**Firewall e IDS**

24/10

**Firewall**

Il firewall è un dispositivo inteso a proteggere una rete o una porzione di rete, che fornisce uno strato di difesa, isolando i sistemi interni (intera rete o una porzione di una rete) dalle reti esterne (internet) o da altre parti della rete interna (esempio: se ho una rete aziendale).

In genere, viene inserito un firewall tra la rete della sede e Internet per stabilire un collegamento controllato ed erigere un muro o perimetro di sicurezza esterno. Per proteggere la rete interna dagli attacchi basati su Internet per fornire un unico punto di strozzatura in cui imporre sicurezza e controllo.

I firewall vengono implementati anche all'interno della rete aziendale per separare parti della rete.

Un firewall serve a proteggere da un certo numero di attacchi, tipicamente esterni ma anche potenzialmente interni, intesi come collaborazione con altri attacchi.

Il firewall è inteso come un singolo punto di controllo, dove posso implementare politiche di sicurezza e di accountability, tracciamento.

La presenza del firewall mi permette di garantire fino a un certo livello servizi di sicurezza e nel momento in cui quei servizi di sicurezza sono bypassati, mi permette di garantire un certo livello di tracciamento delle operazioni, che mi permetterà di andare a capire dove è il problema e come può essere risolto per evitare intrusioni future.

Obiettivi di progetto del firewall:

1. Tutto il traffico dall'interno verso l'esterno e viceversa deve passare attraverso il firewall, tipicamente evitando instradamenti asimmetrici.

Immagine che contiene diagramma, testo, schizzo, clipart

Descrizione generata automaticamente

Supponiamo che la nostra rete abbia due router che agiscano anche da firewall e che garantiscano l’accesso a internet. Se ho delle richieste che entrano dal FW1 e escono dal FW2 mettono in crisi i firewall, perché non riusciranno a mappare le richieste con le risposte, e potrebbero identificare del traffico legittimo come traffico malevolo, in particolare quello della freccia rossa. Perché che arrivi al fw una risposta senza una precedente richiesta potrebbe dar luogo all’implementazione di una politica di sicurezza che blocca tutto il traffico di risposta per cui non ho osservato una richiesta.

È fondamentale il fatto che il fw sia un unico punto di controllo (dal punto di vista logico); se ci sono porzioni di rete diverse devono essere servite da fw diversi. È importante che quando una richiesta passa attraverso un’istanza del fw anche la risposta passi attraverso la stessa istanza del fw.

Se violiamo questa regola il firewall diventa inefficace.

Ciò si ottiene bloccando fisicamente (dal punto di vista telematico) tutti gli accessi alla rete locale che non passino tramite firewall.

1. Sarà consentito il transito solo al traffico autorizzato, come definito dalla politica di sicurezza locale.

Vengono utilizzati vari tipi di firewall, che implementano vari tipi di politiche di sicurezza.

1. Il firewall stesso dovrebbe essere immune alla penetrazione.

Per questo dovrebbe essere realizzato da un sistema “sicuro” e aggiornato costantemente; visto che il fw è il punto di accesso primario dei sistemi della rete, non deve essere il fw il punto di debolezza della rete stessa.

Ciò implica l'uso di un sistema rafforzato con un sistema operativo protetto.

I sistemi informatici affidabili sono adatti per ospitare un firewall e sono spesso richiesti nelle applicazioni governative.

**Tecniche firewall**

Il modo in cui realizzo gli obiettivi di sicurezza è attraverso le politiche di sicurezza, che sono implementate tramite i servizi di sicurezza, e il fw è uno di questi meccanismi abilitanti delle politiche di sicurezza.

È possibile utilizzare quattro tecniche per controllare l’accesso e applicare la politica di sicurezza del sito.

* **Controllo dei servizi**: è possibile accedere ai tipi di servizi Internet, in entrata e in uscita; quali servizi possono essere acceduti.
* **Controllo della direzione di accesso**: direzione in cui può essere avviata una determinata richiesta di servizio. Posso accedere a quali servizi di internet? Utenti di internet posso accedere a servizi interni? Si, no, combinazioni varie.
* **Controllo utente**: quali utenti possono accedere a quali risorse. Spesso applicato agli utenti interni, per gli utenti esterni è richiesta l'autenticazione.
* **Controllo del comportamento**: come viene utilizzato un servizio. Ad esempio, all’interno di un servizio abilitato a determinati utenti potrei decidere di restringere l’accesso ad alcuni contenuti di un servizio abilitato oppure potrei limitare la rate di accesso o la quantità di dati che possono essere scambiati. Posso forzare delle politiche che modellino il comportamento degli utenti.

Antispam o accesso limitato ad alcuni contenuti.

È fondamentale nei fw l’accountability, la possibilità del tracciamento. Abbiamo visto sulla prima parte del corso che uno dei meccanismi principe per effettuare il monitoraggio degli utenti che accedono a un servizio è l’uso dei log del servizio. Il fw tipicamente genererà dei log e il salvataggio dei log nella modalità che l’amministratore di rete ritiene più opportuna e l’accesso ai log e la possibilità di analisi del log sarà uno strumento di fondamentale importanza per realizzare alcune delle politiche di sicurezza. Ad esempio per controllare se le politiche di sicurezza messe in atto funzionano bene oppure se alcune di queste politiche non sono pienamente realizzate.

**Modello generale di FW**

**Immagine che contiene testo, schermata, diagramma, design

Descrizione generata automaticamente**

Il modello del fw è quello di un unico punto di controllo (single choke point) tra una rete e il resto del mondo, che può essere la porzione più esterna della rete che si vuole proteggere o l’intera internet, dal punto di vista concettuale non cambia niente.

Il fw è anche il luogo ideale per monitorare gli eventi, quali l’attività dei propri utenti, sia per monitorare altri tipi di eventi, ad esempio il consumo di risorse del fw stesso.

Proprio perché il fw è il singolo punto di passaggio, bisogna evitare che il fw diventi, dal punto di vista prestazionale, il collo di bottiglia del sistema.

Il fw è tipicamente il punto della rete ideale per realizzare un gateway VPN (Virtual Private Network), permettendo così a utenti esterni di accedere alla rete come se fossero all’interno della rete. Gli utenti esterni possono accedere in qualità di utenti esterni a servizi che sono messi in opera sulla rete interna, quini attraverso il fw, oppure possono accedere come se fossero interni attraverso la VPN. La differenza è sostanziale perché per accedere tramite una vpn gli utenti sono identificati e tracciati in maniera più fine rispetto a un generico utente che accede via internet.

Il fw può essere il punto in cui vengono messe in opera altre funzioni di rete, e questo ha di nuovo a che fare con il problema del dimensionamento delle prestazioni. Se la rete utilizza sia indirizzi pubblici che indirizzi provati ad esempio il fw è il luogo ideale per realizzare servizi di natting. Se la rete utilizza indirizzi dinamici il fw potrebbe essere il punto ideale dove mettere in opera servizi di DHCP, e così via.

* Ci sono delle funzioni che potrebbero essere collocate nel fw e queste potrebbero, in alcuni casi, appesantirne il funzionamento e quindi limitare un po’ le prestazioni.

**Limitazioni del firewall**

1. Il firewall non può proteggere dagli attacchi che bypassano per il firewall, quindi da accessi che passano per canali secondari e che non passano per il fw.

Immagine che contiene diagramma, disegno, schizzo, Line art

Descrizione generata automaticamente

Consideriamo questo schema. Se il fw è realizzato bene e aggiornato, i tentativi di accesso da parte degli attaccanti internet vengono bloccati (frecce blu).

Se all’interno della rete ho un access point wifi in cui la chiave è semplice (tipo “prova”) è ragionevole che uno di questi attaccanti riesca ad entrare nella rete attraverso una connessione wifi mal protetta (freccia rossa).

Se invece abbiamo un utente che deve accedere ad un determinato servizio e il fw è configurato per impedire l’accesso a quel servizio, l’utente invece che effettuare una richiesta per farsi aprire la porta del fw accende il cellulare e passa tramite una connessione 4G/5G e accede a quel servizio, creando un accesso secondario alla rete.

La differenza rispetto a prima è che un access point mal configurato è facile da rilevare perché è un dispositivo fisico, ma un dispositivo ad uso intermittente come un cellulare crea un punto di accesso secondario potenzialmente a intrusi che possono usare questo accesso per fare danni o per utilizzare questa rete per partecipare ad un attacco (spesso succede).

I sistemi interni possono avere funzionalità di chiamata in uscita per connettersi a un ISP.

È possibile che dall'esterno dell'organizzazione sia possibile accedere a una LAN wireless protetta in modo improprio.

1. Il firewall potrebbe non proteggere completamente dalle minacce interne, che siano volontarie o involontarie.

Immagine che contiene diagramma, bianco, schizzo, Line art

Descrizione generata automaticamente

Se la vittima (V) e l’attaccante stanno all’interno della rete, l’attaccante attacca la vittima e questa non può essere protetta dal fw perché sono entrambi dietro il fw.

Questo attacco può essere consapevole o inconsapevole.

È inconsapevole quando si utilizza la pratica del BYOD (Bring Your Own Device), molto diffusa negli ultimi anni: sull’attività lavorativa si utilizzano i propri dispositivi.

Un dispositivo può essere infettato all'esterno della rete aziendale e quindi connesso e utilizzato internamente. È anche difficile poi tracciare come si è diffuso il contagio all’interno dell’azienda.

Questo può essere prevenuto dalle aziende impedendo l’uso di dispositivi non autorizzati all’interno dell’azienda.

L’attacco è consapevole quando qualcuno volontariamente infetta un dispositivo interno per propagare un malware o per realizzare un attacco.

**FW basato su regole o basato su policy**

Possiamo avere fw con comportamento statico o dinamico. I fw con il comportamento statico sono più semplici e più diffusi e sono i fw basati su delle regole. I fw basati sulle policy sono più dinamici.

**Firewall basati su regole**: un fw permette tutto quello che le regole gli permettono di fare; quindi, permette l’accesso ai servizi esplicitamente permessi dalle regole e blocca tutti quelli bloccati dalle regole.

I sistemi firewall basati su regole utilizzano regole per controllare la comunicazione tra host all'interno e all'esterno del firewall.

I sistemi basati su regole sono statici: non possono fare nulla per cui non sono stati espressamente configurati. Deve esserci una riga in uno dei loro file di configurazione da qualche parte che dice loro esattamente cosa fare con ogni pacchetto che scorre attraverso il dispositivo.

(iptables è il classico esempio di firewall rule-based: è organizzato in tabelle, ogni tabella ha una catena e ogni catena ha una regola.)

La configurazione statica può essere aggiornata ma non si aggiorna da sola.

Ciò rende il sistema più semplice da configurare, ma meno flessibile e meno adattivo alle mutevoli circostanze.

**Firewall basati sulle policy**: sono più dinamici perché le policy (che possono essere un concetto di livello più elevato, più astratto) definiscono delle condizioni più generali sotto cui può essere permesso l’accesso ai servizi.

I sistemi basati sulle policy sono più flessibili.

Permettono all'amministratore di definire le condizioni alle quali sono consentiti tipi generali di comunicazione, di specificare quali funzioni e servizi verranno eseguiti per fornire tale comunicazione.

Un sistema basato su policy può impostare dinamicamente la comunicazione consentita verso indirizzi IP casuali.

Un esempio di fw basato sulle policy è che tutti gli utenti che sono stati autenticati accedono a internet o che tutti gli utenti che arrivano tramite VPN possono accedere ai servizi.

In generale tutto quello che riguarda politiche basate su VPN e autenticazione possono essere realizzate tramite fw basati sulle policy.

Qualsiasi sistema che supporti l'intestazione di autenticazione IPsec e l'incapsulamento del payload di sicurezza è considerato un sistema basato su policy.

* Nella realtà abbiamo sempre un mix dei due fw, perché tipicamente abbiamo che anche i fw basati sulle policy, una volta che la policy è stata definita, utilizzano poi delle regole.

Esempio: tutti gli utenti autenticati nel gruppo a hanno accesso a questi servizi tramite queste regole.

Quindi possiamo dire che il livello di astrazione più elevato è realizzato con le policy e il livello di astrazione meno elevato e più stabile è realizzato con le regole.

FINE 24/10

25/10

**Tipi di firewall**

Un firewall può essere una difesa che agisce come un filtro:

* **Positivo**: FW specifica i pacchetti autorizzati a passare
* **Negativo**: FW specifica i pacchetti a cui è stato negato il passaggio

Si va quindi a definire la regola di default e poi i filtri su cui si applica la regola di default, che di fatto rappresentano le eccezioni.

La regola di default del FW positivo è del tipo blocca tutto: impedisce a tutto il traffico di passare, e quindi le regole o le policy stabilite dal fw rappresentano le eccezioni rispetto al comportamento di default, quindi specificano il tipo di traffico che può passare o il tipo di utente che può passare.

Quando invece il FW si comporta come un filtro negativo la regola di default (o policy di default) è di accettare tutto il traffico e noi andiamo a specificare solo il traffico che non è autorizzato a passare.

La prima soluzione è più restrittiva, mentre l’altro caso è chiaramente più permissivo. Nel secondo caso si parte dall’assunzione che non ci saranno problemi, e via via che si verifica un problema si vanno a “curare le falle” restringendo i permessi degli utenti o il permesso di far circolare un determinato traffico.

