UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

Tesi di Laurea Magistrale in

Informatica

Curriculum Sicurezza Informatica

DDoS Attack Mitigation Using BGP Flowspec Rules

Relatori:

Arcangelo Castiglione

Palmieri Francesco

Candidato:

Raffaele Squillante

Mat. 05225 01267

ANNO ACCADEMICO 2022/2023

"I computer sono incredibilmente veloci, accurati e stupidi.
Gli uomini sono incredibilmente lenti, inaccurati e intelligenti.
L'insieme dei due costituisce una forza incalcolabile."

INDICE

CAPITOLO 1

ABSTRACT

Gli attacchi Denial-of-Service Distribuiti (DDoS) mirano a impedire l'uso legittimo di un servizio. Dato che la gravità e la frequenza di questi attacchi stanno aumentando, gli attacchi DDoS stanno diventando un problema sempre più grande per Internet.

BGP Flowspec è un'estensione del protocollo Border Gateway Protocol (BGP), progettata per fornire un approccio granulare alla mitigazione dei DDoS.

BGP Flowspec definisce un flusso di rete basato su informazioni specifiche dei pacchetti, come ad esempio l'origine, la destinazione e altre informazioni. Questo flusso può essere abbinato dinamicamente al traffico in ingresso per eliminare il traffico, inserirlo in un'istanza di forwarding diversa per ulteriori esami, o limitarlo a un tasso desiderato.

La ricerca correlata mostra il potenziale di BGP Flowspec per la mitigazione degli attacchi DDoS. Tuttavia, BGP Flowspec manca di precisione, il che potrebbe comportare il filtraggio di traffico legittimo. Ciò potrebbe avere un impatto negativo sulla rete sottostante.

Pertanto, sorge un problema di minimizzazione e massimizzazione: da un lato, è auspicabile massimizzare la quantità di traffico DDoS bloccato. D'altra parte, l'impatto negativo sulla rete deve essere minimizzato.

Lo scopo di questa ricerca è affrontare questo problema investigando come la mitigazione degli attacchi DDoS possa essere migliorata usando il Blackholing. La ricerca presenta un confronto sullo stato dell'arte delle mitigazioni proposte, e una soluzione che comprenda quante più suddette implementazioni possibili.

CAPITOLO 2

ATTACCHI DDOS

2.1 Introduzione

La società moderna è sempre più **dipendente** da Internet e questo inesorabile trend è sempre in aumento. Dipendiamo da Internet per comunicare, per tenerci aggiornati sulle notizie, per le operazioni bancarie e per fare ormai qualsiasi operazione *banale*. Una minaccia alla crescita delle prestazioni e dell'affidabilità di Internet è la crescente frequenza e portata degli attacchi DDoS (*Distributed Denial of Service*).

In un attacco DDoS, un criminale **sovraccarica** la sua vittima con traffico Internet indesiderato, impedendo al traffico normale di giungere alla destinazione prevista.

Ad esempio, un attacco DDoS può essere visto come un **ingorgo del traffico** causato da centinaia di richieste fittizie ai servizi di taxi. Le richieste sembrano legittime ai gestori dei servizi, che mandano i loro conducenti a prelevare i clienti, bloccando inevitabilmente le strade cittadine e impedendo, in tal modo, al traffico legittimo di arrivare a destinazione.

Durante un attacco DDoS, i malintenzionati sfruttano una grande quantità di macchine e dispositivi connessi su Internet, come dispositivi IoT (*Internet*

of Things), smartphone, personal computer e server di rete, per inviare un afflusso di traffico verso le varie destinazioni.

Un attacco DDoS al sito web di un'azienda, a un'applicazione web, alle API, a una rete o all'infrastruttura di un data center può causare downtime e impedire agli utenti legittimi di acquistare prodotti, fruire di un servizio, ottenere informazioni o qualsiasi altra cosa procurando un **ingente danno economico**, oltre che di *immagine*.

Per sferrare un attacco DDoS, i malintenzionati utilizzano dei malware o sfruttano le vulnerabilità della sicurezza per infettare in maniera dannosa macchine e dispositivi e assumerne il controllo. Ogni computer o dispositivo infettato, detto "*bot*" o "*zombie*", diventa così in grado di diffondere ulteriormente il malware, oltre che di prendere parte ad attacchi DDoS. Questi bot si accumulano formando eserciti interi, detti "*botnet*", che, facendo leva sulla potenza della propria numerosità, amplificano la portata degli attacchi.

E, poiché non si accorgono che i dispositivi IoT sono infetti, proprio come capita quando si scarica il filmetto di turno sugli zombie senza sapere che è infetto, i proprietari dei dispositivi legittimi diventano vittime secondarie o partecipanti inconsapevoli degli attacchi, mentre le organizzazioni colpite hanno difficoltà a identificare i malintenzionati.

Dopo aver creato una botnet, un malintenzionato può inviare istruzioni a ogni bot da remoto, indirizzando un attacco DDoS verso il sistema preso di mira. Quando una botnet attacca una rete o un server, il malintenzionato **ordina** ai singoli bot di inviare richieste all'indirizzo IP della vittima.

A volte le botnet, con le loro reti di dispositivi compromessi, vengono *affittate* per sferrare altri potenziali attacchi tramite servizi di hacking "*su commissione*". Ciò consente anche alle persone malintenzionate, ma **prive di formazione o esperienza** in merito, di sferrare facilmente attacchi DDoS.

2.2 Tipi di Attacco DDoS

Gli attacchi DoS e DDoS possono essere classificati in base al **tipo di attacco** in:

- **Attacchi Volumetrici**
  - Gli attacchi volumetrici (*bandwith saturation*) prevedono la **saturazione** della banda della vittima, allo scopo di negare le risorse principali. A differenza degli attacchi semantici, gli attacchi volumetrici sono di solito molto più difficili da mitigare, poiché abusano di servizi legittimi. Ciò significa che il filtraggio influirebbe anche sul traffico legittimo, inoltre, in molti casi, le risorse della vittima sono limitate, rendendo impossibile difendersi.
- **Attacchi Semantici**
  - Gli attacchi semantici, o non volumetrici, (*resource starvation*) si basano sul generare flussi di **dimensioni limitate** e, quindi, non sono facilmente percepibili da un sistema automatico di rilevamento. Questi flussi sono realizzati in maniera tale da saturare la capacità delle risorse di un server sfruttando delle vulnerabilità o dei bug di livello protocollare o generando pacchetti malformati da mandare poi al sistema target per creare problemi sullo stack TCP/IP.

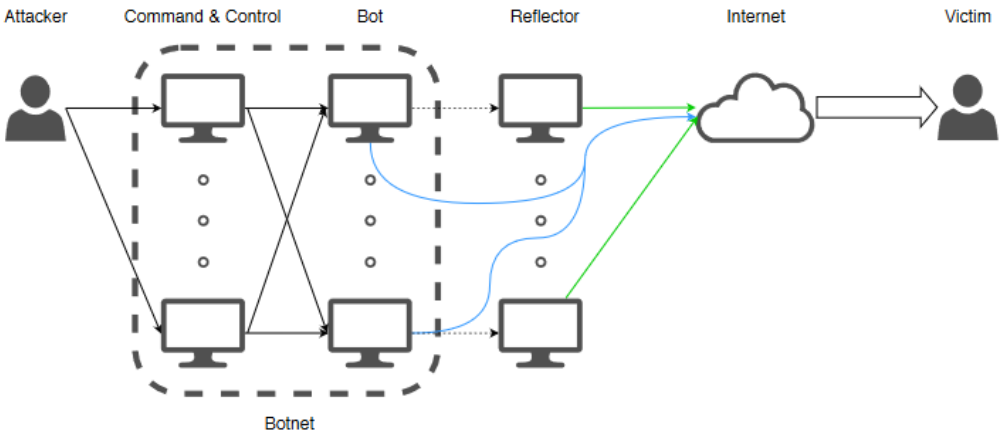

Figura 2.1: DDoS: Attacchi Volumetrici ed Attacchi Semantici

Nella Figura 2.1 è illustrata la differenza tra un attacco DDoS volumetrico e un attacco DDoS semantico. La linea blu indica l'attacco volumetrico, dove il traffico è generato direttamente dai bot, mentre la linea verde indica l'attacco semantico, dove sono usate delle macchine chiamate "reflector". Un attacco di riflessione avviene quando la risposta viene inviata all'origine della richiesta che, però, ha un IP di origine falsificato, così facendo l'aggressore può fare in modo che il server di riflessione invii la risposta alla vittima selezionata. Questa distinzione può essere paragonata alla già citata distinzione tra attacchi volumetrici e semantici, che corrispondono rispettivamente agli attacchi di botnet e di riflessione.
Di seguito sono elencati e categorizzati i dieci attacchi DDoS più usati.

- **Memcached DDoS Attack**

  Un attacco Memcached è un attacco DDoS volumetrico. Un server memcached è un server con un sistema di caching per i database al fine di velocizzare i siti web e le reti. Se questi server sono vulnerabili, gli aggressori possono abusarne inviando richieste spoofed con l'IP del bersaglio negli header, alle quali il server memcached risponderà, inviando una grande quantità di dati al target [7].

- **NTP Amplification Attack**

  Un NTP Amplification Attack è un attacco DDoS volumetrico. In questo tipo di attacco, un aggressore sfrutta le funzionalità di un server NTP per inviare traffico al bersaglio. In particolare, l'aggressore invia una richiesta (con l'indirizzo IP spoofed del target) al server NTP in cui richiede un elenco. Il server risponde inviando l'elenco all'indirizzo IP falsificato. In questo modo, la dimensione della risposta del server è molto più grande della richiesta originale [10].

- **DNS Amplification Attack**

  Un DNS Amplification Attack è un attacco DDoS volumetrico che utilizza un Domain Name System (DNS) server. Si tratta di un altro attacco di riflessione, dove un aggressore invia una richiesta con

l'indirizzo IP spoofed del bersaglio a un resolver DNS. Al fine di creare una grande quantità di traffico, l'attaccante struttura la richiesta in modo tale da generare una risposta il più ampia possibile da parte dei resolver DNS. Di conseguenza, il target riceve amplificato il traffico iniziale dell'aggressore e la sua rete viene intasata da questo traffico [9].

- **SSDP Attack**

  Un attacco SSDP è un attacco DDoS volumetrico che utilizza i dispositivi Universal Plug and Play (UPnP) per eseguire l'attacco. Quando un dispositivo UPnP vuole connettersi a una rete, dopo aver ricevuto un indirizzo IP, il dispositivo invia un messaggio a un determinato indirizzo IP multicast. Successivamente, questo indirizzo comunicherà a tutti i membri della rete le informazioni sul nuovo dispositivo. Quando gli altri dispositivi della rete ricevono queste informazioni, inviano una richiesta al nuovo dispositivo chiedendo un elenco completo delle sue caratteristiche e dei suoi servizi. Un attacco SSDP sfrutta quest'ultima fase, poiché la risposta del nuovo dispositivo genera una grande quantità di traffico. L'aggressore invia pacchetti UDP falsificati ai dispositivi UPnP disponibili, che risponderanno tutti alla vittima, inviando un elenco completo di tutte le funzionalità del dispositivo [12].

- **DNS Flood Attack**

  Un DNS flood è un attacco DDoS semantico dove l'obiettivo è quello di interrompere i servizi dei resolver DNS. Se un dominio non ha una risoluzione DNS, un sito web in esecuzione su quel dominio sarà compromesso. Gli attacchi DNS flood utilizzano molti dispositivi IoT, come le telecamere IP, per inviare richieste al resolver DNS. Questo fa sì che il server DNS venga sopraffatto dal traffico, rendendo l'obiettivo offline. Un attacco DNS flood è particolarmente difficile da mitigare, poiché il traffico proviene spesso da una moltitudine di posizioni uniche, inoltre, le richieste provengono da record reali sul dominio, pertanto,

è difficile per il resolver DNS distinguere il traffico dannoso da quello legittimo [5].

- **HTTP Flood Attack**

  Un HTTP flood è un attacco DDoS semantico, dove l'attaccante (di solito attraverso una botnet) invia una grande quantità di pacchetti HTTP (ad esempio GET, POST, HEAD ecc.) a un server, sovraccaricandolo di traffico [6].

- **SYN Flood Attack**

  Un SYN flood è un attacco DDoS semantico che sfrutta l'handshake nella creazione di una connessione TCP. Normalmente, quando un utente vuole stabilire una connessione, invia un pacchetto SYN al server, chiedendo di stabilire una connessione. Il server risponde con un pacchetto SYN-ACK e lascia una porta aperta in attesa di un pacchetto ACK, che non arriverà mai. L'attaccante invia molte di queste richieste in un breve periodo di tempo. A un certo punto, il server avrà tutte le porte occupate, rendendo le connessioni TCP legittime non disponibili [13].

- **UDP Flood Attack**

  Un UDP flood è un attacco DDoS volumetrico. Per definizione, un UDP flood è un qualsiasi attacco DDoS che inonda un obiettivo con pacchetti User Datagram Protocol (UDP). L'attaccante invia un gran numero di pacchetti UDP a un server mirato con l'obiettivo di sopraffare la capacità del dispositivo di elaborare e rispondere. Quando un server riceve un pacchetto UDP su una particolare porta, controlla se ci sono programmi in esecuzione e in ascolto di richieste su quella porta, se questo non è il caso, invia un pacchetto ICMP che indica che la destinazione non è raggiungibile. Un attacco UDP Flood sfrutta questa situazione facendo sì che il server target invii molte di queste risposte in un breve periodo di tempo, in questo modo, le risorse dell'obiettivo possono esaurirsi. Un attacco UDP Flood di solito ha indirizzi di origine falsificati, in modo che l'identità dell'attaccante rimanga nascosta [14].

