UNIVERSITÀ DI PISA

GESTIONE DI RETI

DETECTION DOMAIN GENEARATION 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 – 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.