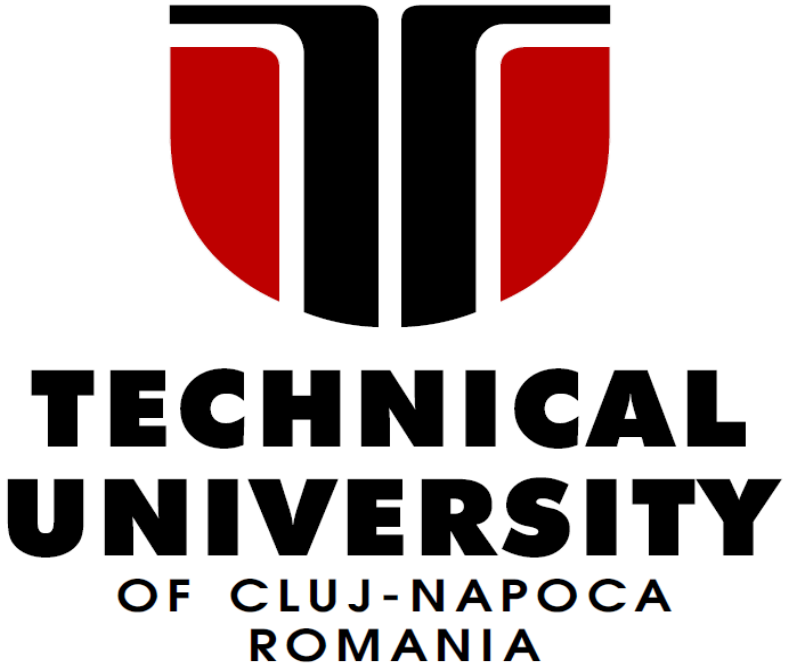

Proiect ASO 2024: Management-ul unui botnet

Documentatie - Etapa 1

Iamnitchi Bogdan - Grupa 30461

Administrarea Sistemelor de Operare
Facultatea de Automatica si Calculatoare
2024-2025

Cerinte rezolvate

Cerințele realizate până acum

1. **Analiza codului sursă Mirai**:
   - Descarcarea codului sursă de pe GitHub.
   - Identificarea și descrierea componentelor principale:
     - Botul (malware-ul) care se instalează pe dispozitive.
     - Serverul de comandă și control (C&C) care primește comenzi de la atacator.
2. **Crearea unei mașini virtuale și instalarea dependințelor**:
   - Configurarea unei mașini virtuale (VM).
   - Instalarea Golang și a altor pachete necesare pentru serverul C&C.
3. **Compilarea binarelor malware-ului**:
   - Modificarea fișierului `table.c` pentru a include IP-ul serverului C&C.
   - Compilarea codului sursă pentru a crea binarele malware-ului.
4. **Rularea serverului C&C**:
   - Compilarea și instalarea serverului de comandă și control.
   - Rularea serverului C&C pe mașina virtuală, fără a porni malware-ul

Modul de rezolvare

---

1. Actualizarea sistemului și instalarea pachetelor necesare
   `sudo apt update -y`: Actualizăm lista pachetelor disponibile pentru a ne asigura că avem cele mai recente versiuni disponibile în repository-uri.
   `sudo apt install -y ...`: Aici instalăm mai multe pachete:

   - **`git`**: Un sistem de control al versiunilor care ne ajută să clonăm codul sursă de pe GitHub.
   - **`build-essential`**: Oferă un set de instrumente necesare pentru compilarea programelor, cum ar fi compilatoarele și bibliotecile standard.
   - **`golang-go`**: Limbajul de programare Go, necesar pentru compilarea serverului C&C.
   - **`mariadb-server`** **și** **`mariadb-client`**: Instalăm serverul și clientul MariaDB, o bază de date care ne va ajuta să gestionăm datele utilizatorilor și comenzile.

2. Configurarea bazei de date, utilizand comenzi SQL

- Creăm un utilizator numit **`asodb`** cu parola **`asodb`** și îi oferim privilegii complete asupra bazei de date.
- Creăm baza de date **`mirai`**.
- Creăm tabelele **`history`**, **`users`**, și **`whitelist`** pentru a stoca informații despre comenzi, utilizatori și adrese permise.
- Inserăm un utilizator default (**`aso`**) în tabelul **`users`**.

3. Activarea și configurarea MariaDB:
- **`sudo systemctl enable mariadb`**: Activăm serviciul MariaDB pentru a porni automat la boot-ul sistemului.
- **`sudo mysql < /tmp/db.sql`**: Executăm comenzile din fișierul SQL creat anterior pentru a configura baza de date.

