Firewall de Aplicação *Web* (WAF)

Yan Victor dos Santos - 14/0033599

Departamento de Ciência da Computação,

Universidade de Brasília

Email: yanvictor_ds@hotmail.com

***Resumo*— Atualmente, numerosos são os servidores *web* que demandam serviços de grande importância, tornando-se atrativos para terceiros com intenções negativas. Neste contexto, é viável o uso de *firewall*, um mecanismo de defesa de que filtra os dados que passam na rede para evitar possíveis ataques. Entretanto, o *firewall* apresenta-se insuficiente para controlar todos os acessos e requisições devido à diversidade de aplicações, necessitando restringir o foco de atuação para mitigar problemas específicos.**

***Firewalls* dedicados à aplicação *web* são grandes mecanismos de defesa contra atacantes que atuam sobre servidores que, em sua grande maioria, demandam serviços importantes e necessitam de proteção imediata. Estes servidores estão sujeitos a diversos tipos de ataques, principalmente se não houver proteção dedicada à maneira de como foram implementados.**

Este trabalho apresenta os principais conceitos de *Web Application Firewall* (WAF), suas diferentes formas de implantação, os principais tipos de vulnerabilidades dos servidores e como ela reduz as ameaças para manter segura a aplicação *web*.

I. INTRODUÇÃO

No mercado, é comum encontrar servidores que oferecem serviços automatizados, sejam estes serviços burocráticos, de compras online, voltados para a educação, entre outros. Servidores que requerem um alto nível de segrança precisam se equipar com ferramentas que impedem ou dificultam ataques direcionados às suas aplicações de maneira precisa. Infelizmente, alguns atacantes possuem uma diversidade de ferramentas que evoluem conforme a segurança evolui, dado a necessidade de explorar vulnerabilidades que foram mitigadas em versões anteriores de sua aplicação.

A confiabilidade de um sistema interefere diretamente no sucesso dos serviços oferecidos. Usuários que não confiam na segurança de uma aplicação tendem a rejeitar os serviços por ela oferecidos, implicando na tendência de tornar inútil o seu desenvolvimento. A automatização de um sistema permite a exploração de vulnerabilidades não detectadas durante a fase de desenvolvimento. Infelizmente, alguns ataques podem ser fatais para o produto final dos desenvolvedores da aplicação, e até mesmo para seus usuários, dependendo do uso da aplicação.

Atacantes exploram diferentes formas de se obter sucesso em suas ações, principalmente se a estrutura base do serviço baseia-se em *frameworks* desconhecidos ou ferramenteas que não possuem módulo de proteção, ou possuem falhas de implementação. Assim como em diversas definições encontradas na Internet comum, a WAF inspeciona requisições ao servidor, verificando a ocorrência de invasões ou tentativas de remover o serviço do ar, como ataques de negação de serviço, contra as vulnerabilidades conhecidas, assim como *cross-site scripting*, injeção SQL, parâmetro *cookie*, etc. Apenas as requisições que são consideradas legítimas são passadas para a aplicação. Dessa forma protegem aplicações que estejam com vulnerabilidades ainda não corrigidas.

O artigo está subdividido da seguinte forma: O capítulo 2 descreve as características da WAF, explicando as principais diferenças entre ela e o *firewall* comum. O capítulo 3 apresenta as opções de implantação. O capítulo 4 apresenta brevemente o funcionamento da análise sintática e o uso de expressões regulares para gerência de requisições na rede. O capítulo 5 descreve as principais vulnerabilidades de aplicações *web*. Por fim, estão apresentadas as concluões e referências bibliográficas no capítulo 6.

II. A ESTRUTURA DA WAF

O *firewall,* cuja tradução significa "Parede de fogo", é um mecanismo de proteção que checa o fluxo de dados com a intenção de proteger uma determinada aplicação. Eles são úteis contra diversos tipos de ataques, proporcionando assim uma conexão segura. Portanto, para entendermos melhor o que é um *firewall de aplicação web*, vamos primeiro entender quais são os 3 principais tipos de *firewalls* existentes, segundo [7].

