UNIVERSITÀ
DEGLI STUDI
DI PALERMO

NDS
NETWORKING AND DISTRIBUTED SYSTEMS
RESEARCH GROUP

Metodi di Machine Learning per l'Identificazione di Anomalie nel Traffico di Backbone

Tesi di Laurea Magistrale in Ingegneria Informatica

Marco Virecci Fana

Relatore: Prof. Giuseppe Lo Re
Correlatore: Ing. Marco Ortolani

Indice

Capitolo 1

Introduzione

Dal 1936, anno in cui Alan Turing inventò il primo modello teorico di calcolatore elettronico, la ricerca ha condotto allo sviluppo di macchine capaci di automatizzare sempre più attività che prima richiedevano l'intervento umano. Tale sviluppo, progressivo e continuo, che secondo la di legge Moore procede con velocità esponenziale, non deriva soltanto dalle prestazioni dei dispositivi hardware o dall'ottimizzazione del software, ma è legato altresì alla nascita di una rete pubblica, Internet, capace di interconnettere i sistemi informatici e gli utenti che li utilizzano. Questa rete, cioè Internet, da una parte ha contribuito all'introduzione di servizi innovativi in numerosi ambiti, come quelli finanziari, medici e industriali. Dall'altra, ha visto la comparsa di una serie di minacce, prima inesistenti, che riguardano la privacy degli utenti e la sicurezza dei dati. Infatti, qualunque sistema informatico per quanto affidabile non sarà mai esente da imperfezioni legate alla sua progettazione o al suo sviluppo che lo rendono vulnerabile ad attacchi di terzi. Tali considerazioni hanno spinto alla ricerca di contromisure al fine di tutelare questi sistemi da possibili attacchi informatici. In particolare, la recente letteratura del settore si è concentrata sull'elaborazione di metodi per garantire l'*integrità* e la *confidenzialità* delle informazioni. Nello specifico caso dell'analisi del traffico di rete l'obiettivo è quello di identificare presunte *anomalie* al fine di isolarle.

I sistemi dedicati a questi compiti rientrano nella classe degli **Intrusion Detection System** (**IDS**) e appartengono a due categorie principali: i sistemi

signature-based e quelli *anomaly-based*.

La prima categoria basa l'efficacia del rilevamento dell'attacco sulle *signature*, ovvero regole che caratterizzano, con un grado variabile di precisione, strutture di comportamenti malevoli, precedentemente documentati. Questi sistemi soffrono di diversi limiti legati all'incapacità di rilevare anomalie non note nella propria base di regole, che di conseguenza richiedono un costante aggiornamento per il rilevamento di attacchi nuovi. La seconda, invece, si basa sull'*anomaly detection*, metodologia applicata a numerosi domini ed in continuo sviluppo, che riguarda, in particolare, l'analisi e il monitoraggio dei dati per il rilevamento di eventuali comportamenti non conformi ad un prefissato modello di normalità. Questa metodologia, negli sviluppi più recenti, si avvale di algoritmi di machine learning che consentono, attraverso metodi supervisionati e non-supervisionati, di effettuare una classificazione dei dati.

Gli algoritmi supervisionati necessitano di una prima fase, detta di *addestramento*, che consente all'algoritmo, sulla base dei dati forniti in ingresso, di generare un modello che li rappresenta. Nel caso che qui verrà considerato, i dati forniti all'algoritmo sono i flussi di traffico etichettati come normali o anomali. Un punto critico di questa fase è la difficoltà di selezionare quelle caratteristiche del traffico, dette anche *features*, che portano un'informazione utile per una corretta classificazione. La maggior parte dei metodi di machine learning applicati al traffico di rete basano la classificazione sulle informazioni presenti nell'header dei pacchetti, in particolare gli indirizzi IP e le porte sorgente e di destinazione. La seconda fase tipica degli algoritmi supervisionati, invece, riguarda il *testing* e consente di valutare la capacità dell'algoritmo di classificare non solo le anomalie note, ma anche quelle originariamente non presenti nei dati usati durante l'addestramento. Il componete software principale di un IDS anomaly-based implementa un particolare algoritmo di machine learning che consente la classificazione del traffico, tale componente viene chiamato *detector*. Negli ultimi anni gli IDS *anomaly-based* sono stati al centro della ricerca perché considerati uno dei principali approcci computazionalmente trattabili per gestire il numero crescente di minacce. Ciononostante, anche

questi sistemi risultano vulnerabili ai più recenti tipi di attacchi, che oggi comprendono quelli noti come Adversarial Machine Learning, ovvero attacchi basati sulla capacità di chi effettua l'intrusione di manipolare il processo di attacco in modo da mascherare quei comportamenti che consentono al detector di rilevarlo.

Il presente lavoro di tesi prende spunto da uno studio dei ricercatori Fontugne e Borgnat che riguarda il progetto MAWILab. La tecnica esposta si basa sulla combinazione di strategie che integrano i risultati ottenuti da diversi detector al fine di ottenere migliori performance complessive. La classificazione fatta dai *detector*, che in una prima fase si limita a contrassegnare il traffico anomalo associandolo ad un allarme, avviene con diversi livelli di granularità a seconda dell'algoritmo implementato. L'uso di questa strategia ha il vantaggio di aumentare lo spettro di anomalie rilevabili, poiché agisce sui dati ad un livello di dettaglio differente. Una conseguenza negativa dell'uso di più *detector* è la possibile sovrapposizione degli allarmi, infatti *detector* differenti possono associare allarmi diversi a flussi di traffico simili o uguali. Per gestire questo inconveniente si utilizza una struttura dati, denominata *grafo di somiglianza*, che associa ad ogni nodo un allarme e ad ogni arco un peso che evidenzia le similarità tra i diversi allarmi. Una volta realizzato il grafo di somiglianza un ulteriore componente si occupa di raggruppare gli allarmi che presentano un elevato valore di similarità. In questo modo si hanno delle macro classi di allarme che includono diversi flussi di traffico valutati mediante un sistema di voto che ne determina l'effettiva pericolosità.

La tecnica appena descritta risulta essere molto efficiente, non solo perché capace di ampliare lo spettro di anomalie intercettabili, ma anche perché capace di rendere meno semplice un attacco di tipo Adversarial Machine Learning.

Durante il lavoro di tesi è stato preso in considerazione un particolare *detector* con lo scopo di implementarlo e valutarne l'efficienza. Il *detector* in questione si basa sull'emergente teoria dei data streams, la quale propone il ricorso a tecniche e modelli matematici atti a creare delle *random traffic projections* (dette anche *sketches*). L'obiettivo di questa tecnica è quello di ridimensionare

lo spazio dei dati e far emergere soltanto le caratteristiche del traffico che portano un'informazione utile per la classificazione. Questo metodo prevede l'uso di diverse funzioni hash che consentono, mediante una particolare struttura dati, di dividere il traffico principale in diversi sotto traffici, senza alterarne significativamente le proprietà. I sotto traffici così ottenuti vengono memorizzati in liste di IP poi classificate mediante un algoritmo non-supervisionato. L'intersezione delle liste di IP classificate come anomale, consente di risalire all'indirizzo IP malevolo. In particolare, il metodo realizzato prevede l'uso della Principal Component Analysis (PCA) come algoritmo per la classificazione. Il lavoro finale della tesi consiste nella valutazione dell'algoritmo realizzato, per fare la quale è stato usato il traffico ottenuto dal dataset MawiLab, una collezione pubblica di tracce di traffico di rete raccolte ogni giorno, da più di dieci anni, da una backbone che collega il Giappone con l'America. Questa tipologia di traffico presenta delle complicazioni legate al confluire, all'interno di una backbone, di diversi traffici appartenenti a diverse sotto reti e quindi apparentemente eterogenei. Un'ulteriore prova del corretto funzionamento del *detector* implementato è data dai test effettuati basandosi su traffico generato artificialmente con una distribuzione degli indirizzi IP definita a priori. Tale sistema potrebbe essere completato con lo sviluppo degli altri detector ed adattato per l'analisi del traffico veicolato dalla dorsale del campus universitario di Palermo.

Capitolo 2

La sicurezza informatica

Possiamo definire sicuro un sistema informatico se garantisce la *confidenzialità*, l'*integrità* e la *disponibilità* dei dati che gestisce.
Con il termine confidenzialità si intende la protezione dei dati e delle informazioni scambiate tra un mittente e uno o più destinatari nei confronti di terze parti. In un sistema che garantisce la confidenzialità, una terza parte che entra in possesso delle informazioni scambiate tra mittente e destinatario, non è in grado di ricavarne alcun contenuto informativo intelligibile.

Figura 2.1: Confidentiality, integrity, availability

Con il termine integrità si intende la protezione dei dati e quindi delle informazioni nei confronti delle modifiche del contenuto effettuate da terzi, essendo compreso nell'alterazione anche il caso limite della generazione ex novo di dati

ed informazioni. Insito nel concetto di integrità vi è la possibilità di verificare con assoluta certezza se un dato o una informazione sono rimasti integri, ossia inalterati nel loro contenuto, durante la loro trasmissione. In un sistema che garantisce l'integrità, l'azione di modifica del contenuto, da parte di una entità non autorizzata, viene quindi rilevata.
Infine con il termine disponibilità si intende la prevenzione della non accessibilità, ai legittimi utilizzatori, sia delle informazioni che delle risorse, quando informazioni e risorse servono. Il concetto quindi, oltre che riguardare dati ed informazioni, è esteso a tutte le possibili risorse che costituiscono un sistema informatico, come, ad esempio, la capacità di calcolo di un elaboratore, la banda di trasmissione di un collegamento e lo spazio utile di memorizzazione dei dati.

2.1 Impatti dell'hacking

Vista la crescente diffusione della connettività ad Internet, alle numerose possibilità che questa offre e alla proliferazione di nuovi scenari che sfruttano intensivamente la rete per lo scambio di informazioni, quali l'*Internet of Things* [1], sempre più sistemi sono vittime di attacchi e tentativi di intrusione. Il fine di questi attacchi è quello di violare una o più condizioni che un sistema informatico sicuro dovrebbe garantire. Come riportato dal Computer Emergency Response Team/Coordination Center (CERT/CC), il numero di attacchi a sistemi informatici ha subito una crescita esponenziale negli ultimi anni (si veda il grafico 2.2).

Inoltre, la complessità e la precisione degli attacchi stanno anch'esse aumentando. Alcuni anni fa chi tentava di violare sistemi informatici aveva una profonda conoscenza della materia, dei sistemi e delle reti che gli permettevano di effettuare attacchi, oggi chiunque può sfruttare una vulnerabilità di un sistema o di una rete utilizzando uno dei numerosi *tool* facilmente disponibili online.
Da uno studio condotto da McAfee [2] si evince che le perdite finanziarie an-

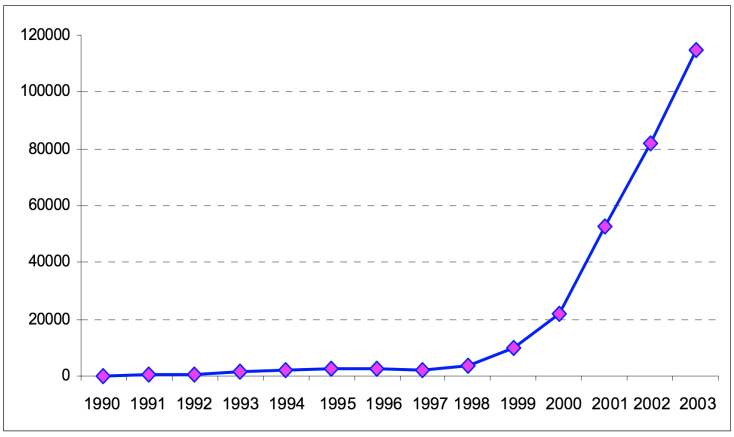


Figura 2.2: Crescita degli incidenti informatici riportati dal Computer Emergency Response Team/Coordination Center (CERT/CC)

nuali dovute a virus ammontano a 100 miliardi di dollari l'anno per l'America e 300 miliardi di dollari per il resto del mondo. I virus e i worm rappresentano una tremenda minaccia per la sicurezza delle organizzazioni e delle aziende, nonostante queste si organizzino in diversi modi per contrastarla.
Nonostante gli innumerevoli investimenti analizzando più nel dettaglio i vari report promulgati, emerge tuttavia un dato preoccupante: soltanto meno dell'1% degli attacchi riesce ad essere individuato dai sistemi di sicurezza. E addirittura il 95% delle intrusioni non autorizzate sono possibili semplicemente rubando le credenziali dei dipendenti dell'azienda, un'attività questa che viene spesso effettuata dall'interno della compagnia stessa e non può quindi essere bloccata dal firewall o da sistemi simili [3].Ciò significa che non è affidandosi solamente agli *antivirus* e ai *firewall* che le compagnie sono in grado di proteggersi dai cyber criminali. Ed è proprio per questo motivo che l'attenzione pubblica si sta focalizzando sempre di più sul settore degli **Intrusion Detection System**.

2.2 Tassonomia degli attacchi

Intrusioni informatiche e attacchi sono spesso considerati sinonimi, ma in letteratura sono state date definizioni di attacco che differenziano i due concetti. Un attacco è un tentativo di intrusione e una intrusione è il risultato di un attacco almeno in parte riuscito.
Sono state fatte numerose classificazioni degli attacchi e delle intrusioni, la tassonomia proposta è ottenuta esaminando e combinando le diverse categorie e tassonomie degli attacchi ai sistemi ed alle reti, pubblicati nella letteratura che riguarda gli Intrusion Detection System [4].
La tassonomia proposta prevede la seguente classificazione:

- Tipo di attacco
- Numero di connessioni alla rete coinvolta nell'attacco
- Ambiente in cui avviene l'attacco
- Livello di automazione

2.2.1 Tipo di Attacco

Il criterio più comune per classificare gli attacchi informatici e le intrusioni si basa sulla tipologia.

- **Denial of Service (DoS)**: questi attacchi colpiscono la disponibilità dei servizi rendendo inutilizzabile la rete, un particolare sistema o le risorse messe a disposizione per degli utenti. Ci sono due tipi di attacchi DoS:
  - attacchi al sistema operativo, le cui vulnerabilità possono essere risolte con l'uso delle patch
  - attacchi alla rete, che sfruttano le limitazioni dei protocolli di rete e delle relative infrastrutture.
- **Probing (surveillance, scanning)**: questi attacchi prevedono scansioni della rete con lo scopo di identificare IP validi sui quali raccogliere

informazioni. Queste informazioni spesso forniscono a chi attacca il sistema una lista di possibili vulnerabilità che possono essere successivamente utili per lanciare attacchi a sistemi e servizi. Questi attacchi sono probabilmente i più comuni in quanto sono i precursori di attacchi di altro tipo.

