INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO CAMPUS BARRETOS

TÉCNOLOGO EM ÁNALISE E DESENVOLVIMENTO DE SISTEMAS

PEN TEST NA PLATAFORMA MOODLE UTILIZADA NO IFSP CAMPUS BARRETOS

RAFAEL LEHNENN OSÓRIO

BARRETOS

2021

O83P Osório, Rafael Lehnenn
Pen test na plataforma moodle utilizada no IFSP Campus Barretos / Rafael Lehnenn Osório. - 2021.
30 f. : il.; 30 cm

Trabalho de conclusão de curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de São Paulo - Campus Barretos, 2021.
Orientação: Prof. Lucas de Araújo Oliveira

1.Kali linux. 2.Moodle. 3.Pen test. 4.Segurança da informação. I. Título.

CDD: 005.3

Ficha Catalográfica elaborada pela bibliotecária Juliana Alpino de Sales CRB 8/8764, com os dados fornecidos pelo(a) autor(a)

PEN TEST NA PLATAFORMA MOODLE UTILIZADA NO IFSP CAMPUS BARRETOS

Trabalho De Conclusão de Curso apresentado ao Instituto Federal de São Paulo, como parte dos requisitos para a obtenção do grau de Tecnólogo

Área de concentração: Segurança da Informação

Orientador: Professor Me. Lucas de Araujo Oliveira

Barretos

2021

RESUMO

Uma rede de computadores é formada a partir da conexão de dois ou mais computadores de forma que consigam trocar informações entre eles de forma autônoma. São incontáveis os dispositivos ligados na rede mundial de computadores internet, e a pergunta que fica é como esses sistemas estão seguros? Assim destacam-se os cibercriminosos, que são indivíduos os quais exploram falhas do sistema afim de tomar vantagem de informações de uma vítima. Como forma de minimizar tais riscos, novas metodologias foram surgindo, como por exemplo o Pen test (Penetration Test), esta metodologia baseia-se em uma sequência de etapas com objetivo de identificar e explorar vulnerabilidades expostas nos sistemas. É feita uma simulação de um ataque real para que possam ser analisadas e encontradas vulnerabilidades, garantindo assim a correção dela. Usufruindo de tal metodologia, pensou-se em verificar as vulnerabilidades mais comuns dos sistemas na plataforma moodle e verificar se ela possui determinado grau de segurança.

Palavras-chave: 1. kali linux. 2. moodle. 3. pen test. 4. segurança da informação. 5. vulnerabilidades.

Sumário

Lista de Ilustrações

1 Introdução

Todos os sistemas os quais se conectam a rede de internet como meio de comunicação possuem uma estrutura dinâmica e poderosa rodando sobre ela. Analistas e gestores trouxeram um item de preocupação, como garantir que tais sistemas estão seguros e fora de ameaça de possíveis ataques virtuais? Segundo Matthew Walker (2014, p.29), existem três pilares que constituem a segurança da informação, sendo eles a confidencialidade, integridade e disponibilidade.

Confidencialidade: A informação só pode ser acessada por entidades autorizadas pelo proprietário. A perda deste pilar resulta quando há quebra do sigilo de um dado, como por exemplo a senha de um usuário ser exposta.

Integridade: A informação deve manter sua característica original, sem alterações. A perda deste pilar resulta em um usuário alterar os dados sem a permissão do proprietário.

Disponibilidade: A informação deve estar acessível quando necessária. A perda deste pilar resulta na informação não estar disponível, como por exemplo um servidor estar offline.

Não podemos garantir que um sistema esteja de fato completamento seguro e confiável, entretanto há maneiras de minimizar esses riscos. Uma maneira é o chamado Pen test conhecido também como teste de invasão. Esta metodologia tem o propósito fundamental de avaliar as consequências que uma vulnerabilidade possa apresentar. Segundo Rodrigo Maués (2019), existem três modelos de Pen test, White Box, Black Box e Grey Box.

O White Box é um dos testes mais simples o qual pode ser executado em uma rede ou aplicação. Esse teste conta com o conhecimento do departamento de TI da empresa e, a equipe de pentesters recebe informações detalhadas sobre a infraestrutura da rede e dados de valor para o ataque. A empresa que efetua o pen test deve ser confiável, pois dados sensíveis da empresa podem ser encontrados. Uma forma que as empresas garantem essa segurança é através do NDA (Acordo de não-divulgação), onde é definido o escopo o teste, onde ele deve ser finalizado e a proteção dos dados contra vazamentos. Resumidamente esse tipo de ataque busca simular um

possível ataque de um membro da empresa que possui muito conhecimento técnico sobre a infraestrutura da empresa.

