Sicurezza Web e Web Application Firewall

Lo straordinario successo di Internet è supportato e dovuto ai miliardi di siti e servizi Web accessibili a chiunque. Ovviamente tutti questi siti e servizi Web sono primari obiettivi di attacco e punti cruciali per la sicurezza in Internet.

Sin dalla fine degli anni ’90 con l’avvio dei primi siti di commercio elettronico fu evidente l’esigenza di misure dedicate di sicurezza a loro protezione. Nacquero quindi i primi Web Application Firewall (ad esempio il progetto Open Source ModSecurity iniziò nel 2002 [1]) il cui scopo è introdurre misure di protezione a livello applicativo, non di rete, a partire dal protocollo HTTP sino alle applicazioni ed alle transazioni svolte dagli utenti.

Ancora oggi però, più di vent’anni dopo la loro nascita, i Web Application Firewall, o WAF, non sono diffusi come ci si potrebbe aspettare e la loro efficacia è spesso messa in dubbio. In questo articolo si cercheranno di capire i punti di forza e di debolezza dei WAF, partendo da una breve descrizione di cosa sono e come vengono utilizzati.

Utilizzo di un WAF

E’ utile partire da una descrizione di come utilizzare un WAF. Per fare questo consideriamo un esempio teorico utile a identificare i punti di interesse. Ovviamente si parte dall’avere un servizio che espone un’interfaccia pubblica in Internet, ovvero un’interfaccia che può essere raggiunta da chiunque da Internet. Questo però non vuol dire che chiunque possa accedere a tutti i servizi e dati dell’applicazione in quanto può essere richiesta un’autenticazione per fare ciò. In questa discussione però non è necessario arrivare a

considerare queste logiche applicative che sono spesso specifiche per ogni applicazione. Per le valutazioni che verranno fatte non è inoltre necessario considerare se il servizio esposto è un classico sito web o di commercio elettronico, il backend di una Mobile App o sono esposte delle Web API. Il punto importante è che il trasferimento di dati in Internet è effettuato utilizzando il protocollo HTTP e codice in HTML, XML, JSON, Javascript ecc.

Considerando una struttura comune di servizio Web, ovvero basato su protocollo HTTP/HTTPS, esposto in Internet, la prima cosa da considerare è dove collocare il WAF nella catena di servizi che compongono l'applicazione.

On-premises, un'architettura logica abbastanza comune per un servizio Web con WAF è descritta in Fig .1.



Fig. 1: Architettura logica di un servizio Web con WAF

L'architettura descritta in Fig.1 è puramente logica in quanto alcuni elementi possono in realtà sia essere realizzati dallo stesso sistema sia da molteplici sistemi. Inoltre le comunicazioni tra i vari componenti sono tipicamente cifrate e sono presenti ulteriori Firewall, appliance di sicurezza dedicate (e router, bilanciatori ecc.) che separano e proteggono i diversi livelli applicativi. Scopo del diagramma è quello di descrivere il principale percorso di un accesso applicativo proveniente da Internet al servizio Web:

- esposto dal Frontend Web Server
- gestito dall'Application Server
- con dati archiviati nel Database.

L'accesso applicativo da Internet utilizza di norma il protocollo HTTPS/TLS e quindi tutti i dati sono cifrati quando attraversano i servizi di sicurezza perimetrali quali il Firewall/IPS presente in figura.

Per proseguire è necessario pertanto che i dati vengano decifrati in modo che possano essere acceduti dai sistemi seguenti e per questo motivo il primo servizio presente è il terminatore TLS che appunto termina la sessione TLS stabilita con l'end-point dell'utente. Questo però non vuol dire che da questo punto in avanti i dati sono trasferiti in chiaro, ma che i canali sono cifrati punto-punto ("hop-by-hop") da un sistema al seguente.

Visto che ogni sistema seguente il TLS Terminator può accedere ai dati in chiaro, il primo elemento presente è proprio il Web Application Firewall, che esamina i dati ricevuti (sia in ingresso dall'utente che in uscita verso l'utente) e ne garantisce la sicurezza. Se il WAF non identifica problemi di sicurezza, i dati sono inoltrati al Frontend Web Server che gestisce le logiche del protocollo HTTP, all'Application Server che implementa le logiche applicative, e al Database che archivia i dati.

