SIMULAZIONE DI ATTACCHI ALLA RETE OT IN AMBIENTE POWER

Documento di progetto

Andrea Ierardi

andreierardi@gmail.com
20018785@studenti.uniupo.it

Sommario

1. Introduzione

In questo progetto sono presenti ben tre attacchi alla sicurezza della rete OT in ambiente power. Gli attacchi sono: Man in the Middle passivo, Websocket Denial of Service (DoS) e Packets Filtering. L'ambiente viene simulato attraverso un software chiamato Docker. Questo utilizza dei container che permettono di simulare le varie componenti del sistema in modo semplice, veloce e pulito dato che non è presente nessuna macchina virtuale ma viene eseguito tutto all'interno dello stesso kernel. Le componenti del sistema sono il client e server MMS e l'attaccante che, in base all'attacco, avrà bisogno di uno o più container.

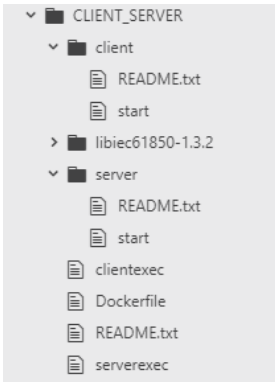
2. Requisiti

- Una distruzione Linux (ho utilizzato Linux Centos7.5)
- Installazione del software di simulazione Docker
- Connessione ad Internet per la costruzione delle immagini Docker e per il download dei software necessari al funzionamento dei container.

3. File e cartelle

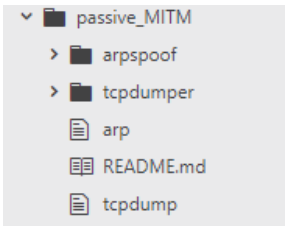
Sono presenti cinque cartelle principali:

- CLIENT_SERVER



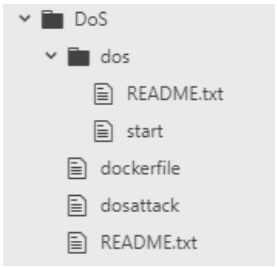
In questa cartella vi sono i file bash “clientexec” e “serverexec” che contengono tutti i comandi da terminale necessari per inizializzare ed eseguire i container Docker per server e client. Il Dockerfile è necessario per la build e l’inizializzazione dei container ed è unico per semplicità, dato che client e server necessitano solo della cartella della libreria libiec6150-1.3.2 (anch’essa presente) per il protocollo MMS. Le cartelle client e server contengono uno script bash necessario per l’esecuzione del client e del server all’interno del container.

- Passive_MITM



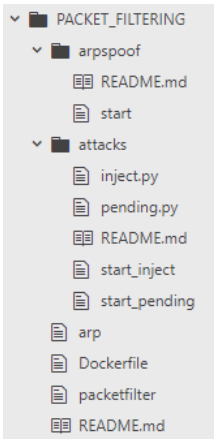
In questa cartella vi sono due script bash “arp” e “tcpdump” che eseguono tutti i comandi da terminale per l’inizializzazione e l’avvio dei container necessari per l’avvio dell’attacco Main in the middle passivo. Inoltre, questi script hanno al loro interno, anche la dichiarazione dell’immagine per la creazione della build per i Docker container in modo da non avere troppi Dockerfile nella stessa cartella. Vi sono due cartelle arpspoof e tcdumper, che contengono gli eseguibili per poter far partire l’attacco ARP e tcpdump all’interno dei container.

- DoS



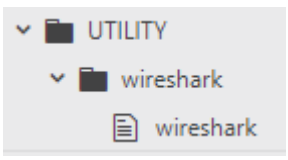
In questa cartella vi è uno script bash “dosattack” che esegue tutti i comandi via terminale necessari per il funzionamento dell’attacco DoS e l’avvio del container. E’ presente un Dockerfile per la configurazione della build del container e una cartella DoS, contenente lo script bash eseguibile all’interno del container.

- PACKET_FILTERING



In questa cartella sono presenti due script bash “arp” e “packetfilter” che permetto l’inizializzazione e la costruzione automatica dei container per l’attacco. E’ presente il Dockerfile per costruire l’immagine del packetfilter container. Nelle varie cartelle sono presenti i file necessari per eseguire gli attacchi all’interno dei container e che quindi dovranno essere avviati per poter lanciare l’attacco.

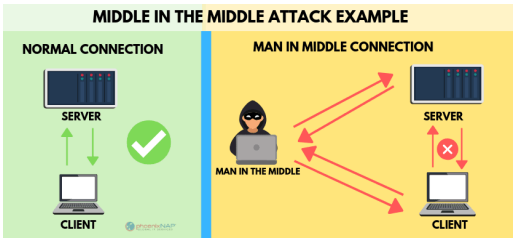
- UTILITY



Infine, nella cartella utility è presente il file bash “wireshark” che consente di avviare wireshark in un container Docker in caso si voglia analizzare il traffico di rete.