La seconda policy è meno utilizzata, di solito si usa la prima quando si mette in piedi un fw.

Un FW può ispezionare:

* le **intestazioni** dei pacchetti, allo strato 2, 3, 4 o applicativo, a seconda del vantage point (che può essere definito sia a livello fisico, cioè la locazione fisica di rete dell’interconnessione del fw, oppure a livello logico, cioè lo strato protocollare in cui il fw opera) in cui è posizionato il fw
* **payload** dei pacchetti: sappiamo che le applicazioni utilizzano tecniche crittografiche per assicurare la confidenzialità dei dati, quindi solo se ho un fw che arriva fino allo strato applicativo posso tipicamente ispezionare il payload dei pacchetti (intendendo come payload come quello del protocollo di strato più alto)
* i **pattern**, cioè l’insieme dei pacchetti all’interno di un flusso. Ci potrebbero essere delle configurazioni all’interno di un flusso che possono rappresentare un tipico segno di malfunzionamento, un’anomalia o un attacco informatico.

Ci possono essere quindi un insieme di azioni che, se prese singolarmente non danno origine a nessuna minaccia, ma se arrivano tutte insieme nell’arco di una finestra temporale limitata possono rappresentare “la firma” di un attacco noto.

Ci sono 4 tipi di FW, che tipicamente solo alternativi, tranne gli ultimi due che spesso lavorano insieme:

* Firewall che effettua il **packet filtering** (es: con netfilter abbiamo specificato delle regole che riguardano dei campi dei pacchetti)
* **Stateful inspection** firewall: non si guardano i singoli pacchetti come prima, ma si vanno a guardare le connessioni, nel caso tcp, o i flussi, nel caso udp
* **Gateway a livello applicativo**
* **Gateway a livello di circuito**

Immagine che contiene testo, schermata, design

Descrizione generata automaticamente

Il primo tipo di FW si applica a tutti i pacchetti che attraversano la nostra macchina, permette di modificare i pacchetti oppure di impedirne il passaggio. Agisce a livello protocollare 3 e 4: in realtà agisce a livello 3, tenendo anche in considerazione informazioni di strato 4. In alcuni casi questi fw possono essere posizionati a livello 2 per non essere visibili né dagli utenti della rete né da utenti esterni alla rete -> posso essere collocati “come uno switch”.

La differenza è che, se operano su un nodo che agisce da router entrambe le porte (di ingresso e di uscita) avranno un indirizzo ip e il pacchetto sarà modificato in ogni caso all’attraversamento del fw; se poi il fw agisce anche da nat ci sarà anche il cambiamento delle porte.

Se invece il FW agisce su un dispositivo di strato 2 il suo comportamento è trasparente, il traffico ammesso passerà indisturbato e non ci si accorgerà della presenza del fw.

Invece, nel caso in cui il traffico soddisfa una delle regole che richiedono modifiche, il traffico sarà modificato o gli sarà impedito di attraversare il nodo.

Vengono utilizzate le informazioni allo strato 3 e allo strato 4: indirizzo ip mittente e destinazione, porta mittente e destinazione, il protocollo di strato 4 (tcp, udp, icmp) e i flag dei vari protocolli o anche l’interfaccia di ingresso o di uscita. In alcuni casi potremmo avere che l’operazione di natting potrebbe non essere effettuata per scarsità di indirizzi ip, ma per mascherare l’identità degli utenti che escono dalla rete.

Per quanto riguarda lo stateful inspection firewall, è “un’evoluzione” del packet filtering firewall. Si basa sul concetto di sessione: una sessione viene creata nel momento in cui viene intercettato il primo pacchetto della sessione stessa.

Nel caso di sessione tcp vanno intercettati i pacchetti SYN, quelli che aprono il three-way handshake. Nel caso di pacchetti udp deve essere intercettato il primo pacchetto che viene aperto da un certo indirizzo ip mittente verso un certo indirizzo ip destinatario, da una certa porta mittente verso una certa porta destinatario.

Una volta identificata una connessione posso definire il comportamento: è fondamentale anche la direzione perché questi fw permettono il controllo sulla direzione di accesso (a differenza del packet filtering fw), proprio perché vado ad intercettare il primo pacchetto e quindi vedo se il pacchetto è originato da un utente interno verso un utente esterno o viceversa. Si prende quindi una decisione sul primo pacchetto, e il comportamento attuato sarà mantenuto per tutti i pacchetti di quello specifico flusso: viene conservata un’informazione di stato, cioè un database interno, che permette l’esecuzione di questo tipo di politiche. Stiamo parlando sempre di fw rule-based, solo che la regola o la policy è applicata al flusso e non al singolo pacchetto.

(Vedremo in laboratorio un esempio su questo: iptables può essere configurata sia come packet filtering fw sia come stateful inspection fw.)

**Packet filtering firewall**

Applica una serie di regole ai pacchetti in entrata/uscita per:

* Inoltrare
* Negare
* Modificare

Informazioni tipiche utilizzate:

* IP di origine/destinazione
* Porta di origine/destinazione
* Campo protocollo IP
* Interfaccia di rete

Definire una politica predefinita (inoltra/ignora).

Immagine che contiene testo, schermata, numero, Parallelo

Descrizione generata automaticamente

Questi sono esempi di alcuni tipi di regole, realizzati tramite tabella grafica, che possono essere facilmente mappati su qualsiasi fw.

La prima regola ci dice di bloccare tutti i tentativi di accesso all’host destinazione che va sotto il nome di SPIGOT. Ci sarà quindi una maschera di rete associata a una determinata rete. Non ci sono vincoli sull’host di ingresso, e anche sulle porte mittente e destinatario.

È invece ammesso il traffico in cui il mittente è il nostro gw e la porta mittente è la 25. Questo permette al server di posta elettronica (è quello che usa la porta 25) di uscire verso qualsiasi altra destinazione.

Questa è una policy classica perché il serve di posta elettronica, ammesso che sia interno come nell’esempio, deve poter inviare e ricevere la posta da tutti gli altri server che stanno in rete: non posso effettuare restrizioni sugli host di destinazione.

Tuttavia, sappiamo che in alcuni casi esistono delle liste di server SMTP che sono compromessi. In quel caso posso applicare queste policy al fw oppure andare a configurare un fw del server stesso che impedisce al server stesso di andare a colloquiare con questi server, oppure all’interno del software che gestisce la posta elettronica posso andare a configurare le liste dei server con cui non bisogna neanche provare a colloquiare.

Quindi nel momento in cui ho una regola aggiuntiva ho abbastanza flessibilità per decidere dove metterla, dipende da quanto è semplice aggiungere regole in questo fw e da quanto è dinamico il processo.

In alcuni casi il ruolo del fw può essere complementato dalla configurazione dei nostri client o dei nostri server, il che non significa necessariamente andare a installare un fw anche sul client e sul server, cosa che può essere fatta senza troppi problemi, a volte è sufficiente imporre delle regole a livello applicativo.

La regola dell’insieme B ci dice che la regola di default è quella di bloccare tutto il traffico: fw di tipo positivo. Qualsiasi altra regola sarà un’eccezione a questa. Una regola di questo tipo deve essere posta in fondo all’elenco: se ho un certo numero di regole e nessuna regola viene eseguita, allora arrivo in fondo all’elenco dove c’è la regola di default che viene eseguita nel momento in cui nessun’altra viene eseguita.

Immagine che contiene testo, Carattere, ricevuta, schermata

Descrizione generata automaticamenteTipicamente ho una tabella con un certo numero di righe: se la regola specificata ad una determinata riga viene eseguita perché viene soddisfatta la condizione specificata nella regola da un pacchetto o da un flusso, allora il resto delle regole non vengono eseguite. Se però nell’esaminare le regole non trovo nessuna corrispondenza e arrivo alla regola di default, quella viene eseguita sicuramente.

In alcuni casi più sofisticati dopo l’esecuzione di una regola posso passare all’esecuzione di un’altra regola. Ad esempio, la prima regola può essere una regola di monitoring, quindi per ogni flusso di questo tipo genera un record, e poi quel tipo di traffico, oltre ad essere loggato, potrebbe anche necessitare di essere bloccato o di garantirgli il transito.

**Possibili attacchi**

In generale, possibili attacchi sono tutti quelli che sfruttano:

* **Vulnerabilità a livello applicativo**: non possiamo fare niente se ho un packet filtering firewall perché viene analizzato lo strato 3 e lo strato 4 e quindi se ho problemi a livello applicativo bypassa completamente il fw e quindi in quel caso ho bisogno di un fw che agisce a livello applicativo, quindi con un diverso vantage point.
* **Configurazione sbagliata**: mi accorgo della configurazione sbagliata quando qualcuno riesce a fare qualcosa che non volevo fosse fatto.
* **Mancanza di autenticazione**: se posso fare autenticare i miei utenti è sempre bene farlo perché questo garantisce un ulteriore livello di sicurezza e spesso garantisce anche la possibilità di differenziare le regole tra utente autenticato e non autenticato.

Attacchi noti:

**IP Spoofing**: consiste nel far impersonare con la propria macchina che ha un determinato indirizzo IP un’altra macchina che ha un indirizzo IP diverso. Questo può essere fatto per partecipare ad un attacco o per effettuare un attacco. La differenza è che nel primo caso, se partecipo ad un attacco, il mio fw mi può impedire di partecipare ma la vittima non è all’interno della mia rete; invece, se la vittima è all’interno della mia rete significa che qualcuno all’esterno della mia rete sta effettuando un attacco di impersonificazione o di mascheramento.

Questo attacco prevede quindi l’utilizzo di un indirizzo ip fasullo: è una tra le cose più semplici che si possono fare.

* Contromisura: controllare l'interfaccia

**Attacchi di source routing**: consistono in quei tentativi in cui all’interno del pacchetto c’è scritta tutta la strada o parti importanti della strada che il pacchetto deve seguire per arrivare a destinazione. È un qualcosa che si specifica molto facilmente con IPv6 perché esistono una lunga serie di opzioni, tra cui quella di specificare un certo numero di nodi intermedi, che possono essere tutti i nodi lungo il percorso o alcuni.

Quindi quando ho un pacchetto che mi dice qual è il prossimo router da raggiungere per arrivare a destinazione, tipicamente l’instradamenti viene fatto verso il prossimo salto e non verso la destinazione reale. Poi una volta arrivati a quel salto, viene tolto il pezzo dell’intestazione e viene instradato il pacchetto verso il prossimo salto. In questo modo è possibile personalizzare l’instradamento del traffico.

Alcune volte questo viene fatto per bypassare alcune sezioni della rete dove ci sono dei fw potenti o che ci sono dei dispositivi che possono intercettare tentativi di attacco. Quindi una delle classiche regole è quella di droppare tutto il traffico che ha opzioni di source routing, proprio perché tipicamente queste opzioni vengono utilizzate tipicamente solo all’interno dei domini e di solito i fw proteggono i bordi dei domini.

* Contromisura: selezionare l'opzione pertinente

**Attacchi con piccoli frammenti**: vediamo come è fatto un pacchetto ip.

Abbiamo una certa intestazione, l’intestazione di strato 4, quella di strato 5 e poi il payload.

Immagine che contiene testo, Carattere, linea, schermata

Descrizione generata automaticamente

Potrei avere delle regole che impediscono a una determinata combinazione di indirizzi ip mittente e destinatario e porta mittente e destinatario di attraversare il fw.

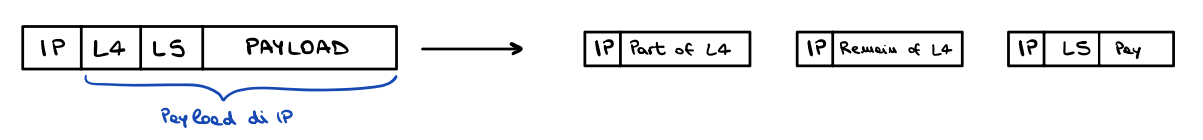
Un attacco che in alcuni casi ha successo è un attacco che si basa sull’estremizzazione dell’opzione di frammentazione dei pacchetti.

Questo pacchetto può essere frammentato in un pacchetto che ha ip, l4, l5 e un pezzetto dei payload (“P”) e poi in un pacchetto che ha ip e il resto del payload (“ayload”). Il payload è stato spezzato in una parte iniziale, rappresentata da P, e poi da tutto il resto della stringa. Posso quindi spezzare tutto il payload di ip in più parti, replicando solo l’intestazione ip attivando l’opzione more fragment e l’offset che ci dice quale parte del pacchetto sto trattando (quale parte del frammento). Questo è il modo classico di fare la frammentazione.

Se ho un problema nell’attraversare il fw dovuto all’applicazione di indirizzi mittente e destinazione allo strato 3 e allo strato 4 quel primo frammento non passerà mai (x rossa).

Molti fw però, se fanno invece passare quel frammento, poi gli altri non li considerano più, e quindi se fanno passare il primo fanno passare tutti i frammenti dello stesso pacchetto. Come faccio a ricostruire i frammenti dello stesso pacchetto? Ho un campo identification all’interno dell’intestazione ip.

In cosa consiste l’attacco? Consiste nel fare frammenti piccolissimi:



Posso quindi spezzare il pacchetto in frammenti piccolissimi, in cui il primo non contiene tutta l’intestazione di strato 4 o non la contiene per niente. Se quel pacchetto passa, passeranno tutti gli altri, e di fatto ho bypassato il fw utilizzando questo schema. È inefficiente dal punto di vista del traffico perché devo generare un elevato numero di pacchetti con un sacco di intestazioni ip, però riesco comunque ad accedere, magari più lentamente ma riesco a passare.