- **Ping (ICMP) Flood Attack**

  Un attacco Ping flood è un attacco DDoS semantico. Il protocollo ICMP (Internet Control Message Protocol) è un protocollo di livello Internet utilizzato in diversi strumenti, ad esempio traceroute e ping, per determinare lo stato di salute e la connettività del dispositivo a cui viene inviata la richiesta. Se molti dispositivi di una botnet inviano richieste ICMP ad un target, quest'ultimo è costretto a utilizzare molte risorse per inviare una risposta ICMP a ciascuna di queste richieste [8].

- **Low and Slow Attack**

  Un attacco Low and Slow è un attacco DDoS semantico che ha come obiettivo i server web basati su thread. L'obiettivo è occupare ogni thread con una richiesta lenta, che si traduce in una negazione del servizio per gli utenti reali. Un esempio di strumento che utilizza questo tipo di attacco è Slowloris. Con questo strumento, l'attaccante invia lentamente richieste HTTP parziali. Il server di destinazione server manterrà la connessione aperta, in attesa del resto dell'intestazione. Se questo viene fatto lentamente su ogni thread, il server sarà occupato nell'attesa, ostruendo tutti i thread . Un altro esempio di attacco Low and Slow è R.U.D.Y. Questo strumento genera richieste HTTP POST per la compilazione di un modulo. In queste richieste, indica al server di destinazione quanti dati può aspettarsi (di solito è una grande quantità). I dati vengono inviati molto lentamente, ma abbastanza velocemente da evitare che il server vada in tilt. Poiché il server si aspetta l'arrivo di altri dati, manterrà aperta la connessione. Anche in questo caso, il risultato può essere l'intasamento dei thread [11].

2.3 Possibili Mitigazioni

In questo capitolo verranno illustrati i vari modi in cui gli attacchi DDoS possono essere mitigati. Quando un attacco DDoS viene avviato da una certa fonte, il traffico di attacco può essere bloccato in varie fasi del percorso del

pacchetto. "Idealmente, gli attacchi DDoS sono mitigati vicino all'attaccante e la mitigazione riguarda solo il traffico dannoso". D'ora in poi chiameremo queste fasi "livelli di mitigazione". L'obiettivo di questo capitolo è fornire una chiara panoramica dei diversi metodi e strumenti di mitigazione. In questo capitolo osserveremo innanzitutto i livelli di mitigazione (2.3.1), elencheremo i livelli Internet e vedremo quali metodi di mitigazione possono essere applicati. Questi metodi di mitigazione saranno spiegati in seguito in 2.3.2. Per ogni metodo, spiegheremo come viene mitigato il traffico DDoS, nonché un elenco di strumenti con cui questi metodi possono essere applicati nella pratica.

2.3.1 Livelli di Mitigazione

In primo luogo, discuteremo i diversi livelli a cui il traffico DDoS può essere mitigato. La Figura 2.2 rappresenta i livelli del percorso in cui è possibile rilevare e mitigare il traffico DDoS. Essi sono rispettivamente il livello dell'attaccante, della botnet, del reflector, dell'IXP, dell'ISP, dell'organizzazione e della macchina target. Per ogni livello, verranno elencati i layer di Internet utilizzati e i metodi di mitigazione che possono essere applicati.

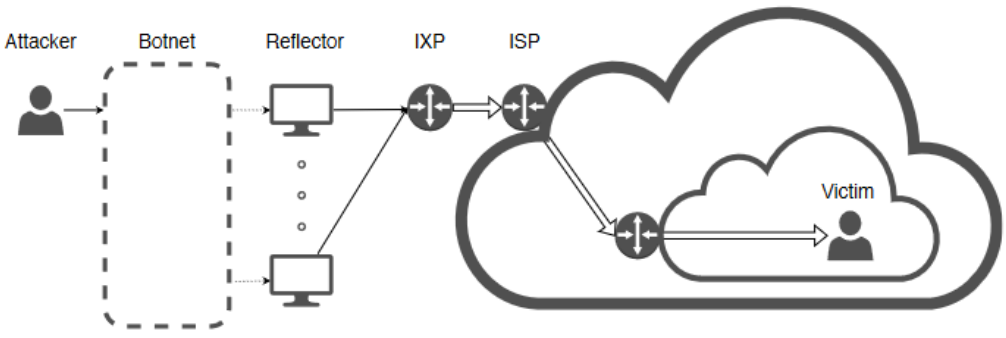

Figura 2.2: Percorso di un pacchetto di attacco DDoS

- **Livello Attaccante**

  La mitigazione degli attacchi DDoS al livello attaccante è possibile. Tuttavia, questo sarebbe un lavoro per ad esempio la polizia, e non rientra nell'ambito di questa ricerca.

- **Livello Botnet**

  La mitigazione a questo livello richiede di impedire l'uso di una botnet. Ciò esula dall'ambito di questa ricerca.

- **Livello Reflector**

  La mitigazione a questo livello richiede di impedire che i reflector vengano utilizzati in un attacco DDoS. Ciò esula dall'ambito di questa ricerca.

- **Livello IXP**

  Prima che il pacchetto entri in un autonomous network, di solito passa attraverso un Internet Exchange Point (IXP). A questo livello, le opzioni di mitigazione del traffico in entrata sono relativamente limitate, in parte a causa del recente dibattito sulla Net Neutrality [3], in cui ci si chiede se un IXP debba avere la possibilità di filtrare il traffico in base al suo contenuto. Anche se non rientra di questa ricerca, è importante che questo argomento sia tenuto in considerazione quando si dotano gli IXP di questa capacità.

Layer Internet	**Metodi di Mitigazione**
Data Link	Blackholing
Network	

- **Livello ISP**

  A livello ISP, potrebbero essere utilizzati tutti i metodi di mitigazione (elencati al punto 2.3.2). Tuttavia, questo in genere non viene fatto, poiché ciò comporterebbe costi elevati per l'utente finale. Inoltre, c'è ancora la questione della Net Neutrality; potrebbe non essere auspicabile che un ISP veda i dettagli del traffico Internet e lo filtri in base a queste informazioni.

Layer Internet	Metodi di Mitigazione
Data Link	Blackholing
Network	Intrusion Detection Systems
Transport	Network firewall
Application	Web Application Firewall

- **Livello Organizzazione**

  A livello organizzazione, in cui si trova il computer target (ad esempio la rete domestica), ci sono molte opzioni per mitigare il traffico DDoS. La mitigazione in questa potrebbe essere meno efficace, poiché l'hardware non è in grado di gestire un throughput così grande come quello di un edge router o di un ISP. Tuttavia, gli strumenti di mitigazione a livello di organizzazione consentono una maggiore granularità rispetto al livello IXP e ISP.

Layer Internet	Metodi di Mitigazione
Network	Intrusion Detection Systems
Transport	Network firewall

- **Livello Macchina Target**

  Sul computer del bersaglio dell'attacco DDoS stesso, è anche possibile bloccare il traffico DDoS.

Layer Internet	Metodi di Mitigazione
Network	Intrusion Detection Systems
Transport	Web Application Firewall
Application	

Per illustrare ulteriormente la differenza tra i livelli, la Figura 2.3 mostra i suddetti livelli di mitigazione e i corrispondenti layer Internet. Allo stesso modo, la Figura 2.4 mostra i metodi di mitigazione che possono essere utilizzati per ciascun livello di mitigazione.

Internet layer	Levels			
	IXP	ISP	Organization	Victim
7 - Application		✓		✓
6 - Presentation				
5 - Session				
4 - Transport		✓	✓	✓
3 - Network	✓	✓	✓	✓
2 - Data link	✓	✓		
1 - Physical				

Figura 2.3: Ogni livello di mitigazione e i livelli Internet in cui può essere applicato

Mitigation methods	Levels			
	IXP	ISP	Organization	Victim
Blackholing	✓	✓		
Intrusion Detection Systems		✓	✓	✓
Network firewall		✓	✓	
Web Application Firewall		✓		✓

Figura 2.4: Ogni livello e i metodi di mitigazione che possono essere utilizzati

2.3.2 Metodi e Strumenti di Mitigazione

Per ogni livello di mitigazione nella Figura 2.3, abbiamo elencato i livelli di mitigazione e i metodi di mitigazione del traffico a quel livello. Nella parte successiva, approfondiremo ogni metodo ed elencheremo i vari strumenti che possono essere utilizzati.

- **DNS Redirection**

  Nel DNS redirection [44], un provider di mitigazione maschera l'indirizzo IP della macchina target come uno del provider stesso. Tutto il traffico viene inviato al provider di mitigazione, che può filtrare il traffico dannoso prima di rinviarlo al target. Tuttavia, poiché questo metodo utilizza un'alterazione del resolver DNS, funziona solo a livello di applicazione.

Pertanto, quando si verifica un attacco diretto all'origine, l'IP della macchina target può ancora essere individuato e quindi bersagliato con traffico DDoS. Per questo motivo, la DNS redirection non è più molto utilizzata.

- **Blackholing**

  Il blackholing [18] è un metodo di mitigazione contro gli attacchi DDoS. Il traffico di rete viene instradato altrove, verso un "buco nero". Ciò significa che tutto il traffico instradato verso il black hole viene interrotto. Ciò può avere conseguenze diverse a seconda del protocollo utilizzato. Quando si utilizza il protocollo TCP, viene inviata una notifica alla fonte per informare che il traffico è stato interrotto, mentre nei protocolli senza connessione, come UDP, questo non avviene. Esistono due tipi di black hole: un Destination Remotely-Triggered Black Hole (D/RTBH) e un Source Remotely-Triggered Black hole (S/RTBH). Entrambi portano essenzialmente a un percorso nullo che interrompe il traffico, ma la differenza tra i due è il traffico che viene filtrato.

  In un D/RTBH, il traffico diretto verso una determinata destinazione viene bloccato. Se un dispositivo sta subendo un attacco DDoS, è possibile impostare un D/RTBH che filtra tutto il traffico con l'IP della macchina target come destinazione. Tuttavia, questo blocca anche tutto il traffico normale che è destinato alla macchina, e ciò significa che l'attacco DDoS è parzialmente riuscito.

  Un modo più favorevole di fare black holing potrebbe essere un S/RTBH. In questo caso, il traffico proveniente da un determinato indirizzo IP di origine viene eliminato. Questo è utile in un attacco DoS non distribuito, poiché il traffico legittimo proveniente da altre fonti non viene bloccato. Tuttavia, nella maggior parte dei moderni attacchi DDoS il traffico proviene da molte fonti diverse. Un tipico attacco DDoS può avere circa 10.000 IP di origine, rendendo impraticabile il blackhole di ciascuno di essi.

  Si può notare subito che il blackholing non è la soluzione ottimale per

mitigare gli attacchi DDoS, poiché in molti casi il traffico legittimo viene bloccato. Tuttavia, è ancora un'opzione ampiamente disponibile per le organizzazioni o gli individui che non hanno accesso ai moderni strumenti di mitigazione DDoS. Può essere utile anche quando l'obiettivo dell'attacco è una macchina o un sito più piccolo che fa parte di una rete più grande. In questo caso, il blackholing del target può impedire che le altre macchine della rete siano colpite dall'attacco.

Tools:

- BGP Flowspec

- **Intrusion Detection Systems**

  Un modo comunemente usato per mitigare il traffico DDoS è l'uso di Intrusion Detection Systems (IDS). Esistono due tipi di IDS: quelli che si basano su informazioni di audit raccolte dagli host della rete che cercano di proteggere e quelli che operano 'stand-alone' osservando il traffico di rete direttamente e passivamente, utilizzando un filtro dei pacchetti. Per semplificare, un IDS si concentra sul traffico di rete in entrata (Network-based IDS (NIDS)) o sul computer dell'host stesso (Host-based IDS (HIDS)).

  Un NIDS viene in genere installato in punti strategici della rete, dove è in grado di rilevare e monitorare efficacemente il traffico diretto a tutti i dispositivi della rete. Un NIDS legge tutti i pacchetti in entrata alla ricerca di eventuali pattern sospetti. Quando vengono rilevate minacce, in base alla loro gravità, il sistema è in grado intervenire, ad esempio avvisando gli amministratori o impedendo all'indirizzo IP di origine di accedere alla rete.

  L'altro tipo di IDS, come già detto, è chiamato Host-based IDS. Come suggerisce il nome, un HIDS è installato sul sistema stesso, in grado di monitorare e analizzare costantemente lo stato della macchina, consentendo di rilevare le modifiche a tale stato. Quando un intruso tenta di ottenere il controllo della macchina, generalmente lascia una

traccia (ad esempio, installando un key logger, installando malware per una botnet, ecc.). Se tale intruso supera un NIDS installato nella rete, l'HIDS ha la possibilità di rilevare lo stato modificato della macchina. In genere gli HIDS lavorano con un database di oggetti di sistema che devono monitorare. Genera una checksum di questi dati, consentendo di verificare facilmente se lo stato del sistema è stato modificato.

NIDS Tools:	**HIDS Tools:**
Snort [35]	OSSEC [20]
Suricata [19]	AIDE [2]
Bro [33]	
IBM QRadar [23]	

Esistono anche sistemi ibridi che combinano i due approcci per fornire una soluzione completa [38] [41] [40].