- **Compromissine**: questi attacchi sfruttano le vulnerabilità conosciute come per esempio buffer-overflow e punti di debolezza dei sistemi, principalmente software, per ottenere un accesso ed eventualmente accrescere i propri privilegi. In base alla sorgente di attacco, interna piuttosto che esterna, la compromissione del sistema può essere classificata in queste due categorie:
  - R2L (Remote to Local) questi tipi di attacchi prevedono che un attaccante che è in grado di mandare pacchetti ad un sistema in rete (ma a cui non ha accesso non avendo un account) riesca ad ottenere un accesso. Esempi tipici di attacchi R2L sono: password-guessing, buffer overflow.
  - U2R (User to Root) questi tipi di attacchi prevedono che un attaccante con un account sul sistema sia in grado di aumentare i propri privilegi sfruttando una vulnerabilità, un bug nel sistema operativo o in una applicazione installata nel sistema. Diversamente dagli attacchi R2L, in cui l'intruso cerca di entrare nel sistema, negli U2R l'attaccante è già un utente del sistema che intende però aumentare i propri privilegi al massimo diventando amministratore (root).

Infine definiamo **Virus/Worms/Trojan horse** tutti i programmi che cercano di replicarsi e propagarsi tramite la rete.
I Virus sono programmi che si riproducono attaccandosi ad altri programmi e infettandoli. Possono causare parecchi danni ai sistemi ma possono anche essere innocui e causare solo fastidiosi imprevisti. Per replicarsi tipicamente necessitano di una interazione umana.
I Worm al contrario sono programmi che si replicano autonomamente e che si

diffondono velocemente attraverso la rete grazie a servizi automatici di invio e ricezione di pacchetti comunemente presenti nella maggior parte dei computer. I Worm possono essere divisi nelle seguenti categorie:

- Worm tradizionali: solitamente utilizzano le comuni connessione di rete per diffondersi e non necessitano dell'intervento di nessun utente
- Email worm: infettano gli altri computer della rete sfruttando i contatti email dell'utente o altre applicazioni client
- Windows file sharing worms: si replicano utilizzando MS Windows peer-to-peer service, che si attiva tutte le volte che nel sistema viene rilevato un nuovo device
- Hybrid worms: tipicamente sfruttano diversi tipi di vulnerabilità.

Alcuni worm recenti vengono anche utilizzati per attacchi di tipo DoS. Infine i Trojan Horse vengono definiti come applicazioni malevoli che tentano di danneggiare la sicurezza di un sistema mascherandosi come applicazioni lecite e non dannose.

2.2.2 Numero delle connessioni

Gli attacchi informatici possono essere classificati in base al numero di connessioni di rete coinvolte. Esempi di attacchi che coinvolgono più connessioni di rete sono DoS, probing e worm. Gli attacchi che coinvolgono una singola connessione o un limitato numero di connessioni di rete di solito sono la causa della compromissione di un singolo sistema, come per esempio gli attacchi di tipo buffer-overflow remoti.
La maggior parte degli attacchi parte da un singolo sistema, ma nei casi di attacchi di Denial of Service distribuiti (DDoS) o altri attacchi organizzati, si hanno diverse sorgenti che possono partecipare all'attacco. Spesso oltre a partire da diversi sistemi, questi attacchi hanno anche diversi obiettivi. Rilevare questi attacchi distribuiti richiede uno studio e una correlazione delle informazioni ottenute dalla rete molto onerosa e non sempre efficace.[4]

2.2.3 Il contesto

Gli attacchi informatici possono essere classificati in base all'ambiente nel quale avvengono.

Le *intrusioni di tipo host* sono rivolte ad un singolo, non sempre connesso ad una rete e possono essere rilevate analizzando le informazioni e i dati presenti sul dispositivo. L'identità di chi ha tentato l'attacco è tipicamente associata ad un username, ed è quindi più semplice rilevare il colpevole.

Le *intrusioni di rete*, invece, sono intrusioni che vengono effettuate tipicamente dall'esterno dell'organizzazione o dell'azienda e il rilevamento di queste avviene analizzando le informazioni contenute nel traffico di rete generato. Tuttavia queste analisi spesso non sono in grado rivelare la precisa identità di chi ha attaccato il sistema, perché non sempre è possibile trovare una corrispondenza biunivoca tra chi attacca e la connessione.

Le *intrusioni nei sistemi peer-to-peer* sono intrusioni che avvengono in sistemi che operano come peer nella rete Internet. Queste applicazioni sono sempre più diffuse, soprattutto per la condivisione di file, proprio la loro diffusione però può agevolare la condivisione di codice eseguibile tra diversi host appartenenti ad organizzazioni o aziende interconnesse.

Le *intrusioni in reti wireless* possono essere generalmente rilevate analizzando il traffico che viene effettuato tra i sistemi coinvolti e il relativo punto di accesso wireless. Tipicamente le minacce delle reti wireless possono essere separate in:

- Eavesdropping: quando viene semplicemente intercettato il traffico di una comunicazione
- Intrusioni: quando si tenta di accedere al sistema o modificare i dati

- Communication hijacking: quando un finto nodo cattura tutto il traffico del canale di comunicazione, o un nodo fa da punto di accesso fittizio per attirare tutti i nodi e raccoglierne i dati confidenziali inviati

- Denial of service o attacco jamming: quando viene disturbato il canale di comunicazione con vari domini di frequenza, oggetti fisici e ostacoli portando alla caduta della comunicazione per quei canali

2.2.4 Livello di automazione

In base al livello di automazione dell'attacco possiamo avere le seguenti categorie:
Attacchi automatici: utilizzano strumenti in grado di effettuare le operazioni basilari di probing e scanning su numerosi IP in tempi brevi. Utilizzando questi tool facilmente reperibili anche un attaccante inesperto è in grado di effettuare un attacco anche sofisticato. Questi tipi di attacco sono probabilmente i più comuni al giorno d'oggi. *Attacchi semi-automatici*: utilizzano script per le fasi di scanning e per compromettere il sistema o la rete, per l'installazione del codice d'attacco, e infine usano il sistema principale o master che gestisce tutti gli altri per specificare il tipo di attacco e l'indirizzo della vittima. *Attacchi manuali*: prevedono una fase di scanning manuale dei sistemi vittima che tipicamente richiede molto tempo e una conoscenza approfondita e una fase in cui conoscendo le vulnerabilità si scrive il codice per l'attacco. Attacchi di questo tipo non sono molto frequenti, ma sono tipicamente più dannosi e difficili da rilevare rispetto ai semiautomatici o gli automatici, in quanto danno all'attaccante più controllo sulle risorse. Esperti e gruppi organizzati di attaccanti generalmente usano questi tipi di attacchi per intrusioni in sistemi di importanza critica.

2.3 Mezzi di contrasto

Sono molti i meccanismi utilizzabili per cercare di garantire il rispetto delle politiche di sicurezza definite in un determinato sistema. Essenzialmente questi possono essere suddivisi in 4 categorie principali:

- Attack prevention: di questa classe fanno parte tutti quei meccanismi ideati per prevenire o difendersi da certi tipi di attacchi, prima che questi possano raggiungere ed avere effetto sul bersaglio. Sicuramente l'elemento più importante di questa categoria è il controllo dell'accesso, un meccanismo applicabile a differenti livelli che limita e regola l'accesso a risorse critiche. Ciò è generalmente fatto identificando o autenticando la parte che richiede la risorsa, e verificando i suoi permessi relativamente all'oggetto richiesto. Un altro meccanismo molto utilizzato appartenente a questa categoria è il firewall, che lavora però ad un livello differente.

- Attack avoidance: i meccanismi di sicurezza presenti in questa categoria partono dall'ipotesi più pessimistica, cioè che un intruso possa accedere alla risorsa desiderata, quindi il loro obiettivo è quello di modificare la risorsa in modo che, anche se un utente non autorizzato riuscisse nel suo tentativo di accedervi, questa risulterebbe inutilizzabile. Un metodo di tale tipo, applicato ad esempio alla protezione dell'informazione in transito su di una rete, prevede che i dati vengano preprocessati dal mittente e quindi postprocessati dal lato del ricevente. In questo modo, mentre l'informazione attraversa la rete, è in grado di resistere agli attacchi, risultando praticamente inutile per un malintenzionato. Un classico esempio di sicurezza tramite attack avoidance è la crittografia.

- Attack detection: questi meccanismi partono dall'ipotesi che un attaccante possa ottenere l'accesso agli obiettivi presi di mira e riesca a violare le politiche di sicurezza del sistema; il loro compito è quello di rilevare il sopraggiungere di una situazione pericolosa e di creare report adeguati, così che l'amministratore del sistema possa rendersi conto di ciò che è

accaduto. Se da un lato queste tecniche hanno il vantaggio di poter operare assumendo il verificarsi del caso pessimo (un attacker che guadagna l'accesso al suo obiettivo), dall'altro non sono ovviamente in grado di garantire in nessun modo la confidenzialità delle informazioni; il loro uso è perciò di grande utilità quando combinato con altri meccanismi di difesa. I membri più importanti di questa classe di tecniche difensive sono ovviamente i sistemi di intrusion detection, sui quali è incentrata questa tesi.

- Attack tolerance: è un meccanismo difensivo che si basa, in prima istanza, sull'attack detection; dopo che un attacco è stato rilevato, nel caso sia andato a segno, i meccanismi di attack tolerance cercano di garantire che il sistema, benché si trovi in una situazione potenzialmente instabile, continui ad erogare un servizio corretto. Poiché, come detto in precedenza, le misure preventive non sono sufficienti a garantire una sicurezza totale, l'attack tolerance aiuta ad avere una certa continuità di servizio anche quando le misure di sicurezza sono state scavalcate.

L'argomento, oltre ad essere estremamente interessante, è intimamente legato all'intrusion detection, per cui, anche se non verrà trattato approfonditamente nel corso della tesi, se ne dà qui un superficiale accenno. Perché un sistema possa essere tollerante ad un attacco, questi deve essere in grado di "contenerlo", garantendo così che l'attacco non possa avere un impatto troppo negativo sulla normale erogazione dei servizi del sistema. Rilevato l'attacco, tre possibili metodologie per raggiungere l'obiettivo specificato sono:

- Rollback recovery: il sistema viene riportato ad un suo stato precedente, in cui l'attacco non era ancora stato effettuato, attraverso una copia salvata in precedenza di tale stato; tale copia viene detto punto di ripristino o checkpoint. Esempi di tale approccio sono: il riavvio del sistema con reinizializzazione dei processi attivi, oppure la chiusura di una connessione TCP, o ancora la re-installazione del sistema operativo.

- Rollforward recovery: in questo caso il sistema viene fatto avanzare di stato in stato, finché non ne venga individuato uno sicuro a partire dal quale questi possa nuovamente operare. Un tale approccio viene ad esempio seguito quando chiavi crittografiche violate sono automaticamente sostituite da nuove chiavi, generate appositamente per porre rimedio alla falla creatasi nei meccanismi di sicurezza, oppure quando il sistema viene fatto evolvere in uno stato, presunto safe, in cui le sue funzionalità sono limitate.

2.4 I Firewall

Da sempre il firewall ha rappresentato la prima linea di difesa per le reti Internet. Si tratta infatti di un dispositivo software, hardware oppure ibrido che si occupa di ispezionare tutti i pacchetti che entrano o escono dalla rete interna, la LAN, e poi a seconda del loro contenuto decide se accettare o meno il pacchetto sulla base di un insieme di regole predefinite che definiscono una *policy* di sicurezza. La politica standard adottata da tutti i firewall è la cosiddetta *default-deny* che consiste nel vietare a priori tutto il traffico, andando a stabilire un insieme di regole per determinare quale sono gli unici tipi di pacchetti che possono essere accettati e fatti passare attraverso il firewall. Se un messaggio non corrisponde a nessuna delle regole prestabilite, verrà bloccato dal firewall.

Attualmente esistono diversi tipi di tecniche che un firewall può utilizzare per bloccare il traffico indesiderato quali ad esempio il *packet filtering*, l'uso di un server proxy, *l'application-level gateway* o il NAT. Molti di questi metodi filtrano i contenuti analizzando singolarmente ogni pacchetto che attraversa il firewall, senza tenere conto dei precedenti e considerando solamente alcune delle informazioni contenute nell'header(l'indirizzo IP della sorgente, l'indirizzo IP della destinazione, la porta della sorgente, la porta della destinazione e il protocollo di trasporto). I firewall che costruiscono le proprie policy di sicurezza basandosi su questo tipo di filtraggio dei pacchetti semplice e leggero

vengono definiti *packet filter firewall* o *stateless firewall*. Altra tipologia molto diffusa di firewall sono gli *application firewall* o *proxy firewall*, che operano invece un filtraggio dei pacchetti di tipo applicativo, andando cioè a filtrare tutto il traffico di una singola applicazione sulla base della conoscenza del suo protocollo. Questo tipo di firewall analizza i pacchetti nella loro interezza considerando anche il loro contenuto ed è quindi in grado di distinguere il traffico di un'applicazione da quello di un'altra, cooperando anche molto spesso con altri sistemi per il rilevamento delle intrusioni (IDS).In base a quanto detto, il firewall è dunque essenzialmente un componente di difesa perimetrale, poichè si trova posto al confine tra due reti per le quali funge da collegamento: una è la rete esterna, che comprende interamente Internet e si suppone non sia né sicura né affidabile; l'altra è la rete interna, o LAN, che può essere invece considerata tale proprio grazie all'azione di difesa del firewall.

Capitolo 3

Intrusion Detection System

L'approccio convenzionale per mettere in sicurezza un sistema informatico è quello di organizzare un'architettura formata da diversi dispositivi hardware e applicativi software, in grado di creare una barriera protettiva che rende vano ogni tentativo di attacco. Fanno parte di questa architettura il firewall, svariati meccanismi di autenticazione, l'uso di Virtual Private Network (VPN) e della crittografia.

Tuttavia, questa infrastruttura, per quanto possa essere progettata con attenzione, non è sempre la soluzione definitiva e non riesce a bloccare svariate tipologie di attacco.

Gli attacchi informatici, infatti, sono il frutto di un costante studio basato sulle vulnerabilità dovute alla cattiva progettazione del software o ad errori di implementazione presenti negli applicativi e nei dispositivi hardware interconnessi in una rete.

Sulla base di quanto appena detto nasce la necessità di una tecnologia che sia in grado di monitorare i sistemi e di individuare i tentativi di intrusione. Questa tecnologia viene chiamata Intrusion Detection System (IDS) ed ha la particolare proprietà di essere indipendente dai sistemi appena descritti o dall'infrastruttura in cui è implementata.

