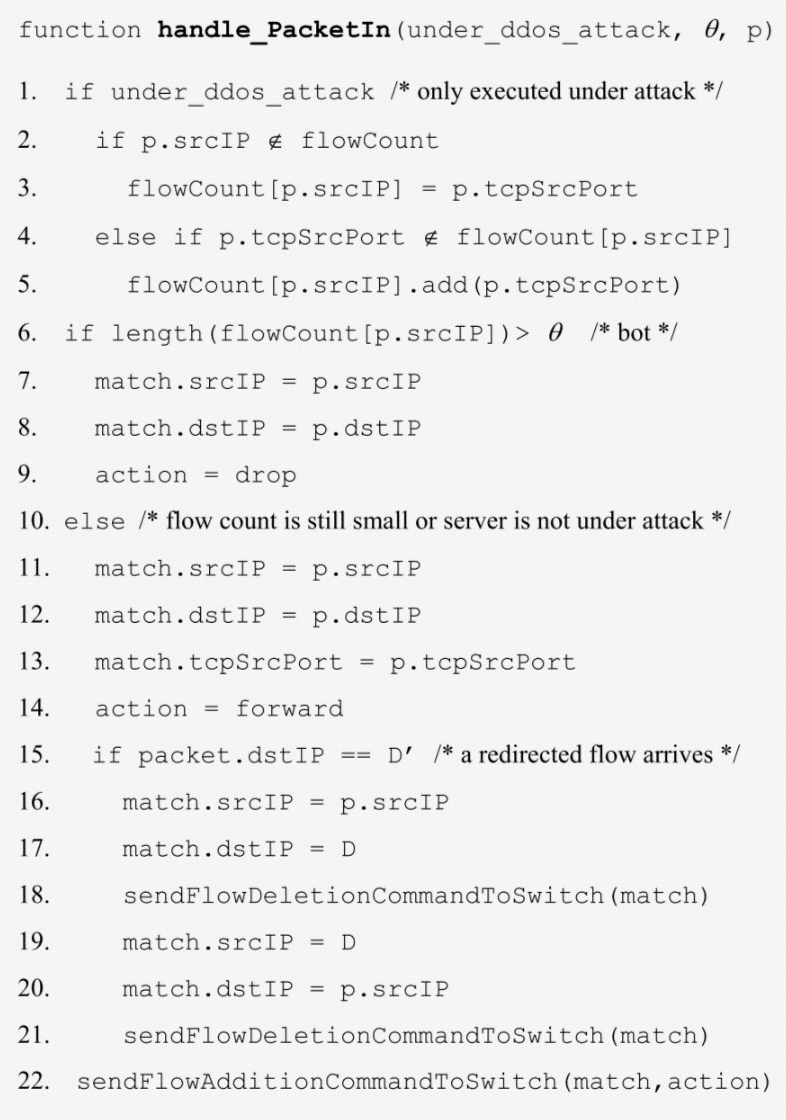
SDN-Oriented DDos Blocking Scheme for Botnet-Based Attacks

Creiamo un controller SDN per la difesa da attacchi DDoS. Attraverso un sistema di API REST un web server può notificare l’inizio di un attacco di tipo DDoS; il controller provvederà a creare un meccanismo di cambio indirizzo da D a D’.

In questo progetto realizziamo:



## IMPLEMENTAZIONE

….Qui Anto lo sai te ☹

## ASSUNZIONI

* Prima parte del paper, in cui si stabilisce il cambio di indirizzo del server da D a D’, non implementato.
* Numero degli indirizzi IP: predefinito e sufficiente per lo scopo del progetto. Il nostro progetto prevede la gestione del numero di indirizzi IP tramite lista circolare per l’assegnazione.
* Il nostro progetto prevede di definire un pool di indirizzi per **…????\***
* Il nostro progetto prevede di definire un servizio di reCAPTCHA: inseriremo un equivalente implementato da noi\*
* Gestione di ARP:
  + Si modifica l’indirizzo IP ma non l’ARP tabl, il server si autoassegna il nuovo indirizzo IP.
  + Questo comportamento è implementato nel modulo forwarding che va incluso
* Gestione della DELETE: cancella la regola precisa dettata dal codice. Se si vuole cancellare più di una dobbiamo gestirlo tenendo traccia delle entry precedenti o inserendo una regola a priorità massima che invalida le altre.