Oltre al fatto che gli IDS possono essere installati sulla rete o sull'host stesso, esiste un modo diverso di classificare gli IDS. In questo caso, variano nel modo in cui sono implementati e rilevano il traffico. Abbiamo sistemi basati sulle firme e sulle anomalie. I sistemi basati sulle firme e sulle anomalie sono simili in termini di funzionamento concettuale e di composizione. Le principali differenze tra queste metodologie sono insite nei concetti di 'attacco' e 'anomalia'. Un attacco può essere definito come "una sequenza di operazioni che mette a rischio la sicurezza di un sistema". Un'anomalia è solo "un evento sospetto dal punto di vista della sicurezza".

Un IDS basato sulle firme cerca pattern specifici, come sequenze di byte nei file o nel traffico di rete. Questi metodi di rilevamento possono essere applicati sia agli HIDS che ai NIDS. In un HIDS, lo strumento esegue una scansione dei file di log e di configurazione alla ricerca di modifiche. In un NIDS, lo strumento analizza le checksum dei pacchetti di rete. Inoltre, un NIDS basato sulle firme dispone generalmente di un database di firme che rappresentano pacchetti dannosi. Poiché molti hacker utilizzano gli stessi

strumenti per raggiungere il loro obiettivo questi strumenti generano sempre le stesse firme di traffico. Questo rende un IDS basato sulle firme un modo adatto per rilevare le azioni dannose. Lo svantaggio degli IDS basati sulle firme è l'impossibilità di rilevare attacchi zero-day, dal momento che non esistono ancora firme.

Questo inconveniente degli IDS basati sulle firme può essere risolto utilizzando un IDS basato sulle anomalie. In questo caso, l'approccio consiste nel classificare il comportamento del traffico. Ciò può essere fatto, ad esempio, utilizzando tecniche di machine learning per costruire un modello che rappresenti il comportamento "normale" del traffico e confrontare qualsiasi nuovo traffico con questo modello. In questo modo è possibile rilevare un'"anomalia", in quanto il traffico non rientra nel modello di traffico normale e viene quindi classificato come comportamento dannoso. Gli IDS basati sulle anomalie possono essere applicati sia agli HIDS che ai NIDS. In un HIDS, un'anomalia potrebbe essere rappresentata da un numero elevato di tentativi di accesso ripetuti e falliti, che suggeriscono che un hacker sta cercando di craccare una password. In un NIDS, un'anomalia può essere qualsiasi pattern di traffico di rete che non corrisponde al pattern di comportamento "buono". Lo svantaggio degli IDS basati sulle anomalie è che esiste la probabilità di soffrire di falsi positivi; il traffico legittimo, precedentemente sconosciuto, potrebbe essere classificato come anomalia perché non corrisponde al modello di machine learning.

- **Network Firewall**

  Un firewall di rete, comunemente chiamato anche packet filter, viene utilizzato per filtrare il traffico di rete in entrata. Viene stabilito un set di regole, in base al quale il firewall permette o nega i pacchetti in arrivo in base a tale set di regole. Un firewall di rete ha regole predefinite, ma l'amministratore di sistema può definire regole arbitrarie. Un firewall in genere crea una barriera tra una rete interna fidata e una rete esterna

non fidata, come Internet.

Tools:

- IPtables [31]
- Berkeley Packet Filter [29]

- **Web Application Firewall**

  Un Web Application Firewall (WAF) è una misura di sicurezza delle applicazioni distribuita tra un client Web e un server Web. Quando un WAF è attivo, esegue un'ispezione del traffico HTTP in entrata e in uscita. Si differenzia da un normale firewall in quanto è in grado di filtrare il contenuto di specifiche applicazioni web, mentre un firewall di rete funge da barriera di sicurezza tra i server. Un WAF è generalmente in grado di prevenire gli attacchi che hanno origine da falle nella sicurezza delle applicazioni web (ad esempio, SQL-Injection e Cross-Site Scripting (XSS)).

  Tools:

  - ModSecurity [34]

Confrontando i metodi e gli strumenti sopra citati, lo strumento più interessante da riconoscere è BGP Flowspec. Il motivo è che può essere applicato a livello di IXP e ISP, dove passa molto traffico (quindi ha una grande capacità di throughput). Inoltre, BGP Flowspec consente una maggiore granularità rispetto al normale blackholing, che è il modo in cui la mitigazione viene attualmente effettuata a questi livelli.

CAPITOLO 3

BGP FLOWSPEC

Dopo aver acquisito le conoscenze sugli attacchi DDoS, sulle fasi e sui metodi di mitigazione, ci concentreremo sull'argomento: BGP Flowspec. Si tratta di uno strumento che può essere utilizzato per la mitigazione dei DDoS a livello di IXP e ISP, come descritto nel capitolo precedente. In questo capitolo, l'obiettivo è quello di acquisire conoscenze su BGP Flowspec e capire come e perché può essere uno strumento efficace per mitigare il traffico DDoS. In primo luogo, verranno fornite informazioni di base in 3.1. In seguito, approfondiremo le limitazioni di BGP Flowspec in 3.2. Infine, descriveremo l'impatto di mitigazione che questo strumento può avere sulla rete sottostante, poiché è possibile bloccare involontariamente il traffico legittimo. Questo aspetto sarà discusso nel paragrafo 3.3.

3.1 Introduzione

Il Border Gateway Protocol (BGP) [22] è il protocollo di routing più importante di Internet, in quanto garantisce la comunicazione tra autonomous network. Esempi di autonomous network che comunicano a questo livello sono gli ISP. Un ISP può scegliere un protocollo di routing interno (come

OSPF o Routing Information Protocol), ma la comunicazione tra autonomous network avviene sempre tramite BGP. In BGP, due router possono diventare reciprocamente "peer" l'uno con l'altro quando iniziano una sessione di comunicazione. Questa sessione viene impostata con TCP e configurata manualmente. Ogni 60 secondi viene inviato un messaggio di keep-alive per mantenere la connessione.

BGP Flowspec è un'estensione del protocollo di routing BGP [28]. La sua caratteristica è quella di consentire il filtraggio del traffico di rete tra un gran numero di router peer BGP. A differenza, ad esempio, del blackholing (in cui tutto il traffico da o verso un determinato host viene eliminato), BGP Flowspec consente un approccio molto più granulare. Permette di costruire regole che corrispondono a un flusso di rete definito offrendo 12 parametri. Tra questi vi sono, ad esempio, l'IP di origine/destinazione, la lunghezza del pacchetto e le flag. Tutti e 12 i parametri sono descritti nella Figura 2.5. I router ai margini della rete di un ISP possono applicare le regole Flowspec in qualsiasi momento. Inoltre, quando arriva del traffico che soddisfa le regole Flowspec, il router può eseguire una delle 3 azioni seguenti:

- **Interrompere completamente il traffico**
- **Reindirizzare il traffico altrove per l'analisi**
- **Consentire il traffico ad un rate ridotto**

I 12 parametri di BGP Flowspec sono definiti come Network Layer Reachability Information (NLRI). Qualsiasi set di NLRI (come definito nella Figura 2.5) definisce un flusso di rete, ossia un gruppo di pacchetti di traffico che possono essere raggruppati ed etichettati di conseguenza. Un pacchetto di rete in arrivo è considerato conforme alla specifica del flusso quando corrisponde a tutti i componenti della specifica. Quando ciò si verifica, viene eseguita una delle 3 azioni sopra descritte.

BGP Flowspec è stato proposto come standard specificato nella RFC 5575 nell'agosto 2009 [28].

NLRI type	QoS match fields	Description	Example value
Type 1	Destination address	Defines the destination prefix to match.	130.89.161.0/24
Type 2	Source address	Defines the source prefix to match.	130.89.161.0/24
Type 3	IP Protocol	Contains a set of {operator, value} pairs that are used to match the IP protocol value byte in IP packets.	1, 3, 5, 17-19
Type 4	Source or destination port	Defines whether TCP, UDP or both will be packets will be influenced	1-80, 443
Type 5	Destination port	Defines the destination port that will be influenced by Flowspec	1-80, 443
Type 6	Source port	Defines the source port that will be influenced by Flowspec	1-80, 443
Type 7	ICMP type	Any (range of) ICMP types	0, 3-5
Type 8	ICMP code	Any (range of) ICMP codes	3, 6-15
Type 9	TCP flags	Any amount of TCP flags	ACK, FIN, PUSH, SYN
Type 10	Packet length	Match on the total IP packet length (excluding Layer 2 but including IP header)	40, 255-1518
Type 11	DSCP	Match on the Class Of Service flag	40, 255-1518
Type 12	Fragmentation bits	Any amount of IP fragmentation flags	dont-fragment, is-fragment

Figura 3.1: Definizione delle tuple BGPFlowspec

3.2 Limitazioni

Vi sono alcune limitazioni pratiche di BGP Flowspec che è importante affrontare. Queste limitazioni derivano dall'implementazione dello standard BGP Flowspec e dall'hardware utilizzato.

Esaminando vari dataset esistenti sugli attacchi DDoS, si può notare che un attacco DDoS ha spesso origine da più fonti. Ad esempio, i dati recuperati da DDoSDB [36] dimostrano che molti attacchi hanno origine da ben 10.000 indirizzi IP di origine diversi. Secondo lo standard BGP Flowspec, l'NLRI di tipo 2 può essere utilizzato per definire un flusso in base a un prefisso di origine. Tuttavia, una regola BGP Flowspec può definire solo un prefisso sorgente. Se vogliamo generare un set di regole che definisca con successo questo attacco DDoS basandosi solo su questo campo NLRI, dovremmo generare 10.000 regole BGP Flowspec. Questo non è scalabile per attacchi più grandi.

Inoltre, ci sono limitazioni sull'hardware che i fornitori forniscono. Ad esempio,

Cisco, uno dei principali fornitori di hardware di rete, ha una linea di hardware progettata per il routing a livello di IXP che utilizza BGP per comunicare con altre reti. Il sistema operativo utilizzato su questi router (IOS XR) prevede un limite di 3000 regole BGP Flowspec. Ricordando l'esempio precedente di un attacco DDoS che proviene da 10.000 indirizzi IP di origine diversi, non è possibile applicare le regole BGP Flowspec per definire questo attacco basandosi solo sugli IP di origine.
Infine, l'implementazione di una configurazione BGP Flowspec è impegnativa, poiché tutti i fornitori di hardware che la supportano (ad es. Cisco, Juniper, Huawei) hanno un linguaggio di configurazione diverso per implementare le regole. Ciò rende difficile la generazione automatica delle regole e il loro scambio.

3.3 Impatto di mitigazione

Per un ISP che applica le regole BGP Flowspec, è molto importante valutare l'impatto che queste regole possono avere sulla rete. L'impatto di mitigazione implica l'impatto negativo che una regola BGP Flowspec ha sulla rete dell'ISP. Questo vale per tutti gli host della rete, compreso l'host di destinazione del traffico DDoS a cui è rivolta la regola Flowspec. BGP Flowspec consente il filtraggio di 12 campi nell'header IP, con una granularità relativamente bassa rispetto, ad esempio, ad un firewall. Per questo motivo, è importante che una regola Flowspec abbia il minor impatto possibile sulla rete. L'impatto è definito da una serie di fattori. Uno di questi fattori è il numero di falsi positivi nei pacchetti filtrati, ad esempio il traffico benigno che viene comunque filtrato dalla regola Flowspec. Questo ha un impatto negativo sulla rete, in quanto il traffico non-DDoS verrà eliminato dai router di bordo dell'ISP.
Una delle sfide principali è ridurre al minimo l'impatto della mitigazione sulla rete. Per risolvere questa sfida, è necessario misurare questo impatto in qualche modo, al fine di quantificarne l'entità. Quando è possibile

quantificare l'impatto della mitigazione, è anche possibile costruire un sistema che auto-distribuisca le regole Flowspec, quantifichi il loro impatto e aggiusti le regole in base alle esigenze della rete. Per esempio, se una serie di regole misura un impatto superiore a una certa soglia, una o più regole possono essere omesse dall'insieme per ridurre l'impatto.

CAPITOLO 4

STATO DELL'ARTE

Al fine di raggiungere l'obiettivo di questo documento, si è visto necessario lo studio della letteratura presente allo stato dell'arte relativa all'obiettivo di questa ricerca. Dopo un'analisi delle pubblicazioni trovate, ci si è focalizzati su dieci in particolare, elencate ed esplicate di seguito:

4.1 Bakker

Bakker [1] propone una soluzione in grado di generare delle regole per BGP Flowspec partendo da:

- Fingerprint di attacchi conosciuti
- Priorità dell'ISP
  - Ovvero pesi da dare a tre relativi criteri, con l'obiettivo di privilegiare determinati aspetti piuttosto che altri
    * Impatto sull'Utente Finale
    * Efficacia
    * Dimensione Fisica delle Regole

La generazione di queste regole è basata sulla *compressione* degli indirizzi sorgenti in prefissi, che possano quindi limitare l'utilizzo di spazio all'interno dell'hardware.

4.2 Jonker et al.

Mattijs Jonker, Aiko Pras, Alberto Dainotti e Anna Sperotto [25] propongono uno studio critico più approfondito sulla mitigazione di attacchi DDoS basati sul BGP Blackholing.
Dalla loro analisi infatti emerge come il meccanisco di difesa del BGP Blackholing si riveli estremamente rapido nel proteggere le reti attaccate. D'altronde, però, ci sono alcune tipologie di Blackholing che si rivelano durare più a lungo, e quindi avere effetti peggiori, rispetto agli attacchi originari.
Essi terminano la loro pubblicazione sostenendo che per comprendere a pieno l'impatto del Blackholing vi è la necessità di più dati, tra cui la misurazione in tempo reale dei DNS. Questo potrebbe portare ad un analisi efficace di quali servizi sono intaccati dall'attacco (che spesso non mira a colpire tutta la rete), e poter pensare di procedere con una sostituzione di IP (per cercare di manternere questi servizi online).