* Contromisura: verificare la dimensione minima di un pacchetto IP, che non può scendere sotto un certo valore.

Gli attacchi di solito si basano su piccole cose, su cavilli, su dettagli, su implementazioni incomplete oppure poco accurate, che poi vengono aggiustate nel momento in cui ci si accorge che quello può diventare un problema per la sicurezza. Quando sono state create magari sono state delle sviste oppure non si è pensato a tutte le possibili casistiche da esplorare per evitare problemi. Quindi viene generata o una nuova versione del protocollo o delle regole sui fw in grado di arginare tentativi di attacco che sfruttino vulnerabilità di protocollo.

**Firewall di ispezione con stato**

Le stesse cose dette sui fw packet filtering si applicano ai fw basati sul concetto di connessione o di flusso.

Nel concetto di connessione e di flusso tipicamente si utilizzano i termini di NEW, RELATED o ESTABLISHED, almeno quando si utilizza iptables.

Le decisioni le effettuo solo sui flussi nuovi, cioè sui primi pacchetti dei flussi nuovi. Tutti gli altri pacchetti relativi a quel flusso saranno etichettati come ESTABLISHED o RELATED.

I pacchetti ESTABLISHED sono quei pacchetti per cui ho effettuato una decisione nel momento in cui la connessione o il flusso sono stati creati. Posso avere un comportamento diverso per i pacchetti ESTABLISHED rispetto ai pacchetti NEW. Questo dipende dalle situazioni. Ad esempio, se sono in una situazione in cui ho un volume di traffico elevatissimo posso decidere se limitare il traffico che eccede la capacità della mia rete o dei miei sistemi terminali. Ovviamente non posso andare ad impattare le connessioni tcp che sono già state create perché altrimenti creo solo un disservizio agli utenti che hanno cominciato ad accedere al servizio e poi subiscono un degradamento del servizio stesso. Quindi o l’utente accede al servizio o non ci accede. Questo implica che se devo prendere una decisione si o no devo farlo sul pacchetto di instaurazione della sessione.

Posso avere delle differenze di trattamento sulle nuove connessioni rispetto al traffico delle vecchie connessioni.

In questi casi, per i protocolli che prevedono il numero di sequenza, cioè TCP, è possibile effettuare anche dei controlli sui numeri di sequenza per evitare che pacchetti che sembrano appartenere a una connessione ma non ci appartengono perché hanno un numero di sequenza non compatibile con quelli che sono stati utilizzati fino ad adesso attraversino il fw.

Può anche ispezionare i dati dei protocolli di controllo per identificare le sessioni associate: traffico SIP e RTP.

Può creare anche uno stato per i flussi UDP.

Vediamo un esempio concreto.

Supponiamo di utilizzare una tecnologia di tipo packet filtering.

Immagine che contiene testo, Carattere, diagramma, calligrafia

Descrizione generata automaticamente

La prima regola di questa tabella permette solo l’attraversamento da sinistra verso destra, solo le richieste, cioè la creazione di una connessione. Le risposte passano grazie alla seconda regola della tabella, se non ci fosse questa regola, la policy di default è drop e quindi non passerebbe nessuna risposta.

In questo modo permetto alle richieste e alle risposte di questo tipo di passare:

Immagine che contiene Carattere, testo, linea, calligrafia

Descrizione generata automaticamente

Supponiamo che questo flusso termini e che un attaccante (quello blu) invii del traffico di questo tipo:

Immagine che contiene Carattere, linea, Blu elettrico, calligrafia

Descrizione generata automaticamente

Secondo queste tre regole questo pacchetto passa perché matcha con la seconda regola. Quindi un packet filtering fw farebbe passare questo pacchetto, che ovviamente non è relativo a nessuna sessione e che ragionevolmente è un tentativo di attacco.

Questa, quindi, è una delle principali vulnerabilità di fw basati sul semplice filtraggio dei pacchetti perché effettuano di fatto un criterio a datagramma, cioè ogni pacchetto per se, non guardano al flusso dei pacchetti ma solo al singolo pacchetto.

Abbiamo già capito quindi che un stateful inspection fw offre maggiori garanzie del fw semplice basato sui singoli pacchetti.

Nel caso dello stateful inspection fw avremo che nella tabella sarà presenta anche un flag.

Immagine che contiene testo, Carattere, calligrafia, bianco

Descrizione generata automaticamente

La prima regola ci dice che tutto il traffico che parte da C e va ad S, dove la porta mittente è superiore di 1024, la porta di destinazione è la 443 ed è un nuovo flusso (ci sarebbe anche da scrivere il campo in cui si indica il protocollo di strato 4, che in questo caso diamo per scontato sia tcp: un nuovo flusso tcp significa che nell’intestazione è marcato il bit SYN) l’azione è accept.

Per tutti gli altri pacchetti che appartengono allo stesso flusso (seconda regola) quindi sono related/established, cioè sono pacchetti successivi, l’azione è accept. Questo significa che questi pacchetti troveranno una corrispondenza nel database che a run time si crea il fw. Quando il fw identifica una entry di tipo new crea un nuovo stato nella sua tabella, e per ogni pacchetto che arriva va a controllare se fa parte di un flusso tra quelli elencati. Quindi i pacchetti successivi saranno identificati come pacchetti che appartengono a quel flusso.

Anche le risposte avranno come flag per passare related/established (terza regola).

Questo nel caso in cui il nostro fw permette di smaltire il traffico che è iniziato dagli utenti che stanno dentro la rete verso i nodi che stanno fuori dalla rete, ma non viceversa. Quindi impedisco ai pacchetti nuovi, che non hanno nessuno stato memorizzato nel database della rete, di passare.

In questo caso questo pacchetto (che era quello inviato dall’attaccante) sarebbe scartato:

Immagine che contiene Carattere, calligrafia, testo, linea

Descrizione generata automaticamente

Esistono però anche altri attacchi un po’ più sofisticati. Supponiamo che qualcuno sia in grado di osservare, nei collegamenti indicati dalle frecce verdi, i pacchetti di una connessione tcp.

Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente

Se si accede al traffico i pacchetti tcp, a meno che non utilizzano tecniche di IPsec, sono in chiaro perché la cifratura parte dallo strato applicativo in poi. Posso quindi leggere nell’intestazione tcp i numeri di porta e i numeri di sequenza.

Se riconosco nel collegamento un flusso che va da C verso S, porta 3000 verso 443, il tentativo di attacco (blu) potrebbe andare a buon fine o meno. Dipende da quanto è smart l’attaccante, perché uno stateful inspection fw va a controllare anche se il numero di sequenza utilizzato in questo pacchetto è compatibile con il numero di sequenza della connessione già stabilita.

Se è compatibile, il pacchetto deve passare per forza; se non è compatibile, ad esempio è un pacchetto successivo ma di poco ovviamente passerà, ma se è troppo in avanti non passerà perché viene fatto un controllo di congruenza anche sul numero di sequenza.

Non sempre questa capacità è presente nei fw, però in alcuni casi si.

(Osservazione: tipicamente tutte queste regole, dal punto di vista del fw, sono regole di tipo statiche. Le regole dinamiche sono quelle che riguardano il bilanciamento del traffico.)

Tipi di attacchi: se un attaccante è smart, riesce a identificare una sessione e inoltra pacchetti che sono compatibili con quella sessione i pacchetti passano, non c’è niente da fare. A quel punto poi sarà l’applicazione che dovrà essere abbastanza robusta da gestire pacchetti che non hanno particolare significato.

Immagine che contiene testo, schermata, numero, Carattere

Descrizione generata automaticamente

**Application level gateway**

Application level gateway è un meccanismo diverso, che agisce a livello di trasporto e applicazioni, di fatto bypassa lo strato IP ed è un tipo di fw che richiede l’autenticazione.

Viene applicato di solito tra la parte esterna (internet o una parte meno protetta sempre della rete aziendale) e la parte interna della rete (quella che voglio proteggere). Funziona che il fw rappresenta un intermediario ed effettua un’operazione tcp split, che termina la sessione tcp tra il nodo richiedente e sé stesso e la riapre tra se stesso e il nodo di destinazione, che tipicamente è un applicativo (un server che fa girare un determinato applicativo).

Immagine che contiene diagramma, linea, schermata, cerchio

Descrizione generata automaticamente

Il client si connette al fw, con un’interfaccia web https o in ssh; il fw dal punto di vista protocollare non è altro che un’applicazione. Una volta connesso il client si autentica e dopo che ha passato la fase di autenticazione, indica l’identità del nodo di destinazione. A questo punto il fw può decidere di creare una sessione tra lui e il nodo di destinazione e rilanciare i pacchetti scambiati su questa sessione sulla sessione tra il client e il fw stesso.

Non avrò accesso diretto al server, ma avrò un accesso in cui all’interno del fw c’è la possibilità di avere uno stateful inspection dei pacchetti. Significa che il fw può analizzare il payload dei pacchetti che scambio con l’applicazione, perché anche se mi sono autenticato sono comunque un utente esterno da tenere sotto osservazione. Quindi di solito viene applicata l’ispezione dei pacchetti; questo processo può limitare in maniera significativa la velocità del fw, perché ispezionare il payload significa ispezionare una quantità significativa di traffico.

Faccio la richiesta sull’applicazione, se passo la fase di autenticazione avrò un relay dall’altra parte e poi le risposte arriveranno all’applicazione. In questa fase avrò inspection, che sarà a livello applicativo perché devo poter guardare il payload dei pacchetti. Se questi passano il controllo vengono rilanciati a sinistra, altrimenti no, e magari viene chiusa quella sessione.

Immagine che contiene Carattere, testo, linea, schermata

Descrizione generata automaticamente

L’application level gateway funziona come un relè del traffico a livello di applicazione, noto anche come proxy dell'applicazione.

Comportamenti tipici:

* L'utente si connette con un protocollo, come Telnet
* Il GW richiede la connessione dell'host remoto
* L'utente si autentica
* Il GW connette l'applicazione utente con l'host remoto e inoltra i pacchetti TCP

Supporto di registrazione più efficiente rispetto ai filtri pkt. Suddivisione della connessione TCP + analisi del traffico.

Svantaggi:

* È possibile inoltrare solo le applicazioni supportate
* Carico di elaborazione aggiuntivo

**Circuit-level gateway**

È una versione depotenziata dell’application level gateway. Tipicamente quello che succede è che viene sempre splittata la connessione tcp, ma senza salire allo strato applicativo. Di solito quindi si permette all’utente esterno di accedere a un server su un utente interno mascherandolo. Qualcuno che osserva non vedrà neanche l’id del server, vedrà solo la connessione al proxy fw.

La differenza fondamentale è che in questo caso non si fa la stateful inspection. Si usa questo quando sono gli utenti interni, quindi quando la connessione parte dal server, e non da un client esterno. Si dice che le connessioni inbound, da fuori verso dentro, vengono gestite con un application proxy fw, mentre le connessioni outbound, da dentro verso fuori, vengono gestite con un circuit-level proxy fw.

Questo perché in un verso non posso fidarmi completamente degli utenti esterni e quindi preferisco appesantire le operazioni del mio fw andando a ispezionare il traffico. Nel secondo caso do per scontato che i miei utenti siano in buona fede e metto sempre un fw ma non effettuo l’ispezione. Si usa in genere questo schema asimmetrico e si assume che gli utenti interni siano trusted, fidati.

Il circuit-level gateway suddivide la connessione TCP verso un host remoto:

* Uno con host interno
* Uno con host esterno
* Nessuna ispezione dei pacchetti

È un sistema autonomo o può essere una funzione specializzata eseguita da un gateway a livello di applicazione per determinate applicazioni. Noto anche come proxy a livello di circuito. L'amministratore si fida degli utenti interni. Può essere combinato con gw a livello di applicazione per il traffico in entrata.

**Firewall e DMZ**

In alcune reti esiste un’area particolare della rete in cui vengono messi a diposizione dei servizi, chiamata zona demilitarizzata, DMZ.

Firewall esterno: protegge l'accesso a Internet

Firewall interno: protegge il nucleo della rete aziendale

Zona demilitarizzata (DMZ):

* Sistema accessibile da Internet
* Ha bisogno di protezione

Immagine che contiene schermata, testo, diagramma, design

Descrizione generata automaticamente

Questa figura rappresenta un classico schema operativo. Avrò un router esterno, un fw perimetrale, che dal punto di vista logico è separato dal router, ma dal punto di vista fisico potrebbe essere collocato con un router, solo che le funzioni sono diverse quindi lo disegniamo diversamente.

Abbiamo poi una prima zona a cui accedo direttamente dopo aver superato il fw perimetrale, quello esterno, che si chiama zona DMZ. Nella zona DMZ ci metto tutti i servizi che voglio che siano accessibili agli utenti esterni. Voglio separare questi servizi dalla parte interna della rete perché la parte interna della rete è la parte più importante della rete che sono interessato a proteggere maggiormente. Per far accedere questi utenti alla rete metto in opera un ulteriore fw che sarà il fw interno, che può essere collocato con un router per connettere questi utenti interno o con uno o più switch.