A. Três Tipos de Firewall

- Firewalls Locais: *Firewalls* focados em um ambiente específico, seja um servidor ou um *desktop*. Você vai encontrar *firewalls* locais em todos os seus dispositivos. Cada dispositivo pode ter a sua própria configuração e exigência de acesso e seu ambiente local torna-se seu próprio ambiente de confiança.
- Firewalls de Rede: são os mais conhecidos. Você pode encontrá-los em roteadores domésticos, protegendo a nossa zona confiável (rede doméstica) de uma zona não confiável (a internet). Em grandes organizações, há configurações similares, mas elas possuem firewalls adicionais para isolar diferentes partes de uma rede, protegendo seus ativos de rede (zonas de confiança). São projetados para analisar as tentativas de conexão

de rede para várias portas de rede, bem como analisar os pacotes de entrada e suas metadados associadas. Com base em regras, o *firewall* determina o que é e o que não é permitido no ambiente que eles estão protegendo.
- Firewalls de Aplicação: possuem a estrutura similar aos *firewalls* locais e de rede. Eles são tecnologias complementares para as instalações de segurança existentes. Firewalls de aplicação vão além dos metadados dos pacotes que estão sendo transferidos ao nível da rede e focam-se na transferência dos dados reais. Eles são projetados para compreender o tipo de dado permitido dentro de protocolos específicos (SMTP ou HTTP). Existem *firewalls* específicos para diferentes aplicações, como e-mail ou *firewalls* de sites.

B. As Principais diferenças entre o Firewall Comum e WAF

Firewalls são a primeira linha de defesa para servidores web e, por extensão, o resto da rede. Os firewalls permitem que as conexões passem, seguindo as regras gerenciadas pelos administradores de rede. O *firewall* trabalha para tentar garantir que a segurança da rede da empresa seja preservada. Ele controla os dados que são transferidos para o computador da empresa ou saem de dentro da empresa através da internet e impede que softwares maliciosos possam invadir as máquinas da rede. No entanto, essas regras são inadequadas à medida que o tempo passa porque é difícil distinguir se pacote é mal-intencionado ou não, portanto, algumas conexões legítimas são bloqueadas e algumas conexões ilegítimas são permitidas. Baseando-se no problema, o AF (Application Firewall) foi desenvolvido, que verifica não apenas a parte do cabeçalho e rodapé do pacote, mas também a parte de dados.

Segundo [2], a função primária é controle de acesso, política ao qual o tráfico da aplicação é autorizada a ir e vir através da barreira de rede implementada. Como ele se baseia unicamente em atributos da camada de rede, a capacidade de controle de acesso de um *firewall* comum não é especialmente granular. Como resultado, o *firewall* nem sempre pode distinguir e, portanto, controlar as aplicações individuais usando uma determinada porta/protocolo. Os *firewalls* de rede comuns não estão equipados para detectar explícitamente ameaças. A única proteção que eles oferecem contra malwares, ataques e outras atividades não autorizadas é um subproduto das políticas de controle de acesso que estão configuradas para reforçar isso. Por exemplo, se uma ameaça depende de um caminho de comunicação que não seja "aberto", será, por padrão, prevenido (sem nunca ser detectado). O resultado líquido é que os firewalls de rede comuns fornecem proteção relativamente limitada para as propriedades da web de uma organização.

C. O Firewall de Aplicações WEB

A WAF é um tipo de firewall que verifica o nível de dados dos pacotes para proteger a camada de aplicação do modelo OSI. Ao verificar a porção de dados dos pacotes, é revelada uma informação mais detalhada que é referida como granularidade de um pacote. Pelo exemplo dado em [2], dentro do HTTP *header* poderia haver http *requests* e dentro de um http request poderia haver agentes de usuário, *coockies* e mais. Agora, podendo ver essas informações, uma decisão mais informada agora é feita no que diz respeito aos controles de segurança para pacotes específicos passados para o aplicativo.

WAF's funcionam como escudo de defesa para aplicações de acesso via HTTP. Eles são capazes de prevenir ataques que *firewalls* da rede ou sistemas de prevenção de intrusão não conseguem. Se encontram à frente da aplicação *web*, monitoram as atividades e alertam ou bloqueiam o tráfico malicioso que não se enquadram em regras específicas. Uma ilustração sobre WAF é representada na Figura 1. A intenção é captar ataques à nível de aplicação, como SQL *injection* e *cross-site scripting*, apenas "manipulando"o comportamento do sistema. Esta defesa não requer modificação do código fonte da aplicação.