E' necessario specificare come il WAF riceve e gestisce i dati della connessione. Come dovrebbe essere chiaro dalla posizione del WAF, questo riceve i dati utilizzando il protocollo HTTP che però viene terminato definitivamente solo sul Frontend Web Server. Quindi la configurazione più semplice per un WAF è quella di Reverse HTTP Proxy[1] in cui il WAF termina la connessione HTTP[2] dell'utente, esamina i dati, e stabilisce una nuova connessione HTTP con il Frontend Web Server. Questa modalità introduce però degli ulteriori carichi e dei ritardi, oltre a mascherare le vere sorgenti degli accessi al Frontend Web Server, per cui di solito si preferisce configurare il WAF come Transparent Reverse HTTP Proxy, modalità in cui il WAF esamina i dati ma non termina le connessioni HTTP, o anche come Transparent Bridge in cui il WAF gestisce i dati a livello 2 Data Link nel modello ISO/OSI (es. Ethernet) e quindi risulta trasparente agli altri componenti del sistema.

Come verrà discusso in seguito, un'altra possibilità è che il WAF riceva solo una copia dei dati e quindi non possa agire

sui dati in linea e nel caso direttamente bloccare la connessione, ma solo inviare degli allarmi.

WAF Tradizionale

In linea di principio, i controlli svolti da un WAF sono facilmente classificabili in due gruppi:

1. verifica dei protocolli utilizzati, a partire da HTTP;
2. verifica dell'assenza di codice nocivo nei dati trasferiti nei linguaggi adottati dal servizio Web che possono comprendere da HTML a XML, Javascript, JSON ecc., ma anche Java, Php, Python, Ruby ecc., ed infine SQL, no-SQL ecc.

E' chiaro che se il primo punto è relativamente semplice, il secondo non lo è per nulla.

D'altra parte il paragone e la similarità con un servizio Anti-Virus è d'obbligo: in entrambi i casi si tratta soprattutto di individuare codice nocivo ed evitare che venga trasmesso o che siano sfruttate delle vulnerabilità (note) dell'applicazione. Infatti il cuore dei servizi WAF è basato, come per gli Anti-Virus, su librerie di "firme" di attacchi. Come si vedrà più avanti, questa è comunque solo la base delle funzionalità di un WAF.

Ad esempio si assuma che un'applicazione Web sia scritta in Java con un database SQL e che utilizzi pagine HTML con Javascript. Il WAF deve essere configurato con le librerie di firme per attacchi relativi a Java e SQL (ad esempio per attacchi di SQL-Injection), HTML e Javascript. Come per gli Anti-Virus, queste librerie devono essere aggiornate frequentemente in modo da avere disponibili al più presto le firme degli attacchi in corso e delle vulnerabilità appena scoperte per tutti i linguaggi e componenti che compongono il servizio. Tra le librerie di firme devono essere anche presenti (ed aggiornate) tutte le vulnerabilità dello

specifico Frontend Web Server, Application Server e Database.

Questo primo livello di funzionalità di un WAF, da alcuni chiamato “WAF Tradizionale”, permette comunque di proteggere l’applicazione Web da molti tipi di attacchi.

I WAF possono essere molto utili anche come misura compensativa di protezione in presenza di vulnerabilità di un componente del servizio in attesa dell’applicazione della relativa patch (questa funzionalità è anche chiamata “Virtual Patching”).

Ad esempio si consideri un servizio Web con un Application Server in Java per il quale viene resa nota una vulnerabilità grave e che la soluzione di questa vulnerabilità richieda sia l’aggiornamento di Java sui sistemi, sia l’applicazione di una patch per l’Application Server e la modifica di parte del codice applicativo. Ovviamente questo processo richiede del tempo per sviluppare e aggiornare il codice, eseguire tutti i test necessari ed installare in produzione la nuova versione. Nel frattempo l’applicazione sarebbe esposta ad attacchi da Internet che sfruttano questa vulnerabilità e che potrebbe portare alla compromissione completa dell’applicazione. Per evitare ciò è sufficiente che sia presente un WAF a protezione dell’applicazione con le firme di questa vulnerabilità. Ovviamente vi è sempre un intervallo di tempo in cui l’applicazione è indifesa da questa vulnerabilità tra il momento della sua scoperta e quello dell’installazione della firma (periodo indicato usualmente come “zero day”), ma è un intervallo di tempo sicuramente molto inferiore (anche ore rispetto a settimane) a quello necessario per installare il patch completo all’applicazione.

