Relazione Progetto Sicurezza Informatica

Matteo Capoferri - Lorenzo Rebuschi

Introduzione

Lo scopo del progetto è quello di analizzare il dataset fornito (*packets-pseudoanonymous.csv*, dataset relativo a interazioni malevole che hanno raggiunto una honeypot in una darknet tramite probing) in modo da ingegnerizzare delle features che possano permettere l'identificazione di modus operandi di attacco attraverso algoritmi di clustering e valutare sperimentalmente la fattibilità di early classification dei modus operandi individuati.

Preprocessing dei dati

Da una prima analisi dei dati è emerso che ci sono valori mancanti nel dataset. L'assenza di valori in un record in alcune delle sue dimensioni dipende dal contenuto della colonna '*proto*'. I protocolli presenti nel dataset sono:

- TCP (corrispondente al valore 6 della dimensione '*proto*')
- UDP (corrispondente al valore 17 della dimensione '*proto*')
- ICMP (corrispondente al valore 1 della dimensione '*proto*')

Il dataset originale è stato quindi spezzato in 3 dataset, ognuno contenente tutti e soli i record di uno dei 3 protocolli.
E' stato poi effettuato un drop sulle dimensioni che risultano a questo punto non contenere valori per i dataset appena creati. In particolare per udp è stata eliminata la dimensione '*tcp_flags*', mentre per icmp sono state eliminate le dimensioni '*src_port*', '*dst_port*', '*tcp_flags*'. In tutti i nuovi dataset è stata eliminata la dimensione '*proto*' che non ha più alcun valore informativo dopo le scelte fatte e la dimensione '*mirai*' non ritenuta utile ai fini delle successive analisi.

Ingegnerizzazione delle Features

Per ognuno dei 3 nuovi dataset creati viene seguito un approccio simile allo scopo di estrarre conoscenza dai dati grezzi presenti. Sulla base della coppia <src_ip, src_port> (per icmp solo <src_ip>) all'interno di una certa finestra temporale vengono individuati gli utenti che hanno portato degli attacchi (dimensione 'user'). Tutte le altre dimensioni derivate dai dati originali sono strettamente legate alla dimensione 'user'. Le nuove dimensioni estratte sono:

- `user` che identifica un particolare utente-attaccante all'interno di una certa finestra temporale
- `num_attack`, ovvero il numero di attacchi portati da uno stesso utente
- `first_port_scan`, cioè la prima porta scansionata da un attaccante (no per icmp)

- `ip_diff`, ovvero il numero di ip destinazione differenti scansionati da uno stesso attaccante
- `port_diff`, ovvero il numero di porte destinazione differenti scansionate da uno stesso attaccante (no per icmp)
- `range_ip` calcolato come differenza tra ip destinazione massimo e ip destinazione minimo
- `range_port` calcolato come differenza tra porta destinazione più alta e porta destinazione più bassa (no per icmp)
- `ip_medium`, cioè l'ip destinazione medio scansionato dall'attaccante
- `ip_std`, cioè la deviazione standard degli ip destinazione scansionati dall'attaccante
- `port_medium`, cioè la porta destinazione media scansionata dall'attaccante (no per icmp)
- `port_std`, cioè la deviazione standard delle porte destinazione scansionate dall'attaccante (no per icmp)
- `pck_medium`, cioè la lunghezza media dei pacchetti usati dall'attaccante
- `pck_std`, cioè la deviazione standard della lunghezza dei pacchetti usati dall'attaccante

Le finestre temporali scelte in UNIX time sono 10, 60, 180, 600. Viene prodotto un nuovo dataset per ogni finestra temporale e per ogni protocollo. I nuovi dataset presentano tutte le caratteristiche sopra elencate e 1 record per ogni utente ('user') individuato.

Questi dataset costituiscono la conoscenza di base per gli algoritmi di clustering applicati.