La discriminante forte è che quelli che stanno nella DMZ sono servizi aperti al pubblico e come tali sono ospitati su server, con la declinazione di server che è colui che aspetta una richiesta di servizio per poi evaderla. Questi server devono poter accettare richieste di servizio che vengono dall’esterno; quindi, il livello di protezione che devono avere queste macchine deve essere necessariamente più blando, perché non è possibile sul fw esterno mettere ad esempio una regola che fa passare solo il traffico che viene da dentro, visto che in questo modo taglierei fuori tutto il traffico che arriva verso i server della DMZ.

La DMZ è un’area protetta, ma non troppo; c’è sempre un fw che protegge la DMZ, ma inteso a limitare tentativi di accesso a servizi che non sono compatibili con quelli che sono della DMZ. Gli stessi tentativi di accesso sulle stesse porte permesse nella DMZ ma su delle workstation che stanno nella parte interna della rete saranno bloccati dal fw interno.

Potrei anche mettere delle regole per cui il fw esterno fa passare i tentativi di accesso solo verso i server e non verso tutti gli altri indirizzi ip. È vero, ma in questo modo eliminerei la barriera del fw interno tra la parte interna della rete e la DMZ. Questo significa che se per caso, attraverso una richiesta di servizio, uno dei server della zona demilitarizzata viene compromesso automaticamente è compromessa tutta la rete, perché mi troverei ad avere un server compromesso all’interno della mia rete, dove cioè il fw non può più fare nulla. Invece, attraverso una segmentazione della rete in più zone, posso far in modo che il fw interno mi protegga anche se quei server vengono compromessi. Alcune aziende, tra l’altro, impediscono agli utenti interni di accedere ai server della DMZ, a seconda delle politiche aziendali questa è una configurazione abbastanza comune. Oppure mi permettono di accedere su porte e indirizzi diversi, non su quelli classici, in modo da limitare il potenziale coinvolgimento di questi nodi in un’azione malevola verso i server della DMZ. Alla fine i server della DMZ possono essere scarificabili, le workstation che stanno all’interno della zona protetta o i dati che stanno all’interno dei nodi della zona protetta tipicamente no.

In generale è possibile mettere in opera un certo numero di fw, a seconda delle caratteristiche interne della rete. Quindi posso avere non solo un fw perimetrale, ma anche più fw intesi a proteggere in maniera diversa diverse sezioni della rete.

All’interno di diverse reti, determinati punti di accesso alla rete posso avere configurazioni diverse, in particolare si tende a rendere più restrittivo l’accesso tramite le connessioni wireless.

Spesso per isolare la rete interna rispetto alla DMZ, i server della DMZ accedono su una diversa porta del fw, che può essere una porta fisica o una porta virtuale. Quindi di solito la DMZ è separata dal resto della rete tramite ad esempio una virtual lan, oltre che da uno o più fw.

Anche il fw esterno serve perché bisogna comunque limitare il più possibile i tentativi di attacco anche ai server della DMZ.

Compiti principali per il FW interno:

* Capacità di filtraggio più rigorosa per proteggere server e workstation aziendali
* Protezione bidirezionale rispetto alla DMZ:
  + Protegge il resto della rete dagli attacchi lanciati dai sistemi DMZ. Tali attacchi potrebbero provenire da worm, rootkit, bot o altri malware depositati in un sistema DMZ
  + Può proteggere i sistemi DMZ dagli attacchi provenienti dalla rete protetta interna
* È possibile utilizzare più firewall interni per proteggere parti della rete interna le une dalle altre. Una pratica comune è posizionare la DMZ su un'interfaccia di rete diversa sul firewall esterno.

**Sistema di rilevamento delle intrusioni**

Nomenclatura per IDS.

**Intrusione**: violazione delle politiche di sicurezza, solitamente caratterizzate da tentativi di compromettere la CIA (cioè uno dei 5 obiettivi: confidenzialità, integrità, disponibilità, accountability, autenticità) di un sistema. Queste violazioni possono provenire da:

* attaccanti che accedono ai sistemi dall’esterno, cioè da Internet o da una porzione esterna della rete
* da utenti autorizzati dei sistemi che tentano di bypassare i loro legittimi livelli di autorizzazione
* da utenti autorizzati dei sistemi che utilizzano il loro legittimo accesso al sistema per condurre attività non autorizzate
* Questo implica che gli attacchi non riguardano solo iniezioni di pacchetti, ma riguardano anche tentativi di accesso di utenti che hanno già guadagnato l’accesso alla rete. Quindi IDS non si preoccuperà di guardare solo al traffico in rete, ma spesso andrà a combinare l’osservazione del traffico con l’osservazione dei log (di un sistema, di un servizio, ecc). Andando a combinare questi due tipi di informazioni cercherà di rilevare le intrusioni.

**Rilevazione delle intrusioni**: il processo di raccolta di informazioni sugli eventi della "rete" e di analisi per individuare segnali di intrusioni.

**IDS**: (Intrusion Detection System) prodotti hardware o software che eseguono il processo di rilevamento delle intrusioni per generare avvisi adeguati.

Tipi di IDS:

* **IDS basati su host**: guardano solo il traffico in ingresso e in uscita dall’host. Un host-based IDS ha un compito più semplice di un networks-based IDS perché il suo vantage point è perfetto visto che sta esattamente sul luogo che bisogna osservare.

Le violazioni sono facili da verificare (vantage point).

* **IDS basati sulla rete**: i punti di vantaggio sono molti e quindi è più complicato rilevare le intrusioni sulla rete. Quanti IDS metto? Dipende da caso a caso, a volte ne basta uno, altre sono necessari più di un IDS.

Un IDS rientra nella classificazione che abbiamo visto sull’analisi del traffico in rete: abbiamo un certo numero di sensori, i collettori, un sistema per fare le query e una stazione di monitoraggio.

Di fatto, la struttura dell’IDS è la stessa, comprende 3 componenti logici:

* **Sensori**: più ne ho e più informazioni avrò, più sarà il carico computazionale dell’IDS.
  + I sensori sono responsabili della raccolta dei dati. L'input per un sensore può essere qualsiasi parte di un sistema che potrebbe contenere prove di un'intrusione.
  + I tipi di input a un sensore includono pacchetti di rete, file di registro e tracce di chiamate di sistema.
  + I sensori raccolgono e trasmettono queste informazioni all'analizzatore.
* **Analizzatori**: stazione di analisi che mette insieme la parte di memorizzazione delle informazioni (grezze o pre-elaborate, dipende dai singoli casi) e le analisi che posso fare su di esse (il sistema delle query)
  + Gli analizzatori ricevono input da uno o più sensori o da altri analizzatori. L'analizzatore ha il compito di determinare se si è verificata un'intrusione.
  + L'output di questo componente indica che si è verificata un'intrusione. L'output può includere prove a sostegno della conclusione che si è verificata un'intrusione.
  + L'analizzatore può fornire indicazioni sulle azioni da intraprendere a seguito dell'intrusione.
* **Interfaccia utente**: può essere un sistema di messaggistica che inoltra l’occorrenza di eventi più o meno sospetti a un amministratore di rete oppure può visualizzare i vari eventi che via via vengono analizzati dall’IDS
  + L'interfaccia utente di un IDS consente a un utente di visualizzare l'output del sistema o di controllare il comportamento del sistema.
  + In alcuni sistemi, l'interfaccia utente può essere un componente manager, direttore o console.

**Perché usare un IDS**

La linea di difesa di base di una rete contro le intrusioni è costituita da un sistema di autenticazione, accoppiato con un controllo dell’accesso e fw.

L'IDS (sistema passivo) / IPS (sistema attivo) è un qualcosa di aggiuntivo:

* Se ho un IDS e questo funziona bene permette di notificare un tentativo di accesso malevolo appena questo si verifica e quindi di limitarne gli effetti dannosi.
* La presenza di un buon IDS può essere un deterrente. In alcuni casi la presenza di un IDS può essere vista anche come una sfida, e su un IDS non è particolarmente performante può rappresentare un’attrazione per un certo numero di soggetti che vogliono sperimentare le proprie attività interne di cybersecurity. Quindi, se devo mantenere un IDS devo essere sicuro che sia performante, cioè che rappresenti veramente un sistema di difesa. In alcuni casi perciò è bene che la presenza di un IDS non sia pubblicizzata perché, a differenza del fw, l’IDS può avere un carico computazionale significativo, quindi non necessariamente l’IDS è in linea con il traffico di data plane. Vedi spiegazione sotto allo schema successivo.
* Consente la raccolta di informazioni sulle strategie di intrusione utili a rafforzare l'IPS (Intrusion Prevention System)

Un possibile IPS potrebbe essere costruito come la combinazione di un IDS e di un fw che viene configurato da una terza parte che riceve in input gli allert dell’IDS e sulla base di questi va a configurare di conseguenza il fw. Il fw sarebbe di tipo statico, però è configurato da una terza parte in maniera automatica.

**Dove posizionare un IDS**

ID significa guardare i pacchetti mentre passano davanti a qualche sensore.

I pacchetti interessanti corrispondono a una firma.

Tre tipi principali:

* firme di stringa
* firme di porta
* firme delle condizioni dell'intestazione, cioè condizioni anomale

Immagine che contiene testo, diagramma, schermata, Piano

Descrizione generata automaticamente

I NIDS (Network IDS) sono piazzati sempre lateralmente rispetto al collegamento che unisce le workstation con i server e con i fw. Il fw quindi sta sempre sul data plane, ovvero il traffico deve attraversare il fw. Nel caso di IDS questi sono sempre posti lateralmente.

Ad esempio, il NIDS numero 1 serve a capire cosa succedere prima del fw, quindi viene piazzato o con il fw, quindi viene collocato fisicamente con il fw anche se è un processo logico diverso, ad esempio analizza tutto il traffico che arriva sull’interfaccia esterna del fw, senza alterarlo, lo osserva e tira le conclusioni, solo per vedere cosa succede e che non vedo da dietro al fw. Nel momento in cui il fw blocca il tentativo di accesso io non lo vedo più e quindi per vederlo o analizzo il traffico o analizzo i log del firewall. Analizzare i log del fw è equivalente a piazzare un IDS “davanti” al firewall.

Oppure posso realizzare l’IDS piazzando uno switch davanti al fw e applico una tecnica di port mirroring, quindi ridireziono tutto il traffico che passa su quello switch verso un terzo che avrà il compito di analizzarlo e generare delle statistiche. Questo comporta l’acquisto di hardware aggiuntivo. La presenza dell’IDS è del tutto trasparente al traffico.

Posso anche piazzare un IDS a valle del firewall: in questo caso ho la possibilità di ispezionare tutto quello che passa e quindi capire se il fw è configurato bene oppure fa ancora passare del traffico “sospetto”, oppure se dentro la rete c’è qualcuno che effettua un’attività sospetta. In questo caso potrei attivare delle azioni, tipo isolare quell’utente o configurare il fw in modo tale da limitare le attività sospette.

Posso avere un IDS dietro il fw in zone critiche della rete. Ad esempio potrei metterlo ad osservare il traffico che è diretto verso la DMZ (nel disegno la service network è di fatto la DMZ) oppure posso metterlo ad osservare il traffico che interessa dei server particolarmente importanti, interni alla rete (NIDS numero 3) o ad osservare quello che combinano gli utenti della rete (NIDS numero 4).

NIDS numero 2, per come è pensato, non vedrà mail traffico che va dalle workstation in basso ai server che stanno in alto. Questo ha senso in una rete di una certa dimensione, non ha senso in una rete di 10 calcolatori e un nodo di accesso, in quel caso se voglio mettere un IDS devo collocarlo sul nodo di accesso.

Principio di funzionamento dell’IDS: analizza il traffico che attraversa la rete, o in particolare che attraversa i link su cui è piazzato. Nel caso di IDS piazzato lateralmente il traffico viene reinoltrato, ad esempio, da uno switch con una porta configurata tramite il mirroring. Tipicamente analizzando il traffico tenta di identificare quella porzione di traffico o quei pacchetti, a seconda di come è configurato, che hanno delle particolari caratteristiche. Queste caratteristiche possono essere orientate a identificare particolari comandi, quindi all’interno di host che posso rilevare attraverso i log del sistema, a identificare particolari combinazioni indirizzi ip-porte (pacchetti con indirizzi ip che non appartengono alla mia rete e che sono originati da dentro la mia rete), oppure combinazioni anomale all’interno dell’intestazione dei protocolli. Una tipica combinazione anomala è quella in cui in uno stesso pacchetto tcp ho impostato a 1 il flag di SYN (inizio della connessione) e il flag di FYN o di RESET (fine della connessione o reset della connessione): è ovvio che non ha senso perché la connessione o la apro o la chiudo, non posso farlo contemporaneamente e quindi è chiaro che è un pacchetto che risulta essere “sospetto” e quindi sarebbe bene generare un allarme per aver rilevato questo tipo di pacchetto. Potrei avere un’applicazione che funziona male, quindi un errore di programmazione, oppure potrei avere un attacco, un bug o qualsiasi cosa.

