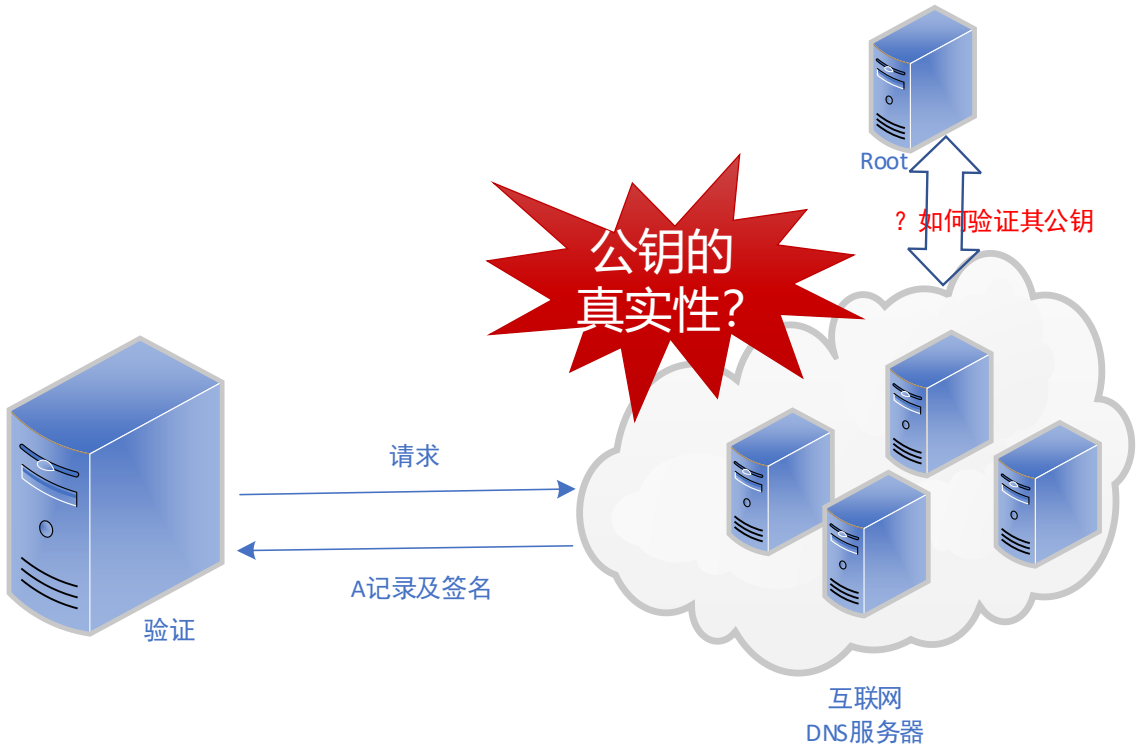
签名过程原理

域名服务器用散列函数计算回复DNS报文内容的散列值，即“内容摘要”，使用私钥对其加密（签名），加密后的信息附加到DNS报文

验证过程原理

本地DNS服务器收到DNSSEC报文，计算报文“内容摘要”，利用公钥解密收到加密“摘要”，对比“摘要”内容

通过非对称加密验证身份—DNSSEC验证公钥



DNSSEC验证需要公钥信任链

- DNSSEC需要一条信任链：支持DNSSEC的本地DNS服务器向支持DNSSEC的权威服务器发起记录请求，得到权威服务器数字签名，签名的正确性（公钥）由上级服务器保证
- 假设DNSSEC 实现了全部署，每个递归服务器只需保留根域名服务器的DNSKEY

