./images/

Alla mia famiglia,
che mi ha sempre sostenuto
in ogni mia scelta.

Introduzione

Questa è l'introduzione.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Lo stato dell'Arte

In questo capitolo si va ad illustrare lo stato dell'Arte delle tecnologie utilizzate. Si illustreranno le principali qualità degli Intrusion Detection Systems e le caratteristiche principali che hanno portato durante i test alla scelta di un software rispetto che un altro. Si passerà poi a presentare sFlow, illustrandone i benefici e le principali differenze con NetFlow e di come esso viene attualmente utilizzato per affiancare un IDS in reti molto estese e complesse. Infine si darà una breve presentazione dello stack ELK, (Elasticsearch-Logstash-Kibana) e di come esso sia utilizzato nell'ambito della Network Security.

1.1 Intrusion Detection System

Un Intrusion Detection System (IDS) [4] è un dispositivo o un' applicazione software che monitora una rete o un sistema per rilevare eventuali attività dannose o violazioni delle policy. Qualsiasi attività o violazione rilevata viene in genere segnalata ad un amministratore o raccolta a livello centrale utilizzando un Security Information and Event Management (SIEM). Un SIEM combina output provenienti da più sorgenti e utilizza tecniche di filtraggio degli allarmi per distinguere le attività dannose dai falsi allarmi.

Esiste un' ampia gamma di IDS, che varia dal software antivirus fino ai sistemi gerarchici che controllano il traffico di un' intera backbone. La classificazione più comune è tra:

- **Network-based Intrusion Detection Firmware (NIDS)**: ad esempio un sistema che monitora il traffico di rete passante attraverso alcuni punti strategici di una rete. Esempi famosi sono: Suricata [6] , Snort [7] e BRO [8] .
- **Host-based Intrusion Detection (HIDS)** : ad esempio un software che monitora alcuni file importanti del sistema operativo su cui è installato. Un esempio famoso di HIDS è OSSec [5]

Il panorama degli Intrusion Detection System (IDS) è al giorno d'oggi in continua evoluzione. Tuttavia è possibile operare una seconda e importante classificazione in base a due criteri principali che ne determinano il funzionamento:

- Sistemi *signature-based*
- Sistemi basati su *anomaly detection*

Un IDS *signature-based* analizza i pacchetti passanti su una rete utilizzando un set di regole appositamente scritte per rilevare un determinato tipo di attacco.

Un IDS *signature-based* analizza i pacchetti passanti su una rete utilizzando il concetto di *signature*: Una signature è un pattern che corrisponde ad un tipo di attacco noto. [**?**] Esempi di signature possono essere:

- un tentativo di connessione a telnet con username "root", che corrisponde ad una violazione delle policy di sicurezza
- un email con oggetto "Immagini gratis!" e un allegato con nome "freepics.exe", che sono caratteristici di un attacco nostro
- tentativi ripetuti nel tempo di connessione ssh ad intervalli sospetti, che identificano un possibile attacco bruteforce su ssh.

Il rilevamento signature-based è molto efficace nel rilevare minacce note, ma in gran parte inefficace nel rilevare minacce precedentemente sconosciute, minacce mascherate dall'uso di tecniche di evasione e molte varianti di minacce note. Per esempio, se un aggressore ha modificato il malware nell' esempio precedente per usare un nome file di "freepics2.exe", una firma che cercava "freepics.exe" non lo corrisponderebbe.

Tradotto con www.DeepL.com/Translator

Questo ci mette al riparo da una grandissima quantità di attacchi conosciuti e si è rilevato nel corso degli anni un sistema efficiente e ampiamente utilizzato.

I secondi invece utilizzano tecniche di intelligenza artificiale per determinare se un comportamento sia lecito oppure no basandosi su quello che è stato definito in fase di installazione come uno "stato normale".

Sebbene nell'ultimo periodo l'intelligenza artificiale stia facendo la sua comparsa in ogni ambito dell'informatica gli IDS signature based rappresentano tuttora un'importante fetta (se non la maggioranza) degli IDS in uso nei più importanti data center del mondo ed è per questo che vale la pena studiarli.

In questo elaborato ci si focalizzerà sugli IDS signature based e se ne analizzeranno le loro prestazioni combinate ad altre tecnologie che verranno introdotte in seguito.

Come anticipato sopra, tra i maggiori esponenti degli IDS attualmente utilizzati abbiamo:

- Snort: Un IDS sviluppato a partire dagli anni '90, acquisito da Cisco nel 2013 e che è tuttora il più utilizzato in ambito enterprise.

- Suricata: Un IDS del nuovo millennio, sviluppato a partire dal 2009 da Open Information Security Foundation (OISF) e che vanta molteplici vantaggi sopra gli altri IDS.