FINE 25/10

31/10

**Recap:**

Eravamo arrivati a discutere dei possibili piazzamenti delle sonde che agiscono come sistema di rilevamento delle intrusioni. Ci sono 3 principali sistemi IDS:

* quelli che guardano al contenuto del traffico, quindi del payload dei pacchetti e vanno a cercare quelle che prendono il nome di **string signatures**. Ad esempio, se abbiamo accesso al treno dei pacchetti HTTP, eventuali contenuti caricati tramite il messaggio di post.
* possiamo avere delle signature basate sulle port (**port signatures**). Ad esempio, una particolare combinazione di indirizzi IP, mittente e destinazione, porta di ingresso e di destinazione, e magari anche il volume dei tentativi di accesso ad un determinato servizio.
* Possiamo avere delle signature basate su delle combinazioni particolari nelle intestazioni dei pacchetti (**header condition signatures**). È sicuramente non un tentativo di accesso normale un pacchetto TCP che porta contemporaneamente il flag SYN e il flag FYN nella stessa intestazione. Il pacchetto serve ad aprire e contemporaneamente a chiudere una connessione, quindi è un pacchetto senza senso, e o è generato da un’applicazione che ha un bug, oppure può essere il sintomo di un tentativo di attacco che magari va a utilizzare una vulnerabilità.

**Approcci per il rilevamento delle intrusioni**

I sistemi di rilevamenti delle intrusioni hanno come obiettivo quello di rilevare i tentativi di intrusione, non di bloccarli.

Possiamo dire che esistono due macro approcci per la rivelazione delle intrusioni.

Nessuno dei due è strettamente migliore dell’altro, la linea che separa i due comportamenti è spesso labile. Hanno usi diversi e spesso sono utilizzati in maniera combinata.

L'assunzione fondamentale è che il comportamento di un utente legittimo è tipicamente diverso dal comportamento di un utente che tenta di fare un attacco o tenta di intrufolarsi in un sistema nel quale non ha il diritto di entrare.

Nel caso in cui questa assunzione venga rispettata, è di fatto impossibile discriminare un utente malevolo da un utente benigno. Quindi se un utente malevolo è un utente autorizzato che è armato solo di cattive intenzioni, si vedrò quello che succede. Quindi, ad esempio, utilizzando solo una signature basata sulle porte (indirizzo mittente e destinazione, porta mittente e destinazione) o basata sulle intestazioni non saremmo mai in grado di rilevarlo. Poi dipenderà anche dai comandi che andrà a dare dal sistema, sempre che sia in grado di dare dei comandi.

Un’altra cosa fondamentale è che i sistemi di rete, sia i servizi sia le reti che connettono questi servizi a internet sono in generale continuamente sotto attacco, solo che la stragrande maggioranza di questi attacchi non rappresentano minacce, e quindi non hanno un uso pericoloso.

Questo rende il sistema di rilevamento delle intrusioni, un sistema particolarmente critico, perché genererà le contromisure agli attacchi. Alcuni di questi allarmi avranno una rilevanza diversa dagli altri, e le capacità di chi analizza questi allarmi, sia esso un sistema informatico, un utente umano, un router, è proprio andare a determinare tra gli attacchi che non sono chissà quanto pericolosi da quelli che non sono pericolosi affatto.

È fondamentale che il comportamento di chi effettua un attacco sia diverso da quello di un utente legittimo.

I due macro approcci sono i seguenti: nomi simili ma sono il contrario uno dell’altro.

* **Rilevamento basato sull’uso anomalo: misuse detection**

Utilizziamo un database di signature quindi di impronte di attacchi noti, e andiamo a classificare il traffico sulla base della differenza del traffico che abbiamo ad osservare con queste impronte. Se c'è una corrispondenza, non solo noi vediamo un attacco, ma sappiamo anche che attacco è. E quindi possiamo prendere la contromisura necessaria, sempre se l'attacco è rilevante e necessita di una contromisura.

Lo scopo della misuse detection è quello di rilevare attacchi noti, facendo uso di database di impronte di attacchi.

* Utilizza algoritmi di tipo pattern-matching: ho un certo tipo di pattern, tutti i tipi di traffico che matchano con quel pattern saranno classificati come quel tipo di attacco. Il traffico che non matcha nessun tipo di attacco, di conseguenza è il traffico legittimo.
  + Migliori prestazioni in termini di falsi positivi. I falsi positivi è il traffico legittimo che viene classificato come attacco.

Proprio perché io ho delle impronte ben precise che sono stati rilevate da attacchi passati e quindi, di solito, questi database vengono aggiornati con regolarità e sono disponibili anche diversi tipi di sistemi IDS.

È difficile che il traffico legittimo possa essere classificato come un attacco, e quindi la % di falsi positivi è molto bassa.

* + L'altra cosa fondamentale è che proprio perché ho un database di impronte di attacchi, di solito questi software riescono a sfruttare in maniera molto efficiente le capacità di calcolo parallelo e quindi sono efficienti in generale dal punto di vista computazionale, sono veloci.

Esempio classico di IDS che utilizza il misuse detection è suricata. È un software open source che fa affidamento su un database che qualcuno ha costantemente aggiornato di impronte di attacchi.

Non riesce a rilevare tutti gli attacchi che non sono nel database delle impronte (gli attacchi nuovi o semplicemente attacchi al di fuori del database, e che siano significativamente diversi dagli attacchi presenti nel database), per questo sono stati definiti gli IDS basati sull’anomaly detection.

* **Rilevamento della anomalia: anomaly detection**
  + Basato sulla ricerca di attività anormale; mentre il misuse detection è allenato sulle impronte di attacchi, quindi sul traffico al normale, questo invece è allenato sul traffico anormale. E quindi tutto ciò che devia da questa pseudonormalità, viene classificato come un attacco.
  + Prestazioni abbastanza bilanciate di falsi positivi – falsi negativi.

I falsi negativi sono il grosso problema del misuse detection. Un falso negativo è un attacco che erroneamente viene classificato come traffico legittimo. Questo succede nel misuse detection quando ho un attacco che non ho mai visto, cioè non è tra quelli riportati nel database.

Nell’anomaly detection posso avere sia i falsi positivi che i falsi negativi.

Ho i falsi positivi perché analizzo il traffico legittimo, però ci possono essere utenti che hanno tutto il diritto di scambiare un certo tipo di traffico che per la loro particolare attività potrebbero essere erroneamente classificati come attaccanti; il loro traffico potrebbe essere classificato come un attacco.

* + Utile per rilevare attacchi 0-day, gli attacchi nuovi, quelli mai visti.

Quindi se ho un meccanismo basato su anomaly detection ci potrebbe essere una buona probabilità che riesco a identificare un tentativo di attacco.

**Differenza** enorme tra i due approcci: nel primo caso rilevo gli attacchi e riesco anche a identificare il tipo di attacco, nel secondo caso posso fare una discriminazione solo binaria, quindi traffico legittimo o anomalia=attacco. Nel primo caso, se un attacco matcha più profili di allarme ci sarà uno score e l’impronta che avrà uno score più elevato sarà quella identificata come il mio attacco.

Se conosco l’attacco sono anche in grado di mettere in piedi una contromisura, se non conosco l’attacco posso solo generare un allarme, poi sarà l'amministratore di sistema ad agire di conseguenza e a cercare di bloccare qualcosa o mettere temporaneamente offline il servizio o la rete, a seconda di quanto è critica l’infrastruttura di cui stiamo parlando.

Dal punto di vista grafico possiamo vederli in questo modo:

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Io mi alleno sul cerchietto, il colore più scuro è il traffico normale, quello più chiaro sono gli attacchi.

Nel caso di misuse detection io conosco gli attacchi, sono quelli che vado a definire e sui cui mi alleno; tutto quello che non è attacco è traffico legittimo e passa. Quindi vado a cercare gli attacchi.

Nel caso di anomaly detection definisco il traffico normale su cui mi alleno, tutto ciò che non è normale lo definisco come attacco, quindi o viene bloccato o genera un allarme. Ricordiamoci che sono schemi IDS, quindi non sono sistemi di prevenzione.

Poiché i bordi non sono mai netti la decisione è critica (le cose non sono sempre facili!) e quindi entra in campo una branca dell’intelligenza artificiale che è il machine learning. È necessario un approccio probabilistico -> il machine learning è uno strumento adatto

I bordi non sono mai netti significa che in generale noi non abbiamo mai una distinzione netta tra quello che è un attacco e quello che è traffico legittimo. Perché ci sono dei casi in cui del traffico legittimo può sembrare un attacco, ma rimane traffico legittimo e invece un attacco può sfuggire all'IDS perché sembra a tutti gli effetti del traffico legittimo. Quindi c'è sempre una certa sovrapposizione tra il comportamento degli intrusi e il comportamento degli utenti legittimi.

Immagine che contiene testo, diagramma, schermata, Diagramma

Descrizione generata automaticamente

In questo grafico in ascissa consideriamo un parametro misurabile che va a caratterizzare il comportamento degli utenti, quello nel gergo del ML prende il nome di feature, quindi questa sarà una delle features. Supponiamo per semplicità grafica che il nostro sistema possa essere descritto da una feature sola.

In ordinata ho una pdf (probability density function), quindi la densità di probabilità.

Queste due campane rappresentano la probabilità condizionata.

La campana di destra rappresenta la probabilità di questo parametro condizionata al fatto che questo parametro è generato dal traffico di un utente legittimo. La feature sarà ad esempio uno dei metadati che vado a catturare con netflow, o una statistica che vado a catturare attraverso una sonda che misura la latenza, o il numero di byte scambiati su una connessione.

La campana di sinistra invece rappresenta la pdf condizionata al fatto che i valori del parametro misurato sono stati generati dal traffico di un utente che tenta di entrare in maniera illegittima sul nostro sistema.

Le cose sarebbero semplici se le due pdf fossero completamente separate. Questo perché la sovrapposizione rappresenterà proprio la causa del nostro errore. Errore: ci sono i falsi positivi e i falsi negativi. Nessun sistema decisionale è in generale esente dagli errori. Qui possiamo avere due tipi di errori.

Se le campane fossero separate per discriminare tra utente legittimo e attacco mi basterebbe piazzarmi in un punto qualsiasi e tirare una linea che rappresenterebbe la mia soglia decisionale.

Immagine che contiene linea, diagramma, Diagramma, design

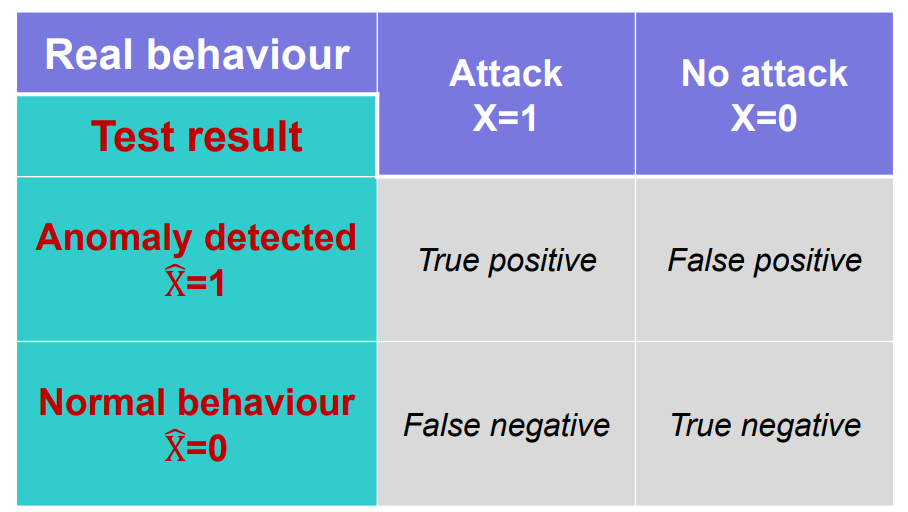
Descrizione generata automaticamente

Invece noi abbiamo questa zona di sovrapposizione. Piazziamo una soglia sull’intersezione delle due campane. Tutta la porzione di traffico legittimo che avrà valori al di sotto della soglia (quella colorata in blu) saranno falsi positivi dal punto di vista della rilevazione dell’attacco, cioè se non ho un attacco non lo sono, sono la coda inferiore della campana di destra. Mentre tutti quei casi in cui il nostro parametro ha un valore più alto della soglia e quindi ricade per definizione nel traffico legittimo ma rappresenta invece la coda superiore della campana di destra parleremo di falsi negativi.

È sempre giusto piazzare la soglia sulla prima intersezione? Dipende dall’importanza relativa del falso positivo e del falso negativo. In genere si fanno sempre esempi di natura medica: tumore. Se ho un falso positivo quello che faccio tipicamente è effettuare nuove analisi per essere sicuro che io ho il tumore, mentre se avessi un falso negativo, io ho il tumore e al primo screening sfugge, poi il tumore cresce e potrebbe portare alla morte.

Quindi nei casi in cui i falsi positivi sono più importanti dei falsi negativi o viceversa, la soglia andrà spostata di conseguenza. Ci saranno delle funzioni di perdita che dovremo utilizzare e saranno quelle che determinano la posizione della soglia.

**Matrice di confusione**



Di solito andiamo a caratterizzare questi algoritmi sulla base della matrice di confusione. In questa matrice sulle colonne ho un comportamento reale e sulle righe il comportamento stimato.

La prima colonna rappresenta l’evento di attacco e la seconda colonna rappresenta l’evento di traffico legittimo.

La prima riga rappresenta l’evento di anomalia rilevata e la seconda riga rappresenta l’evento di anomalia non rilevata.

Positive è sempre relativo al concetto di rilevazione dell’attacco.

True positive: ho un attacco e l’ho rilevato come tale.

True negative: ho traffico legittimo e l’ho classificato come tale.

Gli errori sono sulla diagonale secondaria.

False positive: non ho un attacco e lo classifico come tale.

False negative: ho un attacco ma non lo classifico come tale.