通过非对称加密验证身份--DNSSEC

DNSSEC请求及验证过程

RRSIG（Resource Record Signature）

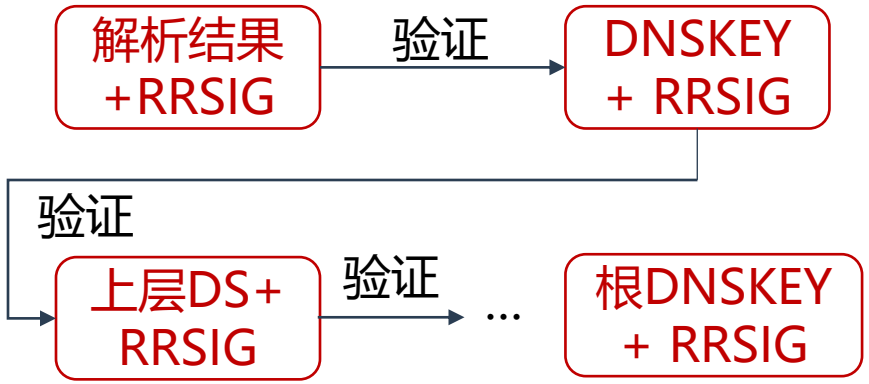
资源记录签名

DNSKEY（DNS Public Key）

公钥记录

DS（Delegation Signer）

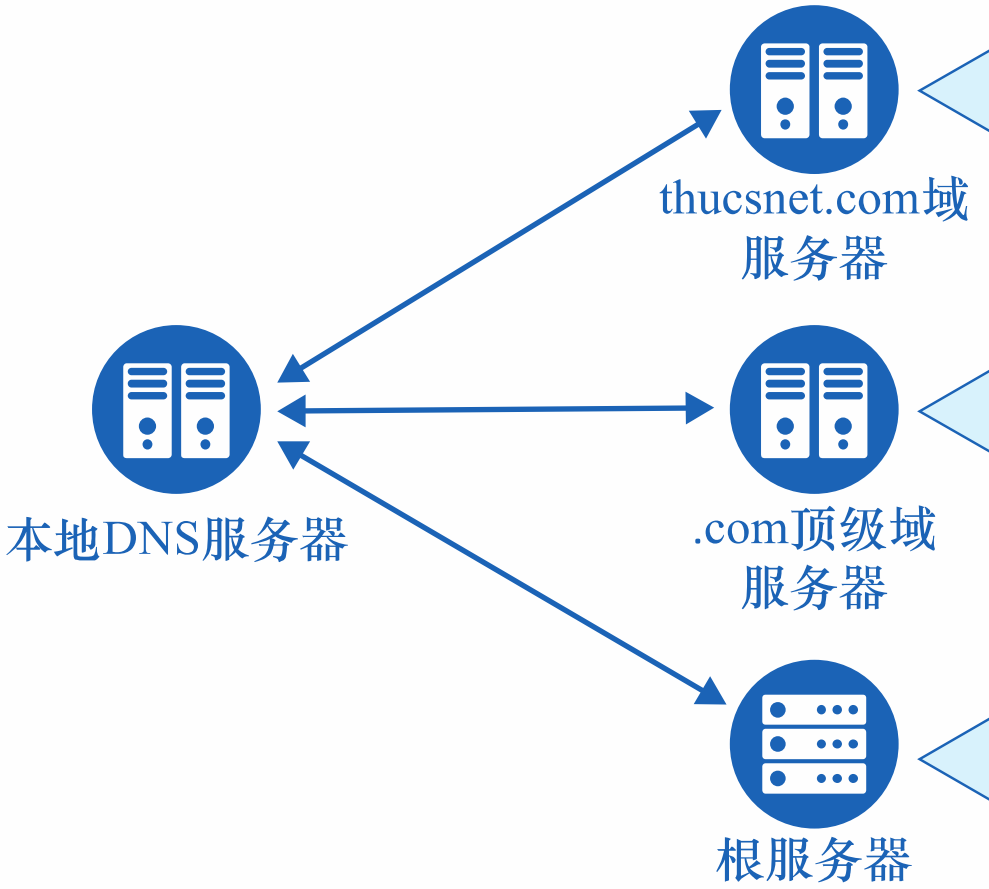
DNSKEY的散列值



解析结果通过签名（RRSIG）验证，RRSIG通过公钥（DNSKEY）验证，公钥通过上级公钥散列（DS）及签名验证

通过非对称加密验证身份--DNSSEC

DNSSEC请求及验证过程

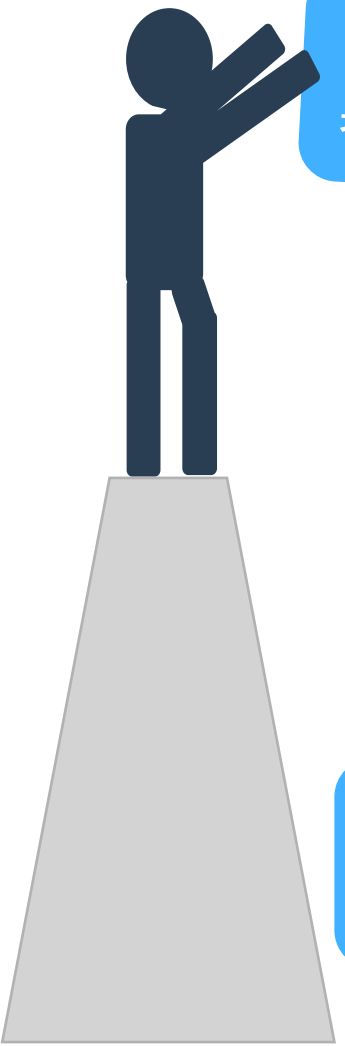


（1）thucsnet.com域服务器返回A记录及RRSIG，通过thucsnet.com域DNSKEY验证RRSIG

（2）向.com域服务器申请thucsnet.com域的DS记录及RRSIG，使用.com域DNSKEY验证RRSIG

（3）向根服务器请求.com域DS记录及RRSIG，通过根服务器DNSKEY验证RRSIG

通过非对称加密验证身份—DNSSEC的不足



可部署性

经济因素制约其部署，大量签名和验证带来严重的负载，影响效率，验证能力取决于递归服务器以及客户端是否对数字签名记录做校验

设计能力

离线存储与频繁使用密钥，私钥的安全无法保证；没有提供机密性和一致性检验，无法抵御重放攻击；只提供单向认证，没有对客户的认证，不能解决缓存中毒； 没有提供DoS防护

引入错误

引入了新的实现和配置错误

DNS-over-TLS（DoT）

可部署性

在DNSSEC被广泛接受之前，需要找到其它解决方案有效阻止DNS攻击造成破坏

信任链

DNSSEC用DNS区域层次结构提供信任链，TLS协议依赖公钥基础设施（PKI），包括证书授权中心（CA）