O *firewall* de aplicações *Web* atua onde outras tecnologias de segurança não conseguem, efetivamente, proporcionando proteção contra ameaças que operam nas camadas mais altas da pilha de computação. As rotinas de aprendizagem automatizadas, complementadas por políticas configuradas manualmente, resultam em uma "compreensão"de alta fidelidade de como funciona cada aplicação web protegida, incluindo todos os recursos personalizados e a lógica comercial. Os desvios subseqüentemente detectados representam o tráfego malicioso suspeito, que é automaticamente descartado - por exemplo, bloqueado, permitido sujeito a restrições ou logado - de acordo com políticas definidas pelo administrador. Comparado à outras tecnologias de segurança, segundo o artigo [2], WAF's são únicos em:

- Validar entradas, parando injeção SQL, cross-site scripting e directory transversal attacks;
- Detectar cookie, sessão ou ataque adulteração de parâmetro;
- Bloquear ataques que exploram vulnerabilidades em propriedades Web personalizadas (Exploits);
- Para a exfiltração de dados sensíveis através de identificação e bloqueio de nível de objeto;
- Inspecionar totalmente o tráfego criptografado SSL para todos os tipos de ameaças incorporadas;
- Evitar ameaças que operam através da exploração de lacunas lógicas em aplicativos empresariais personalizados;
- Proteger contra ataques de negação de camada de aplicativo e distribuição de negação de serviço (DDoS);
- Protege dinamicamente informações de resposta do servidor que são potencialmente úteis para hackers, etc.

III. Opções de implantação

Existem diversas opções de implantação de uma WAF. Cada implantação varia de acordo com a necessidade da aplicação. Suas características devem definir a maneira de como o módulo de segurança se comportará para evitar cenários indesejados durante a execução do serviço. Portanto, assim como descrito em [1], as mais utilizadas são: *blacklist*

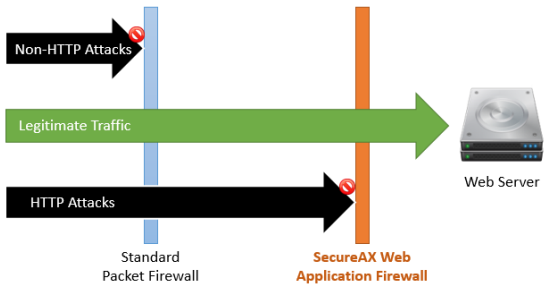

Fig. 1. Ilustração do Funcionamento de uma WAF [6]

e *whitelist*, proxy reverso, ponte de camada 2, *Out-of-band*, servidor hospedeiro e Internet *hosted/cloud*. Abaixo estão descritas as estruturas de cada implantação:

A. Blacklist e whitelist

WAF's comparam requisições a assinaturas de ataques genéricos e políticas específicas de aplicações para a aplicação protegida, e alerta ou bloqueia violações. A Waf pode seguir um modelo de segurança positiva ou negativa para desenvolver as políticas de aplicação. Positiva: Define o que é permitido e rejeita todo o resto. Requer uma ótima compreensão da aplicação. Mais seguro, porém requer um esforço maior. Requer *deep knowledge* da aplicação, para assistir o tráfico da aplicação por um período de tempo e determinar o comportamento padrão dela e entradas, criando um mapeamento de URLs e parâmetros. (*Whitelisting*) Negativa: Define o que não é permitido e aceita implicitamente todo o resto. Não precisa conhecer a fundo a aplicação, apenas monitora o acesso e diz se o acesso viola os padrões. Não protege de ataques desconhecidos. (*Blacklisting*) Todas as WAFs usam o modelo de segurança positiva, negativa ou ambas.

