Implementação e Análise de Técnicas de *Hardening* em Servidores de Hospedagem

Alexandre Pontes Donato
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – (Cefet/RJ)
Nova Friburgo – RJ
alexandre.donato@aluno.cefet-rj.br
ORCID 0009-0001-3772-6206

Bruno Policarpo Toledo Freitas
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – (Cefet/RJ)
Nova Friburgo – RJ
bruno.freitas@cefet-rj.br
ORCID 0009-0006-6825-2361

Nilson Mori Lazarin
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – (Cefet/RJ)
Nova Friburgo – RJ
nilson.lazarin@cefet-rj.br
ORCID 0000-0002-4240-3997

***Resumo*—Com ao avanço das tecnologias e da globalização, as pessoas estão cada vez mais conectadas a Internet, utilizando serviços, consumindo e produzindo conteúdo. A segurança de todos os dados produzidos e consumidos por todos é mais importante do que nunca. A partir disso, a proteção dos servidores de hospedagem que armazenam os dados utilizados por todos nós em nossas relações diárias passa a ser prioridade uma vez que um ataque impede os servidores de operarem corretamente ou rouba os dados, toda uma cadeia de produção e consumo é afetada com repercussões trágicas. Este trabalho analisa diversas combinações de técnicas de proteção para um servidor de hospedagem sob um teste de penetração. Os resultados obtidos demonstram que o uso de um *Web Application Firewall* como *proxy* reverso é preferível na proteção de um servidor de hospedagem, se comparado com um *Intrusion Prevention System*.**

***Palavras-chave*—Hospedagem, Segurança, Proteção.**

I. Introdução

Os Sistemas de Informação devem seguir os princípios de prevenção, detecção e recuperação. Quando uma invasão ocorre, a confiabilidade do sistema é colocada sob questionamento, assim como a capacidade de seu proprietário proteger os dados e manter seus serviços operacionais. Isso pode ser desastroso para empresas que dependem da credibilidade de seus clientes e investidores para realizar seu trabalho de forma adequada [1].

Conforme o *Global Cybersecurity Index* (CGI), mais de 3,5 bilhões de pessoas estão conectadas com o mundo virtual atualmente e riscos à segurança cibernética são uma questão importante que não recebe atenção necessária no Brasil. Uma das medidas propostas pelo CGI é a implementação de uma legislação capaz de identificar as atividades ilegais praticadas online, acompanhada do procedimento necessário para investigar, processar e reforçar essa legislação [2].

No Brasil a Lei Geral de Proteção de Dados (LGPD) regula o armazenamento, utilização e proteção dos dados a todos que tratem dados pessoais de forma *online* ou *offline*, abrangendo várias atividades empresariais dessa forma. Entre os princípios apresentados na lei, estão: o *Princípio da Segurança,* que prevê a utilização de medidas técnicas e administrativas para proteger os dados; e o *Princípio de Responsabilização e Prestação de Contas,* que determina que o possuidor dos dados pessoais consiga demonstrar a adoção de medidas de segurança que protejam os dados, comprovando o atendimento das determinações da lei e da eficácia das medidas [3].

Um ataque é uma tentativa de romper ou fugir da segurança dos computadores, ou da rede. Os ataques podem ser divididos em passivos ou ativos. Um ataque passivo tenta escutar, espionar e aprender sobre um determinado sistema. Alguns ataques ocorrem sem que os sistemas atacados saibam que o ataque está ocorrendo, podendo ser executado como homem do meio, interceptando as comunicações entre cliente e servidor. Exemplos de ataques ativos são o *Denial-of-service* e de força bruta. O primeiro tenta sobrecarregar o servidor de requisições para que este não consiga responder os solicitantes e interrompa a prestação do serviço, e o segundo tenta invadir a máquina servidora, ou a rede com o intuito de extrair dados, ou corromper a estrutura da rede [4].

Ao Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br) foram relatados, em 2020, mais de 665 mil incidentes de segurança, dentre os principais, estão: *scan*, uma varredura em rede de computador que identifica as máquinas e os serviços disponíveis para descobrir vulnerabilidades com essas informações; e *web*, um ataque que objetiva danificar os serviços de um servidor web [5].

Uma das técnicas utilizadas para prevenção de ataques e proteção de rede é o emprego de analisadores de pacotes para controlar o tráfego de dados, verificando as informações que entram e saem da rede para impedir acessos não autorizados. Dentre as ferramentas capazes de impedir um ataque a um servidor de hospedagem podemos destacar o IPS (*Intrusion Prevention System*) que pode funcionar de forma transparente, se colocado à frente de um servidor, analisando todos os pacotes de rede que passam por ele. E também, o WAF (*Web Application Firewall*) que pode funcionar como um *proxy* reverso, se colocado à frente de um servidor, analisando todas as requisições que passam por ele.