数据封装

DNS-over-TLS 协议直接使用传输层安全协议对数据执行加密操作，保证了域名协议交互中信息的完整性与机密性

DNS-over-HTTPS (DoH)

数据封装

DNS-over-HTTPS（DoH）协议与现有域名系统不兼容，采用HTTPS 信道传输域名协议数据

DoH协议流量传输路径

客户端 -> DoH服务器 -> DNS服务器 -> DoH服务器 -> 客户端

通用性

部分浏览器或操作系统直接支持DoH

基于系统管理

规范配置过程

1.规范并梳理DNS配置过程中出现的漏洞

降低攻击影响

1.限制用户在短时间内发起大量DNS查询

2.增加端口猜测难度，如用于查询的UDP端口不再是默认的53，而在UDP端口范围内随机选择(排除预留端口)

3.分布式部署、恶意流量过滤等方案

蜜网和DNS防护系统联合实现BotNet检测及恶意流量过滤

基于系统管理

严格检查（如Bailiwick）

- 附加区记录和问题区问题不在同一域管辖，则不会采信，防范恶意权威DNS虚假记录污染缓存
- 比如，请求www.foo.com时，应答包里出现关于www.bar.com的记录，根据bailiwick检查规则，不会接受此A记录

问题区:
www.foo.com A
权威区:
foo.com NS ns1.foo.com
附加区:
ns1.foo.com A 2.3.4.5
www.bar.com A 6.6.6.6

DNSDB提供bailiwick检查接口

新型架构设计-Blockstack

当前设计

去中心化设计

Blockstack 旨在建立一个去中心化的域名系统及公钥基础设施

1.Muneeb Ali, et.al. Blockstack: A Global Naming and Storage System Secured by Blockchains[C] USENIX ATC 2016: 181-194