4. Clonarea codului sursă Mirai:
- **`git clone ...`**: Clonăm repository-ul de pe GitHub pentru a avea acces la codul sursă al botnet-ului Mirai, daca nu am facut o deja.
- **`pushd Mirai-Source-Code/mirai`**: Intrăm în directorul `mirai` pentru a lucra cu fișierele botului.

5. Inițializarea modulului Go și gestionarea dependențelor
- **`go mod init aso/project`**: Inițializăm un nou modul Go pentru proiectul nostru, stabilind structura de proiect.
- **`go mod tidy`**: Aceasta comandă se asigură că avem toate dependențele necesare pentru a compila codul, eliminând eventualele fișiere nefolosite.

6. Modificarea fișierelor de configurare
- Modificăm fișierul **`main.go`** pentru a schimba utilizatorul root cu **`asodb`**.
- Schimbăm și parola din fișierul **`main.go`** pentru a se potrivi cu utilizatorul creat.

7. Modificări in codul sursa a botului din cauza compilatoarelor care nu mai suporta anumite lucruri:
- **În fișierul `bot/includes.h`,** am găsit linia care zice **`ipv4_t LOCAL_ADDR;`** și am adăugat **`extern`**. Asta ajută ca variabila să fie recunoscută în alte fișiere. Practic, am spus compilatorului că variabila va fi definită în altă parte, astfel încât să o putem folosi fără probleme.
- **Apoi, am deschis fișierul `bot/main.c`.** Acolo, am căutat linia cu **`struct sockaddr_in srv_addr;`** și am înlocuit-o cu: **`ipv4_t LOCAL_ADDR;`**

8. Construirea binarelor:
    - `./build.sh debug telnet`: Compilăm botul Mirai în modul debug, pentru a putea face teste și depanări ușor.

9. Copierea fișierului de configurare (fara de care nu ruleaza cnc-ul):
    - `cp prompt.txt debug/`: Copiem un fișier de prompt în directorul `debug`, care poate fi folosit pentru a interacționa cu botul în timpul testelor.

10. Rularea cnc-ului:
    - M-am asigurat că sunt în folderul corect unde am compilat codul CNC si am rulat serverul CNC din terminal folosind comanda specifică, `./cnc` ceea ce a pornit serverul.
    - După ce serverul a fost activ, m-am conectat la el prin Telnet la adresa `localhost` (de obicei, folosind `telnet localhost <port>`).
    - Odată conectat, am observat că promptul afișat era în limba rusă, ceea ce confirmă că serverul C&C rulează corect și este pregătit să primească comenzi.

Probleme întâlnite și modul de rezolvare

---

1. **Configurarea bazei de date:** A fost destul de complicat să configurez corect baza de date MariaDB. Am avut nevoie de timp pentru a mă asigura că utilizatorul și permisiunile sunt setate corect.
2. **Lipsa fișierului `prompt.txt`:** Am realizat că serverul CNC nu funcționează corect fără fișierul `prompt.txt`. Acesta este esențial pentru ca serverul să comunice eficient. A trebuit să mă asigur că fișierul este prezent și corect configurat în directorul `debug`.
3. **Probleme cu compilarea codului:** La compilarea codului botului, am întâmpinat erori legate de biblioteci lipsă. A fost necesar să instalez pachete suplimentare și să mă asigur că toate dependențele erau corecte.
4. **Conexiuni Telnet refuzate:** Când am încercat să mă conectez la server prin Telnet, am obținut un mesaj de eroare că conexiunea a fost refuzată. A trebuit să verific setările serverului și să mă asigur că acesta era pornit și asculta pe portul corect.

Concluzii

În urma acestui proiect, am reușit să configurez și să rulez serverul CNC cu succes, dobândind astfel o înțelegere practică a funcționării unui botnet. Am învățat ce este un bot, un program care execută comenzi de la un server central, și cum aceste boturi se unesc pentru a forma un botnet, o rețea de dispozitive compromise. De asemenea, am înțeles rolul esențial al serverului de comandă și control (C&C), care coordonează activitățile boturilor și permite atacatorilor să le controleze.

Un alt aspect important a fost să văd pentru prima dată cum arată codul sursă al unui malware, ceea ce m-a ajutat să realizez complexitatea și tehnicile folosite în crearea acestora. Această experiență a fost extrem de educativă, oferindu-mi o perspectivă asupra provocărilor de securitate cibernetică și a importanței protecției împotriva amenințărilor. De asemenea, am realizat cât de important este să înțelegem aceste concepte pentru a putea contracara atacurile cibernetice în viitor.