B. Proxy reverso

WAFs possui um endereço IP. Conexões de entrada para a aplicação são enviadas para a WAF, ao qual faz uma requisição separada para o servidor. Conexões encriptadas são terminadas na camada 7 (aplicação), deixando a WAF decriptar e analizar todo o tráfico web. Esta implementação da ao WAF completo controle de tráfico, permitindo que ele reescreva o conteúdo e injete mecanismos de segurança por política.

C. Ponte de camada 2

WAF atua como switch de camada 2. A WAF executa decriptação SSL passiva, e pode bloquear o tráfico simplesmente dropando pacotes ofensivos. Não terminar ou reproduzir o tráfico limita algumas funcionalidades, como não permitir reescrever os elementos por política. A implementação é parecida com a figura do proxy reverso.

D. Out-of-band

Faz cópia do tráfico via porta de monitoramento em um dispositivo da rede. Pode passivamente decriptar tráfico SSL. A habilidade de bloquear o tráfico é ainda mais limitada. Pode apenas enviar pacotes TCP-reset (reseta a conexão) para interromper o tráfico. Este modo tem uma menor quantidade de impacto na rede e na aplicação, e permite que a WAF seja configurada para alertar somente no tráfego malicioso, eliminando o perigo de bloquear em falsos-positivos ou causar uma interrupção da aplicação. Sua localização está conectada à um switch, e faz uma cópia do tráfico.

E. Servidor hospedeiro

É um software instalado no host rodando o servidor web. Poderia ser implementado como uma aplicação independente, ou um plug-in. Não são tão funcionais quanto os outros, mas removem pontos de falha adicionais da rede. Colocam uma carta extra no servidor, então é importante conhecer a utilização de recursos do servidor antes de implementar.

F. Internet hosted / cloud

Novo, mas cada vez mais popular, usa fornecedor da nuvem pra implementar. Funciona como proxy reverso (o DNS público é configurado para a pontar para o provedor de nuvem, que cria outra conexão com o servidor proprietário). Assim como representado na imagem, é externo ao ambiente corporativo e é "software as a service"(SaaS) na nuvem, onde o cliente acessa o software remotamente.

IV. Funcionamento da análise sintática e expressões regulares para detectar ataque

Uma vez que o *firewall* comum não é completamente viável para a segurança de aplicações específicas apenas por meio de metadados, a WAF consegue mitigar os problemas por meio dos dados da própria requisição, testando suas entradas e comportamentos de acordo com o que está proposto para o serviço. A verificação do bom comportamento se dá por um conjunto de regras definidos exclusivamente para uma aplicação específica, devidamente implementada para autorizar ou rejeitar conexões, verificando a estrutura dos pacotes recebidos.

Controles de acesso são implementados usando listas de controle como regras a permetir ou rejeitar tráficos, assim como descrito em [3]. Segundo este trabalho, a precisa inspeção dos pacotes e a decisão do que fazer com os pacotes é feito no nível do kernel. O aplicativo de *userspace* de *iptables* é usado para filtrar os pacotes a nível do kernel e direcionar os pacotes para passar pela WAF antes de entrar no servidor *web*. Todos os pacotes aceitos são encaminhados para o servidor de aplicação. O Web Application Firewall é capaz de comparar a Lista de Controle de Acesso, que é configurada pelo administrador através de qualquer editor de texto, contra os pacotes HTTP recebidos do tráfego antes de chegar ao próprio servidor web. O algoritmo usado para comparar a carga útil do pacote é simplesmente verificação de padrões com o uso de expressões regulares. Os resultados

dos testes em [3] provam o quão precisa é a WAF na detecção e rejeição de diferentes tipos de ataques de acordo com os principais ataques de aplicações *Web* da OWASP (Open Web Application Security Project), descritas a seguir.

V. As principais Vulnerabilidades da aplicação WEB

Para que seja possível assegurar que falhas no sistema não serão exploradas por atacantes, é necessário conhecer como costumam atacar, e quais os tipos de ameaças mais comuns atualmente. Portanto, Segundo [4], as 10 principais vulnerabilidades da aplicação web em 2017, são:

A. TOP 10 vulnerabilidades da WEB

A *web* possui diversos problemas que podem causar prejuízo para usuários, demonstrando necessidade de mecanismos que contribuiam para aumento da segurança na rede. Abaixo, estão brevemente listados os 10 maiores riscos na aplicação, como apresentado em [4].