Por outro lado, o Black Box acontece quase sem informações repassadas ao pentester, o atacante possui pouca ou nenhuma informação sobre o alvo. O objetivo é simular um ataque real executado por um cibercriminoso, exigindo que o atacante tenha todo o planejamento para o ataque, coletando o maior número de informações possíveis, mas acaba por ser um ataque que requer mais tempo. Esse processo dá a impressão que o resultado não trará tantas informações, mas acaba sendo o contrário caso possua uma boa fase de reconhecimento e um escopo bem definido entre as partes envolvidas.

O Gray Box é a junção dos modelos citados anteriormente, o pentester recebe informações de forma parcial, obrigando-o a coletar o restante delas para maior número de dados para o ataque. O solicitante do pen test deve desenvolver um escopo detalhado sobre o que deve ser explorado, garantindo dessa forma que os profissionais da segurança não ultrapassem limites definidos pela empresa. Esse teste é muito requisitado para verificar a segurança de aplicações web.

Por mais que existam os três tipos de pen test, nenhum deles se sobressai perante os outros. Trata-se apenas de verificar o cenário, objetivos, resultados esperados para decidir qual tipo melhor se encaixa na situação atual.

Um especialista em segurança da informação pode atuar em diversas áreas de pesquisa, são inúmeras possibilidades envolvendo a parte de ataques e a de desenvolvimento de segurança. Uma estratégia muito conhecida nesse meio é a Security Red Team e, Blue Team, são equipes com objetivos específicos atuando com segurança da informação.

O Red Team tem como finalidade simular um ataque cibernético contra uma empresa, buscando vulnerabilidades antes que um cibercriminoso encontre e cause danos os quais podem prejudicar a empresa. Os profissionais começam fazendo uma análise sobre os sistemas utilizados para depois começar simular um ataque real, explorando o maior número de falhas possível. Pode-se utilizar de fatores como elementos físicos, humanos, redes de computadores, sistemas utilizados, entre outros. Após o processo ser concluído, os pentesters apresentam um relatório com as falhas encontradas e sugestões para melhorar a segurança, podendo ser atualização de

sistemas, senhas mais seguras, política de monitoramento das máquinas utilizadas pelos colaboradores, etc.

O Blue Team, por outro lado, tem como objetivo fortalecer as defesas da empresa contra uma invasão. São encarregados de garantir que a ação de ataque de um cibercriminoso não tenha sucesso. Pode-se dizer que os dois times caminham juntos, enquanto o Red Team busca vulnerabilidades, o Blue Team trabalha em desenvolver uma solução para essas falhas. De forma geral, procuram criar estratégias para reduzir os danos e prejuízos da empresa caso sofra um ataque externo.

Além dos dois times apresentados, existe o Purple Team, uma junção dos dois times. Dessa forma a empresa conta tanto com a parte de exploração de vulnerabilidades, como a solução das mesmas. Ambas equipes mantém comunicação a todo instante para que o processo seja desenvolvido da forma mais precisa.

Atualmente, um programa de recompensas vem crescendo continuamente e chamando atenção de muitos profissionais da segurança da informação, o chamado Bug Bounty. Trata-se de plataformas em que empresas se cadastram e dão a liberdade para que os usuários busquem falhas em suas aplicações. Caso encontrada e comprovada que exista uma vulnerabilidade, a empresa oferece um valor em dinheiro como retribuição. Dessa forma, as empresas fortalecem a segurança das suas aplicações enquanto os Bug Hunters ampliam seus conhecimento e técnicas de invasão de forma legal, sem nenhum tipo de infração referentes a leis de segurança virtual. Diversas empresas mundialmente famosas estão listadas nas plataformas de Bug Bounty, como exemplo o Facebook. Além do Google que anunciou uma recompensa de 1,5 milhão de dólares para quem consiga encontrar um bug ou vulnerabilidade em seu sistema.

1.1 Objetivos

1.1.1 Objetivo Geral

Este trabalho tem como objetivo principal executar um pentest na tentativa de explorar algumas vulnerabilidades mais comuns em aplicações web, adotando como alvo a plataforma Moodle a qual utilizamos como meio de ensino no Instituto Federal de São Paulo campus Barretos. Desse modo

testaremos a segurança do sistema a ataques cibernéticos comuns na área de segurança da informação.