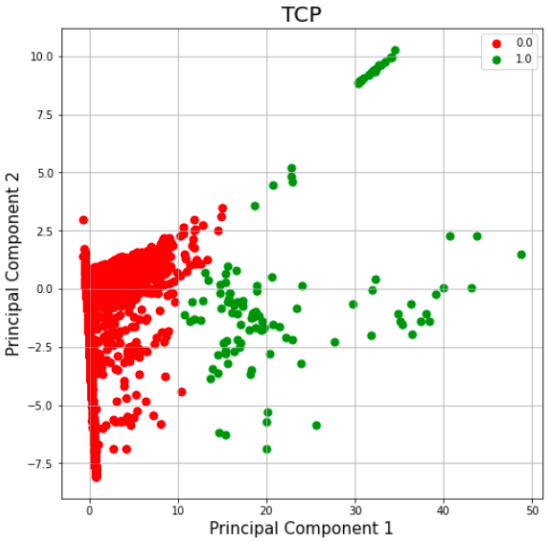
Clustering

Sono scelti 3 diversi algoritmi di clustering con caratteristiche differenti allo scopo di analizzare lo stesso problema da diversi punti di vista e vedere quale approccio meglio si adatta ai dati a disposizione.
In particolare gli algoritmi di clustering usati sono:

- K-Means
- DBSCAN
- BIRCH

Vengono di seguito riportati i migliori risultati ottenuti per ogni algoritmo. Le finestre temporali scelte (tenuto conto del confronto tra le prestazioni ottenute) sono: 180 per TCP, 10 per UDP e 60 per ICMP.

K-Means



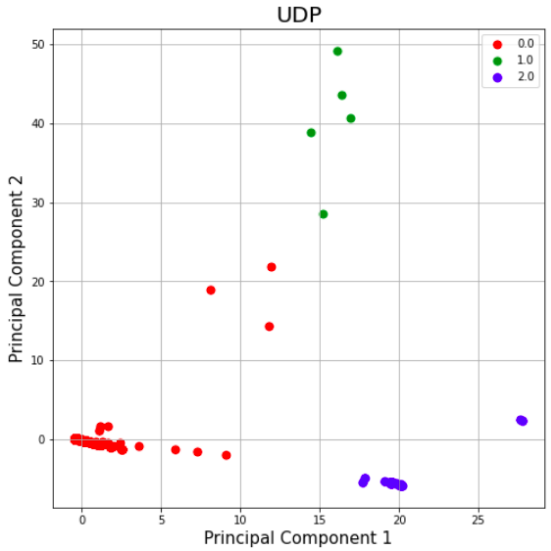
Risultato ottenuto per TCP con finestra temporale 180 e con i seguenti parametri:

- numero cluster=2

Presenta un silhouette score pari a 0.90.
I cluster ottenuti presentano la seguente configurazione:

- cluster 0, 24360 samples

- cluster 1, 117 samples

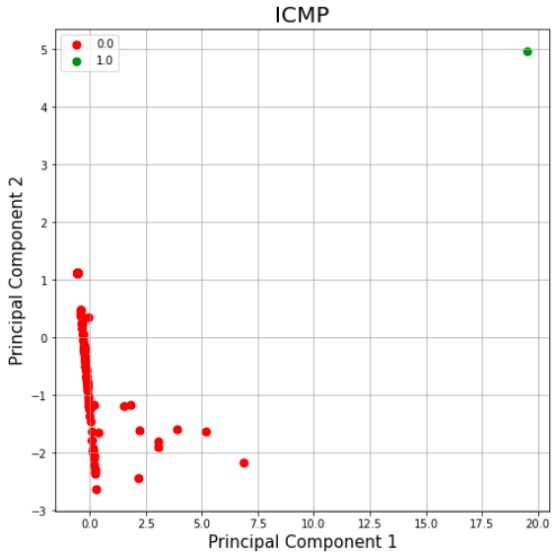


Risultato ottenuto per UDP con finestra temporale 10 e con i seguenti parametri:

- numero cluster=3

Presenta un silhouette score pari a 0.89.
I cluster ottenuti presentano la seguente configurazione:

- cluster 0, 3825 samples
- cluster 1, 5 samples
- cluster 2, 36 samples



Risultato ottenuto per ICMP con finestra temporale 60 e con i seguenti parametri:

- numero cluster=2

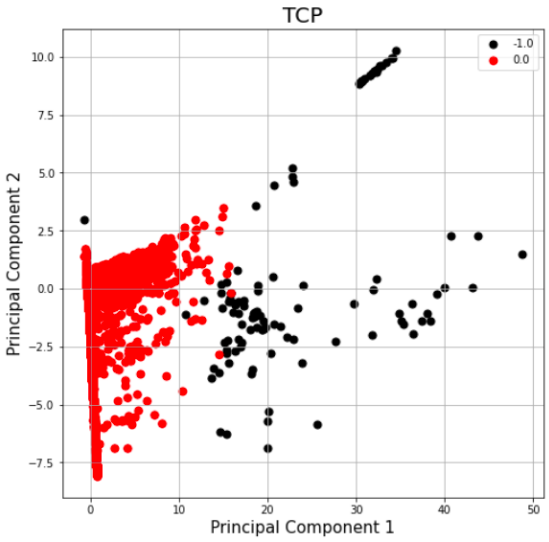
Presenta un silhouette score pari a 0.89.

I cluster ottenuti presentano la seguente configurazione:

- cluster 0, 169 samples
- cluster 1, 1 samples

DBSCAN

Premessa: il cluster -1 indica dati rumorosi.

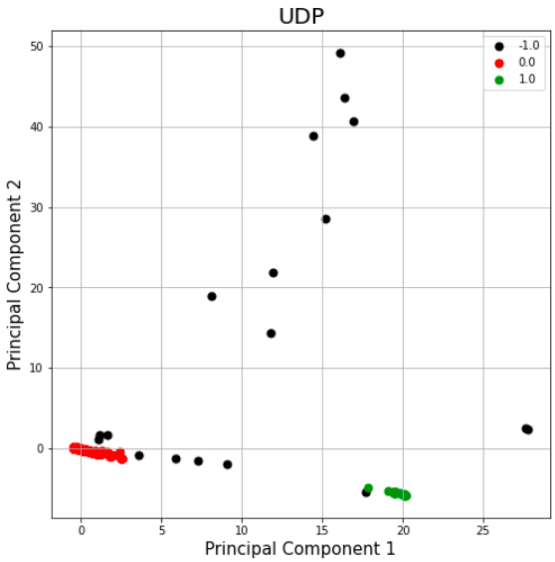


Risultato ottenuto per TCP con finestra temporale 180 e con i seguenti parametri:

- eps = 10
- min_samples = 300

Presenta un silhouette score pari a 0.90.
I cluster ottenuti presentano la seguente configurazione:

- cluster 0, 24369 samples
- cluster -1, 108 samples

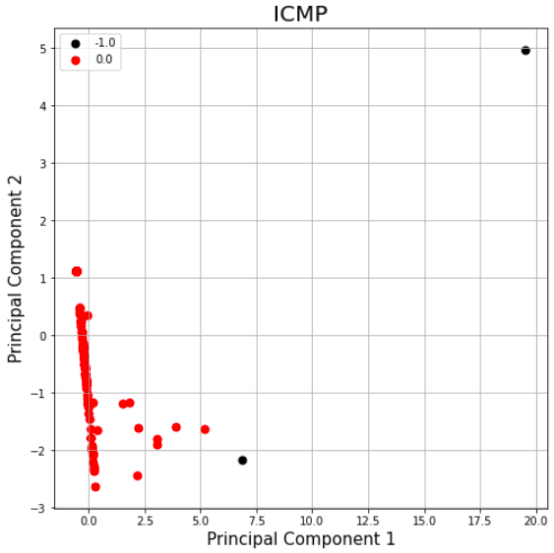


Risultato ottenuto per UDP con finestra temporale 10 e con i seguenti parametri:

