



UNIVERSITÀ DI PISA

## **GESTIONE DI RETI**

### **DETECTION DOMAIN GENERATION ALGORITHM**

Studente

**Francesco Piccinotti**

Professore

**Luca Deri**

## 1 - INTRODUZIONE

I dispositivi connessi in rete possono essere individuati dai protocolli TCP/IP mediante il loro indirizzo IP. Gli utenti generici però preferiscono usare nomi piuttosto che indirizzi numerici. Per questo è necessario un sistema che associ un indirizzo IP ad ogni nome e viceversa.

Vista la dimensione di Internet, un sistema centralizzato non potrebbe gestire le associazioni IP-nome necessarie. Inoltre, se tale sistema centralizzato si dovesse guastare, collasserebbe l'intera rete. La soluzione attualmente in uso consiste nel suddividere questa enorme massa di informazioni e distribuire le varie parti ottenute su calcolatori sparsi per il mondo.

L'host che ha bisogno di associare un indirizzo a un nome, o viceversa, contatta il calcolatore più vicino e gli invia una richiesta opportuna.

Un Domain Name System (DNS) è un database distribuito memorizzato su molti nodi che è implementato mediante una gerarchia di name server e permette agli end host di effettuare query sul database distribuito mediante UDP.

I Domain Generation Algorithm (DGA, algoritmo di generazioni di dominio) sono algoritmi che generano periodicamente nomi di dominio.

Per ridurre la probabilità di essere scoperti, i malware utilizzano un DGA per contattare il proprio server di comando e controllo.

## 2 - SCOPO E STRUTTURA DEL PROGETTO

Lo scopo del progetto è quello di trovare DGA analizzando il traffico DNS di una rete locale. Per far ciò, ogni nome di dominio può essere considerato come una sequenza di bigrammi. Per ogni bigramma si deve considerare quale bigramma lo può seguire. L'obiettivo è dunque di creare un grafo unidirezionale dove i nodi rappresentano i bigrammi e gli archi uscenti da un nodo rappresentano l'insieme delle possibili combinazioni di concatenazione a bigrammi successivi.

L'algoritmo si sviluppa in due fasi.

1. Creazione del grafo. La prima fase consiste nel creare il grafo e di inserire al suo interno bigrammi di nomi di dominio sicuri. Finita la parte di creazione e inserimento iniziale di bigrammi, inizia la seconda fase;
2. Valutazione dei nomi di dominio e apprendimento. La seconda fase consiste nell'analisi dei nomi di dominio. Suddivido il nome di dominio da analizzare in bigrammi e vado a verificare se, all'interno del grafo, ho una corrispondenza. Se il nome di dominio non è verificato, il programma lo segnala. In caso di anomalia si presentano i vari scenari:
  - a. Il nome di dominio è generato da un DGA;
  - b. Il nome di dominio è sicuro ma il programma non lo conosce. Se il nome è sicuro viene inserito all'interno del grafo, dopo una conferma da parte dell'utente.

### 3 – COMPOSIZIONE ED ESECUZIONE DEL PROGETTO

Il progetto è composto da due librerie (creazioneBigrammi e grafoBigrammi) e dal file dga.c che ha due compiti: inizializzare il grafo e catturare i pacchetti.

La libreria creazioneBigrammi è utilizzata per suddividere una stringa in un array di bigrammi.

La libreria grafoBigrammi contiene la struttura dati che implementa il grafo, oltre ai metodi necessari per l'inserimento e la ricerca dei bigrammi all'interno del grafo.

Il programma, per poter funzionare, ha bisogno della libreria libcap.

Il programma può essere compilato con il comando make.

Per eseguire il programma utilizzare: `sudo ./dga`

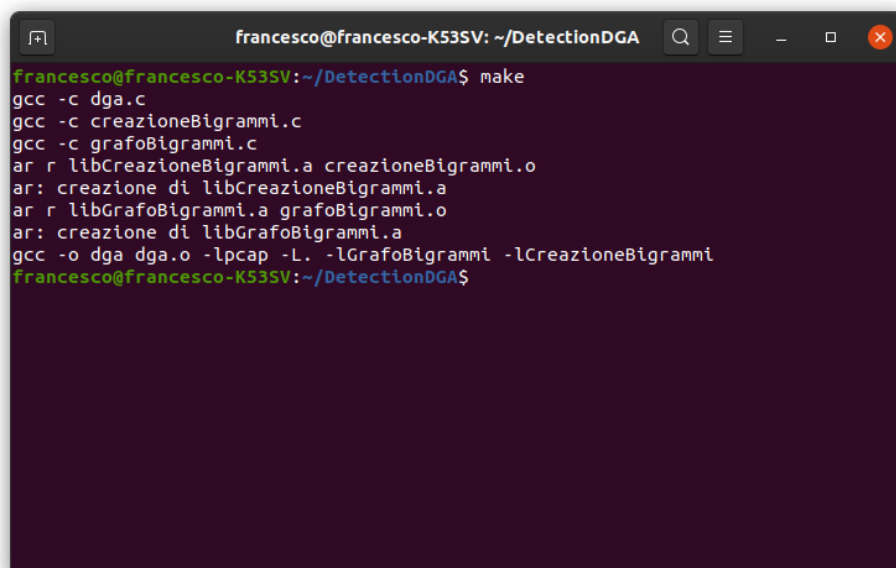
Per conoscere i parametri disponibili utilizzare: `sudo ./dga -h`