Il National Institute of Standard and Technology (NIST) definisce un intrusion detection system come "un processo atto a monitorare gli eventi che avvengono in un sistema o il traffico di una rete, analizzando i quali è possibile rilevare

segni di intrusioni, definiti come tentativi di compromissione della confidenzialità, integrità, disponibilità, o per aggirare i meccanismi di sicurezza di un sistema o di una rete."
In breve, un IDS può essere definito anche come una combinazione di componenti software e/o hardware in grado di monitorare sistemi e di lanciare eccezioni quando avviene una intrusione.
Il primo modello di intrusion detection fu sviluppato dal Dorothy Denning al SRI International nel 1983 , da quel momento in poi, moltissimi modelli di IDS sono stati progettati e sviluppati sia nel mondo commerciale sia nel mondo della ricerca accademica. Tuttavia questi modelli sono estremamente diversi nelle tecniche impiegate e nel modo di acquisire e analizzare i dati, ma molti di essi si basano su di un'architettura standard formata da componenti con i seguenti compiti:

- **Data Gathering**: sono dei sensori che hanno come compito la raccolta delle informazioni presenti nella rete.
- **Detector**: analizzano le informazioni raccolte dai sensori per identificare attività intrusive.
- **Database**: sono delle basi di dati che contengono le informazioni raccolte dai sensori ma in un formato preprocessato.
- **Configurator Device**: contiene le informazioni dello stato corrente dell'IDS.
- **Respose Componen**: quando una intrusione viene rilevata esegue le azioni atte a segnalare l'allarme. Queste risposte possono essere automatizzate (IDS attivo) o possono richiedere l'intervento umano (IDS passivo).

3.1 Tecniche di detection

Nel corso degli anni sono state sviluppate numerose tecniche volte a scoprire anomalie nelle reti. Alla base di ogni tecnica di anomaly detection vi è la natura del dato che si deve elaborare. Ogni istanza può essere descritta usando

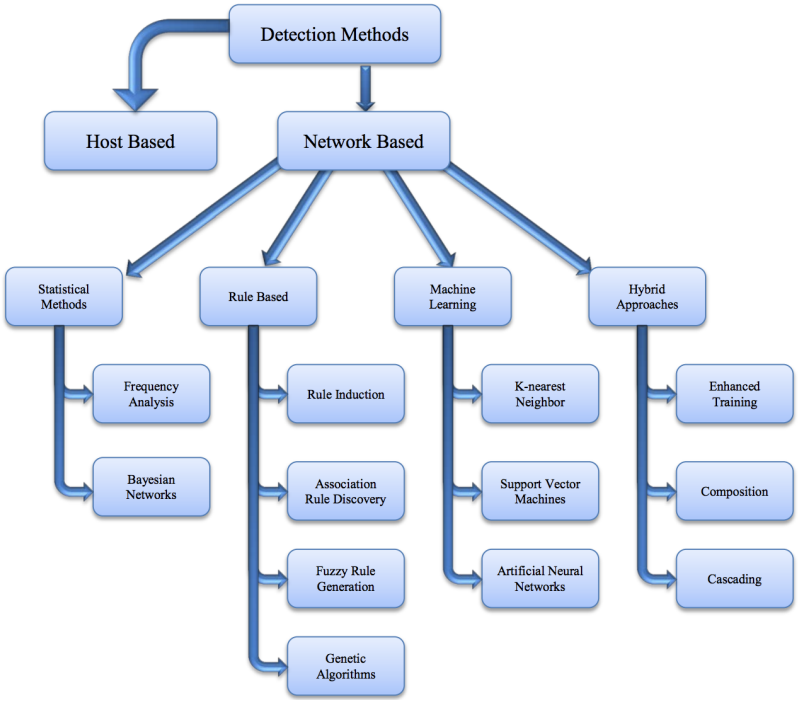


Figura 3.1

un insieme di attributi o features i quali possono essere binari, categorici o continui. Un'istanza può avere un solo attributo o più attributi. È quindi la natura di un attributo a determinare che tipo di tecnica può essere usata. Di seguito se ne riportano solo alcuni esempi:

- **Approccio Statistico**: quando viene sviluppato un algoritmo di detection di tipo statistico vuol dire che siamo interessati ad analizzare proprietà statistiche del traffico come ad esempio la media o la dimensione dei flussi, il rapporto tra flussi entranti e uscenti o la distribuzione degli indirizzi IP. Per scoprire cosa è anomalo, vengono osservate certe proprietà statistiche e se viene osservato un cambiamento delle stesse si può pensare che tale cambiamento sia causato da un'anomalia. Al fine di comprendere se è avvenuta una variazione nei punti di interesse, possono

essere utilizzate molte tecniche, tra cui i modelli regressivi o la Principal Component Analysis (PCA), che verrà trattata in questo lavoro. I metodi statistici si basano sull'assunzione che il comportamento normale avviene con un'elevata probabilità mentre quello anomalo ha una bassa probabilità di accadere.

- **Data Mining-based Methods**: processo computazionale mediante il quale è possibile scoprire patterns all'interno di un elevato numero di dati collezionati. Per rendere tale processo più rapido e automatizzato spesso vengono utilizzate tecniche di machine learning, una metodologia legata principalmente ad attributi di connessione come ad esempio porta sorgente, porta destinazione, indirizzo IP sorgente, indirizzo IP destinazione, protocollo e TCP flag. Conoscendoli è possibile costruire un classificatore. È spesso richiesto un dataset pre-etichettato.

- **Time Frequecy Analysis-based (TFA)**: Spesso utilizzato insieme all'approccio statistico. La differenza è che un TFA analizza le informazioni nel dominio della frequenza invece che nel dominio del tempo come avviene con l'approccio statistico. Tale metodo per prima cosa analizza un parametro statistico del traffico di rete, come ad esempio il tasso dei pacchetti. La TFA trasforma il parametro portandolo nel dominio della frequenza e successivamente cerca di rivelare eventuali anomalie. Quando viene rilevato qualcosa di strano nei pattern di traffico analizzati, viene alzato un alert (ad esempio, se si eccede una soglia pre-impostata). Tale metodo, come anche quello statistico, non forniscono informazioni chiave circa le anomalie riscontrate che possono essere utili per applicare contromisure da parte degli amministratori di rete.

- **Rule Based**: un anomaly detector impara regole che descrivono il normale comportamento del sistema. Un evento che non rientra nelle regole così definite è da considerarsi anomalo. Il primo passo di questa metodologia consiste nell'applicare un algoritmo di rule learning.

- **Machine Learning**: con il termine machine learning si intende l'abilità che ha un programma di migliorare gradualmente le performance di un certo numero di attività che deve svolgere. Esistono numerosi metodi come ad esempio Vector Machine (SVM) e reti neurali.

- Approccio Ibrido: diventa sempre più comune combinare i diversi approcci in un singolo modello. L'idea è quella di combinare i punti di forza dei differenti approcci descritti per massimizzare l'accuratezza e minimizzare i falsi positivi.

3.2 Modalità d'uso

Per monitorare il traffico in una rete alla ricerca di intrusioni, un IDS può essere utilizzato in due diverse modalità, che si differenziano principalmente in base alla posizione in cui il dispositivo viene installato:

- **modalità mirroring**: utilizzando le funzionalità di un router o di uno switch, il traffico di rete viene copiato e ridirezionato sull'IDS. Questo si occupa quindi di analizzare i pacchetti che gli sono stati inviati e, nel caso in cui venga rilevata un'attività sospetta, provvede a notificarla immediatamente all'amministratore del sistema. Operando solamente su una copia del traffico, un IDS di questo tipo non può in alcun modo bloccare o filtrare i pacchetti in ingresso e in uscita né tanto meno modificarli: esso non cerca di bloccare le eventuali intrusioni come ad esempio fa il firewall, ma si limita a rilevarle laddove si verifichino. Per questo motivo tale modalità viene anche più correttamente definita passiva e gli IDS così strutturati sono detti IDS passivi.

- **modalità in-line**: si impone che tutto il traffico di rete passi attraverso l'IDS. Questo può così avere il pieno controllo su tutti i pacchetti analizzati, svolgendo attività di prevention che cercano di bloccare le intrusioni identificate, evitando sul nascere l'attivazione di programmi

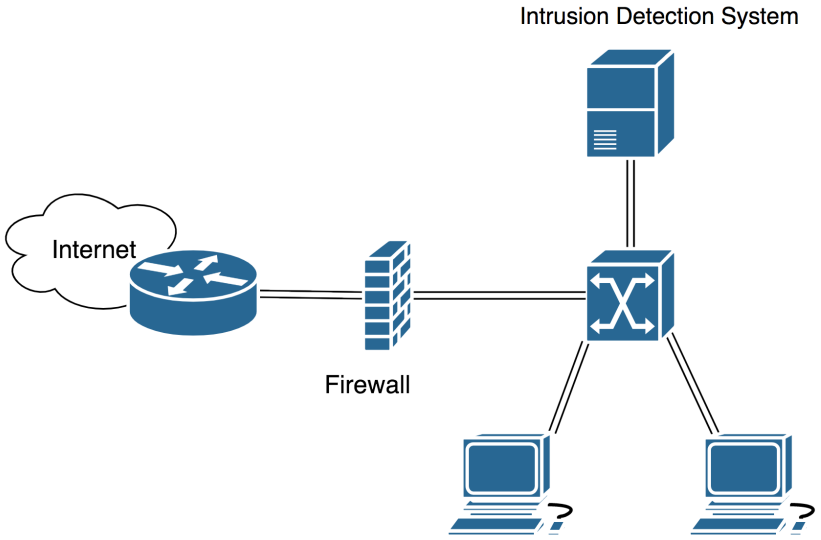


Figura 3.2

potenzialmente dannosi. Azioni di questo tipo che possono essere effettuate includono inviare un allarme (alert), eliminare pacchetti malevoli, resettare le connessioni o bloccare tutto il traffico proveniente da un determinato indirizzo IP. Ciò viene fatto, solitamente, basandosi su una lista di controllo degli accessi simile a quella utilizzata da un firewall, con la differenza che mentre quest'ultimo agisce a basso livello lavorando su porte e IP, un IDS è in grado di operare ad un livello più alto (all'incirca fino al livello 7 del modello ISO/OSI) e ha a che fare principalmente con programmi e utenti. In contrapposizione con gli IDS visti finora, che passivamente si limitano a segnalare una sospetta violazione nella rete senza compiere alcuna azione preventiva, si parla in questo caso di IDS attivi. Più comunemente però, per riferirsi alla capacità che questi sistemi hanno di rispondere alle intrusioni rilevate tramite attività di prevention, si è soliti parlare di intrusion detection and prevention system o IDPS. Gli IDPS sono solitamente considerati un'estensione degli IDS puri: entrambi controllano il traffico e le attività di sistema, ma solo i primi sono abilitati a prevenire e fermare le intrusioni.

In tutte le principali tassonomie dei sistemi di sicurezza, la differenza tra IDS e IDPS è esplicita e resa sempre piuttosto marcata. Ai fini pratici, tut-

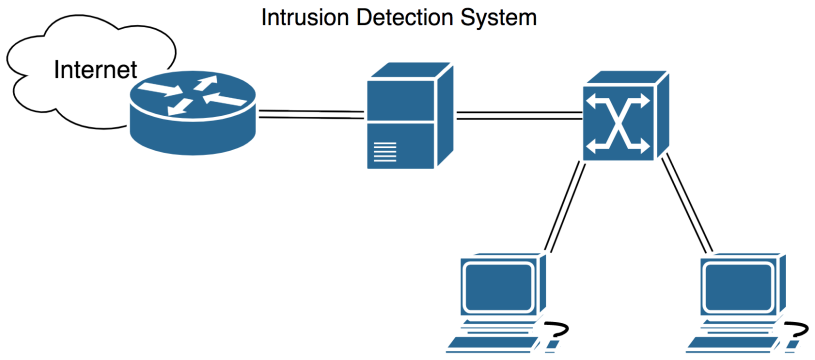


Figura 3.3

tavia, questi due sistemi possono essere tranquillamente assimilati in un'unica entità poiché alla situazione attuale praticamente ogni IDS utilizzato è dotato di capacità di prevention. D'ora in avanti quindi, quando si parlerà di IDS, faremo implicitamente riferimento sempre ad un IDPS. Giunti a questo punto, però, risulta abbastanza naturale farsi una domanda: perché non usare sempre la modalità in-line? Stando a quanto detto finora sembrerebbe infatti che questa sia del tutto identica al mirroring, con in più la possibilità di fare prevention. I motivi alla base di questa differenziazione sono essenzialmente due: Un dispositivo in-line necessita di hardware di rete estremamente avanzato e quindi molto più costoso rispetto alla tecnologia necessaria per operare in mirroring. Inoltre, un IDS che analizza il traffico "in linea" non solo deve essere molto potente ma anche a prova di guasto. Se, infatti, il dispositivo si danneggia ma sta lavorando in mirroring, non si ha nell'immediato alcuna conseguenza; ma se a smettere di funzionare è un sistema inline, allora tutto il traffico si bloccherà in quel punto, provocando gravi disservizi e malfunzionamenti all'intera rete. Di contro la configurazione in mirroring, richiedendo che venga effettuata una copia di tutto il traffico da monitorare, può ritrovarsi a dover sottoporre a sforzi e "stress" notevoli gli apparati di rete di un'azienda, specie se il traffico è molto elevato.

3.3 Misuse Detection e Anomaly Detection

In letteratura esistono due grandi categorie di approcci alla tematica dell'Intrusion Detection:

- **Anomaly Detection**: si basa sull'analisi del comportamento degli utenti e dei sistemi connessi alla rete col fine di tracciare un profilo definito non anomalo e legittimo. Successivamente in fase di detection il detector rileverà in real-time ogni comportamento che devia ciò che è stato definito non anomalo. Il vantaggio di questo tipo di approccio è dato dalla possibilità di rilevare attacchi sconosciuti, infatti questi sistemi non richiedono l'immissione di conoscenza a priori, né richiedono continui aggiornamenti delle "signature" degli attacchi. Questi sistemi sono infatti teoricamente in grado di rilevare i comportamenti anomali sulla base dell'addestramento ricevuto. Tuttavia, per costruire un modello di comportamento non anomalo serve innanzitutto uno studio architetturale preciso su quale tipo di modello usare, e in secondo luogo una fase più o meno prolungata di addestramento in cui il modello viene calibrato sullo specifico utente e sullo specifico sistema. Inoltre, questi sistemi sono affetti ad errori e falsi positivi. Infatti una deviazione dal comportamento normale non significa necessariamente che è avvenuto un attacco.