4. Descrizione degli attacchi

Man in the Middle passivo



L'attacco Man in the Middle prevede che l'attaccante si interponga tra due interlocutori che credono di comunicare tra di loro. In questo caso, l'attaccante, si pone nel mezzo della comunicazione fingendosi il client per il server e il server per il client.

Per la realizzazione di questo attacco sono necessari:

- Un container che esegue l'attacco ARP poisoning, ovvero una tecnica che consiste nell'inviare intenzionalmente e in modo forzato risposte ARP contenenti dati alterati. In questo modo, l'attaccante, può fingersi un host rispetto ad un altro. Nel nostro caso, l'attaccante invierà risposte al client per riferirgli il suo indirizzo MAC per far credere di essere il server e lo stesso procedimento verrà applicato al server fingendosi il client. Per attuare l'ARP poisoning si utilizza il tool Arpspoof.
- Un container che esegue un tool chiamato Tcpdump, che consente di intercettare i pacchetti e le trasmissioni e quindi effettuare il vero e proprio sniffing dei pacchetti.

L'attacco è detto passivo, poiché l'attaccante non altera i pacchetti al passaggio, ma li intercetta, memorizza e inoltra al destinatario.

Websocket Denial of Service

L'attacco Denial of Service consiste nel generare una moltitudine di connessione pendenti con il server in modo tale da saturare tutte le connessioni disponibili, così da non permettere al client di connettersi.

Per la realizzazione di questo attacco è necessario:

- Un container che esegua il tool websocket-bench a cui viene specificato di instaurare 40000 connessioni persistenti totali di cui 2000 al secondo concorrenti. In questo modo il server viene reso praticamente irraggiungibile fino a che il tool rimane in esecuzione.

Packets filtering

L'attacco Packets filtering prevede che venga instaurato un attacco Man in the Middle che però non risulta più passivo poiché si andrà a decidere quali pacchetti inoltrare al destinatario.

Per realizzare l'attacco sono necessari:

- Un container che esegue ARP poisoning come descritto nell'attacco Man in the middle passivo

- Un container che invece esegua gli script di filtraggio dei pacchetti. Sono presenti due Python script che permettono di avviare due tipi di filtraggio differenti. Uno filtra i pacchetti dal 40esimo al 60esimo inviati dal client e questo genera un errore di reporting da parte del client poiché si aspetta un risposta dal server che non arriva. Il secondo script, invece, filtra i pacchetti di numero superiore al 27 in modo tale da far instaurare l'handshake TLS tra i due interlocutori e far rimanere pendente la comunicazione del client. Questo perché il client continuerà a inviare gli stessi pacchetti inziali in loop poiché non ha ancora ricevuto nessuna ACK di conferma dal Server. L'ultimo script è ancora sperimentale poiché non sempre funziona a dovere, dato che il client a volte può andare in errore e terminare. Si potrebbero modificare gli script per consentire anche il filtraggio dei pacchetti per dimensione. Il tool utilizzato è la libreria libnetfilter_queue di Python

5. Realizzazione degli attacchi

Avvio del Client e del Server MMS

Per prima cosa è necessario l'avvio del client e del server. Per evitare problemi a livello di indirizzi IP, consiglio di avviare per primo il container del client e successivamente quello del server.

Per ognuno sono necessari due terminali in cui eseguire il comando:

Terminale1 : $ ./clientexec

Terminale2 : $ ./serverexec

Una volta avviati i container, ci si troverà nella cartella tls_server_example della libreria MMS per il server, mentre per il client nella cartella tls_client_example.

Per instaurare una comunicazione è possibile utilizzare lo script bash "start". Per prima cosa occorre avviare il server che si metterà in ascolto in automatico sulla porta 3782.

Terminale2 : $ ./start

Una volta avviato, per far connettere il client è sufficiente utilizzare lo script bash "start" specificando però l'indirizzo IP del server (rintracciabile utilizzando il comando arp -a oppure direttamente nel server con il comando ifconfig).

Terminale1 : $ ./start [SERVER_IP]
Esempio: ./start 172.17.0.3

Esecuzione Man in the middle passivo

Prima di tutto bisogna avviare l'ARP poisoning. Questo perché al lancio del container di Tcpdump, questo si connetterà alla rete Docker con lo stesso indirizzo IP e MAC dell'attacco ARP. E' necessario avere lo stesso IP, poiché tcpdump dovrà intercettare i pacchetti che passano prima nella rete dell'attaccante che poi verranno inoltrati al destinatario di partenza.

Per avviare l'attacco ARP è necessario lanciare lo script bash in un nuovo terminale (diverso da quello del client e server):

Terminale3 (arp) : $ ./arp

Una volta avviato l'ARP poisoning è necessario avviare il container di tcpdump in un nuovo terminale:

Terminale4 (tcpdump) : $ ./tcpdump

Avviati i due container ci si troverà in una cartella con un file bash "start". Consiglio di avviare prima l'attacco ARP e successivamente tcpdump. Per l'ARP poisoning è necessario specificare l'indirizzo IP del client e del server in qualsiasi ordine.