I parametri disponibili sono:

- p limite probabilità;
- d nome del file contenente i nomi di domini per la creazione iniziale del grafo;
- f nome del file pcap per l'analisi offline;
- n numero pacchetti da sniffare;
- i nome interfaccia;
- m modalità promiscua;
- b nome file black list;
- h help.

### 4 – ESEMPI PRATICI DI FUNZIONAMENTO

Utilizzare il metodo make per la creazione dell'eseguibile. Una volta che il comando make ha terminato la sua esecuzione genera il file eseguibile `./dga`.

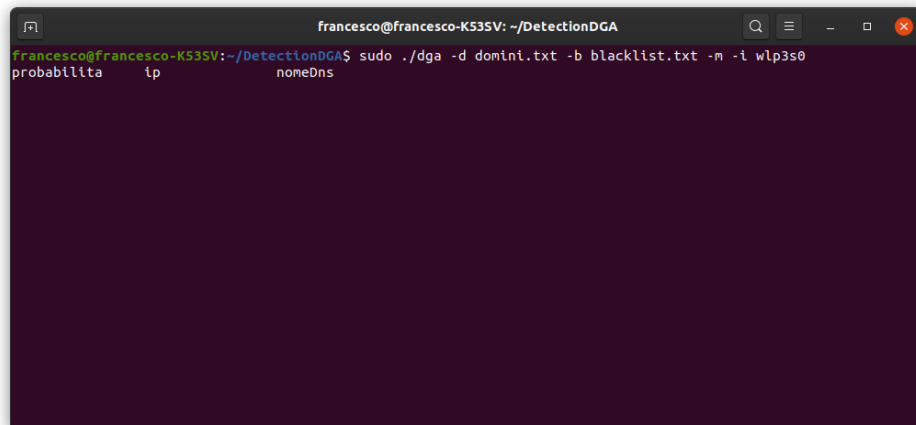


```
francesco@francesco-K53SV: ~/DetectionDGA
francesco@francesco-K53SV:~/DetectionDGA$ make
gcc -c dga.c
gcc -c creazioneBigrammi.c
gcc -c grafoBigrammi.c
ar r libCreazioneBigrammi.a creazioneBigrammi.o
ar: creazione di libCreazioneBigrammi.a
ar r libGrafoBigrammi.a grafoBigrammi.o
ar: creazione di libGrafoBigrammi.a
gcc -o dga dga.o -lpcap -L. -lGrafoBigrammi -lCreazioneBigrammi
francesco@francesco-K53SV:~/DetectionDGA$
```

Il programma può adesso essere avviato in due modalità. La prima modalità (online) permette di analizzare il traffico in tempo reale, invece la seconda modalità (offline) analizza un file in formato .cap o .pcap.

In entrambe le modalità è possibile indicare il numero di pacchetti che il programma può catturare o leggere da file con il comando `-n`.

Una volta avviato il programma (sia in modalità online che offline) appariranno tre colonne:



```
francesco@francesco-K53SV: ~/DetectionDGA
francesco@francesco-K53SV:~/DetectionDGA$ sudo ./dga -d domini.txt -b blacklist.txt -n -i wlp3s0
probabilità   ip           nomeDns
```

La colonna “probabilità” indica la percentuale di riconoscimento del nome di dominio. La colonna “ip” indica l’indirizzo dell’host che ha effettuato la richiesta DNS e infine la colonna “nomeDns” rappresenta il nome di dominio cercato.

## 4.1 Modalità online

Per avviare la modalità online di default è necessario eseguire il comando:

```
sudo ./dga -d domini.txt -b blacklist.txt -i wlp3s0
```

Una volta avviato il programma, esso inizia a catturare i pacchetti della rete.

Il comando `-d` serve per passare al programma una lista di nomi di dominio che si ritengono sicuri. Questo permette di creare le informazioni di base per il nostro grafo.

Il comando `-b` passa al programma una lista di nomi di dominio considerati non sicuri.

Infine, il comando `-i` permette di passare al programma l’interfaccia di rete da dove catturare i pacchetti.

Avviando il programma in modalità online di default, la probabilità limite che differenzia un nome di dominio malevolo da uno non malevolo è di default al 50%: sotto al 50% saranno malevoli e al di sopra sicuri. Questo limite è modificabile passando al programma il nuovo valore con il comando `-p`. Naturalmente un valore basso ci saranno molti falsi positivi, al contrario, un valore alto ci saranno molti falsi negativi.