Este trabalho apresenta um comparativo das diversas combinações no emprego de analisadores de rede para proteção de servidores de hospedagem. Durante os experimentos, um computador intencionalmente vulnerável é atacado e diferentes combinações de proteção são testadas.

Este trabalho está organizado da seguinte forma: na Seção II são apresentados os principais conceitos e ferramentas uti-

lizadas; na Seção III são discutidos alguns trabalhos relacionados; na Seção IV é apresentada a metodologia da análise; na Seção V os experimentos e seus resultados são apresentados; finalmente, na Seção VI, as considerações finais.

II. Fundamentação Teórica

Nesta seção serão apresentados os conceitos, tecnologias e ferramentas de proteção utilizadas na realização dos experimentos. São elas: Snort, ModSecurity, Kali e Metasploitable.

O IPS Snort[1] atua na camada de enlace do modelo ISO/OSI monitorando o tráfego que chega da internet para determinar se existe algum comportamento fora do normal, segundo as configurações da rede. Além disso, ele consegue bloquear o acesso, caso identifique alguma anomalia, e também pode ser utilizado para monitorar todo o tráfego de uma rede, bloqueando tentativas de invasão [6].

O WAF ModSecurity[2] é um sistema utilizado para proteger um servidor web de ataques *Hackers*, *Spammers*, DDoS (*Distributed Denial-of-Service*), injeções de SQL, etc. Funciona como *proxy* reverso posicionado a frente dos servidores web em uma infraestrutura de rede, recebendo as requisições da rede externa e filtrando de acordo com um conjunto de regras quais requisições são seguras e podem ser encaminhadas ao servidor de destino e quais requisições contém ameaças à segurança da rede [7].

O Kali Linux[3] é um sistema operacional baseado em Debian, para realizar auditoria de segurança. Possui mais de 600 ferramentas direcionadas para tarefas de pesquisa de segurança, engenharia reversa e forense digital e testes de penetração. Neste trabalho foram utilizadas as seguintes ferramentas do Kali Linux: Nikto[4], OWASP's Zed Attack Proxy (ZAP)[5] e Armitage[6].

- Nikto: um *scanner* para servidores web capaz de realizar testes para identificar arquivos ou programas perigosos a segurança do servidor, configuração incorreta do servidor e de aplicações e se o servidor e os programas estão desatualizados.
- OWASP's ZAP: uma ferramenta projetada para testar aplicações web, capaz de interceptar e inspecionar as mensagens enviadas pelo navegador, modificar o conteúdo das mensagens, se necessário e em seguida enviar os pacotes para o destino.
- Armitage: uma ferramenta que possibilita a visualização de alvos que recomenda o uso de *exploits* e exibe as informações sobre as máquinas exploradas.

O Metasploitable[7] é uma versão intencionalmente vulnerável do Ubuntu Linux, utilizada para testes de segurança, penetração e demonstração de fragilidades. A maioria dos serviços em execução permitem um ponto de entrada para o sistema, acesso a backdoors, senhas fracas e serviços web frágeis.

[1]https://www.snort.org
[2]https://github.com/SpiderLabs/ModSecurity
[3]https://www.kali.org/
[4]https://github.com/sullo/nikto
[5]https://www.zaproxy.org
[6]https://github.com/r00t0v3rr1d3/armitage
[7]https://sourceforge.net/projects/metasploitable

III. Trabalhos Relacionados

O trabalho de Tabassum et al. (2021) [8], aborda o hacking ético, testes de invasão, experimentos práticos, implementação e uso da estrutura do Metasploit e do Kali Linux. Foi escolhido um ambiente empresarial para a realização dos testes de penetração. Os autores propõem a utilização de software antivírus, atualização regular de sistema operacional, bloqueio de portas não utilizadas e implementação de IPS borda da rede para detectar ataques e prevenir invasões.

O trabalho acima visa educar e informar sobre a utilização do Metasploitable e do Kali Linux, dessa forma se diferencia deste trabalho que objetiva analisar configurações de segurança para servidores de hospedagem por meio de ferramentas como IPS e WAF. As semelhanças são o Metasploitable e o Kali Linux, sistemas utilizados para implementação de ataques.

O trabalho de Silva et al. (2021) [9], discute o desempenho e a eficácia de três WAF *open source*: ModSecurity, Janusec e Naxsi. Os WAF realizaram a proteção de um servidor executando Apache no Debian para hospedar um site WordPress. Os autores recomendam o uso do ModSecurity, por ser o mais eficiente no resguardo das informações sob ataque dos *scanners* Nikto, OWASP's ZAP e WPScan.