I WAF Tradizionali non sono però solo degli strumenti che verificano ciecamente la presenza o meno delle firme degli attacchi nei dati in transito. Come già indicato, verificano anche la correttezza dell’implementazione dei protocolli e della loro sintassi e possono tipicamente applicare delle

logiche basate su regole (“rule-based logic”) che permettono sia di concatenare serie di dati e condizioni sia di applicare verifiche solo a particolari tipi di dati. Ad esempio un costrutto potrebbe essere valido se riferito ad un linguaggio di programmazione ma non valido o maligno se riferito ad un altro linguaggio.

Sino ad ora si è descritta la funzionalità del WAF assumendo che in caso di rilevazione di codice maligno o non valido questi blocchi la connessione. In realtà spesso i WAF sono configurati in modo non attivo ma passivo, ovvero solo come un sistema di rilevamento di attacchi le cui segnalazioni vengono poi integrate nei sistemi SIEM/SOC dell’azienda. Il motivo principale di questa modalità di utilizzo dei WAF è nel rischio di falsi positivi nel rilevamento di attacchi e quindi del blocco di connessioni valide. Per evitare quindi che il WAF blocchi le connessioni ed in pratica effettui un attacco di Denial of Service al servizio Web, si preferisce alle volte utilizzarlo solo in modalità passiva ed intervenire in caso di reale attacco ed intromissione tramite le procedure standard per la gestione degli incidenti di sicurezza informatica. Questo rischio verrà discusso più in dettaglio nel seguito dell’articolo.

Ovviamente, i WAF permettono anche di essere configurati in modalità mista, ovvero sia attiva che passiva, nel qual caso l’amministratore lo configura in modo che certi tipi di rilevazioni siano bloccati mentre altri vengano solo segnalati. Ad esempio una configurazione mista potrebbe prevedere che la connessione sia bloccata in caso di rilevamento di una firma associata ad un attacco o vulnerabilità grave, mentre in caso di vulnerabilità non grave o di violazione della sintassi di un protocollo venga fatta solo una segnalazione al SIEM/SOC. Infatti capita anche che alcune rilevazioni di minore gravità da parte di un WAF siano dovute ad errori di programmazione o configurazione dei servizi applicativi o degli end-point degli utenti che possono

ridurre l’efficacia del servizio ma non veramente danneggiarne la confidenzialità, integrità o disponibilità. Risulta pertanto più utile segnalare queste situazioni al gestore dell’applicazione piuttosto che bloccarle direttamente.

IPS e ulteriori funzionalità dei WAF

La descrizione che è stata data finora di un WAF non si discosta molto da quella di un Intrusion Prevention System (IPS). Gli IPS sono nati come evoluzione degli Intrusion Detection System (IDS) che hanno solo capacità passive di tracciamento di pacchetti di rete nocivi, aggiungendo anche la possibilità di blocco del traffico. Gli IPS sono di base dei servizi di sicurezza di rete che esaminano tutto il traffico a partire di solito dal livello 3 Networking nel modello ISO/OSI ma arrivano ad esaminare anche i più alti livelli applicativi (es. livello 7 Application). Anche gli IPS lavorano con “firme” di pacchetti nocivi e possono verificare la sintassi dei protocolli con logiche basate su regole. Quindi se sostituissimo in Fig. 1 il WAF con un IPS dovrebbe essere possibile intercettare e bloccare buona parte degli attacchi noti ed implementare il Virtual Patching.

Lo scopo dei due strumenti è però molto diverso: un IPS è uno strumento di sicurezza principalmente a livello rete e generico, che può proteggere contemporaneamente molti servizi e applicazioni di tipo diverso. Un WAF invece è uno strumento di sicurezza a livello applicativo e specializzato che può essere personalizzato per proteggere una specifica singola applicazione. Non è detto che un IPS sia in grado di implementare regole di protezione che dipendono molto profondamente dal linguaggio di programmazione adottato nell’applicazione come invece è naturale che sia per un WAF.