- **Misure Detection**: è stato per molti anni l'approccio maggiormente utilizzato dai sistemi commerciali per il rilevamento delle intrusioni. Questo metodo cerca di individuare direttamente un comportamento definito anomalo sulla base di regole, definite "firme" o "signature", presenti in una base di dati. I sistemi di questo tipo vengono anche detti "knowledge based". Tale approccio può essere molto accurato nel rilevare gli attacchi noti, ma nulla può con attacchi nuovi o semplici minacce. Per programmare questi sistemi e renderli efficienti è necessario uno studio costante sulle diverse forme di attacco per la produzione di firme sempre aggiornate. Dalla qualità delle firme e dal loro aggiornamento costante dipende inevitabilmente l'efficacia del sistema, si può notare immediatamente un

parallelo con i meccanismi e le problematiche tipiche del software antivirus. Nonostante questi sistemi sembrano essere più precisi, almeno con gli attacchi noti, stranamente portano spesso a falsi positivi e ad alert indesiderati. Da quanto detto si può anche intuire l'enorme quantità di tempo e denaro necessario per generare e mantenere aggiornata una base di dati con tutti i pattern degli attacchi nuovi.

Alcuni sistemi commerciali ed accademici di Intrusion Detection combinano entrambi questi aspetti, in quanto ciascuno di essi possiede delle caratteristiche molto desiderabili, e sono a tutti gli effetti complementari. Vi è tuttavia un problema di metriche decisionali, e di aumento incontrollato dei falsi positivi, che ha scoraggiato fino ad ora lo sviluppo di soluzioni ibride. Nella stragrande maggioranza dei casi i sistemi di Intrusion Detection commerciali rimangono misuse based.

Signature based IDS	
Pro	• Rappresentano il metodo più semplice ed efficace per rilevare attacchi già noti. • Numero relativamente basso di falsi positivi.
Contro	• Inefficaci contro attacchi sconosciuti o varianti di attacchi già noti. • Scarsa conoscenza di stati e protocolli. • Necessità di mantenere le signature e i vari pattern costantemente aggiornati.

Anomaly based IDS	
Pro	• Efficaci contro vulnerabilità non ancora scoperte. • Indipendenti dal SO utilizzato
Contro	• Inefficaci contro attacchi sconosciuti o varianti di attacchi già noti. • Possono comportare problemi relativi alla selezione delle caratteristiche del sistema da adottare, poiché queste possono variare enormemente a seconda dell'ambiente • Difficoltà a generare allarmi in tempo utile • La presenza di falsi positivi o falsi negativi può avere gravi conseguenze per il sistema

3.4 Network-based e host-based

Quando i primi IDS sono stati progettati, gli ambienti da proteggere erano tipicamente mainframe e tutti gli utenti erano locali rispetto al sistema considerato, quindi il loro compito era quello di analizzare l'informazione messa a disposizione dal mainframe a caccia di eventi sospetti. Questa tipologia di IDS prende il nome di host-based intrusion detection e possiamo considerarla come la prima tipologia di IDS realizzata.

In seguito, poiché l'attenzione nel campo dell'informatica si è massicciamente spostata verso ambienti distribuiti, i sistemi di Intrusion Detection hanno dovuto adattarsi alle nuove necessità.

Il trend iniziale fu quello di far comunicare fra loro IDS host-based: poiché gli utilizzatori di un ambiente del genere possono cambiare continuamente macchina, in alcuni casi cambiando anche utente, gli IDS dovevano poter scambiare

dati con i loro vicini, passandosi direttamente le informazioni di audit oppure generando alert relativi ad un'analisi di tipo locale. Al di là dell'ovvio overhead introdotto seguendo un approccio del genere, l'oggettiva difficoltà a rilevare in tale modo attacchi diretti ai protocolli di rete ha portato allo sviluppo di soluzioni che esaminino direttamente i segmenti di rete da proteggere in modalità real-time, dando origine agli IDS network-based.
Con quest'ultima tecnologia risulta possibile identificare numerosi attacchi diretti a servizi di rete, esaminando a fondo il payload dei pacchetti.

Figure e tabella vantaggi e svantaggi

L'Intrusion Detection network-based si fonda molto più sull'analisi dei pacchetti in transito che sulle informazioni collezionabili attraverso uno specifico protocollo; l'analisi del traffico di rete può essere fatta sostanzialmente in due modi distinti:

- Modalità stateless: prevede un'analisi dei singoli pacchetti prendendone in considerazione solo il loro contenuto grezzo.

- Modalità stateful: in questo caso, prima dell'analisi, si cerca di ricostruire la sessione corrispondente, ad esempio riassemblando pacchetti ip frammentati e ricostruendo lo stream se si tratta di pacchetti tcp, in modo da poter fare un'analisi basata anche sulle specifiche del protocollo di più alto livello trasportato nello stream.

Network based IDS	
Pro	• Permettono di capire quali misure preventive adottare al fine di evitare attacchi futuri • Difficili da rilevare per un hacker • Sono in grado di rilevare anche gli attacchi provenienti dalla rete interna
Contro	• Non sono in grado di monitorare protocolli wireless. • Non individuano attacchi che sfruttano vulnerabilità non ancora rese pubbliche. • Alto tasso di falsi positivi e falsi negativi.

Host based IDS	
Pro	• Possono analizzare una comunicazione end-to-end criptata
Contro	• Consumano le risorse dell'host • Possono entrare in conflitto con altri sistemi di sicurezza • Comportano ritardi nella generazione degli allarmi e nelle creazione dei report per l'amministratore

3.5 Alert

Le azioni che possono essere intraprese da un IDS quando viene identificato un attacco possono essere sia attive che passive. Lo scenario più comune prevede un IDS con risposta passiva, cioè in caso di intrusione rilevata viene semplicemente informato l'amministratore mediante un sistema di alert o scrivendo su un file la diagnosi rilevata, non viene quindi presa alcuna contromisura. L'amministratore che ha accesso agli alert inviati dal'IDS, cioè i messaggi di allarme emessi per segnalare situazioni inattese, si preoccuperà di prendere le corrette contromisure sulla base della diagnosi in modo da limitare i danni. Un problema di questa metodologia è il numero di alert che spesso vengono segnalati, perché nel caso in cui dovessero essere numerosi per l'amministratore non sarebbe semplice gestirli. La situazione ottimale risulta quella in cui il numero di alert è limitato, ma con un alto livello di dettagli. In questo modo è possibile un'identificazione precisa del problema verificatosi e delle conseguenze da esso causate.
L'IDS ideale dovrebbe garantire:

- Basse percentuali riguardo alla segnalazione di falsi positivi, cioè invio di messaggi di alert per situazioni che in realtà non sono rischiose per la sicurezza del sistema.

- Totale assenza di falsi negativi, cioè non dovrebbe mai omettere una segnalazione relativa ad un comportamento pericoloso, ma che in realtà non è stato riconosciuto come tale.

Diversamente altri tipi di IDS prevedono una risposta attiva in caso di evento critico, come per esempio applicando una patch al sistema operativo per contrastare le vulnerabilità, effettuando il log-off dell'utente non autorizzato, riconfigurando i firewall o i router e disconnettendo delle porte. Data la velocità e la frequenza con cui gli attacchi avvengono un IDS ideale dovrebbe automaticamente rispondere agli attacchi senza l'intervento di nessun operatore. Tuttavia questo è irrealizzabile dato l'alto numero di falsi positivi che

comporterebbero variazioni delle regole di rete non controllate e non sempre necessarie. Nonostante ciò, molti IDS prevedono la possibilità di utilizzare un meccanismo attivo di risposta, che può essere utilizzato a discrezione dell'amministratore di sistema.

3.6 Differenze tra Firewall e IDS

Sebbene le attività di prevention svolte da un IDS presentino notevoli somiglianze con quelle di un firewall, esse riguardano in realtà ambiti ben diversi: il firewall agisce unicamente ai livelli più bassi del modello ISO/OSI, lavorando con porte e IP, mentre l'IDS è invece in grado di operare fino al livello di applicazione e quindi di fare ispezione anche dei protocolli. Le differenze, tuttavia, non si fermano qui. Diversamente da tutti gli altri sistemi di difesa visti finora(firewall e antivirus), gli IDS agiscono ad un livello inferiore, operando nelle fasi iniziali del ciclo di vita di un cyber attacco, prima ancora che l'attaccante si infiltri all'interno del sistema e cominci l'attacco vero e proprio. Un IDS, per via della possibilità di analizzare traffico in copia (modalità mirroring), è inoltre in grado di rilevare anche gli attacchi che provengono direttamente dall'interno del sistema su cui è attestato, cosa che invece un firewall non può fare in quanto, trattandosi di un dispositivo completamente in-line, riesce ad analizzare unicamente i pacchetti che lo attraversano. Questo viene fatto molto semplicemente controllando lo stato dei pacchetti che viaggiano già all'interno della rete locale e confrontandolo con situazioni pericolose già accadute o situazioni di anomalia definite dall'amministratore di sistema. Tutto ciò significa che un IDS(da solo) e un firewall(da solo) non possono in alcun modo garantire la completa sicurezza di un sistema. Per ottenere risultati soddisfacenti si deve necessariamente combinare il loro utilizzo.

Capitolo 4

Machine Learning

Il machine learning è una branca della computer science che studia i sistemi e gli algoritmi capaci di apprendere dalla sintesi dei dati. Nel 1959 Arthur Samuel definì l'apprendimento automatico come il campo di studi che dà ai calcolatori la capacità di apprende senza essere programmati esplicitamente [5]. Un sistema basato sull'apprendimento automatico può migliorare la propria conoscenza del contesto in cui si trova sulla base dell'osservazione dei dati di input per poi fornire output più vicini a quelli desiderati. Quindi un metodo di machine learning è un algoritmo capace di apprendere dai dati. Una definizione più precisa di apprendimento viene data da T.M. Mitchell:

Un programma apprende da una certa esperienza E se nel rispetto di una classe di compiti T, con una misura della prestazione P, la prestazione P misurata nello svolgere il compito T è migliorata dall'esperienza E.

I Compiti del machine learning sono spesso descritti in termini di come il sistema possa trattare un esempio, una collezione di *features*, o caratteristiche, quantitativamente misurate da alcuni oggetti o eventi che vogliamo il sistema elabori. Tipicamente l'input viene rappresentato da un vettore dove ogni elemento rappresenta una *feature*. Per esempio le *feature* di un pacchetto di rete possono essere le componenti definite dal protoccolo a cui appartiene.
Per valutare le abilità di un algoritmo di machine learning, dobbiamo stimare una misura quantitativa della sua performance P. Spesso la misura di P è specifica per un certo compito T che deve compiere il sistema.

Per compiti come la classificazione, P si valuta misurando l'*accuratezza* del modello, la percentuale di esempi per i quali il modello elabora un output corretto.
Un altro parametro di riferimento può essere il *rate di errore*, definito invece come la proporzione di esempi per cui il sistema elabora un output sbagliato. E' importante verificare come l'algoritmo riesca a valutare dati che non abbia mai visto.
Misure di performance vengono effettuate usando un insieme di dati chiamato insieme di *test*. Questo insieme di dati è, di solito, diverso dall'insieme delle caratteristiche usate per addestrare il sistema affinchè si possano fare valutazioni più precise ed indipendenti su quanto sia stato efficace l'addestramento. Di solito l'esperienza E corrisponde ad un intero dataset, una collezione di esempi, definiti precedentemente. Sulla base della capacità di *inferire* dall'esperienza il test può fornire risultati migliori o peggiori.

4.1 Metodi di Machine Learning

Tra le tecniche esposte nei precedenti capitoli quella che negli ultimi anni sta interessando il mondo della ricerca e quello industriale riguarda gli Intrusion Detection System *anomaly-based*. Questa tipologia di dispositivi cerca di comprendere il comportamento *normale* dei dati al fine di rilevarne delle deviazioni da segnalare o isolare.
Il motivo che spinge ad investire su queste tecniche è probabilmente legato alla grande quantità di algoritmi intelligenti che possono essere adattati per svolgere operazioni di classificazione del traffico. Inoltre molti di questi dispositivi cercano di generare modelli dei dati che riescono a catturare anomalie non ancora note. Tra le tante soluzioni quelle maggiormente utilizzate nel campo della sicurezza informatica prevedono l'uso di un singolo classificatore o una combinazione di classificatori. In particolare chiameremo questi algoritmi per la classificazione di anomalie di rete *detector*.
Secondo [6] le principali soluzioni implementate negli ultimi anni sono:

- *Single Classifiers* Affrontano il rilevamento delel intrusioni basandosi su un singolo algoritmo di machine learning. In letteratura, tecniche di apprendimento automatico (ad esempio k-nearest neighbor, support vector machines, artificial neural network, decision trees, self-organizing maps, etc.) sono stati usati per risolvere questo problema.

- *Hybrid classifiers* Sono sistemi ibridi progettati con il fine di raggiungere una precisione maggiore nei risultati forniti. L'idea alla base di un classificatore ibrido consiste nel combinare diverse tecniche di apprendimento automatico (machine learning) in modo che le prestazioni del sistema possano essere notevolmente migliri. Esistono diversi approcci per svolgere questo compito. Alcuni vengono detti metodi in *serie* tipicamente i compomenti di questi metodi prendono i dati grezzi e generano dei risultati intermedi da passare ad altri classificatori. Alti invece hanno un comportamento in *parallelo* quindi analizzano tutti gli stessi dati e li valutano in modo differente. Quest'ultimo metodo, che verrà ripreso più avanti, necessità di un ulteriore componenete che si occupa di valutare e mediare i risultati ottenuti da tutti i classificatori.

Un ulteriore distinzione che può essere fatta sui classificatori riguarda il modo in cui riescono ad apprendere e quindi realizzare un modello dei dati. In particolare si possono distinguere metodi di machine learning *supervisionati* e *non supervisionati*.

4.1.1 metodi supervisionati

L'obiettivo di un sistema basato sull'apprendimento supervisionato è quello di produrre un'ipotesi induttiva ossia una funzione in grado di apprendere dai risultati forniti durante la fase di addestramento e in grado di avvicinarsi a dei risultati desiderati per tutti gli esempi non forniti. La casistica degli output può essere molto varia; ciononostante differenzia l'apprendimento di valori quantitativi (comunemente chiamata "regressione") da valori qualitativi (chiamata "classificazione")[7].

4.1.2 metodi non supervisionati

L'apprendimento non supervisionato è una tecnica di apprendimento automatico che consiste nel fornire al sistema una serie di input (esperienza del sistema) che egli riclassificherà ed organizzerà sulla base di caratteristiche comuni per cercare di effettuare ragionamenti e previsioni sugli input successivi. Al contrario dell'apprendimento supervisionato, durante l'apprendimento vengono forniti all'apprendista solo esempi non annotati, in quanto le classi non sono note a priori ma devono essere apprese automaticamente[7].