O trabalho de Claro (2015) [10], apresenta um estudo do consumo de CPU e memória principal no uso do IPS Snort. A ferramenta revelou-se eficiente, detectando as intrusões e gerando alerta durante testes de conectividade, *scan* de vulnerabilidades e ataque de negação de serviço. Além de apresentar baixa influência sobre o desempenho do servidor.

Este trabalho, diferente dos anteriores, apresenta um comparativo da eficácia entre três técnicas diferentes de *hardening*, o uso de IPS, o uso de WAF e a combinação das duas. Para tal, será utilizado o ModSecurity, apontado por [9], e o Snort, apontado por [10], de forma *standalone* em um servidor Debian para a proteção de um servidor de hospedagem com diversas vulnerabilidades. Além disso, serão utilizadas as ferramentas de segurança ofensiva: OWASP's ZAP, tal como [9]; Armitage que utilizada a estrutura do Metasploit, tal como [8]; e o Nikto.

IV. Metodologia

Este trabalho busca implementar uma infraestrutura de rede contendo um Metasploitable, máquina contendo vulnerabilidades utilizada para diversos testes de penetração, uma máquina contendo WAF, uma máquina com IPS e uma máquina com Kai Linux, sistema operacional comumente utilizado para realizar ataques a outras máquinas contendo vários programas com diferentes características para invadir e prejudicar o funcionamento de servidores.

A partir dessa estrutura serão realizados diversos experimentos para avaliar a segurança do servidor. Os testes serão realizados utilizando a máquina Kali Linux e tentando atacar a máquina Metasploitable, inicialmente sem a presença das máquinas com IPS e WAF e posteriormente adicionando esses

sistemas e registrando os resultados das tentativas de invasão para avaliar se a segurança aumenta.

Para avaliar o impacto do uso das ferramentas de proteção, foram analisados três cenários: sem proteção, com proteção na camada de enlace ou de aplicação e com proteção em duas camadas do modelo ISO/OSI.

A. *Cenário 1 - Sem proteção*

No primeiro cenário, apresentado na Figura 1 serão utilizados o Kali e o Metasploitable, dessa forma ficarão evidentes as vulnerabilidades presentes na máquina Metasploitable que representa o servidor de hospedagem no ambiente proposto, servindo como experimento de controle. O objetivo deste experimento é verificar quais são as vulnerabilidades existentes no computador alvo e depois comparar com as vulnerabilidades relatadas nos outros cenários.

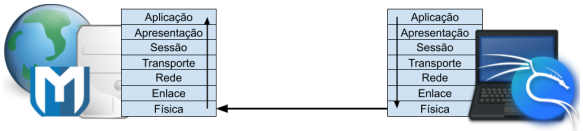

Fig. 1. Cenário 01 - Sem proteção.

B. *Cenário 2 - Um mecanismo de proteção*

No segundo cenário serão realizados dois experimentos, o primeiro incluindo o IPS para monitorar a camada de enlace e impedir a chegada de requisições maliciosas do Kali Linux, conforme apresentado na Figura 2.

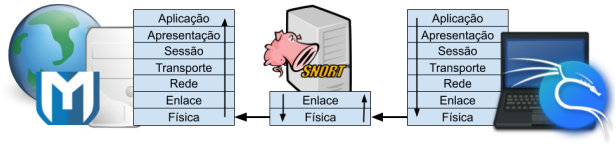

Fig. 2. Cenário 02 - Com proteção do IPS (camada de enlace).

No segundo experimento será utilizado o WAF para filtrar as requisições na camada de aplicação, conforme apresentado na Figura 3. O objetivo deste cenário é avaliar, em que camada do modelo ISO/OSI, uma ferramenta de proteção é mais eficiente.

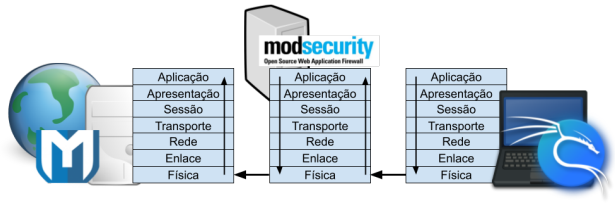

Fig. 3. Cenário 02 - Com proteção do WAF (camada de aplicação).