La specializzazione e personalizzazione è però un’arma a doppio taglio. Come prima cosa ogni applicazione dovrebbe avere il proprio WAF dedicato, od almeno la propria istanza di WAF personalizzata. Questo da solo richiede sia molto lavoro

di configurazione e gestione che l'utilizzo di risorse umane e applicative adeguate, il che include risorse di rete, di calcolo, di memoria senza dimenticarsi dei costi delle licenze delle applicazioni, middleware e sistemi operativi (o appliance fisiche o virtuali) necessarie.

Come sempre quando si progetta un sistema di sicurezza, bisogna decidere quale approccio adottare: bloccare solo il traffico che è noto essere nocivo, o permettere solo il traffico che è considerato lecito. Come ben noto le due logiche sono opposte, la prima è riassumibile in "blocca eventi specifici e permetti tutto il resto" (default-permit) la seconda in "permetti eventi specifici e blocca tutto il resto" (default-block). Tipicamente la logica default-block è utilizzata dai firewall mentre gli IPS utilizzano spesso la logica default-permit. Nell'accezione più avanzata, un WAF dovrebbe adottare la logica default-block, ma questo richiederebbe di specificare esattamente tutti i contenuti leciti che possono transitare alla e dall'applicazione.

Due semplicissimi esempi possono dare l'idea di quanto sia complesso questo problema. Si consideri un sito Web teorico con due pagine banali e super semplificate[3], la prima statica con solo testo HTML e qualche immagine, la seconda con un Form HTML, Javascript e la possibilità per l'utente di inviare tramite il Form dei dati costituiti solo da lettere, spazi e numeri. Quando l'utente richiede la prima pagina, il WAF dovrebbe verificare che la richiesta contiene solo il URL esatto senza alcun parametro o dato allegato e che la pagina sia richiesta solo con il metodo GET. Inoltre la risposta dell'applicazione dovrebbe contenere solo testo HTML e immagini, nessun altro tipo di dati quali documenti, codice Javascript o altro. La seconda pagina è più complessa, inizialmente l'utente la richiede in modalità GET come la prima pagina e con controlli simili, riceve però dall'applicazione un Form HTML e codice Javascript: il WAF deve quindi permettere il transito solo di questi tipi di

dati. L’utente compila il Form e lo invia all’applicazione con il metodo POST. Il WAF deve permettere l’invio all’applicazione solo dei campi previsti dal Form e solo compilati con dati del tipo previsto: ad esempio se sono presenti caratteri di punteggiatura, la comunicazione deve essere bloccata. Per comodità dell’utente, il codice Javascript può essere utilizzato nell’interfaccia cliente sul Browser per verificare che i dati siano permessi prima dell’invio all’applicazione. La verifica deve comunque essere fatta dal WAF (e dall’applicazione) in quanto l’utente può aggirare il codice Javascript ed inviare campi con qualunque dato voglia. Infine il WAF deve verificare che l’applicazione invii all’utente solo i dati previsti come conclusione della transazione, ad esempio solo testo HTML.

Ovviamente non è pensabile poter creare e gestire manualmente una configurazione WAF con questo livello di dettaglio. Una possibile alternativa e via di mezzo tra i due approcci default-permit e default-block è per primo di bloccare i dati nocivi basandosi sulle “firme” e sulla verifica della sintassi dei protocolli, linguaggi di programmazione, applicazioni e database adottati, poi aggiungere regole abbastanza generiche e ampie che permettono il passaggio di dati ritenuti non nocivi per la specifica applicazione, ed infine bloccare tutto il resto. L’approccio è sempre alla default-block, ma la presenza di regole ampie che permettono il traffico, un po’ come nell’approccio più tradizionale della configurazione dei Firewall, rende questa configurazione meno dettagliata e personalizzata e quindi più facilmente gestibile.

Next Generation WAF

Sinora abbiamo descritto i WAF Tradizionali, i loro principali punti di forza ed alcune delle loro criticità, ma come ormai comune nel linguaggio commerciale, i WAF evolvono continuamente in nuove “generazioni”.

Come abbiamo indicato, la gestione di un WAF non è sempre cosa

facile e per questo ormai da tempo sono sorte soluzioni Cloud, ovvero servizi WAF erogati in modalità Software-as-a-Service (SaaS). Le modifiche all'architettura descritta in Fig.1 sono in realtà minime, come si vede in Fig. 2.