新型架构设计-Blockstack

数据层	存储用户加密后的数据
对等网络层	存储数据资源的路由信息
虚拟链层	将数据提取出来呈现给上层用户
区块链层	记录用户操作并达成共识

底层采用区块链搭建，通过将域名哈希值以交易形式存储在区块链，以分布式方式避免篡改，提供可靠DNS服务

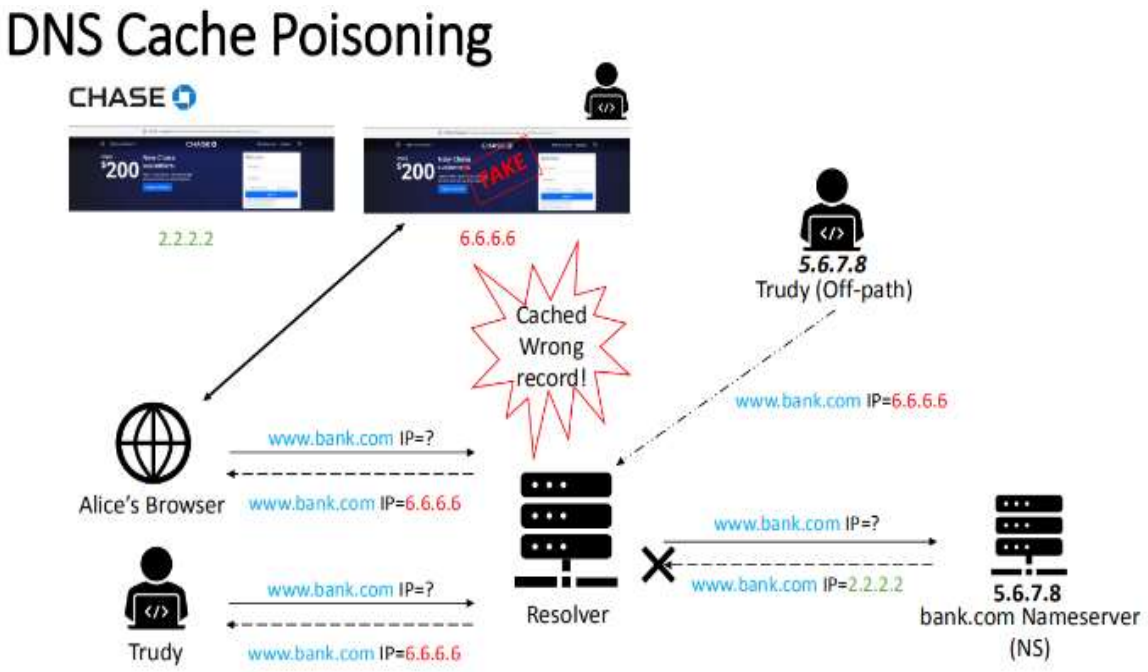
第五节 典型案例分析

- Kaminsky攻击
- 恶意服务器回复伪造攻击
- 拒绝服务攻击

攻击一: Kaminsky攻击

恶意人员发送随机查询请求到DNS服务器，抢在权威应答前伪造应答包发送给服务器，修改授权资源记录

一次出人意料而名留青史的DNS投毒攻击



Kaminsky攻击

成功攻击需要实现三个任务

01 触发DNS服务器（Apollo）发送DNS请求

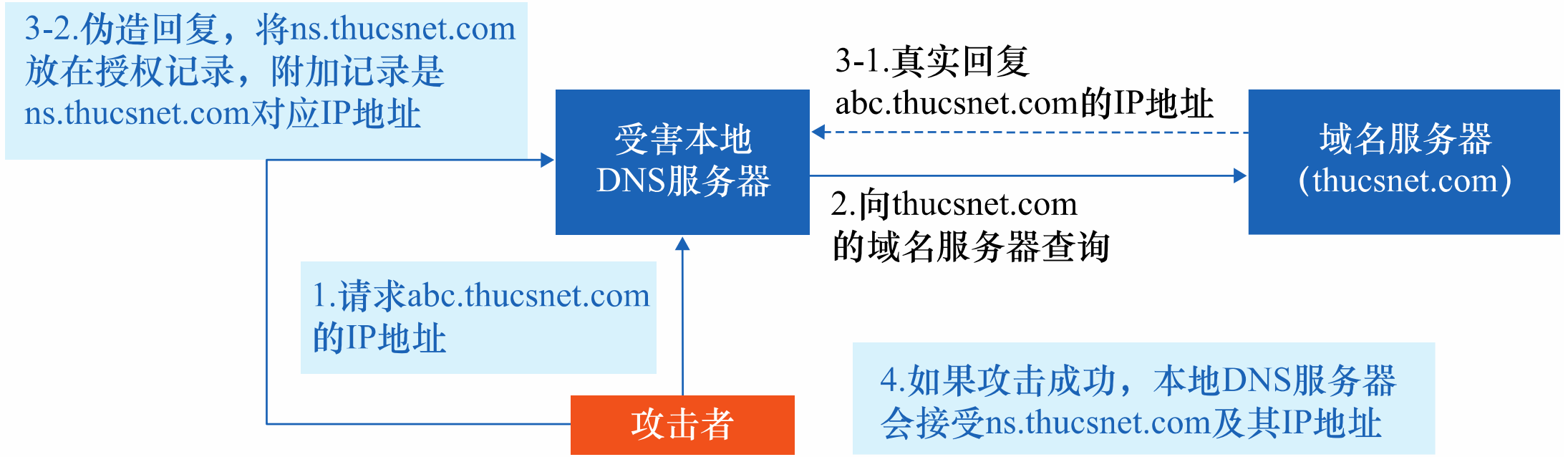
02 发送欺骗回复

03 使缓存失效

前两个任务，攻击者发送DNS请求触发服务器发送请求

第三个任务难度较大，Kaminsky设计了可以持续发起欺骗攻击的方案

Kaminsky攻击



Kaminsky攻击过程

关键点：主动访问不存在的域名，这种域名在本地DNS服务器没有缓存，绕过约束

Kaminsky攻击

成功原因： **在权威区和附加区实施欺骗**

问题区：abc.example.com A

应答区： （空）

权威区： example.com NS www. example.com

附加区：www. example.com A 1.2.3.4