Gli IDS sono algoritmi di classificazione, in grado di classificare o il tipo di attacco o se c’è o meno un attacco; utilizzati sempre e solo per classificare.

**Metriche delle prestazioni**

**Accuratezza**: misura statistica della capacità di un test di classificazione binaria di identificare o escludere correttamente una condizione; prossimità dei risultati della misurazione al valore reale.

Metrica principale ma non sempre utilizzabile. Ci dice quanto le nostre decisioni siano corrette rispetto all'insieme degli eventi. Ci dice di fatto quanti sono i veri negativi e i veri positivi rispetto a tutte le nostre decisioni. L’ottimo è quando ho la diagonale secondaria pari a 0.

Non si utilizza sempre perché non la uso in qui casi in cui ho dataset particolarmente sbilanciati. Soprattutto perché nel caso delle minacce, che noi sappiamo essere molto poche rispetto all’ insieme degli attacchi, quello che può succedere è che su migliaia di attacchi le minacce sono qualche decina o qualche unità. E quindi, essere in grado di andare a rilevare le minacce può essere un task particolarmente difficile, anche perché sono eventi rari. Quando sono eventi rari, il fatto di sempre che i TP siano tanti meno dei TN può fare in modo tale che io ho un'accuratezza modellata anche se i TP sono pari a zero.

**Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente**

Per questo si usano queste metriche secondarie, ma non di secondaria importanza, che sono la precisione e il richiamo.

**Precisione**: noto come valore predittivo positivo; riproducibilità della misura.

Ci dice quanto è buona la nostra stima dei positivi.

**Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente**

**Richiamo**: noto come sensibilità nella classificazione binaria diagnostica.

Ci dice quanti degli eventi di interesse (dei positivi) sono in grado di identificare. Vado quindi ad effettuare la valutazione per colonna: prendo come riferimento non la stima coma prima, ma l’evento reale, cioè la colonna attacco.

**Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente**

**F-score**: la media armonica (ponderata) della precisione e del richiamo.

L’F-score, in particolare questo è l’F1-score combina precisione e richiamo e penalizza molto gli estremi. È una versione bilanciata tra il richiamo e la precisione.

**Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente**

**Una vecchia storia: accuratezza vs precisione**

**Immagine che contiene cerchio, diagramma, testo, linea

Descrizione generata automaticamente**

Dal punto di vista pratico si usa spesso il formalismo del bersaglio.

Diremo che un meccanismo è tanto più preciso quanto i campioni tendono a essere raggruppati. In questo caso ho una buona precisione ma una scarsa accuratezza perché il valore medio è distante dal valore reale.

L’accuratezza ci dice quanto il valor medio della nostra stima è vicino al valore reale. In questo caso ho dei dati che sono distanti tra loro in cui il valor medio è tanto vicino al centro: ho una buona accuratezza e una scarsa precisione.

* Nell’ambito della sicurezza informatica, o in generale anche del network management, diremo che la cosa che più ci interessa è minimizzare i falsi negativi: voglio che tutte le anomalie siano identificate. Dobbiamo però stare attenti che questo non causi un aumento indiscriminato dei falsi positivi, cioè i falsi allarmi. Il falso allarme ha l’effetto di desensibilizzare il sistema perché se ho un numero esagerato di falsi allarmi la tendenza alla fine è quella di non considerare più nessuno. Questo è il modo migliore per tralasciare i veri attacchi. Avere un sistema tarato per identificare tutti i falsi negativi ma che genera un numero enorme di falsi allarmi è quindi un sistema che non funziona.

**Metriche delle prestazioni**

Altre metriche: AUC e ROC.

La ROC rappresenta un modo per identificare la sensibilità del nostro criterio decisionale sulla scelta della soglia. È una curva che ha per ascissa l’FPR o P(FP) (probabilità di falsi positivi).

FPR è definita come falsi allarmi / (falsi allarmi + correct rejections). Ci dice quanti errori vado a considerare e quanti errori commetto nell’identificare il traffico legittimo.

Al numeratore ho il recall, quindi quanti degli attacchi reali riesco ad identificare con il meccanismo di predizione.

Prendiamo quindi la tabella per colonna: la blu è la FPR e la rossa è la TPR.

Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente

La curva può essere interpretata in questo modo: se abbasso molto la mia soglia quello che ho è che cresceranno molto i falsi positivi, invece se metto una soglia troppo alta vado a minimizzare il numero di falsi positivi ma minimizzo anche il numero di attacchi veri che sono in grado di identificare.

Questa curva mi dice, a seconda della mia soglia, qual è la performance del mio algoritmo di classificazione.

Esiste un criterio chiamato AUC, area sotto la curva ROC, che è di fatto una quantificazione invariante rispetto alla scelta della soglia. Può essere definita come la probabilità che un campione positivo scelto a caso abbia uno score inferiore (cioè viene classificato erroneamente) rispetto a un campione negativo scelto a caso.

Immagine che contiene testo, linea, diagramma, Carattere

Descrizione generata automaticamente

La cosa peggiore è quindi quando l’area è il triangolo: quindi la probabilità di falsi positivi è uguale alla probabilità di falsi negativi (come a dire che scelgo lanciando una monetina).

Il comportamento ideale sarebbe questo:

Immagine che contiene linea, schermata, diagramma, bianco

Descrizione generata automaticamente

In questo caso avrò 0% di falsi positivi e 100% di veri positivi, quindi minimizzo gli errori.

Più la mia curva si sposta nella direzione indicata dalla freccia rossa e migliore è la prestazione dell’algoritmo, indipendentemente da come scelgo la soglia.

Immagine che contiene linea, schermata, diagramma, Diagramma

Descrizione generata automaticamente

Il problema di queste metriche è che sono interessanti perché prescindono da alcuni aspetti pratici della configurazione ma tendono, soprattutto l’AUC, a non distinguere più tra falsi positivi e falsi negativi. Quindi se è importante andare a discriminare tra i due allora non vanno bene queste metriche.

Metrica che si sta diffondendo negli ultimi anni:

**Spiegabilità**: il motivo di un risultato di un algoritmo di ML può essere più importante del risultato stesso. Questo perché spesso gli algoritmi di ml sono utilizzati come black box e questo è un trend che bisognerebbe evitare il più possibile. È importante capire perché un algoritmo funziona e perché in alcuni casi funziona male. Non sempre questo è possibile, soprattutto con le reti neurali, che sono sistemi molto molto complessi dal punto di vista matematico, fortemente non lineari e quindi è spesso difficile avere una vera spiegabilità dei risultati che forniscono. Tuttavia, soprattutto nei tool più semplici, è importante capire a priori perché un certo tool si presta meglio a determinate situazioni rispetto ad un altro.

**Rilevamento delle anomalie: supervised ML**

Il rilevamento delle anomalie può essere implementato con il **supervised learning** (SL) quando:

* ho un gran numero di campioni sia di traffico legittimo che di tipi di attacchi con cui addestrare il sistema -> avrò una classificazione multiclasse
* SL sarebbe particolarmente adatto per problemi in cui prevediamo che gli attacchi futuri siano simili agli attacchi passati, presenti nel training set

Può essere difficile trovare un pool rappresentativo di campioni positivi per addestrare bene l'algoritmo.

Le violazioni dei server sono talvolta causate da attacchi zero-day o da vulnerabilità appena rilasciate nel software: sfuggono a questo rilevamento. Però avere la possibilità di un set di attacchi noti è un vantaggio importante.

Per definizione, il metodo di intrusione non può essere previsto.

Spesso quando si mette in piedi un sistema di tipo IDS, soprattutto in ambito enterprise, si accoppiano sistemi basati sul misuso a sistemi basati sull’anomalia: uno cerca di classificare gli attacchi come attacchi noti e l’altro invece cerca solo di capire se c’è un attacco nuovo oppure no.

L’ordine in cui faccio girare gli algoritmi può dare vantaggi o svantaggi.

Ad esempio potrei utilizzare prima un algoritmo di tipo misuse detection e poi un algoritmo di tipo anomaly detection.

Nel primo caso il primo tipo di IDS andrà a classificare tutti gli attacchi noti e farà passare il traffico legittimo e gli attacchi che sfuggono alla classificazione di attacchi noti. Se poi applico a questo presunto traffico legittimo l’anomaly detection è come se l’applicassi a quello che reputo solo traffico legittimo. Se ancora rilevo delle anomalie, questo traffico anomalo sarà potenzialmente traffico dovuto ad attacchi che sfuggono alle impronte presenti del database dell’IDS che funziona con il misuse detection. Questo però non implica che in alcuni casi potrei avere del traffico classificato come attacco ma sbagliando il tipo di classificazione, è comunque un problema.

Se inverto la classificazione, quindi faccio prima anomaly detection e poi misuse detection, assumo che nell’anomaly detection raccolgo tutti gli attacchi e una volta radunato tutti gli attacchi tutto il traffico legittimo passa e tutto il traffico definito come attacco passa in un IDS che lavora con misuse detection. Tutto quello che il misuse detection classifica come attacco è un attacco e quello che classifica come traffico legittimo è ragionevolmente un attacco nuovo che non riesce ad identificare.

Di solito le prestazioni nei due casi sono simili. Altrimenti posso metterli in parallelo e andare a confrontare i risultati.

Problema principale: attacchi nuovi. Utilizzo il SL se è molto importante classificare gli attacchi noti per poi bloccarli il prima possibile e se insieme a questo utilizzo un meccanismo basato su anomaly detection per classificare i nuovi attacchi o se assumo che ci sia una comunità molto attiva che aggiorni in continuazione il database delle impronte degli attacchi e che quindi anche gli attacchi 0-day possano essere classificati il prima possibile. Spero di non essere tra le prime vittime dei nuovi attacchi e che non appena un attacco viene rilevato viene classificato e inserito nel db.

Eventi relativamente rari => classe di problemi di squilibrio non adatta a SL.

**Rilevamento delle anomalie: euristiche**

Le euristiche sono tecniche basate su soglie:

* Come impostare la soglia? La soglia deve essere impostata in modo da minimizzare i nostri errori (falso positivo e falso negativo).
* È uguale per tutti gli utenti? Una cosa fondamentale è che quando uso le euristiche non sempre faccio della analisi di tipo statistiche dei dati ma è importante essere più generali possibili, ad esempio andando a identificare dei casi notevoli. Se ho utenti che hanno permessi diversi io dovrei andare a impostare soglie diverse per utenti diversi.
* Con quale frequenza aggiornare la soglia? Avrò degli algoritmi pseudo deterministici che determinano il valore della soglia. Un aggiornamento continuo può portare instabilità e c’è sempre il problema di fare in modo che il sistema non diventi instabile.
* I metodi basati sulle soglie sono sempre efficaci? Dipende, di solito sono i metodi meno robusti. Anche gli attacchi con un numero basso di tentativi possono avere successo.

La soglia dovrebbe essere robusta:

* Soglia generata dinamicamente attraverso l’analisi dei dati
* Soglia diversa per utenti diversi
* Utilizzare le proprietà statistiche del dataset, non solo le medie: più robusti per gli outliers sono la mediana o i quartili. La media è quanto di più sensibile c’è agli outliers, cioè ai campioni diversi dalla maggior parte dei campioni.

Immagine che contiene schizzo, linea, design

Descrizione generata automaticamenteEsempio numerico:

Immagine che contiene testo, linea, Carattere, diagramma

Descrizione generata automaticamente

La linea rossa è la mediana. Ora se al posto del campione 22 avessimo 220 la media cambierà in maniera significativa mentre la mediana non sarà minimamente interessata dal cambiamento. Lo stesso con i quartili (disegno).

* Anomalie innescate da molteplici violazioni della soglia

**Rilevamento delle anomalie: data driven**

Tutte le volte che effettuo scelte che sono data driven, e quindi che utilizzo meccanismi di machine learning e non di semplici euristiche dovrei cercare di soddisfare questi obiettivi:

* Mantenere gli errori, FP e FN, il più bassi possibile
  + Per ottimizzare le prestazioni del sistema
* I parametri del sistema IDS (diversi dalle features; i parametri sono le soglie, i quartili, ecc) dovrebbero essere facili da configurare, ottimizzare e mantenere
  + Se l’amministratore di sistema non capisce bene come utilizzare lo strumento lascia le impostazioni sempre ai valori di default e come va va
* I meccanismi di IDS si adattano ai trend dei dati
  + Trend dovuti a periodicità o stagionalità (esempio picchi di traffico che possono essere rilevati come anomali o meno a seconda di quando si verificano)
* Gli approcci utilizzati per IDS funzionano bene con dataset di diversa natura, dovrebbero essere il più generale possibili e non fare assunzioni matematiche che non sempre sono realistiche.
  + Classica assunzione: campioni distribuiti come una gaussiana singola per il singolo parametro o multi-variabile per più parametri. Questo non necessariamente è vero e avere un approccio che prescinda da questa assunzione è in generale robusto.
* Sistema efficiente in termini di risorse, adatto per il tempo reale
  + Computazionalmente efficiente
* Meccanismo spiegabile
  + Perché quell’allarme è stato lanciato o no!

Ingegneria delle features:

Quando utilizzo tecniche di machine learning o di intelligenza artificiale è fondamentale la scelta delle features da utilizzare perché ci sono alcune features che sono combinate con le altre e spesso alcune delle feature sono combinazioni lineari di altre features. Il valore aggiunto di queste feature è pari a 0 però peggiorano molto l’efficienza computazionale.