Fig.2 Architettura logica di un servizio Web con WAF in ambiente Cloud SaaS

I punti di forza di un servizio WAF SaaS sono molteplici e simili in generale a quelli di altri servizi SaaS:

- la gestione dell'infrastruttura (network, hardware, software) è in carico al fornitore
- gli aggiornamenti continui del WAF e la sua manutenzione sono in carico al fornitore che sfrutta notevoli economie di scala garantendo al contempo che l'aggiornamento delle firme e delle configurazioni sia molto più veloce e comprensivo per tutti i clienti
- tipicamente il fornitore offre modalità di scalabilità delle prestazioni non possibili quando il WAF è on-premises in appliance hardware, e non facilmente replicabili o così velocemente implementabili per appliance software
- il fornitore ha visibilità diretta del traffico di tutti i clienti e quindi ha la possibilità di valutare se anomalie di traffico sono eventi locali e veramente anomali o se invece sono nuove campagne di attacco che vengono man mano indirizzate a tutti i servizi Web
- il fornitore è tipicamente in grado anche di offrire servizi anti-Distributed-Denial-of-Service (DDoS) specifici per i servizi Web protetti dai WAF, protezione difficilmente replicabile per i singoli servizi Web con le soluzioni WAF on-premises.

Ovviamente valgono anche i comuni aspetti negativi delle soluzioni Cloud SaaS, a partire dalle a volte limitate possibilità di personalizzazione del servizio.

Una delle principali difficoltà di gran parte delle misure di sicurezza attiva è quella di fronteggiare gli zero-days, ovvero il periodo di tempo tra la scoperta di una vulnerabilità e la disponibilità di una misura per contrastarla, tipicamente aggiornando l'applicazione. Come già descritto, un WAF è utile contro gli zero-days come soluzione di Virtual Patching ma solo se è in grado di individuare gli attacchi che sfruttano la vulnerabilità. Oltre alla velocità nella creazione e distribuzione delle firme, i WAF più recenti adottano alcuni ulteriori approcci.

Il primo consiste nella classificazione del traffico che normalmente viene gestito dall'applicazione e dalla creazione di una policy o di un profilo che lo descrive[4]. In altre parole, osservando il traffico ritenuto "normale" e quindi lecito, il WAF ne crea automaticamente una descrizione che traduce in regole che ne permettono il traffico. Viene bloccato tutto il traffico che non rispetta questa descrizione o che fa scattare una regola che descrive traffico maligno (come descritto nella sezione precedente).

Questo approccio rende automatica la creazione della soluzione descritta nella sezione precedente e che dovrebbe essere ottimale: dovrebbe infatti permettere solo il traffico lecito, desunto dal traffico noto, e bloccare tutto il resto. Il problema è che quasi sempre il traffico noto è un sottoinsieme molto ridotto del traffico lecito, e questo per molteplici motivi tra cui:

- il modo in cui gli utenti interagiscono con l'applicazione può essere molto vario sia nei contenuti sia nell'ordine delle azioni e nelle modalità
- gli utenti possono connettersi all'applicazione Web con strumenti software diversi (ad esempio Mobile App, Browser Web su smartphone, tablet e PC, accessi via API o programmatici ecc.) ed anche solo una nuova versione di un Browser Web può introdurre modifiche al comportamento del client rispetto all'applicazione

- le applicazioni sono in costante evoluzione (si pensi solo ai metodi *Agile* di sviluppo) ed introducono costantemente e continuamente nuove funzionalità e dati, il che può in breve tempo rendere superata la descrizione del traffico fatta dal WAF.

Conseguenza di tutto ciò è un alto rischio di blocco di traffico lecito (ovvero di falsi positivi). Per alcune, poche, applicazioni di nicchia e ad alti requisiti di sicurezza, il blocco di traffico lecito può anche essere accettato (se in limitate quantità), ma non può essere accettato per nulla nella maggioranza delle applicazioni di business, dal commercio elettronico ai siti finanziari, di informazione ecc. E' questo il principale motivo per cui i WAF spesso sono utilizzati solo in modalità passiva, ovvero di tracciamento di eventuali attacchi ma non di loro blocco.