4.1.3 Principal Component Analysis

In questo paragrafo verrà introdotta la Principal Component Analysis (PCA), algoritmo di machine learning, non supervisionato, utilizzato in questo lavoro per la classificazione del traffico.
In generale, la PCA viene utilizzata in diverse applicazioni dove ci si trova in presenza di grandi quantità di dati di cui non si conosce la natura.
L'idea alla base della PCA è quella di trovare un nuovo sistema di riferimento in modo da massimizzare la varianza delle variabili rappresentate lungo gli assi. Nel caso in questione le variabili corrispondono alle caratteristiche, *feature*, del traffico di rete. Trovare la direzione lungo la quale i dati esprimono maggiore variabilità ci consente di capire quali elementi del traffico portano un contenuto informativo più interessante ai fini della ricerca delle anomalie.

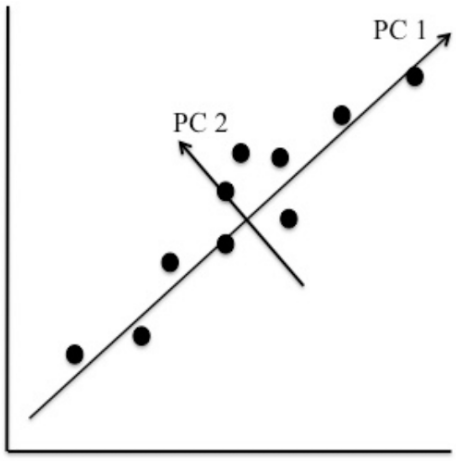


Figura 4.1: I nuovi assi di riferimento, le componenti principali PC_2 e PC_1, descrivono meglio la natura dei dati rispetto al sistema di riferimento inziale.

Per meglio capire lo scopo della PCA in questo contesto, si immagini di provare ad interpretare un flusso di traffico ed individuarne una serie di misure. I dati del flusso possono essere confusi, ridondanti e molto spesso possono mascherare le caratteristiche che veramente ci consentirebbero di inferire delle informazioni utili. La scarsa conoscenza di un fenomeno, come quello del traffico di rete, induce ad effettuare misurazioni che non descrivono al meglio la dinamica del sistema e quindi non consentono di giungere a conclusioni corrette.
Ai vantaggi descritti si contrappongono due problemi:

- I nuovi assi generati dalla PCA fanno perdere il legame semantico che ci consente di associare ad ogni *caratteristica* dei dati un'informazione sul traffico reale.

- La scelta del numero di componenti da tenere è molto importante perché un'eccessiva riduzione delle componenti potrebbe compromettere i risultati a causa della perdita di informazioni su alcuni eventi.

4.2 Dataset

Gli approcci noti per ottenere un maggiore controllo sulle anomalie prevedono l'utilizzo di sistemi addestrati sulla base di grandi raccolte di dati chiamate *dataset*. Queste raccolte sono il frutto di un'attenta elaborazione dei dati proveniente da sensori che rilevano caratteristiche di un certo contesto. I dati raccolti vengono denominati *raw data* e presentano imperfezioni legate al rumore o ad errori che si presentano in fase di acquisizione e che non consentono di utilizzarli senza aver fatto prima la fase di *pre-processing*, cioè una pre-elaborazione che ha il fine di renderli più rappresentativi del contesto che descrivono. Nel caso del traffico di rete si aggiunge un'ulteriore problema legato alle informazioni contenute nel *payload* e nell'header dei pacchetti. Sapere infatti che due indirizzi IP statici comunicano è già un'informazione che potrebbe compromettere la privacy degli utenti, tanto più se i *payload* dei pacchetti non sono criptati. Per queste ragioni individuare un dataset per testare algoritmi di machine learning non è affatto semplice. In letteratura scientifica si trovano pochi dataset e molti di questi non sono molto aggiornati, quindi non descrivono in modo rigoroso le dinamiche di una rete moderna[8]. Il dataset, inoltre, è uno strumento di confronto che consente a chi fa ricerca di comparare i risultati di diverse soluzioni, quindi deve essere pubblico e facilmente accessibile. In questa tesi verranno utilizzati due tipologie di dataset, la prima tipologia deriva dagli archivi Mawi, e presenta tutte le proprietà appena descritte, la seconda è stata realizzata da una sua elaborazione col fine di testare alcune proprietà del software implementato.

4.2.1 Dataset Mawi

Il dataset utilizzato per eseguire i test utili per valutare le prestazioni degli algoritmi implementati in questa tesi è costituito dalle tracci di traffico che si trovano nell'archivio **MAWILab** (acronimo che sta per Measurement and Analysis of the Wide Internet).
Ciascuna traccia è relativa al traffico osservato per 15 minuti, dalle 14:00 al-

le 14:15, ora del Giappone, in uno specifico giorno su un link transoceanico che collega Giappone e Stati Uniti. I dati presenti nell'archivio vengono aggiornati quotidianamente, a partire dall'anno 2001, ad oggi sono disponibili tracce di traffico catalogate fino a Marzo 2019 e sono registrati in file di tipo pcap che contengono informazioni relative ai pacchetti transitati sul link monitorato (principalmente si riferiscono agli header del livello IP e del livello di trasporto). La caratteristica importante delle tracce MAWI è che ciascun flusso è stato catalogato in funzione della probabilità di rappresentare un'anomalia. Forniscono quindi un aiuto per valutare le prestazioni (in termini di falsi positivi e falsi negativi) di un qualsiasi algoritmo di anomaly detection, attraverso un confronto tra le anomalie rivelate da quest'ultimo e quelle registrate nella traccia. Bisogna comunque tenere presente la natura stocastica della classicazione: l'elenco delle anomalie indicate nell'archivio è infatti il risultato dell'impiego congiunto di quattro detector (basati rispettivamente su trasformata di Hough, distribuzione Gamma, Kullback-Leibler divergence e Principal Component Analysis (PCA)), quindi non abbiamo la certezza che siano realmente causate da un attacco.
In particolare, dall'analisi eseguita dagli autori del progetto MAWI, il traffico risulta suddiviso in quattro raggruppamenti:

- **Anomalous**: traffico considerato anomalo con alta probabilità e dovrebbe essere rilevato tale da qualsiasi efficente algoritmo di detection
- **Suspicious**: traffico che molto probabilmente è anomalo, ma non è stato chiaramente identificato come tale con i metodi impiegati
- **notice**: tutto quel traffico che non è stato identificato come anomalo da Mawi-Lab ma è risultato comunque anomalo da almeno un detector utilizzato. Questo traffico non dovrebbe essere identificato anomalo da un algoritmo di detection. Non è stato classificato come benigno in quanto è stato sollevato un alert da uno dei detectors e di questo evento è utile tenerne traccia. Potrebbe trattarsi di attività legittime con comportamenti fuori dal normale

- **benign**: traffico normale, non rivelato da nessuno dei detector.

Le anomalie registrate sono state a loro volta suddivise in tre categorie:

- **attack**: anomalie riconducibili ad attacchi noti
- **special**: anomalie associate ad attività che coinvolgono porte well-known
- **unknown**: anomalie di tipo sconosciuto, che non rientrano nelle due categorie precedenti.

Il trafico etichettato come *anomalous*, *suspicious* o *notice* è elencato in file di tipo xml (due per ogni traccia, uno per le anomalie di tipo anomalous e suspicious, uno per quelle di tipo notice), dove per ogni anomalia individuata vengono riportati [9]:

- T: tipo di anomalia (*anomalous*, *suspicious* o *notice*)
- D_r, D_a:distanze da due punti di riferimento rispettivamente per il traffico regolare e anomalo, calcolate elaborando con un algoritmo opportuno (SCANN)[9] le uscite dei quattro detector; sulla base di tali valori viene deciso il tipo di anomalia:
  - **anomalous** se $D_a \leq D_r$
  - **suspicious** se $D_a > D_r$
  - **anomalous** se $D_a \geq D_r$ e $\frac{D_a}{D_r} - 1 \leq 0.5$
  - **notice** se $\frac{D_a}{D_r} - 1 > 0.5$
- C: categoria dell'anomalia, rappresentata da un valore numerico.
  - *attack* $1 \leq C \leq 500$
  - *special* $500 \leq C \leq 900$
  - *unknown* $C \geq 900$
- V: vettore costituito da 12 elementi binari i cui valori indicano quali detector hanno rivelato l'anomalia; i quattro detector sono stati infatti

applicati ciascuno con tre modalità crescenti di sensibilità: conservative, optimal e sensitive;
ogni elemento del vettore V è associato ad un detector con una determinata modalità, nel seguente ordine: Hough (sensitive, optimal, conservative), Gamma (sensitive, optimal, conservative), KL (sensitive, optimal, conservative), PCA (sensitive, optimal, conservative).

- Filtri che identicano l'anomalia tramite feature del trafico, che possono essere: indirizzo IP sorgente, indirizzo IP destinatario, porta sorgente, porta destinataria e protocollo di trasporto.

- Timestamp che individuano inizio e fine del traffico anomalo.

Si riporta un riepilogo della classificazione delle anomalie nella tabella 4.2. Come si può vedere sono indicate la tipologia (*label*), la categoria più specifica dell'attacco con il corrispondente valore del campo C presente nei file xml e una breve descrizione [10].

Non si riesce però ad avere una copertura completa di tutte le anomalie presenti in rete. Lavori di ricerca portati avanti in questo ambito hanno fatto passi avanti, riuscendo a ridurre il numero di anomalie classificate come "unknown".
La struttura delle tassonomie può essere definita come un albero, dove ogni nodo dell'albero è etichettato. La figura 4.3 è una descrizione di alto livello della struttura ad albero descritta in [10].

Come si può osservare, esistono due principali categorie di eventi: anomali e normali. Gli eventi anomali comprendono eventi che possono essere scatenati da un denial of service o da attività di scanning, mentre gli eventi normali includono i così detti heavy hitter o alpha-flows, point multipoint e altre tipologie di eventi come ad esempio tunnell, point-to-point flows, outages ecc. Alcuni eventi possono essere considerati legittimi o no a seconda del contesto, per esempio un'attività di scanning può essere effettuata per un'attività di ricerca o come fase preliminare di un attacco.

Label	Categoria (C)	Dettagli
Attack	Sasser (1)	Traffico TCP sulle porte 1023, 5554 o 9898
Attack	NetBIOS (2)	Traffico sulla porta TCP 139 o sulla porta UDP 137
Attack	RPC (3)	Traffico TCP sulla porta 135
Attack	SMB (4)	Traffico TCP sulla porta 445
Attack	Ping flooding (20)	Traffico ICMP elevato
Attack	Port scan e altri attacchi (10-SYN, 11-RST, 12-FIN, 51-FTP, 52-SSH, 53-HTTP, 54-HTTPS)	Traffico con più di 7 pacchetti in cui almeno la metà hanno i flag SYN, RST o FIN settati, oppure traffico http, ftp, ssh, dns avente il flag SYN settato per almeno il 30% dei pacchetti
Special	Http (503-HTTP, 504-HTTPS)	Traffico TCP sulle porte 80 o 8080 con meno del 30% dei flag SYN settati
Special	ftp, ssh, dns (501-FTP, 502-SSH)	Traffico TCP sulle porte 20, 21 o 22 oppure TCP & UDP sulla porta 53 con meno del 30% dei flag SYN settati
Unknown	Unknown (901)	Traffico che non corrisponde ai precedenti casi

Figura 4.2: Classificazione delle anomalie nelle tracce MAWI

MawiLab assegna una singola etichetta ad ogni evento. Per prima cosa tende ad assegnare un'etichetta appartenente al sottoalbero la cui radice è etichettata come "anomaly". Se non esiste nessuna corrispondenza, si ripete lo stesso procedimento nel sottoalbero la cui radice è etichettata con "normal". Se ancora non viene trovata una corrispondenza, l'evento è etichettato con "unknown". Tale procedimento viene portato avanti utilizzando delle signatures gestite da MawiLab. Un evento può corrispondere a diverse signatures all'interno di uno dei due sottoalberi: quando ciò avviene, viene scelta la corrispondenza più dettagliata, ovvero quella più lontana dalla radice. Questo consente all'algoritmo di avere un'accuratezza elevata nella scelta dell'etichetta.

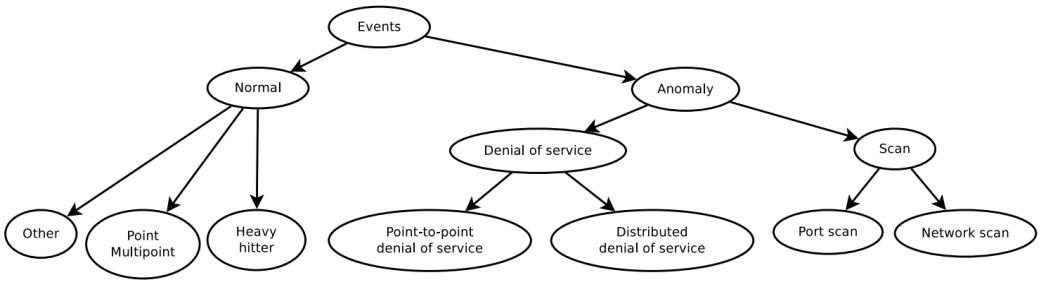


Figura 4.3: Tassonomia

4.3 Report MawiLab

I *report* sono file con estensione .xml o .csv associati ad ogni traccia di traffico catturata ogni giorno. In particolare nei file di report sono presenti le informazioni legati agli attacchi che sono state usate in questo lavoro. Per il formato xml sono riportati:

- **AnomalyType**: sono le label che vengono associate all'anomalia. Sono state trattate nel paragrafo precedente

- **Value**: Il valore contiene una stringa di informazioni separate da virgola. In particolare alcune metriche di distanza risultanti dall'algoritmo di classificazione riassunto precedentemente, un codice di tre cifre compreso tra 500 e 900. Se il valore di tale codice è inferiore a 500 vuol dire che il traffico anomalo sta usando porte sospette ben note (Netbios,smb, SSH attack) o contiene un numero spropositato di pacchetti con i flag TCP del tipo: SYN,RST o FIN.Se il valore è compreso tra 500 e 900, vuol dire che l'anomalia è riscontrata su porti ben noti (FTP, HTTP,SSH). Infine se il codice ha un valore maggiore di 900, significa che l'anomalia è riscontrata su porti sconosciuti. Un altro campo mostra quale detector e con quali parametri ha rilevato l'anomalia. E' un vettore di valori binari dove lo 0 indica che non sono stati riportati alert, 1 viceversa. Infine, nell'ultimo campo vi è la tassonomia dell'anomalia.

- **Slice**: Contiene una serie di attributi, in particolare gli indirizzi IP

sorgente e destinazione, porto destinazione e la durata dell'attacco in secondi.