4.3 Dietzel et al.

Dietzel, Wichtlhuber, Smaragdakis e Feldman [15] propongono una soluzione avanzata di Blackholing, con una granularità molto più fine. Si avvalgono, tra le varie cose, di Flowspec come strumento per ottenere ciò, oltre a sottolineare come gli ISP dovrebbero fidarsi e cooperare l'un l'altro: *"Flowspec, se lanciato su ambienti inter-dominio, è una forma di mitigazione molto popolare, basata sulla fiducia, sulla cooperazione e sulla condivisione di risorse, ma questo è difficile da raggiungere su reti che utilizzano diverse risorse e che hanno politiche e strategie di business diverse l'un l'altra".*
Sostanzialmente la loro ricerca enfatizza come il problema sia la non

condivisione delle risorse di un ISP per la risoluzione di problemi di un altro ISP.
Ragion per cui propongono una soluzione a livello IXP, chiamata Stellar, che sostengono sia molto rapida ed efficace, ma soprattutto scalabile anche ad attacchi di grossa potenza. Essa fornisce, tra le varie cose, anche una telemetria del traffico durante gli attacchi. La soluzione è ancora in sviluppo e l'obiettivo è quella di venderla a quanti più IXP possibili

4.4 Jamous et al.

Jamous, Soltani, Sangduyu e Li [24] propongono un sistema capace di rilevare e diversificare il traffico malevolo su una rete quasi in tempo reale. D'altronde, però, questa ricerca non si focalizza abbastanza sull'impatto finale sulla rete, che si rivela essere molto rilevante. Anzi, non propone BGP Flowspec come soluzione auto-adattiva, questo può significare che determinate regole che impattano in maniera negativa la rete non vengono rilevate e modificate per risolvere il problema.

4.5 Serodio

Serodio [39] discute delle cosidette IDMS, *Intelligent DDoS Mitigation System*, attualmente la seconda tecnica di protezione per attacchi DDoS più utilizzata.
Essa è capace di mitigare anche complessi attacchi a livello applicativo. Essa si basa sull'utilizzo di una risorsa condivisa che ha il compito di ricevere tutto il traffico anomalo riscontrato dalla rete, e, nel caso in cui superi i controlli, ridirezionarlo alle destinazioni originarie.
Tipicamente questo processo avviene con dei prefissi BGP segnalati sulla Global Routing Table, oltre ad un tunneling (o un dominio di routing alternativo) per ridirezionare il traffico pulito verso le destinazioni originarie,

senza correre il rischio di loop.
L'utilizzo di BGP Flowspec in questa soluzione viene ritenuto utile per diversi motivi, tra cui:

- Nessun Cambiamento nella Global Routing Table
  - Le segnalazioni avvengono solo sulla rete interessata, senza andare ad intaccare la GRT
- Nessuna necessità di Tunneling
  - Il traffico pulito viene inviato alla GRT che poi lo indirizzerà alle destinazioni originarie
- Granularità più fine
  - Permette di poter specificare in maniera diretta più parametri, come protocollo, porte, prefissi sorgente e destinazione.

4.6 Nawrocki et al.

Nawrocki, Blendin, Dietzel, Schmidt e Wählisch [30] analizzano i risultati di una ricerca basata sull'utilizzo di BGP Blackholing a livello di punto di scambio europeo. La ricerca si basa su misurazioni della durata di tre mesi, considerando non solo il comportamento dei sistemi automatici di rilevazione e segnalazione, ma anche le analisi delle reti interessate.
La loro ricerca evidenzia come diverse pratiche imperfette, se migliorate, possano migliorare la raggiungibilità dell'infrastruttura di Internet.
La discussione tratta anche analisi sui danni introdotti dall'RTBH (*Remotely Triggered Black Hole*), e della difficoltà nell'avere misure intrinseche concrete, dal momento che, basandosi su pacchetti generici (gli originali sono stati trattati per questioni di privacy), solo una piccola parte di questi poteva essere identificata prima e durante una mitigazione di un attacco DDoS, di fatto rendendo impossibile una quantizzazione dei danni collaterali.

4.7 Dimolianis et al.

Dimolianis, Kalogeras, Kostopoulos e Maglaris [16] propongono un framework per la protezione di attacchi DDoS per infrastrutture interconnesse. Il loro approccio si basa sul paradigma di Federal Learning per la rilevazione collaborativa degli attacchi, nel rispetto della privacy degli utenti, senza quindi richiedere informazioni sensibili alle varie entità.
La mitigazione si basa su un Firewall conforme allo standard VNF, scalabile e programmabile a seconda delle necessità delle vittime. Essa, nel dettaglio, va ad analizzare le firme dei pacchetti in un certo lasso di tempo. Queste vengono poi date come input ad un modello di Machine Learning basato, come anticipato, sul Federal Learning.
A questo punto il traffico segnalato come malevolo viene reindirizzato a dei firewall, e in casi di attacchi molto grossi, segnalano delle richieste di blocco a dei domini *amici* per bloccare il traffico prima che arrivi alla vittima.
Questo framework è stato testato soprattutto per gli attacchi di tipo DNS Amplification, attualmente ritenuti tra i più devastanti.

Un loro ulteriore lavoro [17], con la cooperazione di Pavlidis, propone un framework per la mitigazione di attacchi multi-vettore tramite distribuzione di regole di Access Control.
Il loro lavoro si basa sulla definizione di *livelli di sicurezza* determinati da tutti i device presenti nel percorso di attacco.

4.8 Sriram e Montgomery

Sriram e Montgomery [43] hanno concentrato all'interno di questa pubblicazione una guida che porta raccomandazioni e linee guida per l'installazione di protocolli e tecnologie per la protezione e mitigazione di attacchi inter-dominio. Queste riducono appunto il rischio di arrecare danni con una configurazione approssimativa ed errata delle soluzioni che si vanno ad implementare, ed aiutano, inoltre, ad individuare e prevenire lo spoofing di

indirizzi IP che possano poi portare ad attacchi DDoS.
Queste raccomandazioni coinvolgono principalmente protocolli e tecnologie utilizzati sui router BGP.

4.9 Kock

Kock [26] all'interno della sua tesi da delle risposte a tre domande fondamentali, ovvero:

- Flowspec, teoricamente, come si comporta rispetto alle altre soluzioni di mitigazione degli attacchi DDoS presenti in commercio?
- Come si possono scrivere regole per BGP Flowspec basate sugli attacchi DDoS già noti?
- Quanto sono efficaci le nostre regole di BGP Flowspec?

Le sue conclusioni evidenziano come Flowspec sia uno strumento utile e potente, utilizzato principalmente dagli ISP e dagli IXP. Esso permette di bloccare il traffico con molta più granularità rispetto al classico blackholing. Kock, però, aggiunge che al momento c'è un *vuoto* per quanto riguarda la misurazione dell'impatto sull'utente di questa soluzione.
Per quanto riguarda la generazione di regole, Kock a partire dai dati forniti dal DDoSDB è riuscito ad elaborare un algoritmo che possa ridurre il numero di regole richieste per la definizione di un attacco DDoS, andando così ad avere un impatto migliore sulla rete.
In merito all'efficacia delle regole, Kock in seguito ad una simulazione in ambiente controllato evidenzia come, su un campione di 100 firme di attacchi, 90 di queste ultime abbiano ottenuto un punteggio sulla precisione superiore al 95%.
Secondo la sua valutazione, Flowspec lavora meglio con attacchi DDoS basati su protocollo UDP, rispetto agli altri. Avendo avuto risultati scadenti con attacchi di tipo TCP, egli ritiene che l'introduzione di queste regole abbia un impatto abbastanza medio.

Le sue critiche, infine, si basano sul fatto che le firme utilizzate per la generazione delle regole siano troppo generiche, e quindi non utilizzabili per questo scopo. Andando ad enfatizzare, ancora una volta, come gli attacchi DDoS, essendo molto differenti l'un l'altro, che una soluzione generica per questo problema è molto complicata, se non addirittura impossibile da progettare. Suggerisce quindi di considerare e valutare individualmente i singoli tipi di attacchi, per poi generare delle regole di BGP Flowspec.

4.10 Tabella Riassuntiva

Di seguito una tabella riassuntiva riguardo i lavori presenti allo stato dell'arte, con le features implementate e i relativi campi d'azione.
Dove le colonne:

- Paper
  - Indica la pubblicazione in oggetto
- BH (Blackholing)
  - Indica se la pubblicazione si basi sul Black Hole
- ABH (Advanced Blackholing)
  - Indica se la pubblicazione si basi sull'Advanced Black Hole
- ACL
  - Indica se la pubblicazione fa uso, o meno, delle ACL
- Mitigazione
  - Indica se la pubblicazione tratti, o meno, strategie di mitigazione
- BGPFS
  - Indica se la pubblicazione tratti, o meno, BGP Flowspec

- FSRG
  - Indica se la pubblicazione tratti, o meno, algoritmi di generazione di regole per BGP Flowspec
- MLD
  - Indica se la pubblicazione tratti, o meno, soluzione di Machine Learning per il Detect di Attacchi
- Livello
  - Indica su quale livello si basa la pubblicazione

Paper	**BH**	**ABH**	**ACL**	**Mitigazione**	**BGPFS**	**FSRG**	**MLD**	**Livello**
Bakker	✓			✓	✓	✓		ISP
Jonker	✓			✓	✓			ISP
Dietzel		✓		✓	✓			IXP
Jamous	✓			✓	✓			ISP
Serodio	✓			✓	✓			ISP
Nawrocki	✓				✓			IXP
Dimolianis 1				✓	✓		✓	ISP
Dimolianis 2			✓	✓				ISP
Sriram e Montgomery	✓			✓	✓			ISP
Kock	✓			✓	✓	✓		ISP

Tabella 4.1: Tabella Riassuntiva

CAPITOLO 5

PROGETTAZIONE DEL MODELLO

Dopo aver studiato le soluzioni attualmente utilizzate, nel capitolo precedente, ci siamo concentrati sulla creazione di uno modello generale che potesse essere utilizzato sia per l'identificazione che per la mitigazione degli attacchi DDoS, utilizzando BGP Flowspec. Abbiamo prodotto un modello per automatizzare l'intero processo di detection, mitigation ed inserimento delle regole sui router BGP per attuare la mitigazione. La Figura 5.1 mostra una panoramica semplificata dell'intero processo.

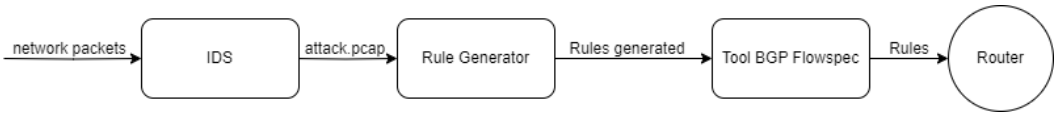

Figura 5.1: Processo di identificazione e mitigazione

Questo modello si divide in 3 parti: un IDS per rilevare gli attacchi, un generatore di regole BGP Flowspec per mitigare gli attacchi ed un tool BGP Flowspec che inserisce in modo automatico le regole create nei router BGP. Analizziamo in maniera più approfondita questi 3 componenti.

5.1 IDS

L'IDS ha il compito di rilevare gli attacchi in tempo reale. Esso riceve in input i pacchetti di rete ed identifica gli attacchi utilizzando tecniche di machine learning, così da rilevare anche possibili attacchi zero-day. Creando una baseline che rappresenti il comportamento normale di una rete è possibile controllare questa baseline per rilevare comportamenti anomali che possono essere frutto di possibili attacchi in corso. Utilizzando tecniche di machine learning possiamo definire se si tratta di una effettiva minaccia, ed in questo caso produciamo in output un file.pcap, che contenga l'attacco stesso, per darlo in input al generatore di regole.

5.2 Generatore di regole

Il generatore di regole prende in input il file.pcap, che contiene l'attacco in corso, dall'IDS, e genera un set di regole BGP Flowspec che possano essere utilizzare per mitigare efficacemente l'attacco stesso. Date le limitazioni di BGP Flowspec, sia sul numero di regole che possono essere inserite su un router BGP, sia sui differenti linguaggi che i produttori di router BGP utilizzano, il generatore avrà un modulo di parsing che permetterà di convertire le regole nei linguaggi utilizzati dai differenti fornitori. Le regole generate verranno utilizzare come input per il tool BGP Flowspec.

5.3 Tool BGP Flowspec

Il tool BGP Flowspec ottiene in input le regole generate per mitigare l'attacco rilevato, e permette di inserire automaticamente queste regole sui router BGP Flowspec in tempo reale. Questo componente ha lo scopo di sviluppare una strategia di contenimento distribuita e cooperativa per la diffusione delle regole sul traffico in tutta la rete a una velocità tale da garantire una reazione tempestiva e utile contro le minacce coinvolte, con un impatto minimo sui protocolli esistenti e sui componenti legacy.

CAPITOLO 6

BGP FLOWSPEC RULE GENERATION

Spiegato il modello generale, nel capitolo 5, questo lavoro di tesi si concentra in particolare sulla creazione del generatore di regole BGP flowspec. Ipotizzando che l'attacco sia già stato rilevato, utilizzeremo file .pcap degli attacchi DDoS esistenti come input al nostro generatore. Il processo di generazione delle regole è suddiviso in 3 parti: sezionare e riassumere le caratteristiche della rete di un attacco DDoS, generare il set di regole BGP Flowspec ed effettuare il parsing del set di regole nei diversi linguaggi dei sistemi operativi dei router.