\*Le assunzioni sono opzionali e verranno implementate a seconda delle difficoltà che incontreremo nel progetto

## PARAMETRI

* Timeout ?
* \tehta = 3
* Cmax = 600
* Cattack = 300
* n = numero di utenti legittimi
* k = numero di bots
* \lamba norm
* \lamba attack
* \mu

## SISTEMA DI TESTING

Il nostro modulo verrà testato attraverso uno script che produrremo che emula il client ed un server. L’architettura scelta per il testing è Mininet.

## DOCUMENTAZIONE FINALE

Al termine di questo progetto saranno prodotti:

* Relazione finale: documento di accompagnamento che illustra le scelte fatte e le motivazioni di tali scelte per l’implementazione del modulo,
* Presentazione sintetica in cui verrà mostrato il lavoro svolto,
* Demo per mostrare il funzionamento del modulo.

Questo verrà implementato, cosa si farà, quali sono le interfacce rest che farà il modulo.

come ? Elenco assunzioni

ci sono delle semplificazioni?

Cosa usi? Plug in? Pyotn?

~~Qui non ci sono assunzioni grandi (ignorare la prima parte del paper).~~

~~Soglia parametri – esponi quali parametri ci sono.~~

~~Testato attraverso un applicazione esterna da scrivere noi o uno script o plug in rest.~~

~~Architettura scelta per il testing dell’implementazione (mininet)~~

Proporre topologia di test

~~Assunzione # indirizzi ip (fai una lista circolare)~~

Se ho più hop per indirizzi, -> mantenere la storia dei passi precedente e corrente (va bene un passaggio solo)

Il server non origina del traffico da solo (differenziazione indirizzi e porta).

~~Pool di indirizzi: implementabile se c’è tempo~~

~~Servizio con capcha: inserire un equivalente , ma anche questo se c’è tempo.~~

~~Relazione:~~

~~documento di accompagnamento che illustra scelte fatte per il codice~~

~~Per l’esame:~~

~~una presentazione in cui presentiamo il lavoro e se vogliamo una demo per far vedere come funziona~~

~~ARP~~

1. ~~Cambi ip ma non cambi l’arp -> non funziona perché arp ha un addr table: due ip diversi avrebbero lo stesso mac addr. Per ogni indirizzo ip che usi devi avere un mac addr~~
2. ~~L’indirizzo ip deve essere gestito:~~ 
   1. ~~Server sotto attacco~~
   2. ~~Arp trasparente -> da non gestire perché il server si autoassegna il nuovo indirizzo D’.~~
   3. ~~Questo comportamento è implementato nel modulo forwarding che va incluso~~

Rotta che viene da fuori (forwarding in floodlight)

Switching su modalità autolearning -> aggiunge una entry nella tabela del forwarding

Mettersi prima del learning sw

Testa: sw con priorità massima – learning switch aggiunge regole

Regola in base alla porta sorgente per contrare le connessioni

~~TIMEOUT~~

~~Scegliere un valore noi, fare keep alive sul client per testare meglio.~~

Usa nc o curl 🡪 io qui non ho capito ☹

Modo per tenere traccia del numero di connessioni avute -> hash table

~~DELETE:~~

~~cancellare tutte le regole che hanno match: serve di guardare nella documentazione o provare. La delete cancella la regola precisa che gli dici. Se vuoi cancellare più di una si deve trovare noi il modo ma carlo non sa se si più fare, se non tenendo traccia delle entry già fatte o inserendo una regola a priorità massima che invalida le altre e levi di mezzo il modulo di sotto.~~

Il modulo sotto che fa autolearning cancella

Lo sw lavora per mac addr, se ho su una porta un host e su un'altra un altro allo sw non importa dell’ip