伪造包并没有包含abc.example.com的A记录，但告诉LDNS可以去www.example.com查询，并且www.example.com 对应的IP是1.2.3.4

Kaminsky攻击

根本原因

- 缺乏端节点身份和发布内容验证
- 数据未采用加密传输
- 协议自身检查采信机制不足

防御策略

- 针对协议内容和通信实体设计可靠的验证方式
- 通过加密等策略确保对用户身份和发布内容进行鉴别，实现有效防御

攻击二:恶意服务器回复伪造攻击

攻击策略

- 攻击者可依赖目标权威服务器或 DNS 软件漏洞，控制特定权威域名服务器，篡改区域文件中的授权数据，形成恶意服务器回复伪造攻击，成功攻破权威域名服务器的难度较大，实际中可行性并不高

- 利用域名系统冗余的架构设计使很多域名之间存在错综复杂的解析依赖，通过控制其中一环，逐步实现劫持特定域名权威服务器

恶意服务器回复伪造攻击

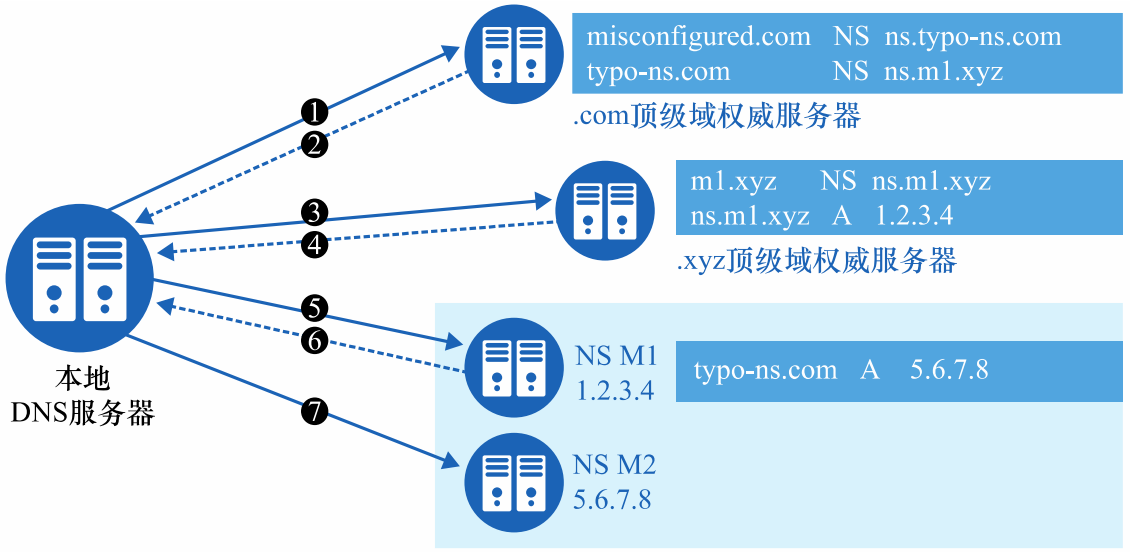
利用误植域名（typosquatting）攻击

- 当误植域名攻击用于域名服务器时，即某个授权记录错误且被攻击者发现，攻击者申请该错误的授权记录对应域名，将查询引导至恶意权威服务器发起伪造回复
- 攻击前提是域名服务器存在依赖关系：如xy.com的授权记录为ns1.xy.com称为符合bailiwick规则，不存在依赖关系；反之称为存在依赖关系，排名前10,000的服务器中有36.4%存在这种现象

1.Vissers T, et al. The wolf of name street: Hijacking domains through their nameservers [C] CCS 2017: 957-970.