- eps = 3
- min_samples = 5

Presenta un silhouette score pari a 0.89.
I cluster ottenuti presentano la seguente configurazione:

- cluster 0, 3815 samples
- cluster 1, 33 samples
- cluster -1 18 samples



Risultato ottenuto per ICMP con finestra temporale 60 e con i seguenti parametri:

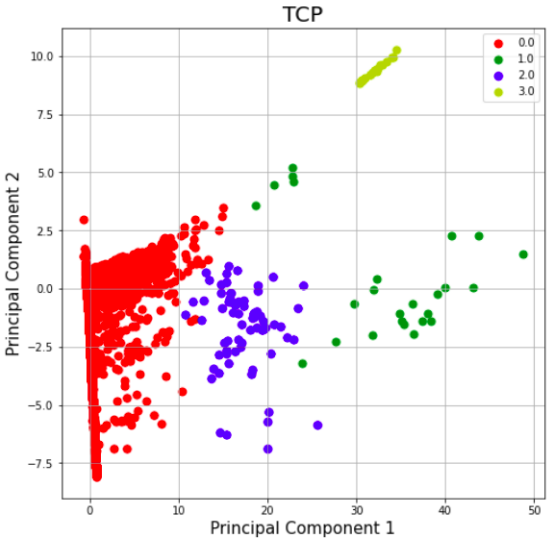
- eps = 3
- min_samples = 5

Presenta un silhouette score pari a 0.86.

I cluster ottenuti presentano la seguente configurazione:

- cluster 0, 168 samples
- cluster -1, 2 samples

BIRCH

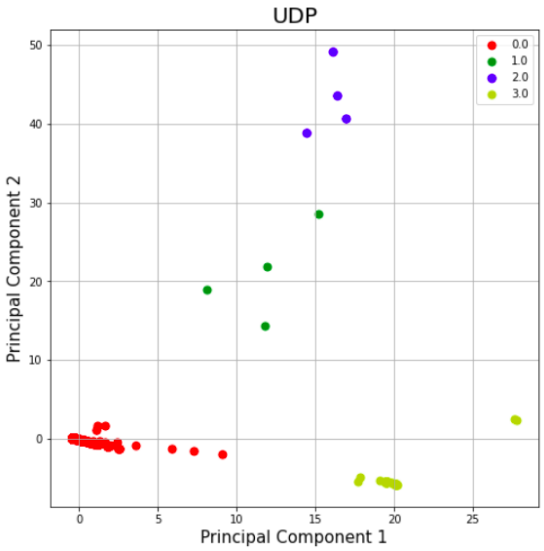


Risultato ottenuto per TCP con finestra temporale 180 e con i seguenti parametri:

- numero cluster=4
- threshold=0.4
- branching factor=50.

Presenta un silhouette score pari a 0.88.
I cluster ottenuti presentano la seguente configurazione:

- cluster 0, 24362 samples
- cluster 1, 25 samples
- cluster 2, 70 samples
- cluster 3, 20 samples

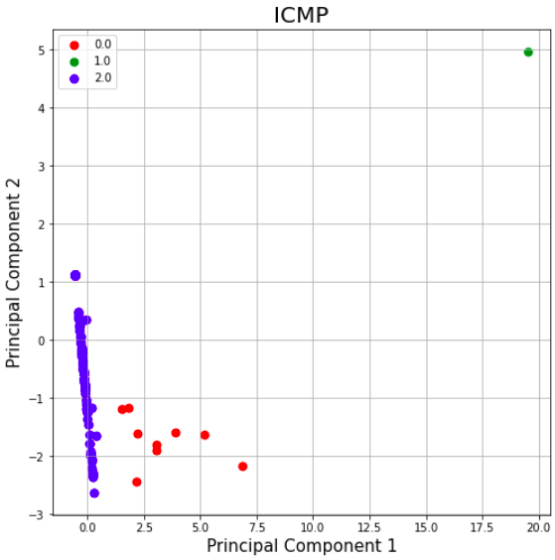


Risultato ottenuto per UDP con finestra temporale 10 e con i seguenti parametri:

- numero cluster=4
- threshold=0.5
- branching factor=50.