Un altro approccio alla configurazione e gestione di un WAF parte dalla creazione di un profilo di rischio dell'applicazione: siti di commercio elettronico, finanziario o di informazione sono soggetti a minacce di tipo diverso e soprattutto a rischi diversi relativi ai differenti dati e transazioni gestite. Questo permette di creare regole diverse a seconda del profilo di rischio, molto strette su un certo tipo di traffico rilevante per l'applicazione e meno strette per tutto il resto. Questo riduce i falsi positivi che nel caso sono maggiormente possibili per transazioni a maggior rischio per le quali il business può essere più disposto ad accettare il danno conseguente al blocco della transazione. D'altra parte questo permette maggior traffico non lecito, e quindi riduce la capacità del sistema di bloccare eventuali zero-day.

Infine, come in tutti i sistemi di sicurezza di rete, sono anche utili gli approcci che integrano nei WAF sistemi di Intelligenza Artificiale, o più precisamente Machine Learning. L'approccio è molto simile a quello appena descritto di creazione di un profilo di traffico lecito, solo che in questo

caso si utilizza un modello ad esempio di Deep Learning che viene addestrato utilizzando traffico lecito e non lecito. Il modello poi è in grado di estrapolare i dati che ha analizzato individuandone le caratteristiche principali in modo da poter correttamente classificare anche dati che non gli sono mai stati sottoposti. Questi modelli tipicamente ricevono in ingresso in blocchi il traffico diretto all'applicazione e indicano la probabilità che un blocco sia nocivo. Vengono poi definite delle soglie, ad esempio il 90% di probabilità, che possono dipendere dal tipo di dato o dalla catena di regole che gli si applicano, al superamento delle quali i dati sono classificati come nocivi dal WAF. I modelli di Machine Learning, in combinazione con gli altri approcci utilizzati dai WAF, da una parte permettono sia di ridurre sia i falsi positivi dovuti a traffico lecito ma nuovo, sia i falsi negativi, ovvero dati nocivi non identificati, si potrebbe dire per analogia con i dati noti. Comunque pur migliorando l'efficacia dei WAF, i modelli di Machine Learning non eliminano del tutto falsi positivi e falsi negativi, sia per la loro natura intrinseca probabilistica, sia perché anch'essi dipendono comunque essenzialmente dai dati leciti e nocivi utilizzati per la loro istruzione.

Efficacia reale dei WAF

Quanto efficaci sono i WAF realmente? Uno studio di Ponemon [2] del 2019 indica che poco più della metà degli intervistati non è soddisfatta dell'efficacia dei propri WAF e che alcuni attacchi reali non sono stati identificati dai WAF stessi. Il costo e la complessità di gestione sono anche indicati come fattori negativi. Infine solo 1/5 degli intervistati utilizza i WAF in modalità di blocco, in quanto ritiene che il rischio di falsi positivi sia troppo alto.

D'altro canto, sarebbe interessante avere statistiche su quanti attacchi sono stati bloccati o segnalati correttamente dai WAF, e quindi quanti incidenti o intrusioni siano state

evitate. Non c’è dubbio che i WAF non sono strumenti perfetti, come tutti gli strumenti di sicurezza informatica, e che sono difficili e spesso costosi da gestire. E’ pertanto difficile fare un rapporto costo benefici, o meglio tra costi e potenziali danni, per l’utilizzo dei WAF.

I WAF sono comunque uno strumento molto utile per la sicurezza delle applicazioni Web esposte in Internet, ed i continui sviluppi li stanno rendendo sempre più efficaci e più semplici da utilizzare.

Riferimenti Bibliografici

Rif. 1: ModSecurity: https://www.modsecurity.org/

Rif. 2: “The State of Web Application Firewalls” https://ponemonsullivanreport.com/2019/07/the-state-of-web-application-firewalls/

Note

[1] Per convenzione un HTTP Proxy è posizionato di fronte ai client degli utenti, mentre un Reverse HTTP Proxy di fronte ai server HTTP / applicativi.

[2] Più precisamente una connessione TCP con protocollo HTTP; per semplicità in questo articolo non vengono considerati HTTP/2 e QUIC+HTTP/3.

[3] Questa descrizione ignora molti dettagli, a partire dalla presenza di link nelle pagine HTML ecc.

[4] Questo approccio è anche chiamato “Application Learning”.

Articolo a cura di **Andrea Pasquinucci**

https://www.ictsecuritymagazine.com/articoli/sicurezza-web-e-web-application-firewall/