Altra cosa fondamentale: i sensori che acquisiscono dati di traffico tipicamente generano dati come se fossero serie temporali. Queste serie temporali sono “immutabili”, ovvero non vengono mai aggiornate perché i dati di traffico sono quelli, non è che cambiano nel tempo. Quindi un record di una funzione di una sessione molto lunga una volta scritto rimane tale. La dimensione temporale di questi dati in alcuni casi può essere non correttamente presa in considerazione dagli algoritmi di intelligenza artificiale e questo è un problema. Questo accade perché non tutti gli algoritmi di intelligenza artificiale utilizzano la dimensione temporale all’interno dei propri meccanismi. Soluzione: invece di dare in pasto al motore di machine learning tutti i dati insieme, posso utilizzare il concetto di località temporale, ovvero utilizzo delle finestre temporali e tengo conto del risultato finestra per finestra.

Diamo ai motori di machine learning:

* Metadati del traffico di rete: quelli che ad esempio estrae netflow o altre features che derivano dalla misura delle prestazioni (numero di pacchetti persi, latenza nel servire una richiesta di servizio). Utilizzare stateful inspection quando possibile.
* In alcuni casi, soprattutto quando ho a che fare non con il traffico vero e proprio, ma con i log di un server che eroga un servizio potrei avere a disposizione o i log o la possibilità di andare ad analizzare il traffico. Se lo faccio sul server che eroga un servizio non mi crea problemi, ma se lo voglio fare su un nodo a monte ho bisogno di tecniche che prendono il nome di deep packet inspections e che hanno bisogni di decifrare il traffico.

Questo implica che se ho una sessione SSL questa assicura sicurezza da estremo a estremo, cioè tra il client e il server. Per analizzare il payload dei dati nell’IDS come faccio? Posso analizzare solo quello che è esterno a SSL, quindi intestazione di strato 3 e 4 ma non posso osservare niente. Devo quindi fare in modo che quando dirotto il traffico sia in grado di terminare la sessione SSL in modo tale da estrarre il traffico e analizzarlo. Questo può però rappresentare una vulnerabilità stessa del sistema.

Immagine che contiene linea, diagramma, cerchio, schermata

Descrizione generata automaticamente

(Quando abbiamo fatto i fw non avevamo visto i proxy level fw in cui la sessione tcp veniva terminata sul proxy e veniva effettuata l’ispezione dei pacchetti, in quel caso dovrei scrivere le funzioni di IDS insieme alle funzioni di fw perché ho il pacchetto in chiaro visto che è terminata la sessione tcp e anche la sessione SSL sul fw e quindi in quel caso è possibile ispezionare il payload del pacchetto. Altrimenti devo far girare l’IDS sul server o sul nodo terzo che però può comportare un pericolo di compromissione del servizio stesso.)

Tipi di attacchi più noti:

* Denial-of-service (dos)
* Network probe (probe): ad esempio quello effettuato da nmap per capire quali porte del sistema sono aperte o sui cui determinati host rispondono
* Remote-to-local (r2l): un utente remoto tenta di entrare dentro il sistema
* User-to-root (u2r): una volta che l’utente è entrato con credenziali di utente semplice tenta di elevare i suoi permessi a livello di root

Attacchi che combinati insieme prendono il nome di escalation: Probe => r2l => u2r: è un pattern noto. È una delle caratteristiche fondamentali di un dataset storico che viene utilizzato per insegnare la sicurezza informatica nelle reti e prende il nome di ?kdb? (lo utilizzeremo nelle prossime esercitazioni).

Features ben note:

* durata della sessione (diversa dalla dimensione temporale perché la dimensione temporale ha a che fare con l’istante di inizio della sessione più che con la durata della sessione; la dimensione temporale non sempre viene utilizzata ma è di particolare importanza)
* tipo di protocollo (tcp, udp, altri protocolli di strato 4)
* il servizio (il campo type of service del protocollo ip o la porta di destinazione dello strato 4, 80 per http, 443 per https, 53 per dns e così via)
* flag (normale o errore)
* src/dst bytes (quantità di byte scambiati in un verso o nell’altro, quindi dal client al server e viceversa)
* land (pacchetti marchiati come oggetti nell’intestazione del tcp)
* frammento sbagliato (numero di frammenti che arrivano sbagliati)
* pacchetto urgente

Features aggiuntive, relative all’accesso di servizio:

* tassi di errore relativi al servizio: ad esempio il numero di sessioni che non riesco a servire da parte del mio server o dei miei server
* % di errori di pacchetti SYN rispetto ai pacchetti REJ (rigettati): quando non riesco a stabilire la sessione tcp
* % di connessioni agli stessi servizi o a diversi servizi

Riduzione delle feature per la pulizia iniziale del dataset:

* Features engineering: rimuovere le feature strettamente legate l’una all’altre in modo da non inondare l’algoritmo con dati utili che non portano però informazione
* Rimuovere le ridondanze: potrei acquisire informazioni da più sensori; se due sensori diversi mi danno la stessa informazione quell’informazione la porto una volta sola, non n volte.

La riduzione dovrebbe comportare una perdita di prestazioni minima o nulla, non devo perdere informazioni perché sennò perdo anche nelle prestazioni del sistema.

Uno degli approcci che possono essere utilizzati in particolare per andare a identificare gli attacchi DDoS o DoS sono le tecniche che vanno sotto il nome di **forecasting**, previsioni, che tipicamente sono gestite con tecniche di supervised learning (SL).

Tipicamente sono serie temporali di una o più features in cui la cosa fondamentale è l’istante temporale in cui la feature viene raccolta. Devono permettere di raccogliere i trends (sul lungo periodo), le stagioni (ripetizioni periodiche, stagionalità) e altri cicli (periodicità secondarie diverse dalla stagionalità).

Ad esempio la stagionalità può essere quella di periodo di lavoro/vacanze e all’interno di questa ci sono dei cicli tipo giorno/notte e giorni lavorativi/finesettimana.

Immagine che contiene linea, Diagramma, Carattere, diagramma

Descrizione generata automaticamente

Modelli che posso utilizzare:

* Modelli autoregressivi (ad esempio ARIMA)

ARIMA sono modelli semplici in cui:

Immagine che contiene calligrafia, Carattere, testo, bianco

Descrizione generata automaticamente

y è in genere funzione lineare, x è l’ingresso.

Si parla di modello ARIMA, modello auto regressivo integrale a media nobile, quando la funzione di trasferimento ottenuta risulta avere dei poli con delle radici pari a 1 o maggiori di 1. Tipicamente quindi non è un modello stazionario proprio perché deve inserire fenomeni di tipo temporale.

* Reti neurali, in particolare le reti neurali ricorrenti, quelle che tengono memoria (LSTM, Long Short Term Memory, sono le più utilizzate).

Proprio perché se includo la dimensione temporale al dataset e so che nel profilo di grafico di richieste di servizio ho dei pattern che tendono a ripetersi, tra cui possono emergere anche dei trend, allora ho bisogno di strumenti che siano caratterizzati dalla possibilità di tenere memoria.

LSTM hanno una memoria sia a breve che a lungo termine e sono in grado di ricordarsi profili diversi.

Quando si utilizza la previsione:

* Metrica a valore singolo nel tempo: quando ho un solo parametro (ad esempio il numero di richieste o il tempo di servizio) questi modelli permettono di stimare il valore futuro del parametro nel tempo e quindi di prevenire situazioni di anomalie. Quando ho più di un parametro potrei avere problemi a trovare un criterio decisionale efficace.
* Non adatto per il rilevamento di valori anomali. È utile per stimare le anomalie solo se il set di training NON contiene valori anomali, altrimenti, si adatterà agli inlier e agli outliers. A quel punto se ho un comportamento diverso dal trend e/o dalla stagionalità o dai cicli noti sarò in grado di rilevare le anomalie. Se invece il dataset ha le anomalie questi modelli integreranno le anomalie all’interno del modello stesso e quindi potenziali attacchi potrebbero essere scambiati per fenomeni attesi, non riuscendo a identificarli.
* Adatto solo quando emergono trend osservabili, con fluttuazione limitata. Anomalie osservate con deviazione RMS (deviazione standard: radice della media dei quadrati)

Immagine che contiene Diagramma, linea, diagramma, schermata

Descrizione generata automaticamente

In questo esempio si vede qual è il valore osservato, quello blu, piuttosto stazionario, e qual è il valore predetto, quello verde. Nei punti in cui ho valori che sono significamente diversi da quelli che ho normalmente la deviazione standard ha un balzo, che è uno scostamento elevato e quindi riesco a rilevare un’anomalia.

Questo funziona bene solo se ho un pattern omogeneo tipo questo (curva blu e verde molto simili), se ho un pattern con grosse fluttuazioni il profilo predetto non riesce a stare ai dati osservati e quindi ho fluttuazioni continue anche della deviazione standard tra quello predetto e quello osservato e non riesco a rilevare le anomalie.

FINE 31/10

Metriche statistiche:

* Media mobile rispetto ad una soglia: se ridotti a un dataset normale standard, i punti che si trovano in [-2,2] sono accettabili
* Median absolute deviation (MAD): mediana della deviazione assoluta dalla mediana della serie.

MAD = median (|x – median(x)|)

La mediana è meno suscettibile della media agli outliers => MAD è robusta nel dataset con outliers

* Molto semplice, ma spiegabile, riproducibile, semplice, facile da regolare e mantenere.

Innanzitutto la bontà di adattamento.

Confrontare la distribuzione prevista (tramite modelli) e quella osservata per rilevare anomalie. Nel mondo reale, dataset di modellazione difficili con distribuzioni semplici e conosciute. Quando la distribuzione attesa è nota, l’approccio è fattibile.

Test noti (ad esempio chi-squared test) applicabili solo per dataset unidimensionali: applicabilità limitata.

Fitting a busta ellittica:

* Solo per dataset distribuiti normalmente, stimando il rapporto anomalo
* Filtra le anomalie nei dati di addestramento
* La dimensione temporale viene trascurata, ma funziona con finestre temporali

Immagine che contiene schizzo, disegno, cerchio

Descrizione generata automaticamente

Unsupervised machine learning (UL)

One-class SVM: adattamento dei dati solo alla classe normale (traffico benigno).

* Nessun meccanismo di robustezza in questo metodo

» Training meno resiliente agli outliers

* Più adatto per il rilevamento di novità (attacchi 0-day) rispetto al rilevamento di valori anomali

» Solo se il training set risulta ripulito da anomalie

Random forests:

* Funziona bene con dataset ad alta dimensione: basato su alberi, quindi computazionalmente efficiente. Il numero di elementi determina l'altezza dell'albero.
* Presupposto: i valori anomali dovrebbero essere più facili da isolare. Lunghezza del percorso viene utilizzata come metrica per il punteggio di anomalia. Le anomalie dovrebbero avere lunghezze di percorso più brevi.

Metodi basati sulla densità:

* Metodi di clustering UL, come k-means.

I metodi basati sulla densità sono adatti per dataset ad alta dimensione.

Difficile da gestire utilizzando le altre classi di metodi di rilevamento delle anomalie.

* L'idea principale alla base di tutti questi è quella di formare una rappresentazione in cluster dei dati di addestramento. Nell'ipotesi che valori anomali o anomalie saranno localizzati in regioni a bassa densità di questa rappresentazione del cluster. Robusto ai valori anomali nei dati di addestramento. Tali casi si troveranno probabilmente anche nelle regioni a bassa densità.
* A questo scopo può essere utilizzato anche k-NN: punti con distanza elevata dal k-esimo vicino più vicino.

26/10 (Laboratorio)

**Attacchi DDoS (Distributed Denial of Service)**

Gli attacchi DDoS sono gli attacchi che mirano a riempire la disponibilità di un servizio o limitando le risorse di calcolo del server (o dei server) che offrono il servizio stesso o saturando le risorse di rete che permettono al servizio di essere connesso a Internet.

Componenti principali di un attacco DDoS:

* Software che permette di effettuare l'attacco, deve essere costruito:
  + in grado di essere eseguito su un gran numero di macchine, tipicamente di tipo eterogeneo (non deve essere specifico per sistemi Linux, Mac o Windows)
  + capace di nascondersi: una volta insidiatosi dentro una macchina infestata dovrebbe essere difficile da rilevare. Tipicamente quello che succede è che, a meno che non si sappia cosa si deve cercare, il software è difficile da vedere
  + in grado di comunicare con l'aggressore o di disporre di un meccanismo di attivazione per avviare l'attacco. Questo perché di solito gli attacchi cominciano tramite un ordine esplicito esterno, tipicamente di chi organizza l’attacco, oppure attraverso un trigger che può essere una data o un particolare evento all’interno del calcolatore. In alcuni casi questi eventi possono essere sincroni. Ad esempio, il 12 dicembre alle 9.25 del mattino, tutte le macchine infestate cominceranno a inviare traffico verso una certa destinazione. In altri casi possono essere anche asincroni. Dopo il 12 dicembre, non appena un utente della macchina lancerà un processo di stampa, ad esempio partirà l'attacco, quindi saranno più o meno asincroni.

La cosa fondamentale è che quando le macchine vengono infestate, il processo di diffusione del contagio dovrebbe passare più sotto traccia possibile, altrimenti quello che succede è che di solito ci si accorge del problema e il problema viene risolto prima ancora che costituisca una minaccia, quindi prima ancora di effettuare l'attacco.