1.1.2 Objetivo específicos

Apresentação das fases do pen test e algumas ferramentas utilizadas para auxiliar na execução de cada etapa. Existem diversas ferramentas disponíveis na rede de internet para ataques virtuais, porém utilizaremos apenas algumas específicas para fins de testes.

2 Fundamentação Teórica

2.1 Fases do pen test

O avanço tecnológico permitiu a comunicação entre dispositivos computacionais, sendo que em 1984 a International Organization for Standardization, órgão de padronização, desenvolveu o modelo OSI (Open System Interconnection) o qual foi mantido como referência. Este modelo baseia-se em uma arquitetura dividida em sete camadas, a física, enlace, rede, transporte, sessão, apresentação e aplicação. Cada camada conta com suas funções bem especificadas e contam com a utilização de diferentes protocolos. Este modelo atua com a parte física, comunicação entre computadores, roteamento de pacotes, troca de pacotes, controle de conexões lógicas, modificação da forma de apresentar os dados até a última etapa de protocolos específicos para cada software particular. A partir do modelo OSI, é possível efetuar a comunicação entre máquinas distintas garantindo compatibilidade entre os computadores.

Segundo Matthew Walker (2014, p.23), o pen test possui cinco fases: reconhecimento, escaneamento, obtenção de acesso, manter o acesso e apagar os rastros. Cada fase conta com um objetivo diferente e deve ser feito na ordem, desde a coleta das informações até chegar de fato na execução do ataque. São inúmeras as ferramentas existentes para auxiliar em cada etapa deste processo, recomendando-se que o próprio atacante desenvolva suas ferramentas para ser o mais efetivo possível com seu objetivo final. Linguagens como o Python e o Shell Script são muito comuns na criação desta ferramenta, podendo desenvolver softwares poderosos e autônomos a partir de scripts.

O reconhecimento é a fase em que o atacante busca detalhadamente informações sobre o alvo para efetuar a intrusão. Deve-se levantar dados como nomes, responsáveis e servidores do domínio, identificar o sistema operacional, descobrir sub-redes, serviços TCP e UDP, topologia de rede,

contas de e-mail, banners que identificam versões e serviços, estrutura de segurança e diversos outros pontos. Existem duas maneiras de recolher esses dados, chamados de footprint passivo e ativo. O passivo ocorre quando não há interação direta com o alvo, podendo ser uma exploração pelo site, busca pelo histórico do site e suas atualizações, e-mails presentes na página e etc. Enquanto o footprint ativo interage com o alvo podendo utilizar servidores DNS para analisar um domínio, usando comandos no kali linux como o “nslookup” que traduz um domínio para IP e vice versa. Além disso o atacante pode explorar por ameaças zero-day, aquelas que são uma falha de segurança no software recém descoberta, mas que não foi solucionada pelo fato dos desenvolvedores não conhecerem sua existência. Esse tipo de ameaça é vendido ilegalmente para que cibercriminosos consigam tirar vantagem do seu alvo.

O escaneamento conta com a varredura da rede, explorando por portas lógicas abertas e serviços rodando no sistema. O nmap é uma ferramenta mundialmente conhecida pelos especialistas da segurança da informação e muito utilizada nesta etapa do pen test. Tal software consegue efetuar essa varredura de forma poderosa, retornando diversas informações e permitindo que o usuário modele o comando para que consiga com precisão apenas as informações relevantes. Em seus comandos é possível definir a opção de explorar diretamente por vulnerabilidades, permitindo ao atacante ter uma visão ampla de suas possibilidades de ataque.

Após a coleta de informações, o atacante entra na fase de obtenção de acesso, procurando meios de ganhar acesso ao sistema em questão. Um exploit é um trecho de código ou sequência de comandos que tem como meta tomar vantagem de uma falha para assumir controle do sistema. Como o nmap na fase de escaneamento, aqui temos o metasploit, um projeto de segurança de computadores que ajuda em testes de penetração. Tal software conta com diversos comandos específicos para inúmeros tipos de alvo, o usuário deve apenas configurar o ataque utilizando as informações de atacante, alvo, tipo de ataque, etc.