Per quanto riguarda il formato .csv, ogni riga consite in un insieme di 4 tuple che descrivono le caratteristiche del traffico o informazioni aggiuntive come le euristiche e le tassonomie, in particolare:

- **AnomalyID**: identificativo dell'anomalia riscontrata. Permette di indetificare le righe del file che si riferiscono alla stessa anomalia.
- **srcIp**: Indirizzo IP sorgente del traffico anomalo
- **dstIp**: indirizzo IP di destinazione del traffico anomalo
- **taxonomy**: tassonomia dell'anomalia riscontrata
- **heuristc**:codice assegnato all'anomalia usando lo stesso criterio visto per il file xml
- **label**: etichetta assegnata all'anomalia. Può essere, anche in questo caso di tre tipi: *anomalous*, *suspicious* o *notice*.

Le anomalie di rete sono estremamente diverse tra loro. Questo è dovuto al fatto che esistono molteplici comportamenti che possono essere considerati anormali. Questi comportamenti però, possono essere caratterizzati applicando un determinato criterio al traffico di rete. MawiLab propone una metodologia con cui classificare i comportamenti anomali per ottenere un insieme di tassonomie, da associare a comportamenti che non risultano essere conformi a determinate regole o signatures. L'obiettivo è quello di caratterizzare completamente i comportamenti anomali. Nella sottosezione seguente viene descritto il processo di etichettamento e cosa si intende per tassonomia.

4.3.1 Signature

Ogni *signature* può essere composta da una o più regole che descrivono la natura del traffico di rete. Ad esempio, può essere utilizzato il numero di

hosts sorgenti e il numero di hosts destinazione per caratterizzare un point-to-point distribuito o dei rapporti che misurano il tasso con cui determinati pacchetti entrano ed escono da una sorgente. La figura 4.4 mostra due esempi di *signature* che identificano un comportamento anomalo:

UDP network scan	$nb\_src\_addr < 5$ $\wedge\ nb\_dst\_addr >= 20$ $\wedge \frac{nb\_packets}{nb\_dst\_addr} < 5$ $\wedge \frac{nb\_udp\_packets}{nb\_packets} >= 0.8$
UDP network scan ICMP response	$nb\_dst\_addr < 5.$ $\wedge \frac{nb\_icmp\_packets}{nb\_packets} >= 0.8$ $\wedge \frac{nb\_destination\_unreachable\_packets}{nb\_icmp\_packets} >= 0.8$ $\wedge$ $nb\_src\_addr < 20$ $\wedge$ $\frac{nb\_network\_host\_unreachable\_packets}{nb\_destination\_unreachable\_packets} >= 0.8$ $\vee \frac{nb\_prohibited\_unreachable\_packets}{\_nb\_destination\_unreachable\_packets} >= 0.8$ $\vee$ $nb\_src\_addr >= 20$ $\wedge \frac{nb\_packets}{nb\_src\_addr} < 20$ $\wedge \frac{nb\_protocol\_port\_unreachable\_packets}{nb\_destination\_unreachable\_packets} >= 0.8$ $\wedge\ nb\_icmp\_src\_addr < 5.$ $\wedge\ nb\_icmp\_dst\_addr >= 20.$ $\wedge \frac{nb\_icmp\_du\_udp\_packets}{nb\_destination\_unreachable\_packets} >= 0.8$

Figura 4.4: Esempio di signature(regole decisionali)

UDP-NetScan: Conoscendo l'attività di scanning è possibile intuire quali signatures siano necessarie per identificare un UDP *network scan*. Vengono ispezionati i pacchetti con un numero piccolo di sorgenti, un elevato numero di destinazioni, un piccolo numero di pacchetti per ogni destinazione (rappresentanti le sonde), ed un'elevato numero di pacchetti UDP ($\geq 80\%$).

UDP-NetScan.Response: Viene identificata da un numero ridotto di destinazioni, questo perché consideriamo che la trasmissione della sonda malevola venga trasmessa da un solo host. La maggior parte dei pacchetti ($\geq 80\%$) è di tipo ICMP, contenenti l'informazione *"destination unreachable"*.

La signature tiene in conto due casi:

a) la rete/hosts sono irragiungibili o che l'accesso è proibito a causa di un fi-

rewall.

b) l'host è attivo ma il protocollo/porta non è disponibile/aperta.

Nel caso a) (marcato in rosso nella figura), è il gateway che invia pacchetti ICMP e quindi provengono da una sorgente singola.

Nel caso b) (marcato in blu), invece, ogni macchina target risponde e quindi il numero di sorgenti è maggiore. Le ultime tre righe della signature descrivono il pacchetto UDP originario incapsulato in un pacchetto ICMP "destination unreachable".

Capitolo 5

Random Traffic Projections

Si definisce un flusso di dati (o *data stream*) una sequenza di segnali digitali utilizzata per rappresentare delle particolari informazioni. Nel caso specifico lo *stream* di dati è quello catturato ed elaborato negli archivi Mawi. Ogni file dell'archivio presenta, in soli 15 minuti di traffico, più di 70 milioni di pacchetti e 100 *features* differenti che corrispondono a circa 4GB di dati. Prendere in considerazione tutte queste informazioni non è l'approccio migliore per trattare il problema dell'*anomaly detection*, serve un metodo capace di far emergere le caratteristiche del *dataset* che forniscono un'informazione utile e che sintetizzi le informazioni da trattare.

Un Intrusion Detection System, come tutti i sistemi informatici, necessita di un tempo proporzionale alla quantità di dati che deve elaborare per determinarne la pericolosità. Questa condizione non consente una risposta real-time, che nel caso di attacchi informatici è molto importante al fine di limitare i danni di un attacco. In questo capitolo si tratteranno le tecniche e i modelli utilizzati in questa tesi per affrontare l'incontrollabile crescita del traffico che ha trasformato in un *big data problem* la *cybersecurity*.

Di seguito verranno descritti gli aspetti formali del modello dei *data streams* ponendo l'attenzione sulle emergenti teorie in questo campo, sull'applicazione di strutture dati tipo Random Projection(*sketch*) per la sintesi dei flussi di informazione e introducendo le proprietà e le funzionalità dell'algoritmo utilizzato in questo lavoro.

5.1 Random Projection

Oggi in rete vengono rilevati sempre più modelli di traffico insoliti. Questi modelli, nel caso di intrusioni, si presentano come varianti di attacchi noti (si pensi a DoS, port scan e worm), molto spesso però anche un utilizzo normale può indurre a modelli di traffico singolari che tendono a somigliare ad anomalie. In questo contesto diventa molto difficile rilevare anomalie in un singolo collegamento, sarebbe necessario controllare l'intera rete per essere in grado di distinguere un attacco reale da un occasionale aumento del flusso del traffico[11]. Controllare l'intera rete è però un'operazione impossibile e appena si cerca di analizzare un numero maggiore di collegamenti la quantità di dati da trattare cresce con legge esponenziale rendendo impossibile effettuare una diagnosi del traffico in tempo reale.
Nel caso di flussi IP considerando soltanto gli indirizzi IP e le porte sorgente e di destinazione si ottiene una stringa di 96bit che risiede in uno spazio a 2^{96} dimensioni, è indispensabile trattare questo spazio mediante tecniche capaci di ridurre le componenti e la quantità di dati in modo efficienti[12]. Le tecniche maggiormente utilizzate per effettuare questa operazione sono:

- **Principal Component Analysi** (PCA)
- **Random Projection** (sketches)

In questo lavoro si utilizzeranno gli *sketches* che basano il loro funzionamento sul lemma di **Johnson–Lindenstrauss** [13].

Lemma 5.1.1. Dato un $\varepsilon \in (0,1)$, un insieme di punti $\{x_1, x_2, \cdots, x_M\} \subset \mathbb{R}^m$ e un numero $n \geq \frac{8}{\varepsilon^2}\log(m)$ esiste un'applicazione lineare $A : \mathbb{R}^m \longrightarrow \mathbb{R}^n$ tale che

$$1 - \varepsilon \leq \frac{\parallel A(x_i) - A(x_j) \parallel}{\parallel x_i - x_j \parallel} \leq 1 + \varepsilon$$

Il lemma di Johnson–Lindenstrauss afferma che, a meno di un errore ε, un insieme di punti in uno spazio Euclideo di dimensione elevata può essere *mappato* in uno spazio Euclideo di dimensioni ridotte mantenendo comunque inalterata la distanza tra i punti di un fattore di $1 \pm \varepsilon$.

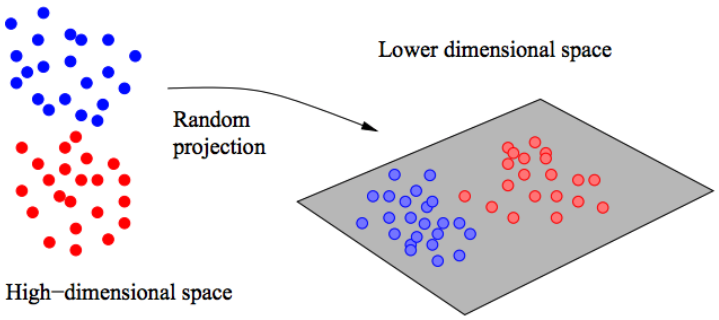


Figura 5.1: Random Projection

5.1.1 Funzioni hash universali RSHash e JSHah

In questo paragrafo si introduce la classe di funzioni hash utilizzate in questo lavoro per realizzare gli *sketches*. Per trattare il concetto di funzioni hash si parte da quello di tabella hash. In informatica una *hash table*, detta anche *hash map*, è una struttura dati usata per mettere in corrispondenza una data chiave con un dato valore. Viene usata per l'implementazione di strutture dati astratte e può usare qualsiasi tipo di dato come indice della tabella. Tutte le operazioni di inserimento possono essere fatte con una complessità pari di $T(n) = (O(1))$, mentre le letture possono avere complessità $T(n) = (O(n))$, con n molto piccolo in assenza di *collisioni*. Esistono vari tipi di algoritmi di hashing molti di questi sono stati studiati a fondo e sviluppati per svolgere diversi compiti legati alla sicurezza informatica.
Il primo passo per realizzare algoritmi di hashing è quello di determinare la funzione di hash: con il metodo hash infatti un elemento con chiave k viene memorizzato nella tabella in posizione $h(k)$ dove la funzione $h()$ è appunto detta funzione hash.

Lo scopo della funzione hash è di definire una corrispondenza tra l'universo U delle chiavi e le posizioni di una tabella hash $T\,[0, \cdots, m-1]$

$$h : U \to \{0, \cdots, m-1\}$$

Necessariamente la funzione hash non può essere *iniettiva*, ovvero due chiavi distinte possono produrre lo stesso valore hash, quando questo accade si dice che si ha una *collisione*. É possibile che la scelta di chiavi sia tale da avere un elevato numero di collisioni. Il caso peggiore è che tutte le chiavi collidano in un unico valore hash come nell'esempio 5.2; in questo caso le prestazioni delle operazioni con tabelle hash, dove la maggior parte delle chiavi hanno un unico valore hash, peggiorano fino a $O(n)$. Una buona funzione hash, come H_1, deve distribuire i valori in modo uniforme.

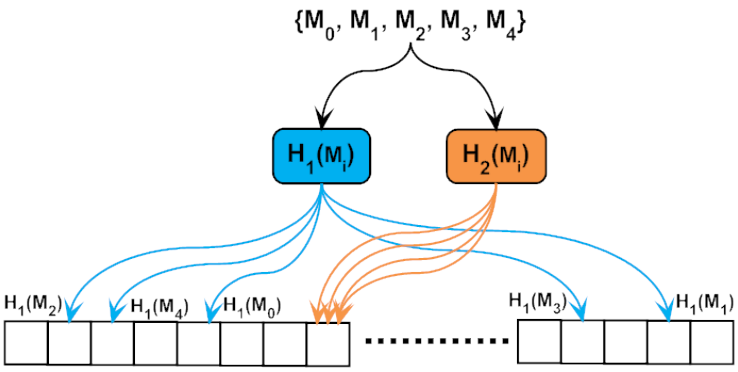
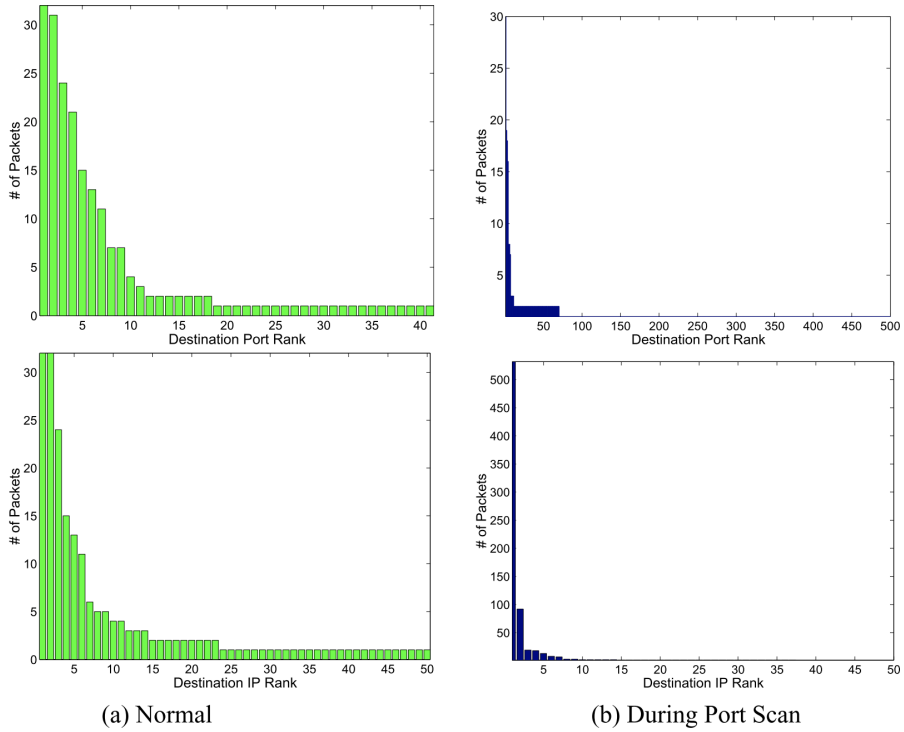


Figura 5.2: Confronto tra funzioni hash

In appendice si riportano le due famiglie di funzioni hash universali utilizzate per l'implementazione degli *sketch* [14].

5.1.2 Count-Min Sketch

Come si è detto nel precedente paragrafo uno dei vantaggi dell'uso delle *Random Projection* è quello di ridurre lo spazio delle *feature* e rendere trattabile un problema complesso. Questa operazione, nel caso del traffico di rete, è trattabile sulla base della seguente osservazione: *nonostante ci siano diverse forme di traffico anomalo si osserva che tutte queste forme condividono una caratteristica comune cioè inducono un cambiamento della distribuzione dei campi dell'header dei pacchetti (IP e porta sorgente e destinazione)*[10]. Per esempio un attacco DoS modifica la distribuzione del traffico aumentando il numero di pacchetti che si riferiscono ad un particolare indirizzo IP di destinazione. Allo stesso modo un attatto di tipo Scan ha una distribuzione delle porte concentrata, mentre la distribuzione degli IP è molto sparsa, si veda esempio in figura 5.3.