恶意服务器回复伪造攻击

攻击者注册typo-ns.com，用户访问misconfigured.com，NS记录有ns.typo-ns.com



①递归解析器解析misconfigured.com

②.com域名服务器需向ns.m1.xyz查询

③向另一个TLD服务器发起域名解析请求

④得到1.2.3.4

⑤解析器需要对NS M1进行查询以获取ns.typo-ns.com的IP地址

⑥得到5.6.7.8

⑦获得misconfigured.com域名服务器（NS M2）的IP地址，并随后向其发出请求

NS M2可以回复伪造信息，实现伪造攻击

恶意服务器回复伪造攻击

根本原因

- 域名系统存在复杂的解析依赖
- DNS管理方面缺乏验证配置内容能力，存在错误输入的记录

防御策略

- 规范域名应用管理，减少使用环节引入的安全威胁
- 对配置内容建立审核机制，确保每一条域名记录的正确性，从根上防止接受恶意信息

拒绝服务攻击

攻击策略

- 向被攻击的服务器发送大量域名解析请求，给服务器带来了很大负载，超过一定数量造成 DNS服务器反应缓慢甚至停止服务

攻击效果

- 攻击者可以通过拒绝服务攻击使得整个国家的因特网受到严重威胁

拒绝服务攻击

亚马逊宣布AWS Shield阻止分布式拒绝服务攻击

发布：昆明沃德软件　发布时间：2016-12-12　浏览次数：451　更多

在最近的 2016 年 re:Invent 大会上，亚马逊宣布了一项叫做 AWS Shield 的新服务，为客户提供针对分布式拒绝服务（DDoS）攻击的防护。

2016年，亚马逊推出AWS Shield，提供DDoS防护

2019年10月23日上午4时，“Route 53和DNS提供商出现间歇性DNS解析错误，正展开调查”

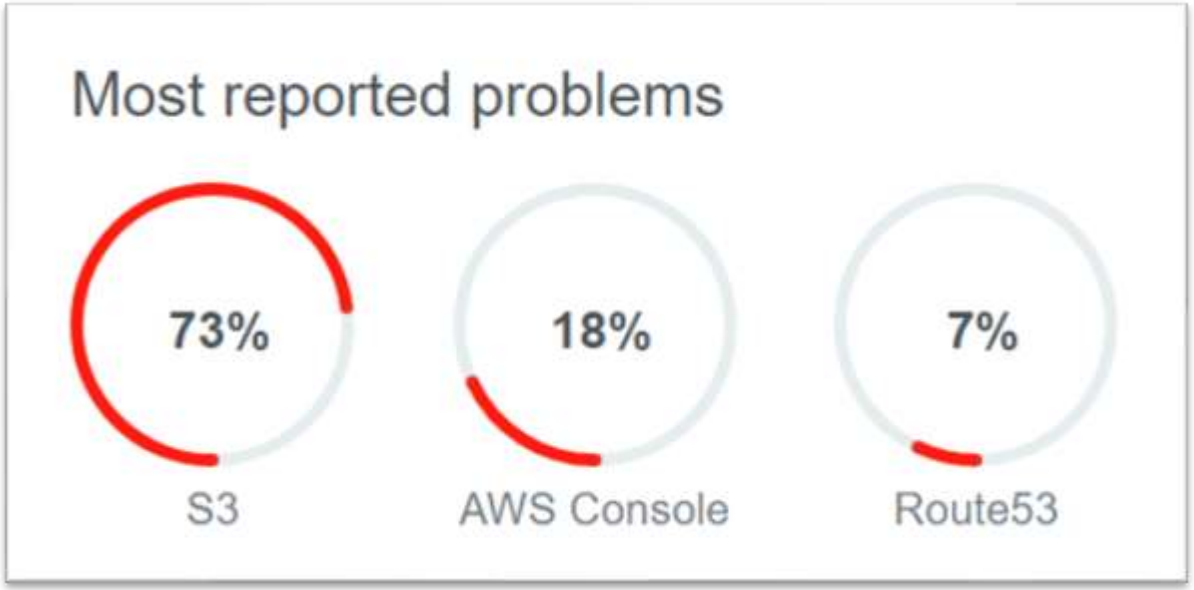
中午12时，“AWS发布公告表示问题已解决”

拒绝服务攻击

Perhaps, the most ironic part of this entire encounter was that Amazon was unable to stop the attack despite it offering its very own DDoS mitigation service named Shield Advanced with multiple plans. However, the company did attribute the reduction in the impact of the incident to it stating that:

"Our DDoS mitigations are absorbing the vast majority of this traffic, but these mitigations are also flagging some legitimate customer queries at this time."

