Pacchetti Server e Client per la Lezione

Questo documento contiene le istruzioni per preparare i pacchetti da distribuire agli studenti e quelli che dovrai eseguire come docente.

Pacchetto per il Docente (Server)

File da includere:

1. `server_vulnerabile.py` - Server web con vulnerabilità per dimostrazioni
2. `secure_comms.py` - Client/server con/senza crittografia
3. `tutorial_docente.md` - Guida completa per il docente

Istruzioni di installazione:

```bash
# Creare una cartella per il server
mkdir server_demo
cd server_demo

# Copiare i file server nel folder
cp server_vulnerabile.py secure_comms.py tutorial_docente.md ./

# Installare le dipendenze necessarie
pip install flask cryptography requests
```

Istruzioni di esecuzione:

1. Per il server vulnerabile:

```bash
python server_vulnerabile.py
```

Il server si avvierà sulla porta 5000 e sarà accessibile da tutti gli host della rete locale all'indirizzo `http://TUO_IP:5000`.

2. Per il server di comunicazione (non sicuro):

```bash
python secure_comms.py server --no-encryption
```

Il server di comunicazione si avvierà sulla porta 8000.

3. Per il server di comunicazione (sicuro):

```bash
python secure_comms.py server
```

Il server di comunicazione sicuro si avvierà sulla porta 8000.

Pacchetto per gli Studenti (Client)

File da includere:

1. `mitre_attack_client.py` - Per esplorare il framework MITRE ATT&CK
2. `port_scanner.py` - Per eseguire scansioni di port su un host
3. `packet_sniffer.py` - Per analizzare il traffico di rete
4. `traffic_generator.py` - Per generare traffico di test
5. `secure_comms.py` - Client per comunicare con il server
6. `istruzioni_studenti.md` - Guida per gli studenti

Istruzioni per gli studenti:

1. Installazione delle dipendenze:

```bash
# Creare un ambiente virtuale (opzionale ma consigliato)
python -m venv cybersec_env
source cybersec_env/bin/activate  # Linux/Mac
# o
cybersec_env\Scripts\activate  # Windows

# Installare le dipendenze
pip install scapy mitreattack-python cryptography requests
```

2. Esplorazione del MITRE ATT&CK framework:

bash

```
python mitre_attack_client.py
```

3. Scanning delle porte del server:

bash

```
python port_scanner.py IP_DEL_DOCENTE
# Scegliere opzione 1 (Porte comuni)
```

4. Generazione di traffico per test:

bash

```
# Richiede privilegi amministrativi
sudo python traffic_generator.py -t IP_DEL_DOCENTE --all
```

5. Analisi del traffico:

bash

```
# Richiede privilegi amministrativi
sudo python packet_sniffer.py
```

6. Connessione al server non sicuro:

bash

```
python secure_comms.py client --host IP_DEL_DOCENTE --no-encryption
```

7. Connessione al server sicuro:

bash

```
python secure_comms.py client --host IP_DEL_DOCENTE
```

Istruzioni per il Docente: Preparazione della Lezione

Prima dell'inizio:

1. **Preparazione dell'ambiente**:
   - Assicurati che il tuo computer abbia Python, Git e tutte le librerie necessarie installate

- Controlla che il firewall non blocchi le connessioni in entrata sulle porte 5000 e 8000
- Testa tutti gli script in anticipo per verificare che funzionino correttamente

2. **Preparazione della rete**:
   - Verifica che tutti i computer degli studenti siano sulla stessa rete locale
   - Annota il tuo indirizzo IP locale
   - Se possibile, configura una rete isolata per la lezione
3. **Preparazione del materiale**:
   - Crea una chiavetta USB con i file per gli studenti o prepara un link per il download
   - Stampa o prepara una presentazione con le istruzioni di base
4. **Test dei server**:
   - Avvia il server vulnerabile e verifica che sia accessibile da un altro computer
   - Avvia il server di comunicazione e verifica che funzioni

Durante la lezione:

1. **Gestione dei server**:
   - Monitora l'output dei server per vedere le richieste degli studenti
   - Sii pronto a riavviare i server in caso di problemi
   - Aiuta gli studenti che hanno difficoltà a connettersi
2. **Dimostrazione Wireshark**:
   - Avvia Wireshark prima di eseguire i server
   - Imposta filtri per mostrare solo il traffico pertinente (es. `tcp port 5000` o `tcp port 8000`)
   - Proietta lo schermo durante la dimostrazione di traffico non criptato vs criptato
3. **Risoluzione problemi comuni**:
   - Problemi di connessione: verifica firewall e rete
   - Errori di permessi: ricorda che packet_sniffer.py e traffic_generator.py richiedono privilegi admin
   - Problemi di installazione: aiuta gli studenti con le dipendenze mancanti

Creazione del Pacchetto Distribuibile

È consigliabile creare un pacchetto zip organizzato per una facile distribuzione:

Pacchetto docente:

bash

```
mkdir server_package
cp server_vulnerabile.py secure_comms.py guida-completa-docente.md server_package/
zip -r server_package.zip server_package/
```

Pacchetto studenti:

bash

```
mkdir client_package
cp mitre_attack_client.py port_scanner.py packet_sniffer.py traffic_generator.py secure_comms.p
zip -r client_package.zip client_package/
```

Riferimenti utili per il Docente

Comandi Wireshark utili:

- Filtro per traffico HTTP: `tcp port 80` o `http`
- Filtro per traffico sulla porta del server vulnerabile: `tcp port 5000`
- Filtro per traffico sul server di comunicazione: `tcp port 8000`
- Filtro per ICMP (ping): `icmp`
- Filtro per DNS: `dns`

Comando per controllare il tuo IP:

bash

```
# Linux/Mac
ifconfig | grep "inet "

# Windows
ipconfig
```

Comandi per verificare le porte in ascolto:

bash

```bash
# Linux/Mac
netstat -tuln

# Windows
netstat -an | findstr "LISTENING"
```