(a) Normal (b) During Port Scan

Figura 5.3: L'anomalia *port-scan* induce una variazione della distribuzione delle *features* che porta dallo stato normale *(a)* a quello anomalo *(b)*

La soluzione sviluppata in questa tesi è una variante della struttura dati probabilistica *count-min sketch*. In questo paragrafo vengono introdotte le proprietà teoriche che servono a comprendere l'implementazione della variante proposta.
I *count-min sketch* basano il loro funzionamento su una famiglia di funzioni hash universali che consente di mappare in uno spazio sub-lineare i dati che analizzano. Questa operazione potrebbe indurre a molteplici collisioni, ma si dimostra che l'errore rimane trascurabile [15].

**** OMISSIS ****

Capitolo 6

Progetto MawiLab

Un Intrusion Detection *signature-based*, nonostante sia uno degli strumenti più utilizzati, presenta una serie di limiti e inconvenienti che negli ultimi anni hanno spinto la ricerca a trovare delle alternative più efficienti.
I limiti, come si è detto in precedenza, sono legati al necessario aggiornamento delle regole e al numero limitato e definito di attacchi rilevabili da questi sistemi. Soluzioni alternative a questo metodo si basano sui *detectors*, questi strumenti solitamente sono algoritmi di machine learning adattati al rilevamento delle anomalie.
Alcune proposte basate sui detectors cercano di ottimizzare le prestazioni in modo da rendere il processo di rilevamento istantaneo (anomaly detection realtime). Una delle proposte più comuni nell'intrusion detection anomaly-based si basa sull'esecuzione in parallelo di uno stesso algoritmo.
In questo capitolo si presenta una valida alternativa proposta dai ricercatori Fontugne e Borgnat nel **progetto MawiLab**[10] realizzato col fine di classificare il traffico di rete degli archivi MAWI attraverso una tecnica alternativa a quelle note in letteratura.
La tecnica si presenta come una combinazione di strategie che hanno come fine quello di mettere insieme i risultati ottenuti da diversi detectors. In questo modo ogni detectos contribuisce in modo diverso al rilevamento delle anomalie e consente quindi di catturarne un numero maggiore, riducendo allo stesso tempo i falsi allarme.

Il metodo ideato dal team del progetto MawiLab per etichettare il traffico di rete, può essere riassunto nei seguenti passi:

- Diversi *anomaly detectors* analizzano il traffico e riportano gli alert riscontrati.
- Vengono studiate le similarità tra gli alert sollevati, in particolare viene introdotto uno stimatore delle similarità che ne effettua un raggruppamento in base alle affinità.
- Ogni *comunità* è analizzata e classificata tramite un ulteriore elemento detto *combiner*.
- Le anomalie sono caratterizzate usando regole di associazione applicate sui risultati forniti dal *combiner*. É in questo stadio che avviene l'etichettamento finale.

Soffermandoci sull'ultimo step, il processo di etichettamento avviene con 4 differenti labels:

- Anomalous: traffico considerato anomalo con alta probabilità e dovrebbe essere rilevato tale da qualsiasi efficiente algoritmo di detection.
- Suspicious: traffico probabilmente anomalo ma non chiaramente identificato dai detectors utilizzati da MawiLab.
- Notice: tutto quel traffico che non è stato identificato come anomalo da Mawi-Lab ma è risultato comunque anomalo da almeno un *detector* utilizzato. Questo traffico non dovrebbe essere identificato anomalo da un algoritmo di detection. Non è stato classificato come benigno in quanto è stato sollevato un alert da uno dei *detectors* e di questo evento è utile tenerne traccia. Potrebbe trattarsi di attività legittime con comportamenti fuori dal normale.
- Benign: tutto il traffico normale che non presenta comportamenti fuori dal comune.

6.1 Metodologia

La metodologia applicata si basa sulla combinazione di diversi *detector* con diversi background teorici e un diverso grado di accuratezza nell'effettuare i rilevamenti su una particolare traccia di dati del dataset MawiLab.
La scelta di utilizzare diversi *detector* deriva dal vantaggio di analizzare il flusso di traffico con una granularità differente. Ogni *detector* infatti, sulla base dell'algoritmo che implementa, agisce sui dati effettuando una valutazione a livello di flusso, pacchetto o host. In questo modo elementi che non sollecitano l'interesse di un detector vengono catturati da altri.
Ogni *detector* fornisce però risultati, che chiameremo allarmi, che, anche se diversi, molto spesso si riferiscono a segnalazioni che a livello di traffico coincidono con quelle fatte da altri *detector*. Da questa sovrapposizione deriva la necessità di trovare un metodo capace di legare diversi rilevamenti anomali in un'unica anomalia. Questa operazione non è affatto semplice, infatti, se per esempio un detector presenta allarmi a livello host, Allarm1 per l'indirizzo IP 134.200.2.44 e un altro detector riporta degli allarmi a livello di flusso, Allarm2 e Allarm3 con [134.200.2.44 , 80, IPy, 384] e [134.200.2.44, 80, IPz, 494], non è assicurato un legame tra i due flussi e l'indirizzo host segnalato come anomalo.

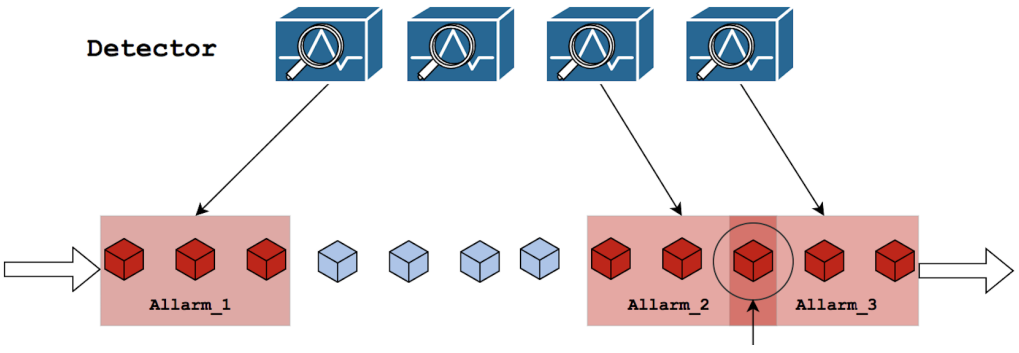


Figura 6.1: Sovrapposizione di allarmi dovuta al rilevamento di diverse anomalie riferite allo stesso traffico di rete.

La 6.1 mostra i quattro *detector* durante l'analisi del flusso di traffico. Ogni detector evidenzia anomalie diverse che, come nel caso *Allarm2* e *Allarm3* possono sovrapporsi.

La metodologia sviluppata può essere divisa in 4 passi:

- Ognuno dei quattro *detector* presentati analizza il traffico in tempo reale e segnala degli allarmi che associa al traffico che ritiene anomalo
- Ogni allarme è riferito ad un sotto traffico, allarmi che si riferiscono a dati simili o uguali vengono raccolti in gruppi denominati *comunità*
- Ogni *comunità* è controllata da un elemento, chiamato *combiner*, che ne stima la pericolosità
- Le *comunità* definite come anomale vengono etichettate sulla base di alcune regole che verranno trattate in dettaglio

6.2 Estrattore di traffico

L'*estrattore di traffico* sulla base delle caratteristiche del traffico contrassegnato anomalo, mantiene un legame tra gli allarmi e il flussi di pacchetti.
Il livello di dettaglio dell'analisi va oltre il singolo pacchetto. Allarmi che a livello di pacchetto sembrano non avere nulla in comune, possono fare parte dello stesso flusso e quindi corrispondere ad uno stesso attacco.

6.3 Graph generator

Il *generatore di grafi* utilizza il traffico ottenuto dall'*estrattore di traffico* per costruire un grafo unidirezionale chiamato *"grafo di similarità"*. Questo grafo ha lo scopo di mettere in evidenza le similarità degli allarmi riportati dai detectors.
Ogni **nodo** corrisponde ad un allarme e ogni **arco** tra due nodi, se è presente, mette in corrispondenza allarmi che hanno traffico in comune. Gli archi del

grafo sono pesati e il valore del peso varia sulla base della quantità di traffico che accomuna i nodi collegati (esempio in figura:6.2). La misura di similarità consente di scartare gli archi che collegano nodi che non hanno un numero sufficientemente elevato di pacchetti sovrapposti e di compattare quelli che rappresentano lo stesso traffico o simile.
In particolare vengono utilizzate tre diverse misure per stimare la similarità tra i nodi del grafo:

- Jaccard index
- Simpson index
- Constant function

La più performante nel svolgere questo compito è la Simpson index.

$$S(E_1, E_2) = \frac{|E_1 \cap E_2|}{min(|E_1|, |E_2|)}$$

Con il termine E_i ci si riferisce al traffico associato all'allarme *i-esimo*. Questa metrica varia tra [0,1], con 0 corrispondente a traffico senza intersezione e 1 traffico identico.

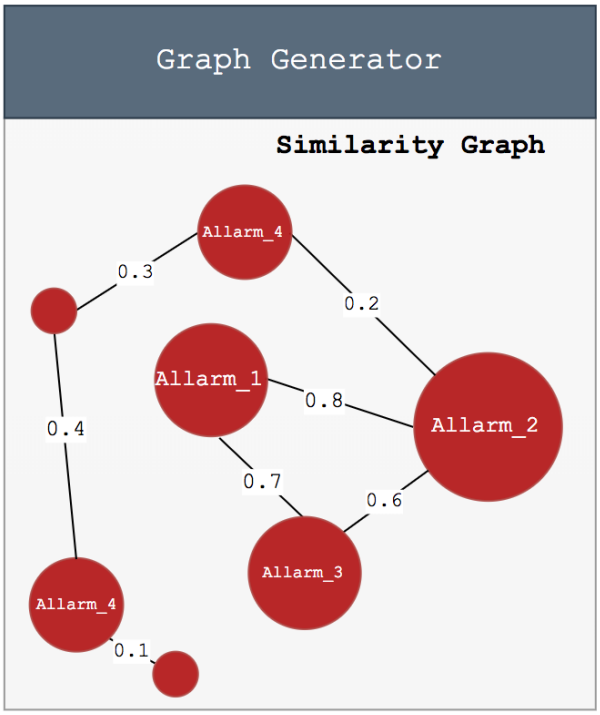


Figura 6.2: Esempio di grafo di similarità.

6.4 Community mining

Si è detto che il grafo di similarità descrive, attraverso nodi e archi, il traffico comune ai diversi allarmi. Sulla base del valore degli archi di questo grafo è possibile determinare quali allarmi (nodi) hanno una quantità maggiore di traffico in comune al fine di raggrupparli in *comunità*.

Sono stati proposti diversi algoritmi per l'individuazione delle comunità in grafi sparsi (un grafo sparso ha molti nodi disconnessi dagli altri). Il metodo maggiormente utilizzato per selezionare le comunità è il Louvain algorithm [16]. Questo algoritmo ha il vantaggio di identificare localmente le comunità. Permettendoci così di raggruppare gli allarmi anche in presenza di grafi sparsi.

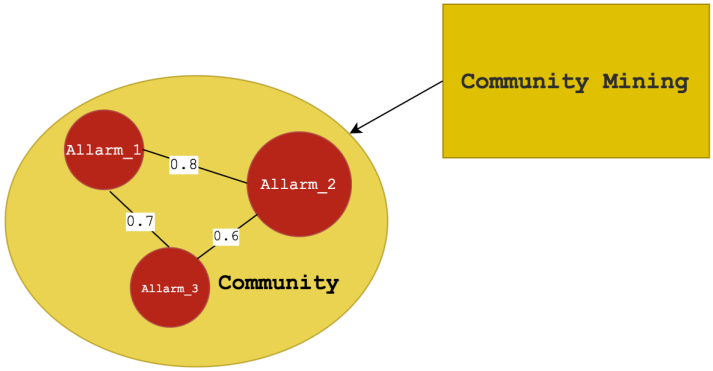


Figura 6.3: Esempio di comunità selezionata dal community mining.

6.5 Combiner

Il ruolo del **combiner** è quello di decidere se ciascuna *comunità*, selezionata dal *community mining*, corrisponde o meno ad un traffico anomalo. Per questo, il **combiner** classifica le comunità in due categorie, *accepted* e *rejected*, che rappresentano rispettivamente le comunità segnalate come anomale o da scartare.

La classe di una *comunità* è determinata tramite una combinazione di strategie basate su tecniche di machine learning e pattern classifiers.

6.5.1 Background:combining detectors

In genere una strategia di *combining* si presenta come una selezione di *detector* o una fusione dei valori ottenuti. La selezione del *detector* consiste nel trovare quello che si presta meglio per la classificazione di un elemento (nel caso in questione, gli elementi da classificare sono le *comunità*). Questo approccio è molto comodo e immediato, tuttavia, la selezione del *detector* non è un'operazione semplice da svolgere.

Una metodologia proposta prevede un sistema di voto, dove ogni *detector* si esprime dando in output un giudizio che valuta una determinata classe. La strategia di *combining* si limita quindi a far emergere la classe votata in maggioranza. La probabilità di prendere la decisione corretta con il voto a maggioran-

za dipende dalla probabilità che ciascun rivelatore fornisca l'output corretto, ovvero:

$$P_{mag}(L) = \sum_{m=\frac{L}{2}+1}^{L} \begin{pmatrix} L \\ m \end{pmatrix} p^m (1-p)^{L-m}$$

Dove L è il numero di *detector* e p è l'*accuracy*. Il risultato, conosciuto anche come il *Condorcet Jury Theorem*, è il seguente:

- Se $p > 0.5 \Rightarrow P_{mag}(L)$ è monotonicamente decrescente in L e $P_{maj}(L) \rightarrow$ inf
- Se $p < 0.5 \Rightarrow P_{mag}(L)$ è monotonicamente crescente in L e $P_{mag}(L) \rightarrow 0$ con $L \rightarrow inf$
- Se $p = 0.5$ allora la $P_{maj}(L) = 0.5$, $\forall L$

Questo teorema evidenzia il vantaggio di combinare più *detector* (con un *accuracy* $p > 0,5$) rispetto ad utilizzarne uno soltanto.