Caso o atacante consiga o acesso, agora ele deve procurar formas de manter esse acesso de forma mais simplificada para não precisar efetuar todo processo novamente. A fase de manter o acesso conta com o backdoor, método em sua maioria secreto para burlar a etapa de autenticação ou criptografia de um sistema ou com o Trojan (cavalo de troia), um malware que garante uma porta para invasões sem a autorização do proprietário. Além destes métodos também existe a possibilidade de utilizar o netcat, um utilitário capaz de ler e gravas dados de conexões de rede utilizando protocolos TCP/UDP.

A fase de apagar os rastros, conta com meios de esconder e excluir todos os registros que possam indicar que o atacante esteve presente no sistema. Comumente efetuada através da exclusão ou modificação dos chamados logs, registros de eventos dos usuários da aplicação. Essa é a última

etapa do pen test, o atacante invade o sistema, garante formas de se manter conectado e esconde todos atos maliciosos executados na aplicação, concluindo o ataque ao alvo.

O pen test conta com uma fase adicional, quando executada por uma empresa com profissionais qualificados. Ao final do ataque, o pentester desenvolve um relatório especificando todas informações, vulnerabilidades e dados que podem ser comprometedores a empresa. Assim o cliente tem em mãos o que deve ser priorizado para manter a melhor segurança de seus sistemas. Algumas empresas de pen test já apresentam as falhas e podem efetuar as manutenções em sequência caso esteja dentro do escopo e seja a vontade da empresa.

2.2 Kali Linux

O kali Linux é um sistema operacional livre o qual possui diversas ferramentas as quais auxiliam no pen test, computação forense e engenharia reversa. Esse sistema conta com diversas ferramentas já instaladas, as quais possuem diversas finalidades e objetivos. Existem ferramentas para ataques de rede wi-fi, mapeamento de rede, execução de código malicioso, quebra de senhas, mapeamento de banco de dados, e diversas outras que podem ser encontradas na internet, desenvolvidas por especialistas da área. Vale lembrar que o cibercriminoso e o profissional de segurança utilizam das mesmas ferramentas, podendo implementar novas funcionalidade para aumentar as possibilidades de execução. O Kali Linux também possui versão para dispositivos móveis, conhecido como Kali NetHunter.

Tal sistema pode ser implantado através de uma máquina virtual no software “Virtual Box”. Assim não é necessário que a própria máquina do usuário possua o sistema instalado.

2.3 Moodle

O Moodle é um software livre, de apoio à aprendizagem, executado num ambiente virtual (aplicação web). O Instituto Federal de São Paulo campus Barretos utiliza de tal software para o ensino, permitindo assistir aulas gravadas e realização de atividades.

2.4 Vulnerabilidades

Vulnerabilidade é qualquer fator que possa contribuir para gerar invasões, roubos de dados ou acessos não autorizados a recursos. A OWASP (Open Web Application Security Project) é uma comunidade global sem fins lucrativos que se empenha em promover a segurança de aplicações web. Em 2020 atualizaram uma lista com as 10 principais vulnerabilidades presentes nestas aplicações, listadas abaixo:

1. Injection;
2. Broken authetication;
3. Sensitive data exposure;
4. XML external entities (XXE);
5. Broken access control;
6. Security misconfigurations;
7. Cross-site scripting (XSS);
8. Insecure deserialization;
9. Using components with known vulnerabilities;
10. Insufficient logging and monitoring.

São inúmeros os sistemas que possuem tais vulnerabilidades nas aplicações comumente utilizadas hoje em dia. Grande parte deve-se ao fato de os desenvolvedores não conhecerem sobre boas práticas para manter a segurança de um sistema. Dessa forma, podemos encontrar diversas falhas em aplicações web de todos os tipos.

3 Desenvolvimento

Este capítulo apresentará o desenvolvimento de teste de segurança da plataforma moodle utilizada no Instituto Federal de São Paulo campus Barretos. Demonstrando a utilização de algumas ferramentas já existentes. Os testes serão executados na plataforma moodle com a URL: moodle.brt.ifsp.edu.br e endereço de IP: 200.133.218.44. Para realização desta etapa, conta-se com a execução de algumas das ferramentas mais famosas no meio de segurança da informação. A partir dos dados coletados, será feita uma análise de vulnerabilidades que podem prejudicar o sistema.