亚马逊使用AWS Shield吸收了大部分攻击流量，但同时也阻止了一些合法用户的访问



AWS服务报障比例

攻击特征分析：

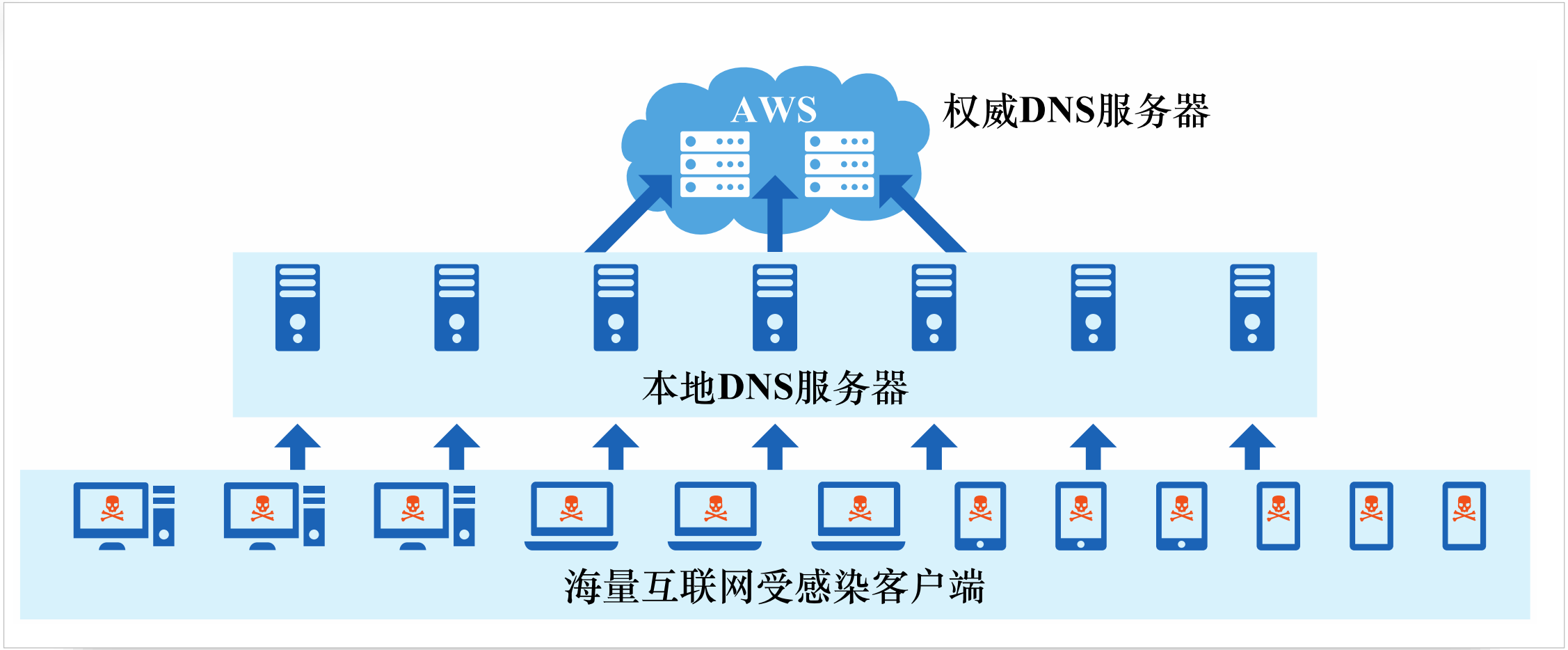
1.攻击域名特点：大量针对s3.amazonaws.com下8位随机前缀域名查询（如gv73dzz0. s3.amazonaws.com）