C. *Cenário 3 - Dois mecanismos de proteção*

O terceiro cenário terá o IPS e o WAF funcionando juntos para proteger o Metasploitable aumentando dessa forma a segurança do servidor de hospedagem de tentativas de invasão da máquina Kali Linux, conforme apresentado na Figura 4. O objetivo deste cenário é analisar a eficiência da proteção em linha de dois mecanismos, atuando em camadas distintas.

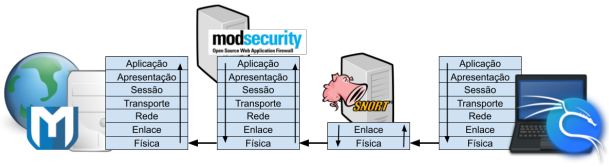

Fig. 4. Cenário 03 - Com proteção do WAF e IPS

V. Experimentos

Os experimentos consistem em atacar um servidor de hospedagem, com vulnerabilidades conhecidas, protegido por um analisador de pacotes. Busca-se avaliar a eficiência das possíveis combinações das técnicas de proteção utilizando WAF e/ou IPS. O servidor alvo executa o Metasploitable, uma distribuição Ubuntu usada para testes de segurança da informação. O computador atacante executa o Kali Linux e as ferramentas Nikto, OWASP's ZAP, e Armitage. Abaixo são descritos as combinações de proteção realizadas de cada cenário e os experimentos realizados em cada um deles.

A. *Experimento 1: Experimento de Controle*

O primeiro experimento realizou três ataques ao servidor web, sem nenhuma defesa, conforme Cenário 1. O objetivo desse experimento é verificar todas as vulnerabilidades possíveis de serem exploradas, servindo como parâmetro de comparação para os experimentos envolvendo as técnicas de proteção. O primeiro ataque foi realizado com a ferramenta Nikto, onde foi possível identificar a versão do Apache e 27 vulnerabilidades, conforme Tabela I.

TABELA I
Relatório Nikto - Cenário 01.

OSVDB-ID	Alerta
OSVDB-877	HTTP TRACE method is active, the host is vulnerable to XST.
OSVDB-3268	/doc/: Directory indexing found.
OSVDB-48	The /doc/ directory is browsable. This may be /usr/doc.
OSVDB-12184	PHP reveals sensitive information via certain HTTP requests that contain specific QUERY strings.
OSVDB-3092	phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
OSVDB-3268	/icons/: Directory indexing found.
OSVDB-3233	PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
OSVDB-3233	/icons/README: Apache default file found. /phpMyAdmin/: phpMyAdmin directory found.
OSVDB-3092	/phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.

O segundo ataque foi realizado com a ferramenta OWASP's ZAP, onde foram reportados 19.380 alertas, sendo: 12 de alto risco; 5.182 de risco médio; 12.134 de risco baixo; e 2.052

Informativos. A descrição detalhada de cada tipo de alerta é apresentada na Tabela II.

TABELA II
RELATÓRIO DO OWASP'S ZAP - CENÁRIO 01.

ID	ALERTA	RISCO	QTD
6	Path Transversal	Alto	12
90022	Application Error Disclosure	Médio	256
10003	Biblioteca JS vulnerável	Médio	1
10020-2	X-Frame-Options Header Not Set	Médio	4925
10202	Absence of Anti-CSRF Tokens	Baixo	6526
10010	Cookie No HttpOnly Flag	Baixo	20
10054	Cookie without SameSite Attribute	Baixo	30
10023	Information Disclosure – Debug Error Messages	Baixo	289
2	Private IP Disclousure	Baixo	138
10037	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Baixo	117
10021	X-Content-Type-Options Header Missing	Baixo	5014
10024	Information Disclosure – Sensitive Information in URL	Informativo	9
10027	Information Disclousure – Suspicious Comment	Informativo	78
10096	Timestamp Disclosure – Unix	Informativo	1965

No terceiro ataque, realizado com o Armitage, foi possível identificar e explorar uma vulnerabilidade no *vsftpd*, conforme Figura 5. Além disso, foi possível acessar remotamente a

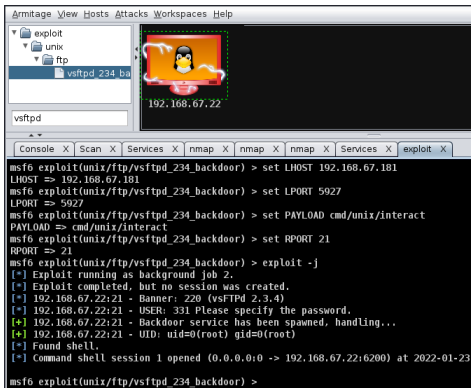

Fig. 5. Exploração de vulnerabilidade no serviço *vstpd*.

máquina via *prompt* de comando, visualizar arquivos em diretórios e identificar o usuário logado, nesse caso o root (administrador), conforme apresentado na Figura 6.