- Injeção SQL: Injeção direta de comandos SQL é uma técnica onde um atacante cria ou altera comandos SQL existentes para expor dados escondidos, ou sobrescrever dados valiosos, ou ainda executar comandos de sistema perigosos no servidor. Isso é possível se a aplicação pegar a entrada do usuário e combinar com parâmetros estáticos para montar uma consulta SQL.
- Quebra de autenticação: Os invasores usam vazamentos ou falhas nas funções de gerenciamento de autenticação ou sessão (por exemplo, contas expostas, senhas, IDs de sessão) para representar temporariamente ou permanentemente usuários.
- *Sensitive data exposures*: Considere quem pode obter acesso a seus dados confidenciais e a qualquer *backups* desses dados. Isso inclui os dados em repouso, em trânsito e até mesmo nos navegadores de seus clientes. Incluir ameaças externas e internas.
- XXE (XML External Entity): Um ataque XML Entity Externo é um tipo de ataque contra um aplicativo que analisa a entrada XML. Esse ataque ocorre quando a entrada XML que contém uma referência a uma entidade externa é processada por um analisador XML fraco configurado. Este ataque pode levar à divulgação de dados confidenciais, negação de serviço, falsificação de solicitação do lado do servidor, varredura de porta a partir da perspectiva da máquina onde o analisador está localizado e outros impactos do sistema.
- Quebrando o controle de acesso: O controle de acesso, às vezes chamado de autorização, é como uma aplicação da Web concede acesso a conteúdo e funções a alguns usuários e não a outros. Essas verificações são realizadas após a autenticação e governam o que os usuários "autorizados"podem fazer. O controle de acesso soa como um problema simples, mas é insidiosamente difícil de implementar corretamente. O modelo de controle de acesso de uma aplicação *web* está intimamente ligado ao conteúdo e às funções que o site fornece. Além disso, os usuários podem se inserir em vários grupos ou funções com diferentes habilidades ou privilégios.
- *Security Misconfiguration*: Considere os invasores externos anônimos, bem como os usuários com suas próprias contas que podem tentar comprometer o sistema. Considere também os insiders que querem disfarçar suas ações.
- *Cross-site scripting* (XSS): Através de um XSS, o hacker injeta códigos JavaScript em um campo texto de uma página já existente e este JavaScript é apresentado para outros usuários, porque persiste na página.
- *Insecure Deserialiazation*: Os dados que não são confiáveis não podem ser confiáveis para serem bem formados. Os dados malformados ou dados inesperados podem ser usados para abusar da lógica do aplicativo, negar o serviço ou executar o código arbitrário, quando desserializados. A deserialização insegura geralmente leva à execução remota de código. Mesmo que as falhas de deserialização não resultem na execução remota de código, elas podem ser usadas para realizar ataques, incluindo ataques de repetição, ataques de injeção e ataques de escalação de privilégios.
- Usando Componentes com vulnerabilidades desconhecidas: Componentes, como bibliotecas, frameworks e outros módulos de software, são executados com os mesmos privilégios que o aplicativo. Se um componente vulnerável for explorado, esse ataque pode facilitar a perda séria de dados ou a aquisição do servidor. Aplicativos e APIs que usam componentes com vulnerabilidades conhecidas podem prejudicar as defesas de aplicativos e ativar vários ataques e impactos.
- Registro e monitormaneto insuficientes: Registro e monitoramento insuficientes é uma falta de controle e não uma vulnerabilidade por si só. É uma importante falta de controle, com certeza, mas um desenvolvedor não é responsável por manter os logs das aplicações, etc. Não cria uma vulnerabilidade, restringe e limita a resposta ao incidente e as investigações.

Além do mais, podemos usar WAF para mitigar ataque DoS. Além de ser totalmente gerenciado, o Web Security Manager também é uma WAF tecnologicamente avançada, descrita em [8], e inclui muitos recursos para proteger sites da DoS e outros ataques. Esses incluem:

A proteção a nível de rede inclui a limitação da taxa de solicitação e a concorrência por IPs de bloqueio direto que excedem os limites.