In questo elaborato si è preferito utilizzare per motivi di performance e di implementazione, Suricata. I dettagli di questa scelta saranno chiari

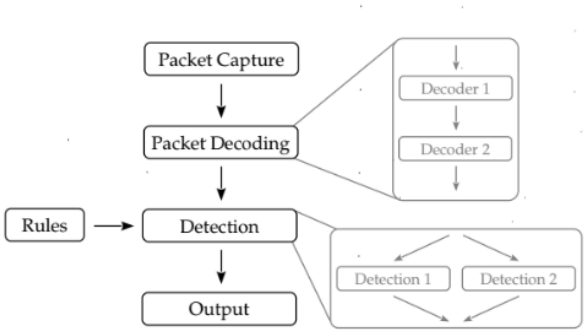

Figura 1.1: Funzionamento del pattern matching

più avanti quando saranno state introdotte le principali caratteristiche di Suricata.

1.1.1 Suricata, una breve introduzione

Suricata è un IDS che fa uso di pattern matching per il riconoscimento dei threat capace di effettuare un'analisi dei pacchetti molto approfondita. Il funzionamento del pattern matching può essere riassunto dalla figura 1.1. [2] Ogni pacchetto viene quindi decodificato e poi analizzato parallelamente per riscontrare similitudini con più pattern. E' stato dimostrato infatti (trovare il paper) che Suricata, data la sua natura multi threaded, è di gran lunga più veloce di Snort (in ambienti multi-core), il quale invece è single thread. Mentre una tipica regola per il suddetto patter matching è fatta in questo modo:

```
rule header    alert tcp any any -> 192.168.1.0/24 111
```

La praticità dell'utilizzare questa sintassi sta nel fatto che essa è quasi del tutto identica a quella di Snort. Per cui l'utilizzo dell'uno o dell'altro software, almeno nel caso di studio di questa tesi, non è determinante. E' possibile definire regole personalizzate, così come scaricarne di già confezionate molto accurate.

Infine una delle caratteristiche fondamentali di Suricata sta nel fatto che esso può funzionare in due modi distinti:

- In modalità online: viene monitorata una interfaccia specifica in modalita‘ promisqua, ossia tutti i pacchetti passanti per quella determinata interfaccia vengono decodificati e analizzati.
- In modalita offline: viene monitorato un file pcap contenente del traffico "registrato" in precedenza e che costituisce un punto di riferimento per l'analisi prestazionale delle regole o dell'istanza di Suricata da analizzare.

1.2 sFlow

sFlow [1] è una tecnologia per monitorare il traffico in reti contenenti switches e routers che utilizza il capionamento di pacchetti. In particolare, esso definisce i meccanismi di campionamento implementati in un *sFlow Agent* e il formato dei dati campionati mandati da tale Agent.

Il monitoraggio si compone di due elementi fondamentali:

- **sFlow Agent**: ovvero un qualsiasi apparato in grado di campionare i pacchetti secondo le specifiche di sFlow e di inviarli ad un *Collector*. l'Agent è un componente molto versatile ed estremamente performante dell'architettura sFlow che può essere impersonato anche da uno switch o da un router, senza degradarne le prestazioni. Il campionamento e la raccolta dei dati del nodo viene fatta in hardware e non presenta overhead nemmeno su reti Gigabit.
- **Collector**: ovvero una macchina in qualsiasi parte del mondo in grado di raccogliere i dati sFlow e di elaborarli.

L'architettura e le modalità di campionamento usati in sFlow offrono numerosi vantaggi tra cui quello di avere una visione di tutta la rete (*network-wide*) in tempo reale. Infatti, i pacchetti campionati vengono mandati al

collector non appena arrivano all'agent. Inoltre l'architettura è estremamente scalabile e permette di posizionare agent in diversi punti della rete, o anche in reti diverse.

1.2.1 Campionamento

Il campionamento è la parte fondamentale del protocollo sFlow, ed è anche il motivo per cui esso si distacca da altre tecnologie simili come NetFlow.

Avvengono due tipi di campionamento:

- un campionamento: in cui viene salvato 1 pacchetto ogni N, con N configurabile
- un troncamento: in cui il pacchetto designato per il salvataggio viene troncato, solitamente ad una dimensione di 128 Byte (tuttavia configurabile)

Questi due meccanismi insieme permettono di avere una visione della rete sufficientemente accurata o comunque di una accuratezza configurabile.

sFlow vs Netflow

Un'altra tecnologia molto simile ad sFlow è NetFlow, sviluppato da Cisco. Essi si differenziano per un motivo importantissimo e che non permette di usare efficacemente NetFlow per monitorare la rete a fianco di un IDS; ovvero che Netflow fornisce solo dati aggregati e non fornisce un troncamento parametrizzabile. Esso infatti esporta solo il Layer 3 del pacchetto, questo fa perdere tutte le informazioni che, nel nostro particolare campo di applicazione, sarebbero state utili, come l'header dei livelli superiori.