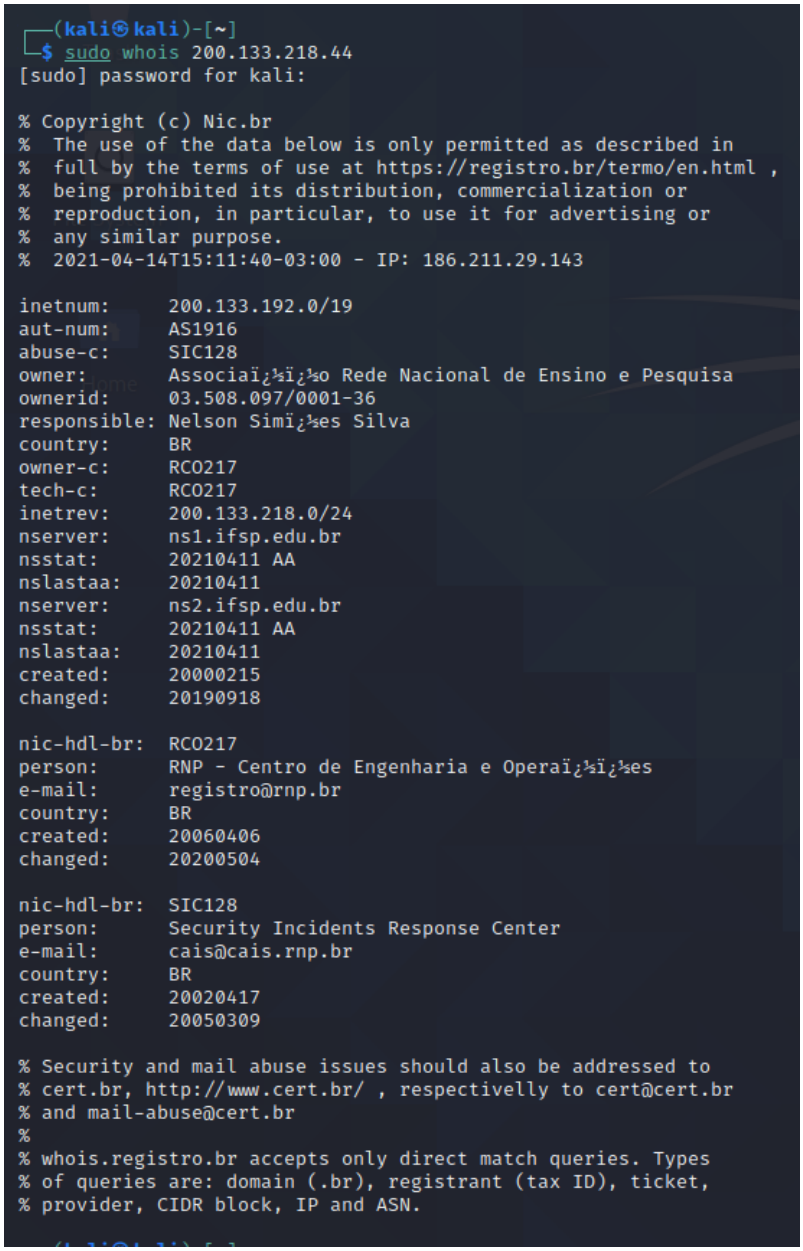
Abaixo temos explicações de algumas ferramentas e o que elas retornaram após sua execução no moodle. Na fase de reconhecimento foram utilizadas as ferramentas: whois e wappalyzer. Já na fase de escaneamento, as ferramentas utilizadas foram: nmap, nikto e nessus.

3.1 Whois

WHOIS é um protocolo específico para consultar informações de contato e DNS sobre entidades na internet. Essas informações são de acesso público, porém algumas empresas internacionais através de pagamentos, podem garantir um whois privado e protegido do acesso de pessoas não autorizadas.

Utilizando do protocolo Whois, podemos analisar diversos dados sobre esse endereço, exibidos abaixo:

Figura 1 – Retorno do comando Whois no Kali Linux.



Fonte: Autor (2021)

Esse tipo de informação também pode ser consultado pela internet através do site registro.br. Nota-se um retorno diferente para coletar mais informações sobre o alvo.

Figura 2 – Retorno do comando Whois no registro.br

Bloco **200.133.192.0/19**

ASN	AS1916
CONTATO DE ABUSO	SIC128
TITULAR	Associação Rede Nacional de Ensino e Pesquisa
DOCUMENTO	03.508.097/0001-36
RESPONSÁVEL	Nelson Simões Silva
PAÍS	BR
CONTATO DO TITULAR	RCO217
CONTATO TÉCNICO	RCO217
CRIADO	15/02/2000
ALTERADO	18/09/2019

Delegações

200.133.218.0/24

SERVIDOR DNS	ns1.ifsp.edu.br
SERVIDOR DNS	ns2.ifsp.edu.br

Fonte: Autor (2021)

3.2 Wappalyzer

O wappalyzer é uma extensão do Google Chrome a qual descobre tecnologias usadas em sites.