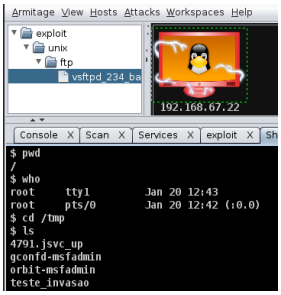

Fig. 6. Invasão com acesso remoto via *prompt*.

B. Experimento 2: Protegido por um IPS

Neste experimento foram realizados três ataques contra a máquina vulnerável, agora protegida por um IPS, conforme Cenário 2. O IPS foi instalado separadamente em uma máquina posicionada entre a máquina atacante e a máquina vulnerável. O IPS utilizado nos experimentos é o Snort.

O primeiro ataque, realizado com o Nikto, identificou a versão do Apache e reportou 7 vulnerabilidades, conforme apresentado na Figura 7.

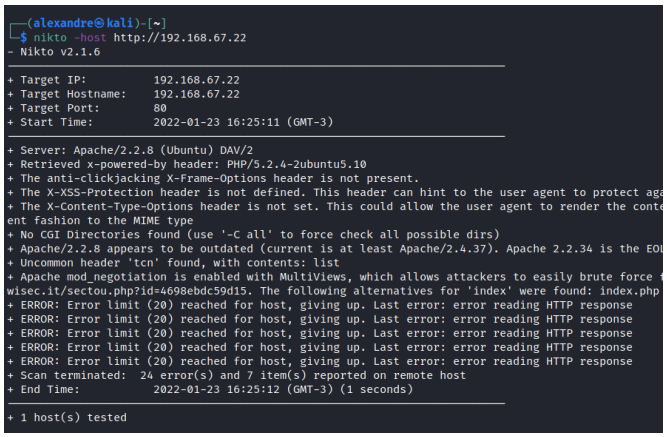

Fig. 7. Relatório do Nikto - Cenário 02, com proteção na camada de enlace.

No segundo ataque foi utilizado o OWASP's ZAP, onde foram reportados 10.756 alertas, sendo: 2.786 de risco médio; 6.524 de risco baixo; e 1.446 Informativos. Neste ataque o OWASP's ZAP não conseguiu reportar alertas de alto risco. A descrição de cada tipo de alerta é apresentado na Tabela III.

TABELA III
RELATÓRIO DO OWASP'S ZAP - CENÁRIO 02, COM PROTEÇÃO NA CAMADA DE ENLACE.

ID	ALERTA	RISCO	QTD
90022	Application Error Disclosure	Médio	140
10003	Biblioteca JS vulnerável	Médio	1
10020-2	X-Frame-Options Header Not Set	Médio	2645
10202	Absence of Anti-CSRF Tokens	Baixo	3475
10010	Cookie No HttpOnly Flag	Baixo	18
10054	Cookie without SameSite Attribute	Baixo	28
10023	Information Disclosure – Debug Error Messages	Baixo	149
2	Private IP Disclousure	Baixo	25
10037	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Baixo	115
10021	X-Content-Type-Options Header Missing	Baixo	2714
10024	Information Disclosure – Sensitive Information in URL	Informativo	9
10027	Information Disclousure – Suspicious Comment	Informativo	78
10096	Timestamp Disclosure – Unix	Informativo	1359

Finalmente, a máquina vulnerável protegida pelo IPS foi atacada com a ferramenta Armitage. Foi possível identificar os serviços disponíveis no servidor, mas não foi possível realizar a invasão, conforme Figura 8.

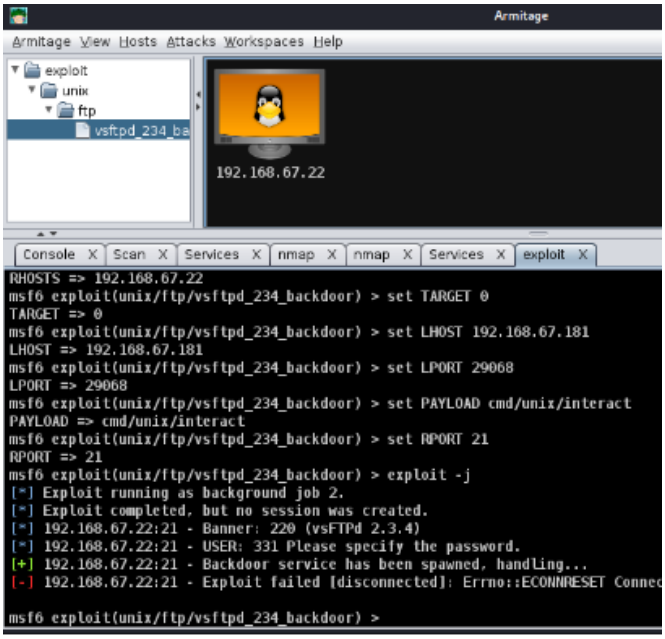

Fig. 8. Tentativa de execução de *exploit*.