Qui è possibile vedere un esempio di un output del comando *nfdump*:

```
Date flow start          Duration Proto   Src IP Addr:Port      Dst IP Addr:Port     Packets    Bytes Flows
2010-09-01 00:00:00.459     0.000 UDP     127.0.0.1:24920   ->  192.168.0.1:22126        1       46     1
2010-09-01 00:00:00.363     0.000 UDP     192.168.0.1:22126 ->  127.0.0.1:24920          1       80     1
```

Figura 1.2:

1.3 Packet sampling e IDS

La sicurezza delle reti, specie di quelle molto estese, è un problema tutt'oggi presente e di un importanza vitale in qualsiasi azienda che operi nel settore ICT (Information and Communications Technologies). Le minacce possono avvenire in qualunque momento [3] ed essere generate sia dall'interno che dall'esterno. Riuscire ad identificare queste minacce in tempo è il primo passo verso la soluzione di questo difficile problema. Per fare ciò è necessario avere un'ampia e continua sorveglianza della rete. Storicamente la sorveglianza di una grande rete era (ed è tuttora) affidata a sonde (*probes*), posizionate in punti strategici della rete. Questo è stato sufficientemente accurato fino ad ora, quando si sta verificando un aumento dell'installazione di switched point to point network. Quindi implementare il monitoraggio già dall'interno di uno switch o di router sta diventando sempre più una necessità. Tuttavia le esigenze di mercato tendono a preferire l'ampiezza di banda sulla sicurezza, per cui la funzione di monitoraggio deve essere relegata come funzione secondaria all'interno di questi apparati di rete. E' necessario quindi che questa funzione operi con il minimo overhead possibile, al fine di non degradare le prestazioni dell'apparato. E' qui che entra in gioco la tecnologia sFlow, essa permette di delegare la parte di analisi del traffico ad un altro componente della rete (*il collector*), lasciando allo switch o al router le risorse per effettuare le loro decisioni di smistamento dei pacchetti.

Vediamo allora come sflow garantisce l'efficacia del sistema di monitoraggio richiesto:

- Sorveglianza continua network-wide: sFlow può essere configurato su ogni apparato di rete

- I dati devono essere sempre disponibili per rispondere efficacemente: sFlow manda immediatamente i pacchetti campionati al Collector con UDP

- I dati devono essere sufficientemente dettagliati per caratterizzare l'attacco: sFlow permette di esportare più di 128 Byte (comprendendo quindi ad esempio anche header di livello 7)
- Il sistema di monitoraggio non deve esporre gli apparato ad attacchi: sFlow impatta sulle prestazioni degli apparati poichè viene effettuato in hardware inoltre il consumo di banda è limitato poichè i datagram sFlow sono compressi

1.4 Stack ELK

Lo stack ELK è uno stack formato dai tre programmi (Elasticsearch, Logstash e Kibana) i quali uniti insieme forniscono una metodologia di centralizzazione, esplorazione e visualizzazione dei log. Esso si compone di tre componenti fondamentali:

- **Logstash** : il quale si occupa di raccogliere i log, manipolarli così da renderli consumabili da elasticsearch
- **Elasticsearch** : un database No-SQL particolarmente ottimizzato per la ricerca
- **Kibana** : un visualizzatore dei dati contenuti in Elasticsearch che permette di effettuare ricerche e aggregare i dati in modo da fornirne una visualizzazione utile dei dati di log raccolti

Lo stack ELK è ad oggi lo standard *de facto* per l'esplorazione dei log, esso si va ad affiancare a tutti i programmi critici all'interno di una infrastruttura e risulta particolarmente utile nell'esplorazione dei log degli IDS. Esso permette infatti acquisire in pochissimi secondi conoscenza su cosa accade nella rete e su che tipo di attacchi sono in corso. I dati vengono fruiti all'operatore in maniera veloce e comprensibile tramite dashboard, fornendo una fotografia estremamente informativa dello stato attuale

Conclusioni

Queste sono le conclusioni.
In queste conclusioni voglio fare un riferimento alla bibliografia: questo è il mio riferimento [3, 4].

Appendice A

Prima Appendice

In questa Appendice non si è utilizzato il comando: `\clearpage{\pagestyle{empty}\cleardoublepage}`, ed infatti l'ultima pagina 8 ha l'intestazione con il numero di pagina in alto.

Appendice B

Seconda Appendice

Bibliografia

[1] RFC 3176 InMon Corporation's sFlow for monitoring traffic in Switched and Routed Networks

[2] A survery on Network Security Monitoring System, Ibrahim Ghafir, Vaclav Prenosil, Jakub Svoboda, Mohammad Hammoudeh, 2016 4th International Conference on Future Internet of Things and Cloud Workshops

[3] Traffic Monitoring with Packet-Based Sampling for Defense against Security Threats, Joseph Reves ans Sonia Panchen

[4] "NIST â Guide to Intrusion Detection and Prevention Systems (IDPS)" (PDF). February 2007.

[5] https://ossec.github.io/

[6] https://github.com/OISF/suricata

[7] https://www.snort.org/

[8] https://www.bro.org

Ringraziamenti

Qui possiamo ringraziare il mondo intero!!!!!!!!!!
Ovviamente solo se uno vuole, non è obbligatorio.