La Figura 6.1 mostra una panoramica semplificata del processo di generazione delle regole. Si può notare che il processo è suddiviso in 3 componenti principali: il DDoS dissector, il generatore di regole ed il parser. In questo capitolo, ci concentreremo su ciascun componente e spiegheremo i dati utilizzati come input, il processo che si svolge e i dati di output generati.

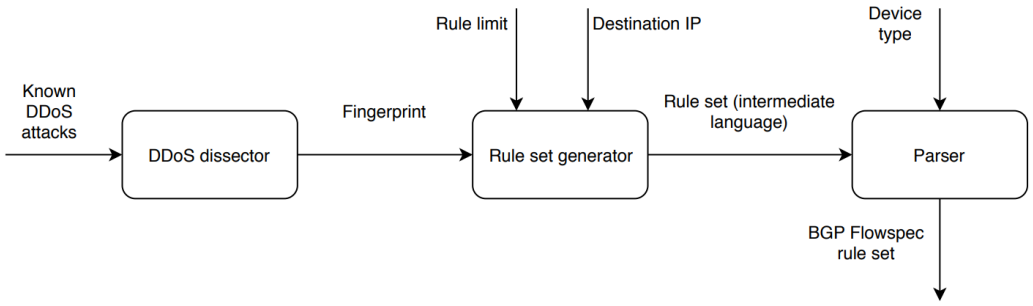

Figura 6.1: Processo di generazione regole BGP Flowspec

6.1 DDoS dissector

Il primo componente del processo di generazione delle regole è il DDoS dissector. Si tratta di una funzione responsabile della generazione di un recap delle caratteristiche della rete a partire dal traffico noto degli attacchi DDoS. Questo recap è chiamato "fingerprint" o firma dell'attacco. Va notato che il dissector è un modulo separato e fa parte di DDoSDB [36]. DDoSDB è una piattaforma progettata per aiutare le vittime di attacchi DDoS e la comunità accademica ad accedere alle informazioni sugli stessi attacchi. Raccoglie i dati da collaboratori (spesso vittime di attacchi DDoS) che hanno raccolto dati dagli attacchi DDoS che hanno subito.

Per questa ricerca, il dissector non è stato sviluppato dagli autori ed è quindi considerato correttamente funzionante. Pertanto, gli autori hanno solo apportato modifiche al dissector in base ai requisiti del generatore di regole. Per questo motivo è stato scelto il dissector su DDoSDB. Analizzeremo qui di seguito il dissector in modo più dettagliato.

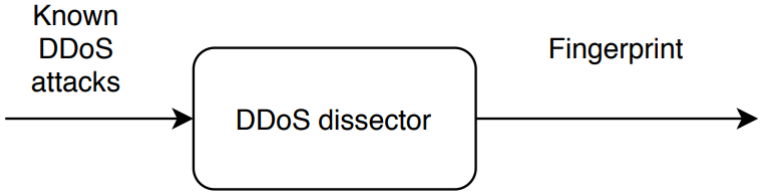

Figura 6.2: DDoSDB dissector

Va notato che l'uso di un dissector, come descritto in questa sezione, non è necessario per il nostro algoritmo di generazione delle regole. Il nostro algoritmo per la generazione delle regole BGP Flowspec richiede le caratteristiche di rete dell'attacco, che possono anche essere estratte manualmente dal traffico dell'attacco. Per quanto ne sappiamo, non esistono altri strumenti che estraggono automaticamente le caratteristiche di rete degli attacchi DDoS, pertanto, viene utilizzato per il nostro algoritmo di generazione delle regole.

6.1.1 Input

L'input del DDoS dissector è una raccolta di attacchi DDoS noti. Per questa ricerca, il traffico DDoS richiesto sarà recuperato da una repository di GitHub [21], che contiene una raccolta di packet captures di attachi DDoS reali. I dati presenti sulla repository sono pubblicati sotto forma di traccia dell'attacco. La traccia può essere letta in due diversi formati: pcap e pcapng. L'obiettivo che si pone il creatore di questa repository, fornendo queste catture, è quello di stimolare la crescita della ricerca sugli attacchi DDoS, in quanto le catture DDoS del mondo reale sono difficili da reperire e questo rende più difficile la ricerca.

6.1.2 Processo

Come già detto, il dissector è responsabile della generazione di una firma di un vettore di attacco. La firma è definita come una sintesi delle caratteristiche della rete di quel vettore di attacco. Il dissector inoltre anonimizza i dati della firma per proteggere l'identità delle vittime, il che significa che l'indirizzo IP di destinazione viene omesso. Inoltre, va notato che se in una traccia di attacco DDoS sono presenti più vettori di attacco, viene generata una firma contenente tutti i vettori di attacco appartenenti alla stessa traccia; questo è il caso di attacco multi-vector. Il codice sorgente del DDoS dissector è pubblicamente disponibile su GitHub [37].

6.1.3 Output

L'output del dissector è la firma in formato JSON. Questo rende i campi facilmente leggibili e facili da convertire in una regola BGP Flowspec. Di seguito è riportato un esempio semplificato di firma con l'ID univoco "05fe9a1689202cb5d0e2dd2e320826c5".


```
{
    "attack_vectors": [
        {
            "service": null,
            "protocol": "TCP",
            "fraction_of_attack": 0.447,
            "source_port": "random",
            "destination_ports": {
                "9069": 0.451,
                "9070": 0.225,
                "others": 0.324
            },
            "tcp_flags": {
                "..........S.": 0.945,
                "others": 0.055
            },
            "nr_packets": 364,
            "nr_megabytes": 0,
            "time_start": "2021-06-20T19:42:57.294010+00:00",
            "duration_seconds": 817,
            "source_ips": [
                "136.243.174.154",
                "163.158.248.5",
                ...
                ...
            ],
            "ethernet_type": {
                "IPv4": 1.0
            },
            "frame_len": {
                "74": 0.717,
```

```
                "60": 0.124,
                "66": 0.115,
                "others": 0.044
            },
            "fragmentation_offset": {
                "0": 1.0
            },
            "ttl": {
                "60": 0.451,
                "56": 0.239,
                "others": 0.31
            }
        },
        {
            "service": "FTP",
            "protocol": "TCP",
            "fraction_of_attack": 0.553,
            "source_port": 21,
            "destination_ports": {
                "21": 1.0
            },
            "tcp_flags": {
                ".......A..S.": 1.0
            },
            "nr_packets": 532,
            "nr_megabytes": 0,
            "time_start": "2021-06-20T19:42:57.409323+00:00",
            "duration_seconds": 818,
            "source_ips": [
                "75.136.225.254",
                "93.114.150.139"
            ],
            "ethernet_type": {
                "IPv4": 1.0
            },
            "frame_len": {
                "60": 1.0
```

```
            },
            "fragmentation_offset": {
                "0": 1.0
            },
            "ttl": {
                "236": 0.744,
                "245": 0.256
            }
        }
    ],
    "target": "Anonymized",
    "tags": [
        "TCP flood attack",
        "TCP",
        "TCP flag attack",
        "Multi-vector attack",
        "TCP SYN ACK flag attack"
    ],
    "key": "05fe9a1689202cb5d0e2dd2e320826c5",
    "time_start": "2021-06-20T19:42:57.294010+00:00",
    "time_end": "2021-06-20T19:56:35.453656+00:00",
    "duration_seconds": 818,
    "total_packets": 896,
    "total_megabytes": 0,
    "total_ips": 60,
    "avg_bps": 564,
    "avg_pps": 1,
    "avg_Bpp": 64
}
```


Questo JSON rappresenta informazioni su un attacco di tipo TCP flood con attacchi di tipo TCP flag attack e Multi-vector attack. Ecco una descrizione delle chiavi principali:

- **attack_vectors:** Questo è un array che contiene informazioni su diversi vettori di attacco. Ogni oggetto all'interno di questo array rappresenta un vettore di attacco specifico.

- **service:** Indica il servizio coinvolto nell'attacco. Nel primo attacco è nullo (null), mentre nel secondo attacco è "FTP" (File Transfer Protocol).

- **protocol:** Specifica il protocollo utilizzato per l'attacco. In entrambi gli attacchi è "TCP" (Transmission Control Protocol).

- **fraction_of_attack:** Indica la frazione di attacco attribuita al vettore di attacco specifico. Nel primo attacco è 0.447 (corrispondente al 44.7%), mentre nel secondo attacco è 0.553 (corrispondente al 55.3%).

- **source_port:** Indica la porta sorgente dell'attacco. Nel primo attacco è "random" (casuale), mentre nel secondo attacco è 21 (porta FTP).

- **destination_ports:** Questo oggetto contiene le informazioni sulle porte di destinazione coinvolte nell'attacco e la loro distribuzione percentuale. Nel primo attacco, la porta 9069 ha una percentuale di 0.451 (45.1%), la porta 9070 ha una percentuale di 0.225 (22.5%), mentre le "others" (altre porte) hanno una percentuale di 0.324 (32.4%). Nel secondo attacco, la porta di destinazione è solo la 21 (porta FPT) con una percentuale di 1.0 (100%).

- **tcp_flags:** Questo oggetto contiene informazioni sui flag TCP coinvolti nell'attacco e la loro distribuzione percentuale. Nel primo attacco, il flag ".....S.", ovvero syn, ha una percentuale di 0.945 (94.5%), mentre gli "others" (altri flag) hanno una percentuale di 0.055 (5.5%). Nel secondo attacco, ci sono due flag ".....A..S.", ovvero syn e ack, con una percentuale di 1.0 (100%).

- **nr_packets:** Indica il numero totale di pacchetti coinvolti nell'attacco per il vettore specifico.

- **nr_megabytes:** Indica il numero totale di megabyte coinvolti nell'attacco per il vettore specifico.

- **time_start:** Specifica il momento di inizio dell'attacco.

- **duration_seconds:** Indica la durata totale dell'attacco in secondi per il vettore specifico.
- **source_ips:** Questo array contiene gli indirizzi IP di origine coinvolti nell'attacco per il vettore specifico.
- **ethernet_type:** Indica il tipo di Ethernet coinvolto nell'attacco. In entrambi gli attacchi, è "IPv4" con una percentuale del 100% (1.0).
- **frame_len:** Questo oggetto contiene informazioni sulla lunghezza dei frame coinvolti nell'attacco e la loro distribuzione percentuale.
- **fragmentation_offset:** Indica l'offset di frammentazione coinvolto nell'attacco. In entrambi gli attacchi, l'offset è "0" con una percentuale del 100% (1.0).
- **ttl:** Questo oggetto contiene informazioni sul Time-To-Live (TTL) coinvolto nell'attacco e la sua distribuzione percentuale.
- **target:** Indica il bersaglio dell'attacco (è stato anonimizzato).
- **tags:** Una lista di etichette che descrivono l'attacco. Le etichette includono "TCP flood attack", "TCP", "TCP flag attack", "Multi-vector attack" e "TCP SYN ACK flag attack".
- **key:** Una chiave univoca che identifica l'attacco.
- **time_end:** Specifica il momento di fine dell'attacco.
- **total_packets:** Indica il numero totale di pacchetti coinvolti in tutti i vettori di attacco, che è 896.
- **total_megabytes:** Indica il numero totale di megabyte coinvolti in tutti i vettori di attacco.
- **total_ips:** Indica il numero totale di indirizzi IP coinvolti in tutti i vettori di attacco, che è 60.

- **avg_bps:** Indica la velocità media in bit al secondo dell'attacco, che è 564 bps.
- **avg_pps:** Indica il numero medio di pacchetti al secondo dell'attacco, che è 1 pps.
- **avg_Bpp:** Indica la dimensione media dei pacchetti in byte dell'attacco, che è 64 byte per pacchetto.

6.2 Generatore di regole

Il componente successivo è il generatore del set di regole basato sulla firma. Questo componente genera tutte le regole BGP Flowspec necessarie per specificare il flusso di attacco descritto nella firma.