No nível de aplicação, o Gerenciador de segurança da Web pode diminuir os pedidos e proteger contra tentativas de força bruta e outros ataques baseados em taxas através da limitação de conexão e limitação de solicitação HTTP. Essas técnicas também ajudam a garantir que os recursos sejam otimizados em situações de pico.

Para ataques do DoS que exaurem os recursos do servidor, tornando os pedidos extremamente lentos (abrindo os recursos do servidor que aguardam exaustivamente a entrada do cliente), o Web Security Manager combate esses

ataques por:

- Aplicando limites de tempo limite para o cabeçalho de pedido do cliente e o corpo do pedido do cliente. Se o cabeçalho da solicitação ou o corpo da solicitação não for recebido dentro de um tempo limite definido, a conexão será fechada. Ataques como Slowloris e slow HTTP POST que são notáveis para o envio de pedidos lentos podem ser tratados através deste tipo de configuração de tempo limite;
- *Bufferizar* todos os pedidos de clientes antes de enviá-los para o servidor *backend*. Por exemplo, o Web Security Manager pode garantir que GETs ou POSTs lentos sejam recebidos na íntegra antes de serem enviados para o servidor backend.

VI. CONCLUSÕES

Todos os servidores de aplicação *web* precisam de proteção contra ataques que comprometam sua integridade. Se tal premissa não fosse levada a sério, parte dos sistemas hoje implementados teriam sido gravemente comprometidos dentro da Internet. Os usuários jamais iriam querer usar o serviço de um sistema que não garante a confidencialidade de seus dados, disponibilidade do sistema, etc. Portanto, *firewall* de aplicação *web* é uma ótima ferramenta para reduzir o problema de ataques, dando mais segurança ao usuário e uma grande satisfação aos desenvolvedores do sistema.

Embora a WAF não seja uma defesa perfeita, possuindo vulnerabilidades, como as apresentadas em [5], ela tenta garantir que a aplicação esteja o máximo protegida possível, segundo as regras definidas para aquela aplicação específica. Sua efetividade supera a de um *firewall* comum dentro do contexto de aplicações *web*, permitindo a validade de sua integração como produto de segurança válido para quaisquer que seja o sistema. Sua estrutura de implantação define as principais necessidades de segurança do cliente, enquanto resolve grande parte dos ataques frequentes à servidores *web*.

REFERÊNCIAS

[1] J. Pubal and B. Filkins, Web Application Firewall, SANS Institute, March 13, 2015. *Disponível em:* https://www.sans.org/reading-room/whitepapers/application/web-application-firewalls-35817

[2] Web application firewall – delivering must-have protection for the modern enterprise, 5 Citrix Systems. *Disponível em:* https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/web-application-firewall-delivering-must-have-protection-for-the-modern-enterprise.pdf

[3] A. Endraca, B. King, G. Nodalo, M. Sta. Maria and Isaac Sabas, Web Application Firewall (WAF), International Journal of e-Education, e-Business, e-Management and e-Learning, December, 2013. *Disponível em:* http://www.ijeeee.org/Papers/277-A0045.pdf

[4] OWASP. (2017). Owasp top 10 - 2010: The ten most critical web application security risks. [Online]. *Disponível em:* https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

[5] OWASP. (2017). The Truth about Web Application Firewalls: What the vendors do NOT want you to knowm TROOPERS 09 – Munich, April 2009. *Disponível em:* https://www.troopers.de/media/filer_public/13/a5/13a57f8a-2634-4ac5-a00c-efbc5d82b35f/troopers09_gauci_henrique_web_application_firewalls.pdf

[6] Secure AX. Figura 1. *Disponível em:* https://www.secureax.com/specialist-solutions/managed-web-application-firewall/

[7] Sucuri Blog, "Diferenças entre Firewalls de Segurança". *Disponível em:* https://blog.sucuri.net/portugues/2016/04/diferencas-entre-firewalls-de-seguranca.html

[8] Alert Logic, Using a Web Application Firewall (WAF) to Mitigate Denial of Service (DoS) Attacks. *Disponível em:* https://www.alertlogic.com/blog/using-a-web-application-firewall-waf-to-mitigate-denial-of-service-dos-attacks/