C. Experimento 3: Protegido por um WAF

Neste experimento, a máquina vulnerável agora é protegida por um WAF, conforme Cenário 02. O WAF foi posicionado em uma máquina entre a máquina vulnerável e a máquina atacante. O WAF utilizado foi o ModSecurity.

No ataque com o Nikto, foi possível identificar a versão do Apache e reportadas 3 vulnerabilidades, conforme Figura 9.

```
┌──(alexandre㉿kali)-[~]
└─$ nikto -host http://192.168.67.22
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.67.22
+ Target Hostname:    192.168.67.22
+ Target Port:        80
+ Start Time:         2022-01-25 17:55:20 (GMT-3)
---------------------------------------------------------------------------
+ Server: Apache/2.4.38 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the
+ The X-Content-Type-Options header is not set. This could allow the user
ME type
+ All CGI directories 'found', use '-C none' to test none
+ 26545 requests: 0 error(s) and 3 item(s) reported on remote host
+ End Time:           2022-01-25 17:57:35 (GMT-3) (135 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Fig. 9. Relatório do Nikto - Cenário 02, com proteção na camada de aplicação.

O ataque com o OWASP's ZAP, diferentemente dos dois experimentos anteriores onde foram encontradas diversas vulnerabilidades e apresentado diversos alertas, dessa vez retornou um erro *403 Forbidden*.

No ataque com o Armitage, a ferramenta apenas conseguiu listar um serviço disponível, o Apache na porta 80, não tendo êxito em nenhum método de invasão sugerido para essa porta.

D. Experimento 4: Protegido por WAF + IPS

Neste experimento, a máquina vulnerável agora é protegida pelo IPS e WAF em linha, conforme Cenário 3. O WAF foi posicionado em uma máquina entre a máquina vulnerável e a máquina atacante. Por sua vez, o IPS recebe a saída do WAF, filtrando o seu conteúdo e repassando a requisição à máquina vulnerável. Assim como nos experimentos anteriores, o WAF utilizado foi o ModSecurity e o IPS foi o Snort.

Com proteção tanto do IPS quanto do WAF, os ataques realizados pelas ferramentas não obtiveram sucesso: o Nikto não conseguiu encontrar formas de explorar os serviços da máquina Metasploitable; o OWASP's ZAP não conseguiu atacar a máquina vulnerável; e o Armitage apenas conseguiu listar um serviço disponível, o HTTPD na porta 80, não tendo êxito em nenhum método de invasão sugerido para essa porta.

E. Comparação

Um comparativo geral entre os experimentos, exibindo as ferramentas utilizadas para atacar a máquina Metasploitable, os cenários possíveis e um resumo dos resultados obtidos em cada ataque é apresentado na Tabela IV.

TABELA IV
COMPARATIVO DOS EXPERIMENTOS

Cenário	Nikto	OWASP's ZAP	Armitage
Sem proteção	27 itens	14 tipos de alertas	Êxito na invasão
IPS	7 itens	13 tipos de alertas	Falha na invasão
WAF	3 itens	Falhou na tentativa de Scan	Falha na busca de vulnerabilidades
IPS+WAF	3 itens	Falhou na tentativa de Scan	Falha na busca de vulnerabilidades

Na Tabela V é apresentado um resumo comparativo dos ataques envolvendo a ferramenta OWASP's ZAP. A presença do WAF mostrou-se ser suficiente para prover um nível de segurança maior do que o IPS.

TABELA V
COMPARAÇÃO DOS RESULTADOS DOS ATAQUES DO OWASP'S ZAP DE ACORDO COM AS CONFIGURAÇÕES DE PROTEÇÃO DA MÁQUINA VULNERÁVEL

ID	ALERTA	RISCO	Sem proteção	IPS	WAF	IPS + WAF
6	Path Transversal	Alto	12	-	-	-
90022	Application Error Disclosure	Médio	256	140	-	-
10003	Biblioteca JS vulnerável	Médio	1	1	-	-
10020-2	X-Frame-Options Header Not Set	Médio	4925	2645	-	-
10202	Absence of Anti-CSRF Tokens	Baixo	6526	3475	-	-
10010	Cookie No HttpOnly Flag	Baixo	20	18	-	-
10054	Cookie without SameSite Attribute	Baixo	30	28	-	-
10023	Information Disclosure – Debug Error Messages	Baixo	289	149	-	-
2	Private IP Disclousure	Baixo	138	25	-	-
10037	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Baixo	117	115	-	-