Presenta un silhouette score pari a 0.898. I cluster ottenuti presentano la seguente configurazione:

- cluster 0, 3822 samples
- cluster 1, 4 samples
- cluster 2, 4 samples
- cluster 3, 36 samples



Risultato ottenuto per ICMP con finestra temporale 60 e con i seguenti parametri:

- numero cluster=3
- threshold=0.5
- branching factor=50.

Presenta un silhouette score pari a 0.666. I cluster ottenuti presentano la seguente configurazione:

- cluster 0, 9 samples
- cluster 1, 1 samples
- cluster 2, 160 samples

Commenti

Il processo di feature engineering ha permesso l'identificazione di modus operandi d'attacco che altrimenti, usando solo i dati presenti nel dataset originale, non sarebbe stato possibile identificare (per ulteriori dettagli consultare la sezione *Old Data* dei notebooks che si occupano di clustering).
Le migliori prestazioni complessive sembrano essere date da Birch che consente di trovare rispetto agli altri algoritmi utilizzati un numero maggiore di cluster a parità di prestazioni (per iCMP le prestazioni sono inferiori). Ciò potrebbe dipendere dal fatto che questo algoritmo di clustering meglio si adatta alle features ingegnerizzate.
In tutti gli esempi analizzati viene identificato 1 cluster a cui appartengono la maggior parte dei samples e pochi cluster piccoli: questa tendenza potrebbe essere legata al fatto che nella finestra temporale in cui sono stati registrati gli eventi verificatisi nella honeypot effettivamente solo pochi utenti hanno usato un pattern d'attacco diverso da quello utilizzato dalla maggioranza degli altri utenti oppure questi samples sono semplicemente del rumore come suggerirebbe dbscan (il quale identifica 1 solo cluster per TCP e ICMP, 2 per UDP).

Early Classification

Vengono di seguito riportati brevi commenti riassuntivi sulle performance

registrate in materia di early classification di modus operandi con riferimento ai cluster presentati al punto precedente. In particolare viene valutata l'accuracy con riferimento a first_port_scan poiché unica feature tra quelle utilizzate per fare clustering che può essere effettivamente usata per early classification. Tutte le altre features estratte sono relative a dati consuntivi.

K-Means
L'accuracy del modus operandi con più samples risulta essere molto elevata sia per TCP (0.99) che per UDP (0.99) . Contrariamente l'accuracy per i restanti modus operandi è molto scarsa.

DBSCAN
L'accuracy del modus operandi con più samples risulta essere molto elevata sia per TCP (0.99) che per UDP (0.98) . Contrariamente l'accuracy per i restanti modus operandi è molto scarsa.

BIRCH
L'accuracy del modus operandi con più samples risulta essere molto elevata sia per TCP che per UDP (0.99 circa). Contrariamente l'accuracy per i restanti modus operandi è molto scarsa.

Commenti
In generale la early classification basata sulla feature ***first_port_scan*** funziona bene sia per TCP che per UDP indipendentemente dalla finestra temporale scelta, solamente, però, per il cluster più denso. Essendo il cluster più denso il modus operandi più probabile a cui un'istanza d'attacco possa appartenere l'utilità di questa previsione è limitata.

References

- https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers
- https://scikit-learn.org/stable/modules/clustering.html
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html#sklearn.cluster.DBSCAN
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html#sklearn.cluster.Birch