Relazione_Gestione_Reti

Riconoscitore di Protocolli Non Standard

Autore: Lorenzo Iannarella
Email: l.iannarela@studenti.unpi.it

Introduzione

Il documento presente descrive `recognition_nsp.lua`, un plugin Wireshark scritto in Lua. Questo strumento è progettato per riconoscere i flussi di dati generati dalle chiamate e videochiamate effettuate tramite le applicazioni come Telegram, WhatsApp, Teams, Meet e Zoom.
Questa applicazioni utlizzano molti protocolli, ma tra questi vi è il protocollo STUN che viene usato per stabilire connessioni tra gli utenti (in quanto permette ad un host di conoscere il proprio indirizzo pubblico).
I pacchetti STUN hanno permesso la realizzazione di questo plugin, in quanto l'identificazioni di questi protocolli non standard si basa prorpio sull'analisi dei diversi campi contenuti nel layer STUN.

Descrizione

Il plugin `recognition_nsp.lua` si basa su un post-dissector, un tipo specifico di dissector Wireshark, che si occupa dell'analisi dei pacchetti. In particolare, questo plugin si concentra sull'analisi dei pacchetti UDP, poiché le applicazioni menzionate precedentemente utilizzano principalmente il protocollo di trasporto UDP per la trasmissione dei dati.

Per ogni pacchetto UDP, `recognition_nsp.lua` inizia verificando se il pacchetto è un pacchetto STUN. Se lo è, il pacchetto viene analizzato ulteriormente per determinare quale protocollo non standard rappresenta. Successivamente, il flusso - identificato tramite ***IP sorgente, porta sorgente, IP destinazione e porta destinazione*** - viene registrato nella `flows_table`. Se il pacchetto non è un pacchetto STUN, il plugin verifica se appartiene a un flusso già riconosciuto, cercandolo nella `flows_table`.

Dopo aver completato l'analisi di un pacchetto, vengono inserite nell'albero di dissezione di Wireshark informazioni quali il nome del protocollo non standard identificato. Questo permette di visualizzare tale informazione anche nella *Packet List* di Wireshark.

Strutture dati utilizzate

```
local  request_table = {}

local  flows_table = {}

local  processed_packets = {}
```

La `flows_table` è stata impiegata per registrare i flussi e i relativi protocolli applicativi non standard associati.

La `request_table` è stata impiegata per conservare informazioni relative ai pacchetti STUN che rappresentano delle *Bind Request*. Questi pacchetti hanno permesso di identificare il protocollo applicativo sottostante. Inoltre, hanno facilitato l'identificazione del protocollo dei pacchetti di risposta, ovvero le *Binding Success Response*.

La `processed_packets` è stata impiegata per tenere traccia dei pacchetti che sono stati già analizzati. Questo è stato fatto per prevenire che il post-dissector li elaborasse nuovamente in

maniera diversa, a causa dell'evoluzione delle informazioni raccolte attraverso le analisi effettuate sui pacchetti precedenti e sui flussi identificati.

Definizione del layer

```
local l7_proto = Proto("l7", "Layer 7 Protocol")
l7_proto.fields = {}
local l7_fds = l7_proto.fields
l7_fds.proto = ProtoField.new("Protocol Name", "l7.proto", ftypes.STRING)
```

Per aggiungere informazioni sai nella *Packet List* che nella *Details packet* è stato definito un protocollo ,ovvero **"Layer 7 Protocol"** e gli è stato aggiunto un campo,**"Protocol Name"**.

Aggiunta dell'informazione all'albero di dissezione

```
local  subtree = tree:add(l7_proto, tvb(), "Application Protocol")
```

Questa istruzione ha consentito l'integrazione del layer appena creato nell'albero di dissezione. Di conseguenza, il layer è ora visibile nella sezione *Details Packet* di Wireshark. Questo assicura che tutte le informazioni pertinenti siano facilmente accessibili e visualizzabili anche nella *Packet List*.

```
subtree:add(l7_fds.proto, protocol)
```

que

Prerequisiti

Avere **Wireshark** installato sul pc.

Istruzioni

Aggiungere una colonna alla Packet List

1. **Vai su** `Edit`.
2. **Clicca su** `Preferences`.
3. **Clicca su** `Columns`.
   3.1 aggiungi una colonna premendo sul `+`.
   3.2 seleziona la spunta sulla riga appena creata .
   3.3 Nella colonna `Title` scrivi `Application Protocol`.
   3.4 Nella colonna `Type` seleziona `Custom`.
   3.5 Nella colonna `Fields` scrivi `l7.proto`.
4. **Clicca** `Ok`.

Caricamento del Plugin

Caricamento dello script da Terminale

1 Trovare la Cartella dei Plugin di Wireshark

Wireshark carica automaticamente gli script Lua dalla sua directory di plugin. Ecco dove puoi trovare questa directory su diversi sistemi operativi:

- **Windows**: `C:\Program Files\Wireshark\plugins\<version>\`
- **macOS**: `/Applications/Wireshark.app/Contents/PlugIns/wireshark/<version>/`
- **Linux**: `/usr/lib/wireshark/plugins/<version>/`
-

2 Copiare lo Script Lua nella Cartella dei Plugin

Copia il tuo script Lua (`recognition_nsp.lua`) nella directory dei plugin di Wireshark. Se stai usando la directory personale, assicurati che esista e crea la cartella `plugins` se necessario.

Caricare lo Script Lua tramite la GUI di Wireshark

1. **Apri Wireshark**.
2. **Clicca su** `Help`.
3. **Clicca** `About Wireshark`:
4. **Vai in** `Folders`:
5. **Clicca sul link individuato da** `Personal Plugin`
6. **Inserisci lo script lua**

Riavvia Wireshark

- Chiudi Wireshark e riaprilo per caricare il nuovo script Lua, questa operazione indimentemente la modalitàdi caricamento utilizzata.