* Vulnerabilità in un gran numero di sistemi: la parte preliminare di un attacco di DDoS consiste nell'identificare delle specifiche vulnerabilità delle macchine e di trovare quelle macchine che ancora esibiscono queste vulnerabilità. Quindi questa vulnerabilità per mancanza di una soluzione software o per pigrizia dell'amministratore di sistema non sono state risolte.
  + Identificare una tale vulnerabilità
  + Diventeranno macchine zombie
* Scansione per individuare questi sistemi vulnerabili: per trovare le vulnerabilità è necessario effettuare una scansione “a tappeto” dei sistemi.

Di solito il processo è un processo multi-step, nel senso che prima vengono infettate alcune macchine, poi da queste viene fatto uno scan per trovarne delle altre e poi da queste nuove vengono fatti altri scan per trovarne delle altre e così via. E questa è tutta una fase preliminare tipicamente abbastanza lunga, proprio per evitare che il contagio continui a non essere “completo” nel momento in cui arriverà il trigger che scateni l'attacco. Perché qual è il grande vantaggio degli attacchi DDoS? Il grande vantaggio è che l’attacco è distribuito, e quindi viene da un numero molteplice di sorgenti. Più le sorgenti sono tante, più ciascuna sorgente avrà la possibilità di rimanere nascosta, ovvero di contribuire con una porzione di traffico minimale che può sfuggire ai sistemi di rilevamento delle intrusioni o ai firewall, passando per traffico legittimo e quindi è difficile da bloccare, almeno nella primissima fase. Poi chiaramente nel momento in cui il danno è fatto, sarà bloccato dai sistemi di protezione o di recupero della rete, ma la cosa fondamentale sarebbe bloccarlo.

Procedura di scansione: di solito avviene attraverso delle liste. Queste liste possono essere costituite da indirizzi IP a caso, che vengono testati ad esempio con software stile a nmap per vedere se hanno delle porte aperte e se hanno delle porte aperte qual è il sistema operativo. Quando abbiamo visto nmap abbiamo visto che dando delle opzioni molto semplici si riesce a vedere la versione del sistema operativo, il tipo di sistema operativo, le porte aperte e così via. Questo è il modo più lungo, cioè avere degli indirizzi IP a caso, altrimenti magari sulla base di passate esperienze possono circolare delle liste. Le liste non possono essere scandite tutte nell'arco di un'ora, altrimenti il processo di scansione viene rilevato e quindi viene bloccato nella fase iniziale del contagio.

Quindi questa fase è tipicamente piuttosto lunga e che cosa succede? Se io ho una lista molto lunga, comincio a distribuire questa lista tra le macchine infette facendo una partizione, e poi ogni macchina che riesce a infettarne un'altra ridivide la lista in liste più piccole, assegnando a una macchina nuovamente infettata un pezzettino della lista, così che lo scan avviene da tante macchine diverse e anche questo tende a passare sotto braccio.

* Sondaggio casuale dell'indirizzo IP: processo potenzialmente lungo, altrimenti potrebbe essere facile da rilevare
* Lista dei successi:
  + Ad ogni macchina infetta viene fornita una parte dell'elenco da scansionare
  + Processo potenzialmente lungo

Lo scan può essere topologico o locale.

* **Topologico**: utilizza le informazioni della macchina infetta per trovare altri host.

Ad esempio, le utility di windows che permettono di vedere tutti i calcolatori nella propria sottorete. Quindi quelli saranno dei calcolatori che sono delle potenziali sono delle potenziali vittime, oppure possono non essere nella stessa sottorete ma magari condividere una cartella condivisa. Quella è un'informazione topologica interessante che può essere tentata dal processo di diffusione del contagio, prima ancora di scatenare l’attacco. Oppure la cosa più semplice di tutte, dal punto di vista topologico, è fare uno scan a tappeto sulle macchine in LAN con la mia, perché il numero di macchine in LAN sarà comunque limitato e poiché di solito all'interno della singola LAN non ho nessun firewall proprio perché il firewall protegge la LAN ma non analizza il traffico che passa all’interno della LAN, di solito queste operazioni non vengono facilmente rilevate.

* **Locale**: è un sottoinsieme dello scan topologico.

se un host viene infettato dietro un firewall, cerca i candidati zombie nella sottorete locale

Questa è tutta la fase preliminare, che ha il compito di creare le macchine infette che tipicamente prendono il nome di zombie. Gli zombie possono essere di primo, secondo, terzo livello, diciamo che è un processo multilivello. Noi avremo un unico attaccante che infetta un certo numero di macchine, dalle quali poi vengono infettate altre macchine. I cosiddetti agenti zombie rappresentano il livello più basso della gerarchia e gli agenti zombie di solito inviano poi il traffico verso una vittima.

Questo è il cosiddetto **attacco diretto**, ovvero ciascun agente zombie contribuisce con una porzione di traffico più o meno significativa all'attacco verso la vittima. A seconda del tipo di traffico che gli invierò sarà un attacco volto a esaurire le sue risorse computazionali oppure le risorse di rete che lo connettono ad internet.

Immagine che contiene diagramma, linea, design

Descrizione generata automaticamente

Esiste anche un altro tipo di attacco sempre del DDoS che è l'**attacco di reflection** che utilizza il sistema dei riflettori. Questo perché nel caso dell'attacco diretto nel momento in cui la vittima vede arrivare un’elevata quantità di traffico da un certo numero di macchine, la cosa più semplice da fare è bloccare tutte quelle macchine che stanno inviando del traffico magari a ripetizione, anche se è modesto però con un pattern facilmente identificabile. Quindi basta che blocco gli zombie e l'attacco finisce così.

L'attacco di reflection è invece più ostile da bloccare perché l’attacco consiste nel generare del traffico che sembra legittimo, attraverso l'uso di reflettori. I reflettori sono dei server non infetti che rispondono al traffico delle macchine infette, quindi degli zombie. Questo perché il traffico che inviano gli zombie sembra traffico legittimo che proviene dalla vittima, e quindi questi server, che tipicamente sono o server web o server DNS o altri server di questo tipo ad esempio server NTP quelli che danno la sincronizzazione temporale, sono di solito server ad alta affidabilità, quindi che sono replicati, ridondati eccetera, che supportano anche moli di traffico molto consistenti. Pensate ad esempio un DNS di Google.

Questi server tipicamente non è una buona idea bloccarli da parte della vittima, proprio perché forniscono dei servizi che a volte anche la vittima stessa può utilizzare e quindi è più complicato bloccare un attacco di questo tipo senza privare la vittima di alcuni tipi di servizi. Quindi quello che si può fare per bloccare un attacco di questo tipo è cercare di limitare al massimo la quantità di traffico che viene che viene affetta la vittima stessa. In qualche caso questo comporterà anche la perdita di traffico legittimo, cioè traffico che viene da un utente legittimo.

In questo caso lo schema è che gli zombie inviano del traffico ai reflettori utilizzando come indirizzo IP sorgente quello della vittima; i reflettori rispondono all'indirizzo IP sorgente, che pensano essere quello della vittima, rispondendo alla vittima quindi tutte le risposte invadono la vittima.

Immagine che contiene linea, diagramma, origami

Descrizione generata automaticamente

Quindi i due schemi sono: inondo la vittima con delle richieste oppure faccio inondare la vittima con delle risposte da parte di reflettori che sono server pubblici, che difficilmente possono essere bloccate che non fanno niente di illegale e niente di sbagliato, rispondono a delle richieste legittime.

Entrambi questi tipi di attacco di solito utilizzano la tecnica dell’ip spoofing, che consiste nell’utilizzare come indirizzo IP mittente non il proprio, ma l’indirizzo di qualcun altro. Nel caso dell'attacco diretto gli zombie possono essere configurati per utilizzare gli indirizzi IP anche casuali, perché nel momento in cui arriva molto traffico alla vittima, se il traffico arriva da 6 macchine, quindi da 6 indirizzi IP, e il traffico comincia a crescere è chiaro che è facile portare quei 6 indirizzi IP. Ma se ciascuno di queste 6 macchine zombie utilizza 100 indirizzi IP generati a caso come indirizzi mittenti, allora il traffico non arriverà da 6 macchine ma da 600 macchine e quindi la porzione di traffico che vedrà la vittima, associata a ciascuna di queste 600 macchine, potrebbe essere modesta e quindi sarà più complesso bloccare questo traffico (600 macchina o 6000).

La stessa cosa per quello che riguarda questo schema qui, solo che per questo schema l'indirizzo mittente è sempre quello della vittima, quindi per fare il modo che l'attacco, soprattutto nelle fasi iniziali non sia rilevato, la cosa fondamentale è utilizzare un numero molto elevato di reflettori. Se ho tanti reflettori ogni reflettore, ad esempio non si insospettirà per un elevato traffico che arriva da un indirizzo IP della vittima, perché ogni reflettore pensa di rispondere al traffico generato dalla vittima.

Di solito quindi abbiamo due tipi di attacchi: l'attacco di sopra, che è un attacco alle risorse di calcolo della vittima, ad esempio tramite traffico TCP: i miei zombie generano in maniera diretta l'attacco inviando traffico TCP. Inviano pacchetti TCP SYN, la vittima risponde con SYNACK quindi apre il socket e aspetta il terzo messaggio per chiudere il 3 way handshake, e questo passa a curare ad esempio la RAM della vittima stessa.

Oppure posso avere un attacco alle risorse di rete che interconnettono la vittima con il resto del mondo. Per far questo tipicamente si utilizza o traffico ICMP e l'attacco è di reflection di solito. Traffico ICMP ad esempio gli zombie che sono questi laterali in verticale inviano dei ping ai reflettori che rispondono a quello che pensano essere il mittente del ping, cioè la macchina che può essere un router o anche la vittima stessa. Magari quindi l'obiettivo è di intasare il router a monte, senza neanche andare a fare il ping verso la vittima.

Oppure, l'altra cosa che si può fare è inviare il traffico UDP, perché il traffico UDP lo invio e basta, non ho bisogno di sessioni. Posso inviare un certo numero di pacchetti a frequenza anche elevata, in modo tale da cercare di intasare il router. Ad esempio posso inviare i pacchetti non tutti allo stesso indirizzo IP ma a un certo numero di indirizzi IP che sono tutti connessi al router. Poi quelle macchine quando riceveranno questo traffico lo scarteranno, però intanto questo traffico dovrà attraversare il router andando a rendere indisponibile la sua connessione dati verso l’esterno.

Immagine che contiene testo, diagramma, schermata, design

Descrizione generata automaticamente

Abbiamo quindi due tipi di attacchi

Gli zombie generano in mamnei9ra diretta l’attacco inviando traffico tcp ape il socket e aspetta il terzo messaggio e questo va a saturare

L’altra cosa che si può fare è inviare traffico udp per intaseare il rpuer. Ad e smepipo possno inviare pacchetti non tuti allo stesso indirizzo ip però intanto tutto il traffico dovrà andare ad attraversare il router rendendo

Possibili contromisure:

* Prevenzione e prelazione degli attacchi (**prima** dell'attacco): si possono mettere delle regole che tentino di imitare il più possibile l'esecuzione dell’attacco; spesso sono di modesta efficacia.

L'importante è che queste regole non creino un disservizio al traffico legittimo.

* + Policy, sistemi e protocolli di patching, macchine di backup
* Rilevamento e filtraggio degli attacchi (**durante** l'attacco): si può cominciare a bloccare volumi di traffico sospetto. Per bloccare volumi di traffico sospetto ho necessità di monitorare la mia rete o i miei servizi. Gli schemi basati sui dati necessitano di monitoraggio costante e continuo; fare monitoraggio solo durante l'attacco può essere abbastanza rapido. Se ho il sistema di monitoraggio permanentemente riuscirò a filtrare parte del traffico durante l’attacco e quindi a mitigare l'attacco.
  + Risposta immediata
* Tracciabilità e identificazione dell'attacco (**durante e dopo** l'attacco): se io ho un sistema di monitoraggio permanentemente dopo l'attacco devo riuscire a capire da dove è arrivato l'attacco e se ci sono delle contromisure più efficaci di quelle che ho messo in piedi durante l'attacco, per fare in modo che l'attacco non si ripeta.

Di solito è molto difficile risalire a chi ha scatenato veramente l'attacco perché, se la tecnica di IP spoofing è utilizzata in modo opportuno, a un certo punto arrivano dei pacchetti con indirizzi IP casuali e non riusciamo a risalire a chi lo ha scatenato. Però magari guardando al profilo dell'attacco potrebbe essere possibile trovare delle strategie che permettano di rilevarlo e bloccarlo nel prossimo futuro nelle fasi iniziali.

* + Cerca di evitare attacchi futuri
  + Non veloce e molto difficile da implementare
* Quindi buona parte di questi attacchi, così come gli attacchi di mascheramento o di impersonificazione o a volte anche quelli di codifica di dati, prevedono spesso la tecnica dell’ip spoofing. Quindi creare pacchetti in cui l'indirizzo IP non è il proprio, e per far questo sono necessari dei software appositi.

Vedremo oggi in laboratorio software che sfruttano una libreria in python che permette di generare in maniera del tutto legittima del traffico arbitrario e faremo delle prove per vedere come un firewall, configurato in maniera più o meno semplice tramite iptables, di solito non è robusto rispetto alla tecnica dell’IP spoofing, e quali sono gli accorgimenti da adottare per fare in modo che al momento non succeda.

FINE TEORIA 26/10, POI LABORATORIO