Indice:

1. Introduzione
   1. Definizione del problema
   2. Struttura della tesi
2. Traffico di rete
   1. Pacchetti di dati
   2. Zeek
3. Process discovery
   1. Trace di eventi (fornire
   2. Conversione di pacchetti in Trace
   3. Reti di Petri
   4. Fitness di un trace
4. Anomaly detection
   1. Definizione del problema
   2. Isolation Forest
5. Validazione empirica
   1. CICID 2017
   2. Risultati dell’estrazione dei trace
   3. Etichettatura dei trace
   4. Metriche di valutazione e setting sperimentale
   5. Risultati
6. Conclusioni

Appendice A - Diagramma delle classi

1. Zeek
2. Creazione dei trace dai pacchetti
3. Anomaly detection

Appendice B -Manuale Utente

1. Download e installazione dei software necessari
2. Download del Dataset
3. Generazione di un file xes
4. Avvio sperimentazione

Bibliografia

1. Introduzione
   1. Definizione del problema

La sperimentazione nasce con l’obiettivo di **rilevare connessioni anomale**, cioè dove si è effettuato un attacco a livello di sicurezza, in un traffico di rete, mediante l’uso di analisi delle connessioni tramite algoritmi di *process mining*, coadiuvato da algoritmi di *Anomaly detection*.

Inizialmente è necessario ottenere i pacchetti e memorizzarli all’interno di *log* di connessioni; quindi, le cui righe del *log* corrispondono ad una connessione (quadrupla indirizzo IP e porta dell’*host* mittente e indirizzo e porta dell’*host* destinatario), e ogni connessione è composta da pacchetti inviati e ricevuti dai due *host*. Le connessioni possono essere catturate tramite un software *sniffer* di pacchetti come *Wireshark*, che una volta terminata la cattura dei pacchetti, questi vengono memorizzati in dei file *.pcap*. Questi file vengono poi analizzati da software specifici che permettono di convertirli file *log*. Il software utilizzato in questa sperimentazione è *Zeek*, un software a linea di comando che dato in input un file *.pcap*, genera dei file *log*, ognuno dei quali contiene informazioni specifiche per i diversi protocolli adoperati da ogni connessione, e un file *conn.log* che contiene tutte le connessioni catturate nel file *.pcap* e le informazioni comuni a tutte le connessioni.

A questo punto è stato creato un applicativo che converte i file *log* di *Zeek*, in file *xes*, che è una specifica di file *xml*, contenente una serie di *trace*, i quali raggruppano una serie di *event*, specifici per il *process mining*. I trace in questo caso sono identificati dalla quadrupla indirizzo IP sorgente, porta sorgente, indirizzo IP destinazione, porta di destinazione, mentre, gli eventi, sono tutte le connessioni, o righe del *log*, che sono accomunate dalla quadrupla del trace.

La sezione di ***process mining***consisterà di due fasi: generazione delle *PetriNet* e calcolo dei livelli di *conformance*.

La **prima fase** ha come obiettivo quello di modellare le *PetriNet* generando dei file *xes* con *trace* ed *event* usando le connessioni normali. Successivamente si calcolano i livelli di *conformance* di ulteriori connessioni normali.

La **seconda fase** ha l’obiettivo di calcolare i livelli di *conformance* di connessioni di tipo normale e anomalo con le *PetriNet* modellate durante la prima fase.

La sezione di ***Anomaly detection***consiste anch’essa di due fasi: la fase di apprendimento del modello di *anomaly detector* e quella di predizione delle anomalie.

Durante la **fase di apprendimento** si utilizzano i livelli di *conformance* calcolati con connessioni normali per allenare il modello di una *Isolation forest.*

Durante la **fase di predizione** si predicono le connessioni anomale con i livelli di conformance calcolati nella seconda fase del *process mining*.

* 1. Struttura della tesi

1. Traffico di rete
   1. Pacchetti di dati
   2. *Zeek*