2.攻击源数量庞大

3.攻击通过递归侧

4.攻击持续时间长

拒绝服务攻击



攻击者通过控制大量用户发起DDoS攻击，受害域名服务器可以通过分布式部署的方式，提升自己的处理能力

拒绝服务攻击

根本原因

- 大量恶意请求到达被攻击服务器并被攻击服务器接受

防御策略

- 安全的网络体系架构，如采用真实网络地址保障每一台接入网络计算机的安全性，建立快速DDoS溯源机制
- 专用DNS请求过滤系统，针对DNS数据包进行恶意流量进行过滤；分布式部署降低攻击对性能的影响

第六节 总结和展望

域名系统安全问题

你觉得 域名系统存在哪些安全问题 ？

01 攻击面

- 用户机
- 本地DNS服务器
- 互联网DNS服务器

02 常见攻击

- DNS缓存中毒
- 恶意DNS服务器回复伪造
- 拒绝服务

03 攻击成功原因

- 缺乏端验证及加密传输
- 系统配置管理不足
- 缺乏过滤及分布式部署能力

04 典型案例

- Kaminsky攻击
- 恶意服务器回复伪造
- 2019年AWS拒绝服务攻击

总结

熟悉了DNS的演进历史、域名结构、区域形式、使用原理及基本解析过程，思考了DNS潜在的安全问题，学习了常见的DNS攻击技术和防御手段，探究了具体攻击案例

第一节
DNS概述

- DNS的演进
- DNS域名结构及区域组织形式

第二节
DNS使用及解析过程

- DNS使用
- DNS解析过程

熟悉DNS运行原理

思考DNS安全问题

DNS在设计之初缺乏安全考虑

第三节
DNS攻击

- 缓存中毒攻击
- 恶意DNS服务器回复伪造
- 拒绝服务攻击

第四节
DNS攻击预防策略

- 基于密码技术
- 基于系统管理
- 新型架构设计

掌握DNS攻击技术

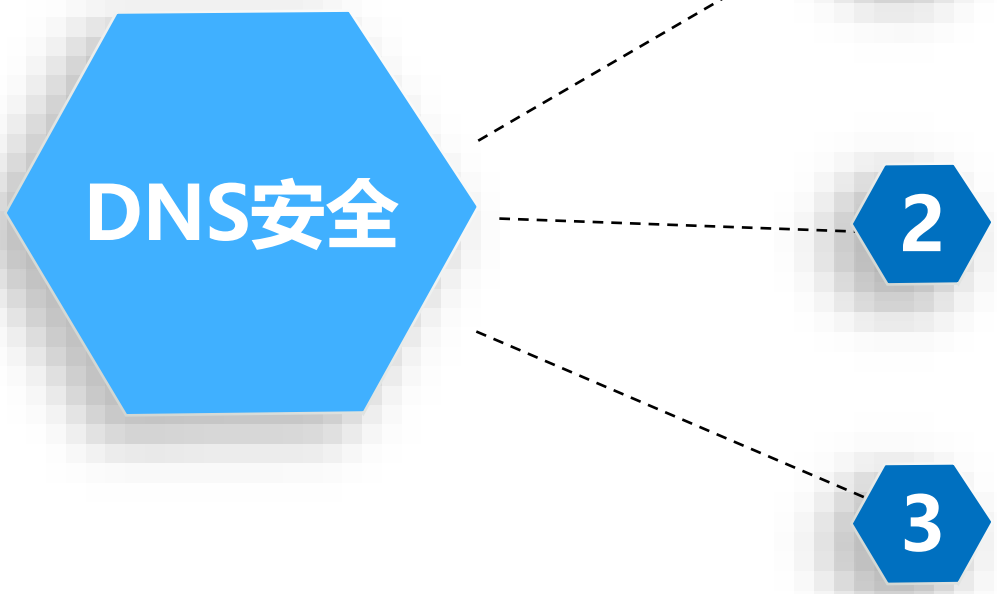
了解DNS防御策略

多种攻击技术需要特定的防御策略进行防护

第五节
典型案例分析

- Kaminsky攻击
- 恶意服务器回复伪造攻击
- 拒绝服务攻击

展望



现有DNS协议修改完善

基于签名、加密技术提升协议机制安全性

针对新型安全威胁寻找解决方案

研究国际化域名，以及DNS转发器等引入的新威胁

新型架构设计

从根源上解决DNS安全问题，部署较为困难，但部分思想有助于提升DNS系统的安全能力