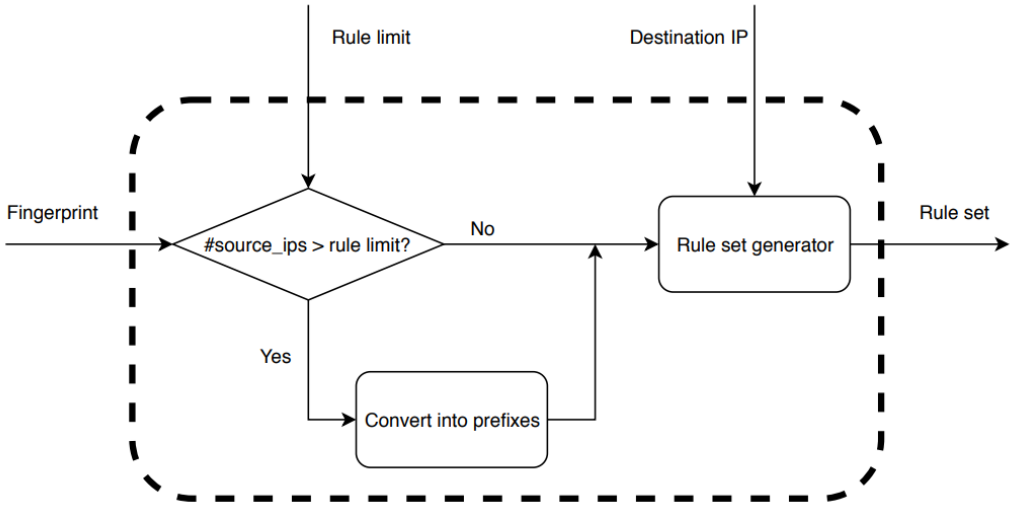

Figura 6.3: Generatore di regole

6.2.1 Input

Il modulo di generazione delle regole ha diversi valori di input: la firma generata dal dissector (in formato JSON), il numero massimo di regole che possono essere installate sul router in una sola volta e l'IP di destinazione per ogni regola (poiché non è presente nella firma).
Il limite delle regole del router viene utilizzato come input per questo modulo,

poiché i diversi produttori di hardware hanno limiti di regole diversi. Per esempio, Cisco, uno dei principali fornitori di hardware di rete, ha una linea di hardware progettato per il routing a livello di IXP. Il sistema operativo utilizzato su questi router (IOS XR) prevede un limite di 3.000 regole BGP Flowspec che possono essere installate in un determinato momento.
Esaminando vari dataset esistenti sugli attacchi DDoS, si può notare che un attacco DDoS ha spesso origine da molte fonti. Ad esempio, i dati recuperati dalla repository GitHub [21] mostrano che molti attacchi hanno origine da ben 10.000 IP diversi. Secondo lo standard BGP Flowspec [22], l'NLRI di tipo 2 può essere utilizzato per definire un flusso in base a un prefisso di origine. Tuttavia, una data regola BGP Flowspec può definire solo un prefisso sorgente. Ciò significa che un vettore di attacco con n indirizzi IP di origine richiede un set di regole di dimensioni n. Pertanto, il filtraggio di 10.000 indirizzi IP di origine non è fattibile sui suddetti router Cisco.
Esiste un'ulteriore documentazione su Junos OS [32], il sistema operativo utilizzato dai router Juniper. Qui viene specificato un limite di 6000 regole BGP Flowspec.
Per garantire la riproducibilità di questa ricerca, il limite di regole del router viene utilizzato come input generico. Va detto che è possibile ridurre il numero di regole combinando gli indirizzi IP in prefissi più grandi. Tecnicamente, utilizzando questo metodo, qualsiasi insieme di indirizzi IP può essere ridotto a qualsiasi dimensione. Tuttavia, questo aumenta l'impatto della mitigazione, poiché i prefissi più grandi hanno una maggiore probabilità di includere anche indirizzi IP legittimi.

6.2.2 Processo

Prima di generare il set di regole, è necessario affrontare la limitazione del numero di regole. Come già detto, in ogni regola è possibile specificare un solo prefisso di origine. Ciò significa che, quando l'insieme di IP sorgente è più grande del numero massimo di regole installabili, questo insieme di IP sorgente deve essere ridotto. Inoltre, c'è da ricordare che in ogni firma possono essere

presenti più vettori di attacco, quindi non possiamo assegnare lo stesso limite a tutti i vettori presenti, altrimenti si supererebbe il limite massimo di regole generate.

Spieghiamo questo concetto con un esempio: data una firma che contiene 5 vettori di attacco ognuno con 1000 indirizzi IP sorgente. Per contrastare questo attacco dobbiamo generare 5000 regole BGP diverse. Poniamo il caso di avere un numero massimo di 3000 regole, se diamo in input all'algoritmo che raggruppa gli IP in prefissi lo stesso limite di regole, quindi 3000, riusciamo a generare correttamente le 1000 regole sia per il primo vettore di attacco, sia per il secondo che per il terzo. Avendo raggiunto il numero massimo di regole possibili non riusciamo a contrastare il quarto ed il quindo attacco. Per questo motivo dobbiamo calcolare un numero massimo di regole generabili per il singolo vettore di attacco in modo da porterli bloccare tutti in maniera proporzionale ed efficace. Per superare questo ostacolo, dobbiamo tener conto della grandezza dell'attacco stesso e del numero di IP sorgenti per ogni attacco, calcolando un limite proporzionale per ogni vettore di attacco. Nell'esempio specifico, avendo un totale di 5000 IP sorgenti con un limite di 3000 regole BGP, dobbiamo assegnare ad ogni vettore di attacco un limite proporzionale di 600 regole. Sommando tutte le regole generate con questo nuovo limite riusciamo ad eguagliare il limite prefissato di 3000 e riusciamo a contrastare tutti e 5 i vettori di attacco.

Nell'esempio sopra citato, ogni vettore di attacco ha 1000 IP sorgente per facilitare la spiegazione, ma nello scenario reale, nel caso in cui i singoli vettori di attacco avessero un set di IP sorgente differente, verrano generate le regole in maniera proporzionale, così da contrastare loro equalmente, senza privilegiare uno rispetto che l'altro.

Per questo è stato creato un algoritmo che genera un limite massimo di regole diverso per ogni vettore di attacco presente nella firma, in proporzione alla quantità di indirizzi IP sorgente che esso contiene rispetto al totale dei vettori di attacco. Lo pseudocodice di questo algoritmo è mostrato di seguito:

Algoritmo: Calcolare i limiti percentuali per gli attacchi

INPUT: Firma dell'attacco *fingerprint*
INPUT: Limite massimo di regole *rule_limit*
INPUT: Numero totale di indirizzi IP *total_ips*
proportional_limits = lista vuota
if *total_ips* <= *rule_limit* **then**
OUTPUT: *rule_limit*
end if
while *attack_vector* in *fingerprint['attack_vectors']* **do**
Calcola la percentuale proporzionale *percentage_limit*
Aggiungi *percentage_limit* alla lista *proportional_limits*
end while
total_proportional_limit = somma di tutti gli elementi in *proportional_limits*
while *total_proportional_limit* > *rule_limit*
Decrementa il limite massimo percentuale del vettore con il valore massimo in *proportional_limits*
Decrementa *total_proportional_limit* di 1
end while
OUTPUT: Lista dei limiti percentuali calcolati *proportional_limits*

L'input dell'algoritmo consiste in tre elementi: la firma che contiene la lista di vettori di attacco, il limite massimo di regole desiderato ed il numero totale di indirizzi IP dell'intera firma. Inizialmente, controlla se il numero totale di indirizzi IP è inferiore o uguale al limite massimo di regole *rule_limit*. Se questo è il caso, poiché ci sono abbastanza regole disponibili per coprire tutti gli indirizzi IP, l'algoritmo restituisce direttamente il valore *rule_limit*, altrimenti l'algoritmo procede a calcolare le percentuali proporzionali per ciascun vettore di attacco. Dopo aver calcolato i limiti percentuali per tutti i vettori di attacco, l'algoritmo calcola la somma

totale di tutti i limiti percentuali e memorizza questa somma nella variabile *total_proportional_limit*. A questo punto, l'algoritmo esegue un ciclo che, controlla se il totale dei limiti proporzionali calcolati supera il *rule_limit*, e decrementa il valore totale delle percentuali finchè la somma di queste ultime non è più maggiore del limite massimo di regole desiderato. Alla fine dell'algoritmo, la lista *proportional_limits* conterrà i limiti percentuali corretti, in modo che la somma di tutti i limiti percentuali sia esattamente uguale al limite massimo di regole. Questa lista viene restituita come output dell'algoritmo.

A questo punto è possibile realizzare la riduzione in prefissi, e per far ciò è stato creato un algoritmo che prende in input un insieme di indirizzi IP e un numero massimo di regole e produce un nuovo insieme di prefissi IP con una dimensione inferiore alla quantità massima di regole. Lo pseudocodice di questo algoritmo può essere visto di seguito:

Algoritmo: Ridurre l'elenco degli IP di origine

INPUT: Insieme di indirizzi IP elenco ip *ip_list*
INPUT: Quantità massima di regole *max_rulex*
  current_prefix = 32
  Ordina *ip_list*
  result_list = *ip_list*
  while len(*result_list*) >*max_rules* **do**
    current_prefix = *current_prefix* - 1
    Converti *ip_list* in *current_prefix*
    if ci sono duplicati nella lista **then**
      Gruppa i duplicati e aggiungi alla *result_list*
    end if
  end while
OUTPUT: Insieme di prefissi IP che non supera il limite massimo di regole

L'algoritmo si comporta come segue: parte da un elenco di indirizzi IP e

da una quantità massima di regole. In primo luogo, ordina l'elenco di indirizzi IP in modo da percorrerlo in loop. Quindi, a ogni iterazione, controlla se la dimensione dell'elenco risultante è superiore alla quantità massima di regole. Se ciò è vero, l'insieme di IP deve essere ulteriormente ridotto. Il *current_prefix* inizia a /32 e conta alla rovescia. Alla prima iterazione, a /31, ogni indirizzo IP viene convertito in un prefisso /31. L'algoritmo controlla quindi questo elenco per verificare la presenza di duplicati. Se questi si verificano, gli indirizzi IP possono essere raggruppati in un singolo prefisso /31. I duplicati di questa iterazione vengono aggiunti all'elenco risultante. L'algoritmo continua a eseguire questo processo (riducendo la dimensione del prefisso corrente e raggruppando gli indirizzi) fino a quando la dimensione dell'elenco risultante è inferiore alla quantità massima di regole.
Il codice di questi due algoritmi è disponibile pubblicamente su GitHub [42].
È importante notare che l'esecuzione del suddetto algoritmo ha potenzialmente un risultato sull'impatto della mitigazione. Per illustrare questo aspetto, ricordiamo il seguente esempio: un'azienda con indirizzo IP 130.89.10.1 sta subendo un traffico di attacchi DDoS proveniente da oltre 10.000 IP. L'azienda dispone di un sistema che analizza il traffico in entrata e costruisce regole BGP Flowspec in base alle caratteristiche del traffico in entrata, in questo caso l'IP di origine. Tuttavia, poiché questo particolare attacco ha molti indirizzi di origine, potrebbero essere necessarie più di 10.000 regole Flowspec per definire correttamente un flusso di rete corrispondente al traffico DDoS. Come discusso nelle limitazioni (3.2), alcuni router possono gestire solo un massimo di 3000 regole BGP Flowspec installate. Per questo motivo, definire questo flusso di rete con gli IP di origine non è fattibile. Possiamo usare i prefissi nelle nostre regole per "raggruppare" più IP di origine insieme. Ad esempio, se 100 degli IP di origine iniziano con 50.62.24.xx, possiamo raggrupparli creando una regola Flowspec con campo di tipo 2 "50.62.24.0/24". In questa situazione, tutto il traffico proveniente da questo prefisso viene bloccato. Se l'azienda riceve un pacchetto dati legittimo dall'indirizzo IP 50.62.24.11, questo pacchetto dati soddisfa la condizione di trovarsi nel prefisso 50.62.24.0/24. Di conseguenza,

il pacchetto viene bloccato perchè viene classificato come parte del flusso di traffico DDoS e quindi viene eliminato all'ingresso nella rete dell'ISP. Sebbene questo algoritmo consenta di ottenere un set di regole più piccolo, può avere un impatto negativo sulla rete.
Successivamente, viene generato il set di regole. Ricordiamo i 12 campi NLRI che BGP Flowspec può utilizzare per filtrare il traffico. Questi campi sono elencati nella Tabella 6.1.

NLRI Type	**QoS fields**
Type 1	Destination prefix
Type 2	Source prefix
Type 3	Ip protocol
Type 4	Source or destination port
Type 5	Destination port
Type 6	Source port
Type 7	ICMP type
Type 8	ICMP code
Type 9	TCP flags
Type 10	Packet length
Type 11	DSCP
Type 12	Fragmentation bits

Tabella 6.1: Campi NLRI BGP Flowespec

Una regola BGP Flowspec è definita come un qualsiasi sottoinsieme dei 12 campi NLRI presentati in questa tabella. Come discusso in precedenza, all'interno di una regola è possibile utilizzare un'istanza di ciascun campo. Al fine di generare regole BGP Flowspec efficaci per mitigare gli attacchi DDoS, è necessario che l'insieme delle regole assomigli il più possibile al flusso di traffico dell'attacco DDoS. Questo riduce al minimo il rischio di blocco del traffico legittimo. Analizzeremo ogni campo NLRI e osserveremo se può essere

utilizzato per generare una regola utilizzando i dati delle firme generate dal DDoS dissector.

1. **Destination Prefix**

   Come già detto, l'indirizzo IP di destinazione nella firma generata dal DDoS dissector è anonimizzato per proteggere l'identità delle vittime. Ciò significa che l'IP di destinazione dell'attacco non è incluso nella firma. Tuttavia, in uno scenario reale, un ISP utilizzerebbe un set di regole BGP Flowspec per bloccare il traffico in entrata, il che significa che la destinazione del traffico si trova da qualche parte all'interno della rete dell'ISP. Pertanto, è ragionevole supporre che l'IP di destinazione sia noto al momento dell'applicazione del set di regole e possa quindi essere incluso in ogni regola. Anche se questo campo non è disponibile nei nostri dati di partenza, procederemo con l'ipotesi che questo dato sia disponibile, poiché questo sarebbe il caso in un ambiente reale, quindi, prendiamo il prefisso di destinazione come input per il nostro algoritmo di generazione di regole.

2. **Source Prefix**

   Come detto in precedenza, solo un'istanza di ciascun campo può essere inclusa in una regola BGP Flowspec. Gli attacchi DDoS spesso provengono da più fonti, il che richiede una regola per ogni IP di origine. Tuttavia, poiché questo campo specifica un prefisso di origine, è possibile raggruppare più indirizzi IP in un unico prefisso. In questo modo si otterrà un set di regole più piccolo, ma aumenterà il rischio di includere indirizzi IP legittimi nel flusso di traffico filtrato.