Terminale3 (arp) : $ ./start [SERVER_IP] [CLIENT_IP]
Esempio: ./start 172.17.0.3 172.17.0.2

Terminale4 (tcpdump) : $ ./start

Una volta avviato l'attacco ARP e tcpdump si può testare subito l'attacco avviando la comunicazione tra client e server.

Terminale1 (client) : $ ./start [SERVER_IP]

Una volta terminata la comunicazione, se si interrompe l'esecuzione della start di tcpdump (con il comando CRT + C), si può notare che è stato creato il file sniffed.pcap che potrà successivamente essere letto con ssldump (già installato nel container) o con altri tool come Wireshark o Tshark.

Eseguiti questi due, quando il client eseguirà lo start (quindi si connetterà al server) Arpspoof eseguire l'ARP poisoning delle tabelle ARP in modo tale che l'attaccante si finga il server per il client e il client per il server.

NB. E' possibile verificare che l'attacco ARP sia andato a buon fine con il comando *arp -a* dal client. Se vengono visualizzati due indirizzi MAC uguali associati a due IP differenti, implica che l'indirizzo MAC del server è stato sostituto correttamente da quello dell'attaccante (sono due uguali perché l'indirizzo MAC dell'attaccante è anche associato al suo IP oltre che a quello del server).

Esecuzione Websocket Denial of Service

Prima di tutto bisogna avviare il container dell'attacco lanciando lo script bash "dosattack" (da un terminale differente da quello del Server e del Client):

Terminale3 (dos) : $ ./dosattack

Una volta avviato, all'interno vi sarà un altro script bash "start" che dovrà essere eseguito per lanciare l'attacco specificando l'IP del server:

Terminale3 (dos) : $ ./start [IP_SERVER]

Una volta lanciato l'attacco, quando il client cercherà di connettersi al server:

Terminale1 (client) : $ ./start [SERVER_IP]

questo non riuscirà a connettersi poiché l'attacco instaurerà talmente tante connessioni socket col server da non essere in grado di reggerne ulteriori. Il client, infatti, terminerà con un errore.

Esecuzione Packets Filtering

Come per il Man in the Middle passivo, c'è bisogno dell'attacco ARP. Prima di tutto bisogna eseguire lo script bash "packetfilter" ed è necessario farlo prima del container "arp" per lo stesso motivo del Man in the Middle passivo (il container arp si connetterà con stesso MAC e IP del container packetfilter per risultare una macchina singola).

Per questo attacco sono necessari due terminali: uno per eseguire l'ARP poisoning e uno per eseguire il vero e proprio packets filtering.

Terminale3 (packetfilter) : $ ./packetfilter

Una volta eseguito, avviamo il container per l'attacco ARP:

Terminale4 (arp) : $ ./arp

Una volta avviato, eseguire l'ARP poisoning lanciando lo script start:

Terminale4 (arp) : $ ./start [IP_SERVER] [IP_CLIENT]

A questo punto ci sono due alternative, avviare lo script bash:

- **Terminale3(packetfilter)** : $ ./start_inject
- **Terminale3 (packetfilter)** : $ ./start_pending

Lo script *start_inject* farà in modo di generare un errore di reporting nel client bloccando pacchetti dal 40 al 60 del client. Lo script *start_pending*, invece, farà in modo di far instaurare la connessione TLS tra client e server e poi bloccherà tutti i pacchetti successivi (va a bloccare dal 27° pacchetti in poi) poiché quelli precedenti sono i pacchetti che client e server utilizzato per l'handshake TLS.

NB. Il problema è che il MAC del client da filtrare è impostato guardando i due bit finali che corrisponderanno a x02. Il container del client (se avviato per primo) di default avrà indirizzo IP 172.17.0.2 e indirizzo MAC contenente x02 all'interno. Per questo motivo, consiglio di eseguire il client prima di tutti i container in modo tale da non dover reimpostare l'indirizzo IP del client (con ifconfig eth0 [IP]) e dover cambiare lo script. Per questo è ancora sperimentale e modificabile, bisognerebbe rendere possibile specificare il MAC del client.

6. Risoluzione problemi

- L’attacco ARP prevede di mandare parecchie risposte ARP che verranno poi stampate a terminale, rendendo difficile lo stop del container. Per questo motivo, consiglio prima di tutto di eseguire CTR + C per fermare il processo e digitare il comando exit più volte per interrompere il comando /bin/bash per terminare il container.

- Molto spesso, interrompendo i vari attacchi, è possibile che rimanga in background il loro processo in esecuzione. In questi casi consiglio di utilizzare il comando ps e successivamente terminare tutti i processi (con il comando kill -9 [PID_PROCESSO]) che non siano ps e bash.

- In caso di problemi di esecuzione perché il passaggio dei parametri bash non funziona correttamente oppure si vogliono specificare parametri differenti ad alcuni tool, è possibile stampare a video i comandi lanciati negli script bash con il comando cat [nomefilebash] e ripetere gli stessi con parametri differenti.

- In caso non si riuscisse ad eseguire i file bash all’interno dei containers, probabilmente mancano i permessi di esecuzione. E’ possibile dare i permessi con il comando chmod +x [nomefile].