10021	X-Content-Type-Options Header Missing	Baixo	5014	2714	-	-
10024	Information Disclosure – Sensitive Information in URL	Informativo	9	9	-	-
10027	Information Disclousure – Suspicious Comment	Informativo	78	78	-	-
10096	Timestamp Disclosure – Unix	Informativo	1965	1359	-	-

Por fim, na Tabela VI é apresentado um resumo comparativo para os testes com a ferramenta Nikto envolvendo todas as configurações dos testes: máquina vulnerável sem proteção; protegida apenas pelo WAF; protegida apenas pelo IPS; e protegida por ambas. Nela, percebe-se que a proteção conferida pelo WAF consegue ser melhor que a conferida pelo IPS.

TABELA VI
COMPARAÇÃO DOS RESULTADOS DOS ATAQUES DO NIKTO DE ACORDO COM AS CONFIGURAÇÕES DE PROTEÇÃO DA MÁQUINA VULNERÁVEL

DESCRIÇÃO	Sem proteção	IPS	WAF	IPS+WAF
OSVDB-48: The /doc/ directory is browsable. This may be /usr/doc.	1	-	-	-
OSVDB-877: HTTP TRACE method is active, the host is vulnerable to XST.	1	-	-	-
OSVDB-3092: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.	5	-	-	-
OSVDB-3233 : PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.	2	-	-	-
OSVDB-3268: Directory indexing found.	4	-	-	-
OSVDB-12184 : PHP reveals sensitive information via certain HTTP requests that contain specific QUERY strings.	4	-	-	-
/phpinfo.php – Output from the phoinfo() functions was found.	1	-	-	-
/phpMyAdmin directory found	1	-	-	-
Server may leak inodes via Etags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mtime: Tue Dec 9 15:24:00 2008	1	-	-	-
Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.	1	1	-	-
The anti-clickjacking X-Frame-Options header is not present.	1	1	1	1
The X-XSS-Protection header is not defined. Protection against some forms of XSS.	1	1	1	1
Apache/2.2.8 appears to be outdated.	1	1	-	-
Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names.	1	1	-	-
The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a diferent fashion to the MIME type.	1	1	1	1
Uncommon header 'tcn' found, with contents: list	1	1	-	-
Server: Apache/2.2.8 (Ubuntu) DAV/2	Identificado	Identificado	Identificado	Identificado
Itens relatados	27	7	3	3

VI. CONSIDERAÇÕES FINAIS

É possível observar que no cenário sem proteção a máquina com Kali Linux consegue acessar várias vulnerabilidades disponíveis no Metasploitable. Nos outros cenários, com a inclusão do IPS podemos perceber uma redução na efetividade dos ataques, o mesmo ocorreu no terceiro cenário com a inclusão do WAF. O quarto cenário teve um resultado similar ao terceiro, o que traz o questionamento sobre a utilidade do IPS. Uma vez que o WAF recebe todas as solicitações na porta 80 e conta com um aparato de filtragem e segurança contra invasões e está separado da máquina servidora o mesmo poderia ser capaz de proteger o ambiente sem a ajuda do IPS.

Com os experimentos realizados neste trabalho foi possível identificar e proteger diversas vulnerabilidades presentes em um servidor de hospedagem e dessa forma criar uma metodologia para configuração de segurança nesses ambientes, além de trazer um alerta sobre o quão importante é a segurança nos serviços disponíveis na Internet. Com as configurações de segurança implementadas, inclusão do IPS e do WAF ficou evidente o aumento da segurança e da proteção no servidor de hospedagem.

Os resultados obtidos sugerem também que a presença de WAF confere um nível de proteção bastante alto em comparação ao IPS. Ou seja, caso seja necessário escolher entre um dos dois métodos, o WAF é preferível ao IPS. Por atuar como *proxy* reverso, o WAF possuí outra vantagem frente ao IPS, pois é capaz de analisar as requisições HTTP que chegam ao servidor de hospedagem. Ao analisarmos os sites de e-commerce, percebe-se que 27% deles utilizam o WooCommerce, uma solução open-source aderente à LGPD, entretanto, 39% dos sites de e-commerce do mundo utilizam soluções próprias [11] [12]. No caso do uso de soluções próprias, é muito importante utilizar um WAF, pois muitos desenvolvedores não se sentem adequadamente preparados para lidar com questões de segurança, e vulnerabilidades como *Credential Stuffing* ou *Session Hijacking* são dificilmente identificados por desenvolvedores web [13].