3. **IP protocol**

   Il protocollo IP dell'attacco è sempre costante per un vettore di attacco. Ciò significa che può essere essere sempre aggiunto a una regola BGP Flowspec senza costi aggiuntivi. Se sono necessarie più regole per definire un vettore di attacco, il protocollo IP può essere aggiunto a ogni regola.

4. **Source or destination port**

   La distribuzione delle porte utilizzate dipende dal tipo di attacco. Ad esempio, in un attacco DNS la porta di origine di un pacchetto è sempre 53 e la porta di destinazione è casuale. Il dissector analizza questo dato e genera un elenco di porte di origine e un elenco di porte di destinazione. Ne risultano quattro possibili casi di distribuzione delle porte:

   - **Uno-a-uno**, dove c'è una sola porta di origine e una sola porta di destinazione;
   - **Uno-a-molti**, dove c'è una sola porta di origine e più di una porta di destinazione;
   - **Molti-a-uno**, quando ci sono più porte di origine e una di destinazione;
   - **Molti-a-molti**, dove c'è più di una porta di origine e più di una porta di destinazione.

   Quando è stata eseguita l'analisi su ogni firma per osservare la distribuzione delle porte, si è notato che l'elenco delle porte di origine o di destinazione ha una lunghezza di 1. Inoltre, ogni volta che c'è più di una porta in uno dei due elenchi, l'elenco contiene molte porte distribuite in modo casuale. Poiché l'inclusione di un elenco di n porte richiederebbe un insieme di n regole, non è possibile includere un elenco di porte di tali dimensioni. Per questo motivo, l'elenco delle porte di destinazione sarà incluso solo se ha una lunghezza pari a 1.

   Lo standard BGP Flowspec consente il filtraggio delle porte con i tipi di NLRI 4-6, che consentono di specificare sia la porta di origine che quella di destinazione. Poiché gli ultimi due tipi annullano il campo di tipo 4, verranno utilizzati solo i tipi 5 e 6.

5. **Destination port**

   Come menzionato in precedenza, questo campo viene utilizzato solo se l'attacco DDoS è mirato a 1 porta di destinazione.

6. **Source port**

   Come menzionato in precedenza, questo campo verrà utilizzato solo se l'attacco DDoS proviene da 1 porta di origine.

7. **ICMP type**

   Il campo del tipo ICMP è disponibile nella firma. Pertanto, se il protocollo IP utilizzato è ICMP, questo campo sarà incluso in ogni regola del nostro set di regole.

8. **ICMP code**

   Analogamente al tipo ICMP, l'algoritmo includerà questo campo se applicabile.

9. **TCP flags**

   Il dissector è in grado di leggere questo campo dalla traccia dell'attacco. Se il protocollo IP è TCP, i flag TCP saranno inclusi in ogni regola del nostro set di regole.

10. **Packet length**

    Analogamente alla distribuzione delle porte, la firma generata dal dissector DDoSDB genera un elenco di tutte le lunghezze dei pacchetti che rientrano nel vettore di attacco. Quando questo elenco ha una lunghezza di 1, la lunghezza del pacchetto sarà inclusa in ogni regola del nostro set di regole.

11. **DSCP**

    Il campo DSCP non è disponibile nei nostri dati di partenza. Pertanto, non sarà incluso.

12. **Fragmentation bits**

    Il campo dei bit di frammentazione non è disponibile nei nostri dati di partenza. Pertanto, non sarà incluso.

Ricordiamo che il dissector genera una firma contenente ogni vettore di attacco appartenente alla stessa traccia DDoS. Osservando ogni campo NLRI nello

standard BGP Flowspec, si può notare che, ad eccezione dell'NLRI di tipo 2, tutti i campi sono costanti per un vettore di attacco. Ciò significa che questi campi possono essere inclusi in ogni regola, indipendentemente dal numero di regole. Inoltre, le limitazioni hardware riguardano solo il numero di regole che possono essere installate su un router in un determinato momento. Per quanto ne sappiamo, non esistono limiti al numero di campi NLRI all'interno di una regola. Di conseguenza, il campo type 2 è quello che determina la dimensione dell'insieme di regole. Per esempio, quando l'insieme degli indirizzi IP di origine è ridotto a 100 prefissi nel nostro algoritmo (raggruppando gli indirizzi IP in prefissi), la dimensione dell'insieme di regole di BGP Flowspec sarà di 100 regole. Ogni regola avrà un prefisso sorgente diverso, mentre tutti gli altri campi rimarranno costanti per lo stesso vettore di attacco.
Eseguendo i passaggi sopra descritti, si otterrà un set di regole che non supera il numero massimo di regole consentito, pur utilizzando tutte le informazioni dello standard BGP Flowspec.

6.2.3 Output

L'output del nostro generatore di regole sarà formattato un linguaggio intermedio. Ogni campo nella nostra regola corrisponde ai tipi in linea con l'RFC 5575 [27]. Questo è un modo temporaneo di memorizzare informazioni del set di regole. Di conseguenza, il set di regole può essere facilmente analizzato in un linguaggio del sistema operativo di qualsiasi router. Questo rende il nostro algoritmo di generazione delle regole riproducibile, consentendo di analizzare il set di regole per i sistemi operativi di diversi fornitori di router. Ricordiamo l'esempio di firma presentato in precedenza in questo capitolo:

```
{
    "attack_vectors": [
        {
            "service": null,
            "protocol": "TCP",
            "fraction_of_attack": 0.447,
            "source_port": "random",
```

```
            "destination_ports": {
                "9069": 0.451,
                "9070": 0.225,
                "others": 0.324
            },
            "tcp_flags": {
                ".........S.": 0.945,
                "others": 0.055
            },
            "nr_packets": 364,
            "nr_megabytes": 0,
            "time_start": "2021-06-20T19:42:57.294010+00:00",
            "duration_seconds": 817,
            "source_ips": [
                "136.243.174.154",
                "163.158.248.5",
                ...
                ...
            ],
            "ethernet_type": {
                "IPv4": 1.0
            },
            "frame_len": {
                "74": 0.717,
                "60": 0.124,
                "66": 0.115,
                "others": 0.044
            },
            "fragmentation_offset": {
                "0": 1.0
            },
            "ttl": {
                "60": 0.451,
                "56": 0.239,
                "others": 0.31
            }
        },
```

```
        {
            "service": "FTP",
            "protocol": "TCP",
            "fraction_of_attack": 0.553,
            "source_port": 21,
            "destination_ports": {
                "21": 1.0
            },
            "tcp_flags": {
                ".......A..S.": 1.0
            },
            "nr_packets": 532,
            "nr_megabytes": 0,
            "time_start": "2021-06-20T19:42:57.409323+00:00",
            "duration_seconds": 818,
            "source_ips": [
                "75.136.225.254",
                "93.114.150.139"
            ],
            "ethernet_type": {
                "IPv4": 1.0
            },
            "frame_len": {
                "60": 1.0
            },
            "fragmentation_offset": {
                "0": 1.0
            },
            "ttl": {
                "236": 0.744,
                "245": 0.256
            }
        }
    ],
    "target": "Anonymized",
    "tags": [
        "TCP flood attack",
```

```
        "TCP",
        "TCP flag attack",
        "Multi-vector attack",
        "TCP SYN ACK flag attack"
    ],
    "key": "05fe9a1689202cb5d0e2dd2e320826c5",
    "time_start": "2021-06-20T19:42:57.294010+00:00",
    "time_end": "2021-06-20T19:56:35.453656+00:00",
    "duration_seconds": 818,
    "total_packets": 896,
    "total_megabytes": 0,
    "total_ips": 60,
    "avg_bps": 564,
    "avg_pps": 1,
    "avg_Bpp": 64
}
```

Il nostro algoritmo utilizza tutti i campi possibili per generare il set di regole. Prima di generare il set di regole, viene valutato il set di IP di origine. In totale, in questa firma sono presenti 60 indirizzi IP di origine. Questo è un attacco dimostrativo semplicemente per spiegare il funzionamento del nostro generatore. In generale, come sopra descritto, un attacco DDoS può avere anche più di 10.000 indirizzi sorgente diversi. In quest'ultimo caso, se un operatore di rete volesse allocare un numero inferiore di regole per bloccare questo attacco DDoS, l'algoritmo che genera i prefissi (come descritto sopra) verrà eseguito. In questo modo si ottiene un insieme di prefissi che verrà utilizzato per il nostro set di regole. Per ogni prefisso IP sorgente, viene generata una regola. Un esempio semplificato di regole generate per i due vettori di attacco di questa firma può essere visto qui:

```
{
    'type1': '1.1.1.1/32',           //Indirizzo destinazione
    'type2': '136.243.174.154/32',   //Prefisso sorgente
    'type3' : [6],            //Protocollo, in questo caso TCP
    'type9': ['syn']                 //Set di flag TCP
}
```

```
...
{
    'type1': '1.1.1.1/32',           //Indirizzo destinazione
    'type2': '75.136.225.254/32',    //Prefisso sorgente
    'type3' : [6],            //Protocollo, in questo caso TCP
    'type5': '21',                   //Porta destinazione
    'type6': '21',                   //Porta sorgente
    'type9': ['ack', 'syn']          //Set di flag TCP
}
```

Nel primo vettore di attacco di questa firma ci sono 4 campi che possono essere utilizzati da BGP Flowspec: l'IP di destinazione (tipo 1), l'IP di origine (tipo 2), il protocollo (tipo 3) e i flag TCP (tipo 9), mentre nel secondo vettore di attacco ci sono 6 campi che possono essere utilizzati da BGP Flowspec: l'IP di destinazione (tipo 1), l'IP di origine (tipo 2), il protocollo (tipo 3), la porta di destinazione (tipo 5), la porta di origine (tipo 6) e i flag TCP (tipo 9). Ciascuno di questi campi viene incluso nella regola risultante. Si noti che la firma ha più indirizzi IP di origine. In questo caso, verrà generata un'istanza della regola per ogni prefisso IP di origine. In ogni regola, il campo IP di origine sarà diverso, mentre tutti gli altri campi rimarranno uguali per ogni vettore di attacco.

6.3 Parser

L'ultimo componente del processo di generazione delle regole è il parser. La regola è stata generata e salvata nel formato sopra menzionato. Per i diversi fornitori di hardware, è possibile implementare funzioni di parser che traducono il set di regole nella sintassi corretta.

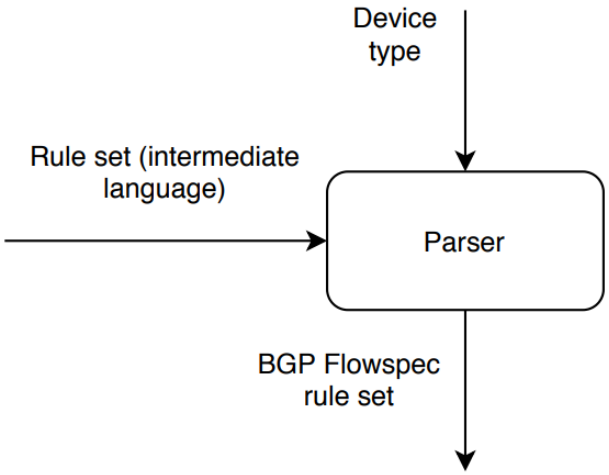

Figura 6.4: Parser

6.3.1 Input

L'ingresso di questa funzione è l'output del generatore di set di regole, cioè il set di regole in formato dizionario.

6.3.2 Processo

Siccome l'output del parser dovrà essere l'input del tool BGP Flowspec, di cui abbiamo parlano nel capitolo 5.3, le regole in formato stardard vengono convertite ed analizzate sia in ACL estese che in un formato personalizzato perchè potrebbero essere utili per l'implementazione di questo tool. Dato che il generatore può essere utilizzato indipendemente dal tool BGP Flowspec che verrà realizzato, vengono generati file di configurazione che possono essere installati direttamente su router Juniper e Cisco che supportano BGP Flowspec. Le regole BGP Flowspec in formato standard, quindi, vengono convertite in JunOS, il sistema operativo dei router Juniper [32], ed in IOS XR, il sistema operativo dei router Cisco [4]. Le funzioni di parser implementate, iterano tutte le regole e recuperano i singoli campi di ogni regola. Il codice sorgente di questo componente è pubblicamente disponibile su GitHub [42].

6.3.3 Output

L'output di una configurazione di Juniper OS è definito come una "flow route", che corrisponde a una regola BGP Flowspec. Ogni rotta ha due componenti: un campo "match" e un'"azione". Il campo match è un elenco di tutti i campi NLRI utilizzati per il filtraggio del traffico in quella regola e i valori corrispondenti. L'azione descrive ciò che il router deve fare con il traffico che soddisfa le condizioni. In IOS XR di Cisco, la configurazione di BGP Flowspec è simile a Juniper OS. In IOS XR, si utilizzano le "BGP FlowSpec Rules" per definire regole di filtraggio del traffico basate su specifiche caratteristiche del pacchetto. Ogni regola BGP FlowSpec è composta da un "campo di corrispondenza" (match field) e un'"azione" (action), così come in JunOs.

Ricordiamo le due regole di esempio citate in precedenza. La prima regola risultante, che è correlata al primo vettore di attacco, filtra il traffico con i seguenti campi: IP di destinazione (tipo 1), IP di origine (tipo 2), protocollo (tipo 3) e i flag TCP (tipo 9), la seconda regola risultante, che invece è correlata al secondo vettore di attacco, filtra il traffico con i seguenti campi: IP di destinazione (tipo 1), IP di origine (tipo 2), protocollo (tipo 3), porta di destinazione (tipo 5), porta di origine (tipo 6) e i flag TCP (tipo 9). Per semplicità, l'esempio sopra citato ha 60 prefissi IP di origine /32, ma vengono riportate solamente 2 regole dimostrative, per ogni output del parser implementato, una per ogni vettore di attacco presente nell'esempio.

