

Corso di Laurea Triennale in Ingegneria e Scienze Informatiche

DGA Domain Generation Algorithm

Tesi di laurea in:
PROGRAMMAZIONE A OGGETTI

Relatore

Prof. Mirko Viroli

Candidato

Simone Collorà

Correlatori

Dott. CoSupervisor 1

Dott. CoSupervisor 2

Abstract

Max 2000 characters, strict.

Optional. Max a few lines.

Contents

Abstract	iii
1 Introduction	1
2 Uso di Machine Learning e AI per rilevare i DGA	3
2.1 Botnets e C&C	3
2.1.1 Botnets	3
2.1.2 Command and Control	4
2.2 Some cool topic	7
3 Contribution	9
3.1 Fancy formulas here	9
	11
Bibliography	11

CONTENTS

List of Figures

2.1	Ciclo di vita di un botnet (da [6])	4
2.2	Esempio di server C&C. (a) centralizzato, (b) decentralizzato (da [5])	5
2.3	esempio del funzionamento di un DGA	6

LIST OF FIGURES

List of Listings

listings/HelloWorld.java	9
------------------------------------	---

LIST OF LISTINGS

Chapter 1

Introduction

Write your intro here.

La sicurezza informatica è un argomento di crescente importanza nel mondo moderno. Con il passare del tempo, i sistemi di protezione sono diventati sempre più sofisticati e potenti ma, allo stesso tempo, anche gli hackers hanno sviluppato tecniche sempre più avanzate per eludere i sistemi di protezione. Tra queste vi è sicuramente l'uso di Botnets dei Command and Control (C&C) servers. I C&C sono dei server che manipolano i computer infetti da malwares, i Botnets, permettendo all'attaccante di eseguire codice malevolo da remoto. Il malware, però, deve conoscere un indirizzo IP o un dominio per contattare il server. L'attaccante potrebbe inserire in modo brutto l'indirizzo IP del server nel codice del malware, ma questo metodo è facilmente rilevabile e bloccabile. Gli hackers, quindi, preferiscono utilizzare dei domini generati in modo pseudo casuale per nascondere i loro server chiamati Domain Generation Algorithm (DGA) servers.

Structure of the Thesis

Chapter 2

Uso di Machine Learning e AI per rilevare i DGA

2.1 Botnets e C&C

I suggest referencing stuff as follows: fig. 2.3 or Figure 2.3

2.1.1 Botnets

I Botnets sono reti di computer infetti da malware, chiamati bot, che possono essere controllati da un attaccante, il botmaster. La vita di un botnet di solito sono questi:

1. **Infezione e propagazione:** Questo è il primo passaggio. L'attaccante cerca di infettare un computer tramite vari metodi come email con link malevoli o Peer to Peer (P2P) sharing. Una volta infettato un dispositivo, il malware cerca di infettare altri dispositivi nella rete.
2. **Rallying:** i bots cercano di contattare per la prima volta il server C&C per far capire all'attaccante che l'attacco è andato a buon fine.
3. **Commands and Reports:** il malware esegue le istruzioni ricevute dal server C&C e invia i risultati al botmaster. I bots ascoltano i comandi dal server C&C o si connettono ad esso periodicamente. Appena ricevono un

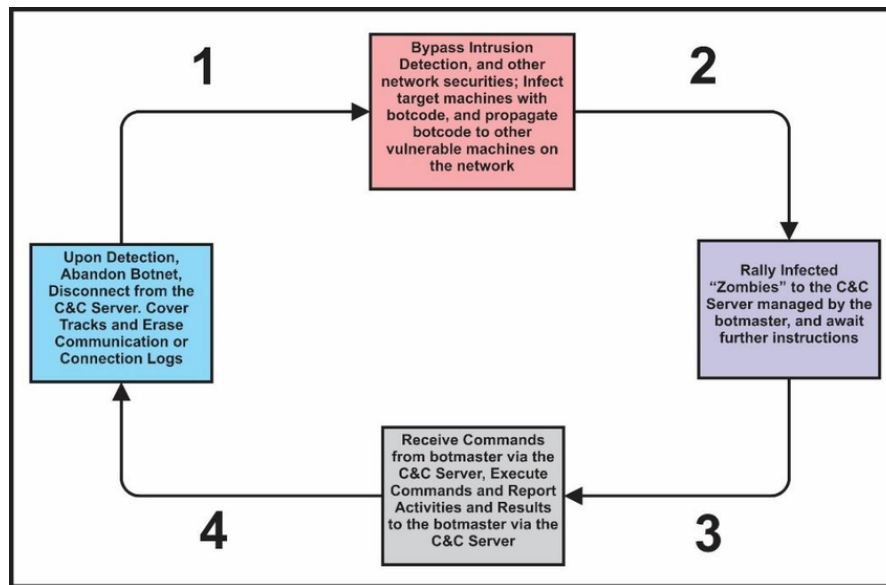


Figure 2.1: Ciclo di vita di un botnet (da [6])