6.5.2 Confidence Score

Ogni *anomaly detector* emette un valore binario che indica se il traffico analizzato è anomalo oppure no. Ciò significa che per ogni *comunità* in un *grafo di somiglianza* un *detector* effettua una votazione indiretta per dire se quella *comunità* contiene traffico anomalo oppure no. In pratica se in una *comunità* è presente un allarme emesso da un *detector*, quel detector ha contribuito, mediante il suo voto, ad etichettare la *comunità* come anomala. Sebbene questo sia sufficiente per il calcolo della votazione a maggioranza, questo valore binario è poco preciso per ottenere un risultato esatto. Inoltre, i voti dei *detector* possono variare in modo significativo, a seconda dei valori assegnati ai loro parametri. Per evitare queste difficoltà, si attua una tecnica di valutazione che aggiunge un fattore di *fiducia* ad ogni voto emesso da un *detector*.
Questo valore di fiducia, detto anche ***confidence score*** per un *detector* d e una *comunità* c, può essere definito come:

$$\varphi_d(c) = \frac{\phi(c)}{T_d}$$

dove T_d è il numero totale di configurazioni possibili per il *detector* d e $\phi(c)$ è il numero di queste configurazioni che riportano almeno un allarme per la configurazione c. Il valore di fiducia (*confidence score*) è continuo e varia nell'intervallo $[0, 1]$, dove 0 ci comunica che il *detector* non conosce la *comunità*, mentre 1 indica che tutte le configurazioni del *detector* evidenziano anomalie per una cerca *comunità* c.

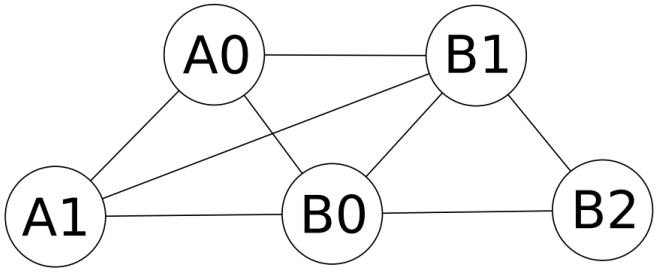


Figura 6.4: Esempio di community c_{ex} composta da cinque allarmi. Assunto che l'input del *similarity estimator*, X_i, consiste dell'output di tre *detector* X=A,B,C con tre differenti set di parametri i=0,1,2, i valori di **confidence scores** sono:$\varphi_A(c_{ex}) = 0.66, \varphi_B(c_{ex}) = 0.1.0, \varphi_C(c_{ex}) = 0.0$

6.5.3 Strategie di combining

Ora che è stato definita una metrica per valutare il grado di accuratezza degli output forniti dai singoli *detector* si utilizzano delle strategie capaci di combinare i **confidence scores** precedentemente ottenuti per una data *comunità* c in un valore, $\mu(c)$ che accetta la *comunità* in esame come anomala se il suo valore è maggiore di 0.5.

L'aggregazione del punteggio di confidenza di una *comunità* in media ci consente di fare affidamento in egual misura sui voti di tutti i *detector*. Formalmente, per una *comunità* c e usando L *detector*, la media è:

$$\mu(c) = \sum_{i=1}^{L} \varphi_i(c)$$

Nell'esempio mostrato in Fig.6.4 la media di tutti i punteggi equivale a 5/9, quindi questa strategia di *combining* classifica la *comunità* c_{ex} come *accepted*, quindi anomala.
Esistono anche altre strategie, come la *minimum confidence score* che accetta come anomale solo le *comunità* che sono state definite anomale da tutti i *detector*. Nel caso valutato in Fig.6.4 se applicassimo questa strategia,la *comunità* verrebbè scartata perché non tutti i *detector* si sono pronunciati allo stesso modo.

6.6 Anomaly detector

Il progetto MawiLab implementa quattro *anomaly detectors* non supervisionati, basati su distinti metodi di machine learning. I parametri di ogni *detector*, trattati nel precedente, vengono impostati in tre diverse modalità, in modo da ottenere diversi *confidence score* per ogni *detector*.Quindi, per l'esperimento, l'input per il metodo proposto consiste nelle 12 uscite di tutte le configurazioni (4 *detector* che utilizzano 3 set di parametri differenti). I rilevatori utilizzati sono:

- **Principal component analysis (PCA)**
- **Distribuzione gamma**
- **Trasformata di Hough**
- **Kullback-Leibler**

Questa tesi si concentra nell'implementazione del *detector* che integra la PCA per estrae le principali *features* dei dati col fine di definire comportamenti di traffico normali. Da questo è possibile individuare quello anomalo per

esclusione. Un problema di questo metodo è la difficoltà di mantenere un legame tra i risultati della PCA e i dati originali del traffico. Verrà presentata più avanti una soluzione implementata che consente di superare questo problema.

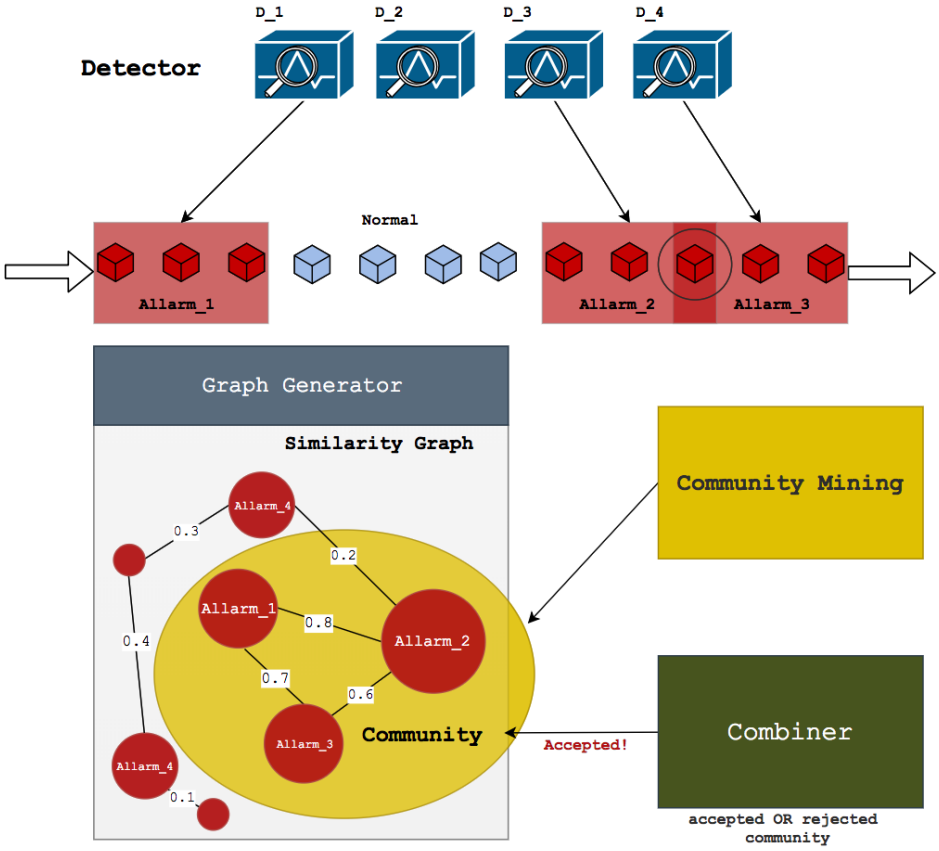


Figura 6.5: Struttura del progetto MawiLab

Capitolo 7

Sviluppo e Risultati

La parte sperimentale di questa tesi si concentra sullo sviluppo di un particolare *detector* del progetto MawiLab.
Nella prima parte del lavoro è stato analizzato il *dataset* col fine di comprenderne la struttura, il contenuto e le caratteristiche dei flussi di traffico che lo compongono. Successivamente sono state sviluppate le componenti del *detector* seguendo le indicazioni trovate in [17][18][19].
Il *detector* scelto basa il suo funzionamento sulla combinazione delle *Random Projection*, esposta in 5.1 e della PCA. Dopo lo sviluppo sono stati eseguiti diverse prove col fine di verificarne il corretto funzionamento.

7.1 Analisi del dataset

Il dataset Mawi è accessibile dal sito internet del progetto MawiLab (http://www.fukuda-lab.org/mawilab) dal quale è possibile, attraverso uno script, selezionare giorno, mese e anno della cattura che si vuole acquisire.

**** OMISSIS ****

7.2 Sviluppo del detector

Dopo aver descritto le soluzioni implementate per trattare il dataset Mawi è stato selezionato un *detector* presente nel progetto MawiLab col fine di svilup-

parlo e verificarne il funzionamento.

Tra i quattro possibili *detector* si è scelto di implementare quello che combina la Principal Component Analysis (PCA) e le *random traffic projection* (dette anche *sketch*) introdotte nei precedenti capitoli. La scelta di questo particolare *detector* deriva dalle seguenti considerazioni:

- La Principal Component Analysis (PCA) si presta bene per mappare grandi quantità di dati in uno spazio di dimensioni ridotte.
- La PCA è una delle tecniche più utilizzate nell'analisi del traffico quindi è semplice trovare soluzioni che consentono di confrontare i risultati ottenuti
- Anche se meno note, le *random traffic projection* sono sempre più utilizzate, non solo per trattare problemi legati ai *big-data*, ma anche per rilevare traffico anomalo.

7.2.1 Implementazione sketches

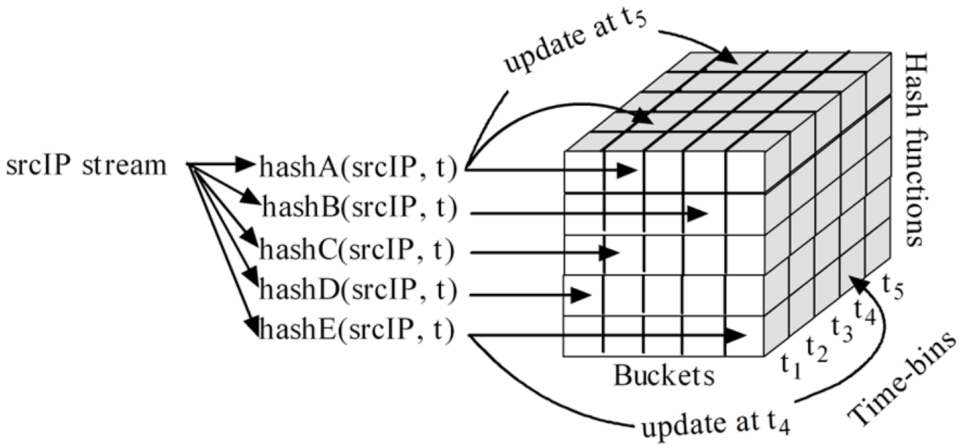


Figura 7.1

Gli *sketches* vengono utilizzati per dividere il traffico principale in N sotto traffici. Per fare questo è stata utilizzata una particolare struttura dati simile a quella riportata in figura 7.1.

*** OMISSIS ***

7.2.2 Applicazione della PCA

La seconda parte del lavoro si è concentrata sullo sviluppo della Principal Component Analysis che per essere applicata ai risultati ottenuti dalla tecnica degli *sketch* ha richiesto particolari modifiche e un'attenta scelta dei parametri.
La soluzione implementata, che può essere consultata in appendice, segue la metodologia utilizzata nell'articolo [17].

*** OMISSIS ***

Appendice A

Appendice - Script Python

**** OMISSIS ****

A.1 Filtraggio File Report MawiLab

**** OMISSIS ****

A.2 Scapy

Scapy è disponibile sia come programma che come modulo Python e consente all'utente di inviare, sniffare, sezionare e modificare i pacchetti di rete. Queste funzionalità sono state utili per analizzare e modificare il traffico del dataset Mawi. Scapy può facilmente gestire la maggior parte dei compiti classici come scansione, tracerouting, sondaggio, unit test, attacchi o analisi della rete.

Bibliografia

[1] S. Gaglio e G. Lo Re. *Advances onto the Internet of Things: How Ontologies Make the Internet of Things Meaningful*. Springer, 2014.

[2] C. for Strategic e I. S. (2013). *The economic impact of cybercrime and cyber espionage. Technical report. McAfee*. URL: `http://www.mcafee.com/us/resources/reports/rp-economic-impact-cybercrime.pdf`.

[3] F. Tonacci. *Cybercrime, il web sotto attacco per banche, aziende e Stati un conto da 575 miliardi, 2015*. URL: `http://www.repubblica.it/economia/affari-efinanza/2015/11/09/news/cybercrime_il_web_sotto_attacco_per_banche_aziende_e_stati_un_conto_da_575_miliardi-127010094/`.

[4] F. Sabahi e A. Movaghar. "Intrusion Detection: A Survey". In: *2008 Third International Conference on Systems and Networks Communications*. IEEE, 2008, pp. 23–26.

[5] A. L. Samuel. "IBM Journal of research and development 3.3". In: *Some studies in machine learning using the game of checkers*. IEEE, 1959, pp. 210–229.

[6] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin e W.-Y. Lin. "Intrusion detection by machine learning: A review". In: ().

[7] R. T. e. J. H. F. Trevor Hastie. Rapp. tecn. 1959, p. 10.

[8] M. Tavallaee, E. Bagheri, W. Lu e A. A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set". In: *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*. 2009.

[9] MAWILab. URL: `http://www.fukuda-lab.org/mawilab/`.

[10] R. Fontugne, P. Borgnat, P. Abry e K. Fukuda. *MAWILab : Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking*. Rapp. tecn. 2010, p. 552071.

[11] M. C. A. Lakhina e C. Diot. "Diagnosing network-wide traffic anomalies. In Proceedings of ACM SIGCOMM". In: (2004). Diagnosing network-wide traffic anomalies. In Proceedings of ACM SIGCOMM, Aug. 2004.

[12] X. Li *et al. Detection and Identification of Network Anomalies Using Sketch Subspaces General Terms*. Rapp. tecn. 2006.

[13] K. Park. *The Johnson-Lindenstrauss Lemma*. Rapp. tecn.

[14] Partow. *General Purpose Hash Function Algorithms*. URL: `http://www.partow.net/programming/hashfunctions`.

[15] G. Cormode e S. Muthukrishnan. *An Improved Data Stream Summary: The Count-Min Sketch and its Applications*. Rapp. tecn.

[16] V. D. Blondel, J.-L. Guillaume, R. Lambiotte e E. Lefebvre. *Fast unfolding of communities in large networks*. Rapp. tecn. 2008.

[17] Y. Kanda, K. Fukuda e T. Sugawara. "Evaluation of anomaly detection based on sketch and PCA". In: *GLOBECOM - IEEE Global Telecommunications Conference* December 2010 (2010).

[18] S. Pukkawanna, H. Hazeyama, Y. Kadobayashi e S. Yamaguchi. "Detecting anomalies in massive traffic with sketches." In: *Cfi* January 2015 (2014), 14:1–14:2.

[19] C. Science. "Mining Anomalies Using Traffic Feature Distributions". In: (2005).