L'output delle regole convertite in JunOS di questo esempio può essere visto di seguito:

```
flow {
    term-order standard;
    route 15526 {
        match {
            destination 1.1.1.1/32;
            source 136.243.174.154/32;
            protocol tcp;
            tcp-flag syn;
```

```
9
10          }
11          then discard;
12      }
13  }
14  ...
15  flow {
16      term-order standard;
17      route 307899 {
18          match {
19              destination 1.1.1.1/32;
20              source 75.136.225.254/32;
21              protocol tcp;
22              destination-port 21;
23              source-port 21;
24              tcp-flag syn ack;
25
26          }
27          then discard;
28      }
29  }
```

Le regole precedenti hanno rispettivamente le etichette "15526" e "307899". Queste etichette viengono utilizzate dal software JunOS per identificare le regole e sono generate in modo casuale. La singola regola ha le condizioni di corrispondenza descritte nel campo match, e sono diverse per i due vettori di attacco dato che l'esempio riportato è un multi-vector attack. Quando il traffico che soddisfa queste condizioni entra nella rete del router, essendo l'action "discard" viene scartato.

L'output delle regole convertite in Cisco IOS XR di questo esempio può essere visto di seguito:

```
1  router bgp 10000
2   address-family ipv4 flowspec
3    match
4              destination-address 1.1.1.1/32;
5              source-address 136.243.174.154/32;
```

```
            transport protocol tcp;
            tcp-flag syn;

  action drop
...
router bgp 10000
 address-family ipv4 flowspec
  match
            destination-address 1.1.1.1/32;
            source-address 75.136.225.254/32;
            transport protocol tcp;
            destination-port 21;
            source-port 21;
            tcp-flag ack syn;

  action drop
```

Per le BGP Flowspec rules di Cisco IOS XR, non è necessario definire un identificativo specifico per ogni regola come in Juniper OS (Junos). Cisco IOS XR permette di elencare i criteri di corrispondenza (match) e le azioni (action) di ciascuna BGP Flowspec rule all'interno della stessa sezione di configurazione BGP Flowspec. Quello che però bisogna definire è l'"AS_NUMBER", ovvero l'etichetta "10000" presente nelle regole appena citate. Questo identifica il numero di autonomous system (AS) reale della propria configurazione di rete, e dato che dipende da rete a rete, viene preso come input dal generatore nel momento in cui si richiede di generare regole BGP Flowspec per Cisco IOS XR. Così come per le regole generate per JunOS, quando il traffico soddisfa le condizioni (match), essendo l'action "drop", viene scartato.

Di seguito vengono elencate le altre due conversioni che sono state realizzate per facilitare lo sviluppo del tool BGP Flowspec.

Output della conversione in ACL estese:

```
ip access-list extended ATTACK_RULE_35
            destination 1.1.1.1/32;
            source 136.243.174.154/32;
            protocol tcp;
```

```
            tcp-flag syn;

deny ip host 136.243.174.154/32 host 1.1.1.1/32 syn
...
ip access-list extended ATTACK_RULE_59
            destination 1.1.1.1/32;
            source 75.136.225.254/32;
            protocol tcp;
            destination-port 21;
            source-port 21;
            tcp-flag ack syn;

deny ip host 75.136.225.254/32 host 1.1.1.1/32 ack syn
```

Così come per JunOS anche per le ACL estese abbiamo bisogno di un identificativo, che in questo caso è rispettivamente "ATTACK_RULE_35" e "ATTACK_RULE_59" e viene generato sequenzialmente così da essere sicuri di generare sempre un id diverso per ogni regola. Anche per le ACL estese abbiamo un campo match, contenente le condizioni di corrispondenza, che se sono rispettate scaturiscono l'action "deny", quindi il traffico viene scartato. Output della conversione in formato personalizzato:

```
{{destination 1.1.1.1/32} {source 136.243.174.154/32}
{protocol tcp} {tcp-flags syn}}
...
{{destination 1.1.1.1/32} {source 75.136.225.254/32}
{protocol tcp} {destination-port 21} {source-port 21}
{tcp-flags ack,syn}}
```

Questo output è una versione semplificata che raggrupa tutte le regole generate, ognuna in una riga corrispondente, per ogni vettore di attacco. Questo formato viene realizzato per rendere più agevole la lettura e la compresione delle regole ed anche per essere facilmente utilizzato durante lo sviluppo futuro del tool BGP Flowspec.

CAPITOLO 7

SVILUPPI FUTURI

Come spiegato nel capitolo 6, questa tesi di ricerca si concentra sullo sviluppo del generatore di regole BGP Flowspec ed ipotizzando che l'attacco sia già stato rilevato, sono stati utilizzati file .pcap di attacchi DDoS esistenti come input al nostro generatore. Rispetto all'intero modello presentato nel capitolo 5 lasciamo come sviluppi futuri:

- **IDS:** lo sviluppo di un IDS per rilevare gli attacchi in tempo reale utilizzando tecniche di machine learning per identificare anche possibili attacchi zero-day. L'output di questo componente dovrà essere un file.pcap contenente i pacchetti dell'attacco rilevato, che dovrà essere dato in input al nostro generatore.

- **Tool BGP Flowspec:** lo sviluppo del tool BGP Flowspec che, date le regole generate da questo lavoro di tesi, permetta di installare in modo automatico queste regole sui router BGP in tempo reale.

- **Parser:** l'implementazione esistente del parser ci permette di convertire le regole per router BGP sia per JunOS che per Cisco IOS XR. Dato che questo generatore può essere utilizzato indipendentemente dal tool BGP Flowspec, che si trova attualmente in fase di implementazione, si

potrebbe considerare di estendere il parser per permettere la conversione ad altri sistemi operativi di diversi produttori di router BGP.

- **DDoS Dissector:** il DDoS dissector utilizzato per questo lavoro è stato preso dal DDoSDB per estrarre le caratteristiche di rete dai file .pcap, contenente gli attacchi, in un file Json che rende facile la conversione dei campi in una regola BGP Flowspec. Dato che il dissector è un modulo separato non progettato dagli autuori della tesi, si potrebbe pensare di sviluppare un modulo simile per collegare sia l'IDS che il generatore di regole, creando un unico modulo che effettua sia il detection che la mitigation dell'attacco.

CAPITOLO 8

RINGRAZIAMENTI

Desidero dedicare questo spazio per esprimere la mia profonda gratitudine a tutte le persone che hanno contribuito alla realizzazione di questa tesi. Questo percorso accademico è stato un viaggio lungo e impegnativo, reso più significativo grazie al sostegno e all'apporto prezioso di molti.

Innanzitutto, un ringraziamento speciale va ai miei genitori, Maria e Giuseppe, per il loro amore e il loro costante sostegno. Grazie per aver creduto in me e per avermi sostenuto in ogni fase di questo percorso.

Ringrazio la mia fidanzata, Francesca, una persona straordinaria che ha reso questa esperienza ancora più preziosa. Il tuo costante appoggio, la tua comprensione e la tua presenza hanno reso ogni sfida più gestibile.

Ai miei amici e compagni di corso, voglio esprimere la mia profonda gratitudine. Le vostre parole di incoraggiamento, le lunghe discussioni, il vostro supporto, le risate e i momenti condivisi con voi hanno reso questo viaggio più significativo.

Infine, ma non per importanza, voglio ringraziare una persona che, anche se non può essere qui fisicamente per condividere questo momento con me, so che mi sta guardato dall'alto con orgoglio.

BIBLIOGRAFIA

[1] Diederik Bakker. «Impact-based optimisation of BGP Flowspec rules for DDoS attack mitigation». In: (2019).

[2] R. Lehti P. Virolainen R. van den Berg e H. von Haugwitz. «AIDE - Advanced Intrusion Detection Environment». In: (2023). URL: http://aide.sourceforge.net/.

[3] J. Pil Choi e B.-C. Kim. «Net neutrality and investment incentives». In: (2010). URL: http://doi.wiley.com/10.1111/j.1756-2171.2010.00107.x.

[4] Cisco. «IOS XR». In: (2023). URL: https://www.cisco.com/c/en/us/products/ios-nx-os-software/iosxrsoftware/index.html.

[5] Cloudflare. «DNS Flood DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/dns-flood-ddos-attack/.

[6] Cloudflare. «HTTP Flood DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/.

[7] Cloudflare. «Memcached DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/.

[8] Cloudflare. «Memcached DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/.

[9] Cloudflare. «NS Amplification DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/.

[10] Cloudflare. «NTP Amplification DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/.

[11] Cloudflare. «Ping (ICMP) Flood DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/ping-icmp-flood-ddos-attack/.

[12] Cloudflare. «SSDP DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/ssdp-ddos-attack/.

[13] Cloudflare. «SYN Flood DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/.

[14] Cloudflare. «UDP Flood DDoS Attack». In: (2023). URL: https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/.

[15] Christoph Dietzel et al. «Stellar: Network Attack Mitigation using Advanced Blackholing». In: (2018).

[16] Marinos Dimolianis et al. «DDoS Attack Detection via Privacy-aware Federated Learning and Collaborative Mitigation in Multi-domain Cyber Infrastructures». In: (2022).

[17] Marinos Dimolianis et al. «Mitigation of Multi-vector Network Attacks via Orchestration of Distributed Rule Placement». In: (2019).

[18] C. Dietzel A. Feldmann e T. King. «Blackholing at IXPs: On the Effectiveness of DDoS Mitigation in the Wild». In: (2016). URL: http://link.springer.com/10.1007/978-3-319-30505-9%2024.

[19] Open Information Security Foundation. «Suricata — Open Source IDS / IPS / NSM engine». In: (2023). URL: https://suricata-ids.org/.

[20] OSSEC Foundation. «OSSEC — Open Source HIDS Security». In: (2023). URL: https://www.ossec.net/.

[21] L F Haaijer. «DDoS Packet Capture Collection». In: (2023). URL: https://github.com/StopDDoS/packet-captures.

[22] Y. Rekhter S. Hares e T. Li. «Border Gateway Protocol 4 (BGP-4)». In: (2006). URL: https://rfc-editor.org/rfc/rfc4271.txt.

[23] IBM. «IBM QRadar Security Intelligence». In: (2023). URL: https://www.ibm.com/nl-nl/security/security-intelligence/qradar.

[24] Ziad El Jamous et al. «RADAR: An Automated System for Near Real-Time Detection and Diversion of Malicious Network». In: (2016).

[25] Mattijs Jonker et al. «A First Joint Look at DoS Attacks and BGP Blackholing in the Wild». In: (2018).

[26] Joeri Kock. «A Signature-Based Approach to DDoS Attack Mitigation Using BGP Flowspec Rules». In: (2019).

[27] P Marques N Sheth R Raszuk B Greene J Mauch e D McPherson. «Dissemination of flow specification rules». In: (2009).

[28] P. Marques N. Sheth R. Raszuk B. Greene J. Mauch e D. McPherson. «Dissemination of flow specification rules». In: (2009).

[29] S. McCanne e V. Jacobson. «The BSD Packet Filter: A New Architecture for User-level Packet Capture». In: (). URL: https://www.usenix.org/legacy/publications/library/proceedings/sd93/mccanne.pdf.

[30] Marcin Nawrocki et al. «Down the Black Hole: Dismantling Operational Practices of BGP Blackholing at IXPs». In: (2019).

[31] Netfilter. «iptables - iptables tree». In: (2023). URL: https://git.netfilter.org/iptables/.

[32] Juniper Networks. «Junos OS». In: (2023). URL: https://www.juniper.net/us/en/products-services/nos/junos/.

[33] V. Paxson. «Bro: a system for detecting network intruders in real-time». In: (1999). URL: https://suricata-ids.org/.

[34] Ristic. «ModSecurity: Open Source Web Application Firewall». In: (2023). URL: https://modsecurity.org/.

[35] M. Roesch et al. «Snort: Lightweight intrusion detection for networks». In: (1999).

[36] J J Santanna. «DDoSDB - GitHub». In: (2023). URL: https://github.com/ddos-clearing-house/ddosdb.

[37] J J Santanna. «Dissector DDoSDB - GitHub». In: (2023). URL: https://github.com/ddos-clearing-house/ddos_dissector.

[38] Quadrant Information Security. «Sagan Main Wiki». In: (2023). URL: https://wiki.quadrantsec.com/bin/view/Main/SaganMain.

[39] Leonardo Serodio. «Traffic Diversion Techniques for DDoS Mitigation using BGP Flowspec». In: (2013).

[40] SolarWinds. «Log and Event Manager». In: (2023). URL: https://www.solarwinds.com/log-event-manager-software.

[41] Security Onion Solutions. «Security Onion». In: (2023). URL: https://securityonion.net/.

[42] Raffaele Squillante. «DDoS Attack Mitigation Using BGP Flowspec Rules». In: (2023). URL: https://github.com/Mr-Rafo/Master-Thesis-RaffaeleSquillante.

[43] Kotikalapudi Sriram e Doug Montgomery. «Resilient Interdomain Traffic Exchange: BGP Security and DDoS Mitigation». In: (2019).

[44] TechTarget. «What is DNS redirection?» In: (2023). URL: https://searchsecurity.techtarget.com/definition/Web-application-firewall-WAF.