comando lo eseguono, inviano i risultati al botmaster e aspettano un nuovo comando.

4. **Abbandono:** Quando un bot non è più utile o utilizzabile, il botmaster può decidere di abbandonarlo. Il botnet, invece, sarà completamente distrutto quando tutti i bot saranno abbandonati o bloccati dalla vittima o quando il C&C server verrà bloccato

2.1.2 Command and Control

Il meccanismo del C&C crea un canale di comunicazione tra il botmaster e i bot. Questo è essenziale per il funzionamento del botnet. Ci sono tre tipi di server C&C:

- **Centralizzati:** In questo tipo di server, il botmaster controlla tutti i bot tramite un server centrale. Questo è il metodo più semplice e veloce per controllare i bot ma è anche il più vulnerabile. Se il server centrale viene bloccato, tutti i bot non possono più ricevere comandi. Questo a sua volta è diviso in due categorie:

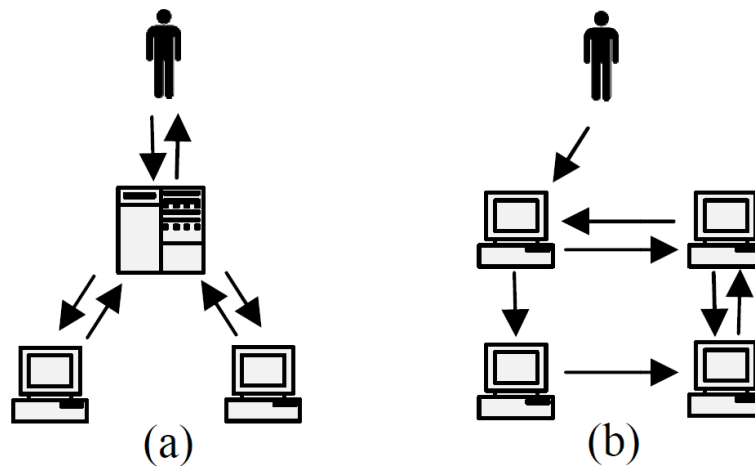


Figure 2.2: Esempio di server C&C. (a) centralizzato, (b) decentralizzato (da [5])

- **IRC** : Internet Relay Chat (IRC) è un sistema di chat usato per comunicare tra i bot e il botmaster in tempo reale. Questo era più usato nella prima generazione di botnet. I bot si connettono al server IRC e aspettano i comandi dal botmaster. I bot seguono un approccio PUSH ovvero quando un bot si connette ad un determinato canale, esso rimane connesso.
- **HTTP**: Il più usato. Con questa tecnica, i bot usano un URL o IP per contattare il server C&C. Qui invece i bot seguono un approccio PULL. I bot si connettono al server C&C periodicamente e controllano se ci sono nuovi comandi. Questo processo va ad intervalli regolari definiti dal botmaster.
- **Decentralizzati**: In questo tipo di server, i botmaster controllano i bot tramite più server. Questo rende più difficile bloccare il server ma è anche più complesso da gestire.
- **Ibridi**:

I DGA sono algoritmi che generano migliaia di domini in modo pseudo casuale. Prima viene scelto un seed, di solito la data odierna o anche le previsioni meteo [1] e, tramite un algoritmo di hashing, vengono generati i domini. Questi domini vengono poi utilizzati per contattare i server C&C. Non tutti i domini generati però