Para trabalhos futuros, outras técnicas de invasão e outras ferramentas de proteção como Firewall de Camada de Rede e atualizações de segurança podem ser utilizadas para analisar o nível de proteção que podem oferecer e identificar outras vulnerabilidades não encontradas neste trabalho.

REFERENCES

[1] J. M. d. S. Pinheiro, "Ameaças e Ataques aos Sistemas de Informação: Prevenir e Antecipar," *Cadernos UniFOA*, vol. 3, no. 5, pp. 11–21, Mar. 2017. [Online]. Available: https://doi.org/10.47385/cadunifoa.v3.n5.885

[2] ITU, *Global Cybersecurity Index 2020*. Geneva - Switzerland: International Telecommunication Union (ITU), 2021. [Online]. Available: https://www.itu.int/epublications/publication/global-cybersecurity-index-2020/

[3] H. d. S. Gomes, "Lei Geral De Proteção de Dados (LGPD): uma análise dos impactos da lei na cultura e tratamento de dados no Brasil," Graduação, UNISUL, Florianópolis, 2019. [Online]. Available: https://repositorio.animaeducacao.com.br/handle/ANIMA/11112

[4] M. Hyppönen, "Securing a Linux Server Against Cyber Attacks," Mestrado, Tampere University, Tampere, 2021. [Online]. Available: https://trepo.tuni.fi/handle/10024/131119

[5] CERT.br, "Incidentes Reportados ao CERT.br – Janeiro a Dezembro de 2020," 2021, https://www.cert.br/stats/incidentes/. [Online]. Available: https://www.cert.br/stats/incidentes/2020-jan-dec/tipos-ataque.html

[6] L. J. Rohling, *Segurança de redes de computadores*, 1st ed. Curitiba: Contentus, 2020.

[7] F. Melchior, D. Kreutz, M. Fiorenza, F. Flora, I. Ferrao, R. Fernandes, T. Escarrone, and D. Macedo, "Introdução à Web Application Firewalls (WAFs): Teoria e Prática," in *Minicursos da XVII Escola Regional de Redes de Computadores*. Porto Alegre: SBC, 2019, p. 1. [Online]. Available: https://doi.org/10.5753/sbc.5929.0.5

[8] M. Tabassum, O. Muscat, T. Sharma, and S. Mohanan, "Ethical Hacking and Penetrate Testing using Kali and Metasploit Framework," *International Journal of Innovation in Computational Science and Engineering (IJICSE)*, vol. 2, no. 1, pp. 9–22, 2021. [Online]. Available: https://webportal.hct.edu.om/ijicse/pages/publishedjournal/volume2.html

[9] E. Silva, L. da Silva, M. J. Frez, F. M. H. Malara, and N. M. Lazarin, "Waf: Uma análise de desempenho e eficácia," in *Anais Estendidos do XVII Simpósio Brasileiro de Sistemas de Informação*. Porto Alegre, RS, Brasil: SBC, 2021, pp. 21–24. [Online]. Available: https://doi.org/10.5753/sbsi.2021.15347

[10] J. R. Claro, "Sistemas IDS e IPS – estudo e aplicação de ferramenta open source em ambiente linux," Tecnologia em Sistemas para Internet (Graduação), Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense - IFSUL, Passo Fundo, 2015. [Online]. Available: https://inf.passofundo.ifsul.edu.br/graduacao/monografias-defendidas

[11] C. da Costa, O. T. Junior, W. Morete, and N. M. Lazarin, "Woocommerce e lgpd: Uma análise de uso e conformidade," in *Anais da VII Escola Regional de Sistemas de Informação do Rio de Janeiro*. Porto Alegre, RS, Brasil: SBC, 2021, pp. 100–103. [Online]. Available: https://sol.sbc.org.br/index.php/ersi-rj/article/view/16985

[12] BuiltWith Pty Ltd, "eCommerce technologies Web Usage Distribution," 2022, https://trends.builtwith.com/shop. [Online]. Available: https://trends.builtwith.com/shop

[13] P. V. Costa, W. I. Gonçalves, E. D. Gonçalves, and N. M. Lazarin, "Nível de conhecimento de desenvolvedores sobre segurança em aplicações web: Pesquisa e análise," in *Anais da V Escola Regional de Sistemas de Informação do Rio de Janeiro*. Porto Alegre, RS, Brasil: SBC, 2018, pp. 92–99. [Online]. Available: https://doi.org/10.5753/ersirj.2018.4661