Per attivare la modalità promiscua (catturare il traffico proveniente da altri host della rete) si utilizza il comando `-m`:

```
sudo ./dga -d domini.txt -b blacklist.txt -i wlp3s0 -m
```

## 4.2 Modalità offline

Come nella modalità online, bisogna specificare il file che contiene i nomi di dominio che si ritengono sicuri e il file che contiene i nomi di dominio considerati non sicuri (comandi `-d` e `-b` rispettivamente).

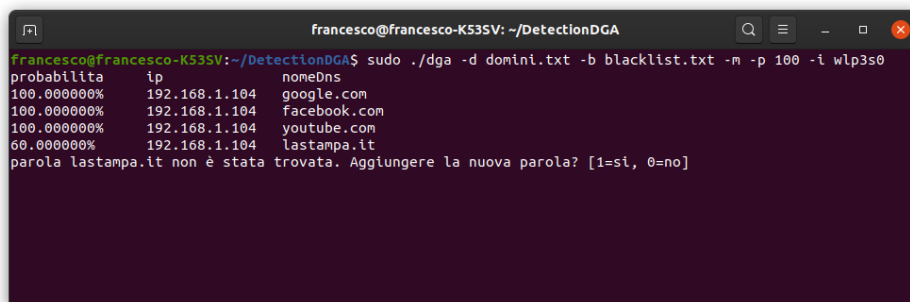
Nella modalità offline si utilizza il comando `-f` per passare al programma un file in formato `.pcap` o `.cap`.

```
./dga -d domini.txt -b blacklist.txt -f dns.cap
```

## 4.3 Esempi di utilizzo in casi limite

### Probabilità limite alta

Facciamo un esempio sull'utilizzo del programma in modalità online cambiando il valore della probabilità limite passata per parametro. Nel primo esempio è impostata la probabilità limite al 100%.



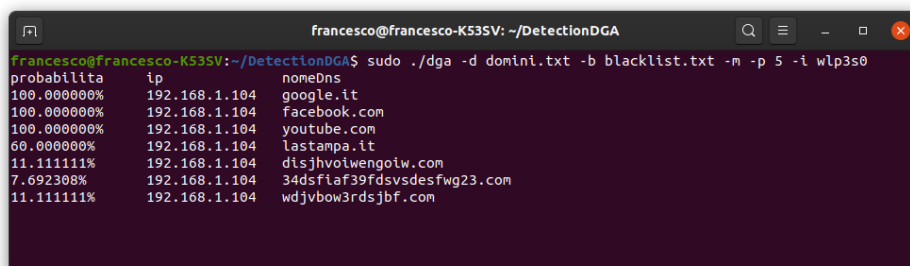
```
francesco@francesco-K53SV: ~/DetectionDGA
francesco@francesco-K53SV:~/DetectionDGA$ sudo ./dga -d domini.txt -b blacklist.txt -m -p 100 -i wlp3s0
probabilita ip nomeDns
100.000000% 192.168.1.104 google.com
100.000000% 192.168.1.104 facebook.com
100.000000% 192.168.1.104 youtube.com
60.000000% 192.168.1.104 lastampa.it
parola lastampa.it non è stata trovata. Aggiungere la nuova parola? [1=si, 0=no]
```

In questo caso, dove è stata impostata una probabilità limite alta (100%), verranno considerati corretti soltanto i nomi di dominio con una correttezza del 100%, qualsiasi nome al disotto verrà segnalato.

Come possiamo vedere dall'immagine sopra, il sito "lastampa.it" è stato riconosciuto al 60% segnalandolo come probabile DGA.

### Probabilità limite bassa

Adesso consideriamo un esempio opposto cioè con un limite di probabilità molto basso, del 5%.



```
francesco@francesco-K53SV: ~/DetectionDGA
francesco@francesco-K53SV:~/DetectionDGA$ sudo ./dga -d domini.txt -b blacklist.txt -m -p 5 -i wlp3s0
probabilita ip nomeDns
100.000000% 192.168.1.104 google.it
100.000000% 192.168.1.104 facebook.com
100.000000% 192.168.1.104 youtube.com
60.000000% 192.168.1.104 lastampa.it
11.111111% 192.168.1.104 disjhvoiwengoiw.com
7.692308% 192.168.1.104 34dsflaf39fdsvsdesfwg23.com
11.111111% 192.168.1.104 wdjvbow3rdsjbf.com
```

In questo caso invece non viene fatta nessuna segnalazione neanche su possibili DGA.

## 5 – CONCLUSIONI

Il programma creato è in grado di dare un indice di probabilità espresso in percentuali ai nomi di dominio che cattura dalla rete.

La probabilità limite che differenzia un nome di dominio malevolo da uno non malevolo è di default al 50%: al di sotto di questa soglia, il nome di dominio viene segnalato.

Questo valore limite può essere anche modificato dall'utente mediante il comando -p.

La probabile presenza di un malware viene segnalata all'utente, che può accrescere il grafo del programma accettando nomi di dominio dubbi (con bassa probabilità).

Più il programma viene usato, più diventa efficiente.