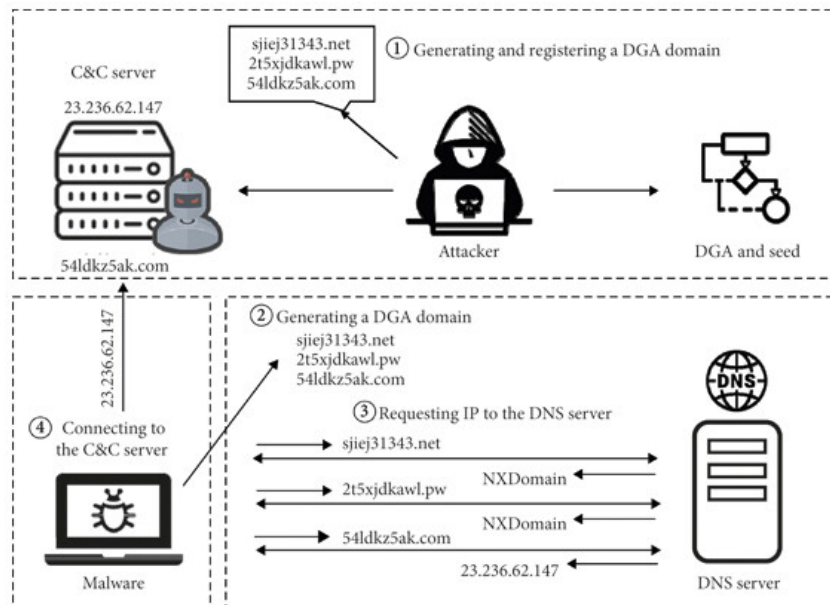


Figure 2.3: esempio del funzionamento di un DGA

sono registrati. Il computer infetto, tramite i DNS locali, cercherà di tradurre un dominio in un indirizzo IP. Se non riesce a contattarlo con un determinato dominio, proverà con il successivo finché non troverà un dominio valido che permetterà al malware di comunicare con il server C&C [2]. In questo modo, diventa più difficile per i sistemi di protezione rilevare e bloccare i loro attacchi. Si potrebbe pensare di bloccare direttamente i domini tramite una blacklist ma questo metodo risulta inefficace poiché vengono generati migliaia di domini continuamente. Si pensi che Conficker C, un famoso malware che utilizza DGA, è in grado di generare fino a 50.000 domini pseudo casuali al giorno [3].

Un altro modo per contrastare ciò potrebbe essere quello di fare reverse engineering del DGA per capire quale seed viene utilizzato per generare i domini. Questo però risulta lento e dispendioso e possibilmente inefficace [4].

Per contrastare i DGA, sono stati sviluppati vari metodi di machine learning in grado di rilevare i domini generati. Questi metodi hanno due lati positivi:

- Non richiedono un lungo processo di reverse engineering.
- Essendo l'AI una blackbox, è molto difficile per gli hackers eseguire un reverse

engineering del modello.

2.2 Some cool topic

Chapter 3

Contribution

You may also put some code snippet (which is NOT float by default), eg: chapter 3.

3.1 Fancy formulas here

```
1 public class HelloWorld {
2     public static void main(String[] args) {
3         // Prints "Hello, World" to the terminal window.
4         System.out.println("Hello, World");
5     }
6 }
```

Bibliography

- [1] R. Sivaguru, C. Choudhary, B. Yu, V. Tymchenko, A. Nascimento, and M. D. Cock, “An evaluation of dga classifiers,” in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 5058–5067.
- [2] B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock, “Character level based detection of dga domain names,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.
- [3] G. Alley-Young, “Conficker worm,” in *The Handbook of Homeland Security*. CRC Press, 2023, p. 175.
- [4] J. Namgung, S. Son, and Y.-S. Moon, “Efficient deep learning models for dga domain detection,” *Security and Communication Networks*, vol. 2021, no. 1, 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/8887881>
- [5] M. Eslahi, R. Salleh, and N. B. Anuar, “Bots and botnets: An overview of characteristics, detection and challenges,” in *2012 IEEE International Conference on Control System, Computing and Engineering*, 2012, pp. 349–354.
- [6] E. Ogu, N. Vrakas, C. Ogu, and A.-I. B.M., “On the internal workings of botnets: A review,” *International Journal of Computer Applications*, vol. 138, pp. 975–8887, 04 2016.
- [7] X. Ma, X. Guan, J. Tao, Q. Zheng, Y. Guo, L. Liu, and S. Zhao, “A novel irc botnet detection method based on packet size sequence,” in *2010 IEEE International Conference on Communications*, 2010, pp. 1–5.

Acknowledgements

Optional. Max 1 page.