Ao utilizarmos tal extensão no ambiente de testes, conseguimos adquirir algumas informações sobre o site, tais como:

Figura 3 - Informações do Wappalyzer no moodle.

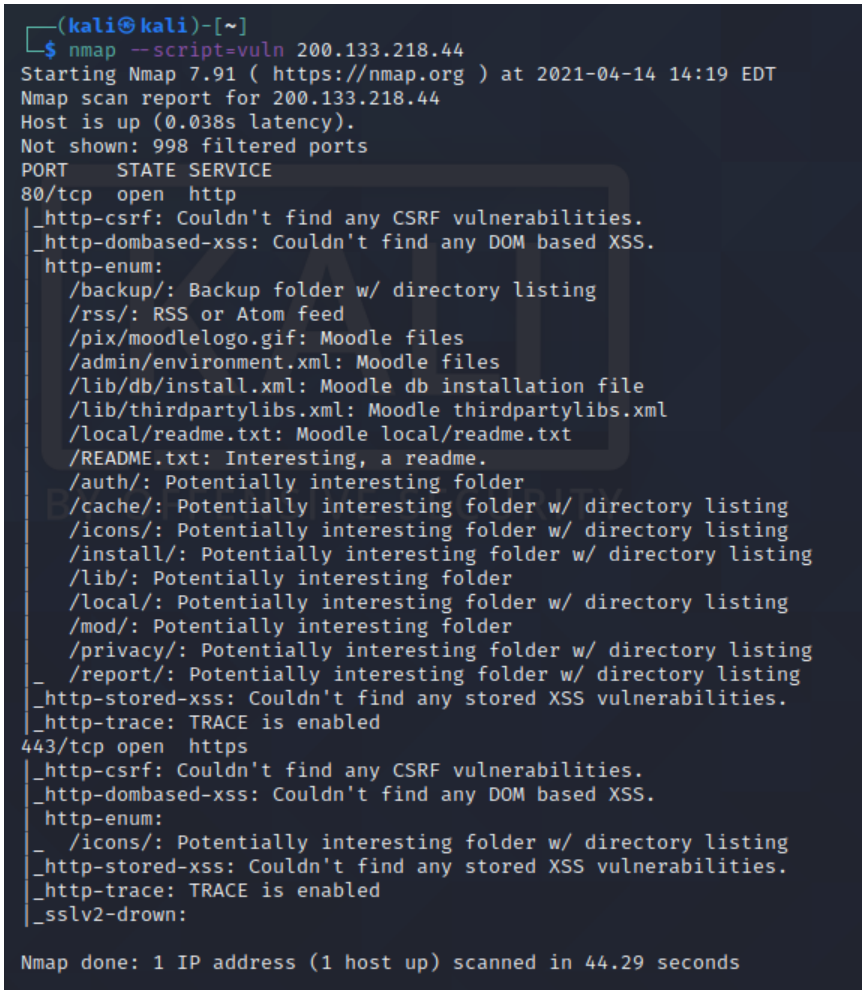
Fonte: Autor (2021)

Para o atacante, essas informações são muito valiosas para estudar e buscar uma vulnerabilidade. Analisando o retorno da ferramenta, nota-se diversas extensões, sistema operacional, entre outros componentes. Uma técnica muito comum é pesquisar se as versões estão atualizadas. Diversas versões de todos os tipos de softwares possuem falhas já expostas na rede de internet, permitindo que o invasor encontre uma possível vulnerabilidade sem muito esforço.

3.3 Nmap

O nmap é um utilitário gratuito e de código aberto para descoberta de rede e auditoria de segurança. Tal ferramenta realiza o escaneamento de portas, avaliando a segurança dos computadores pois permite descobrir serviços ou servidores de uma rede. É um dos softwares mais popular entre os especialistas, sendo indispensável para a coleta de informações. São diversos os comandos e possibilidades de escaneamento embutidos em tal programa. Essa ferramenta destaca-se por ser flexível, conseguindo mapear redes cheias de filtros IP, firewall, roteadores e outros obstáculos. Além de características de ser poderoso, funciona em diversos sistemas operacionais, gratuito e de excelente documentação para os usuários.

Figura 4 – Escaneamento de vulnerabilidades com nmap.



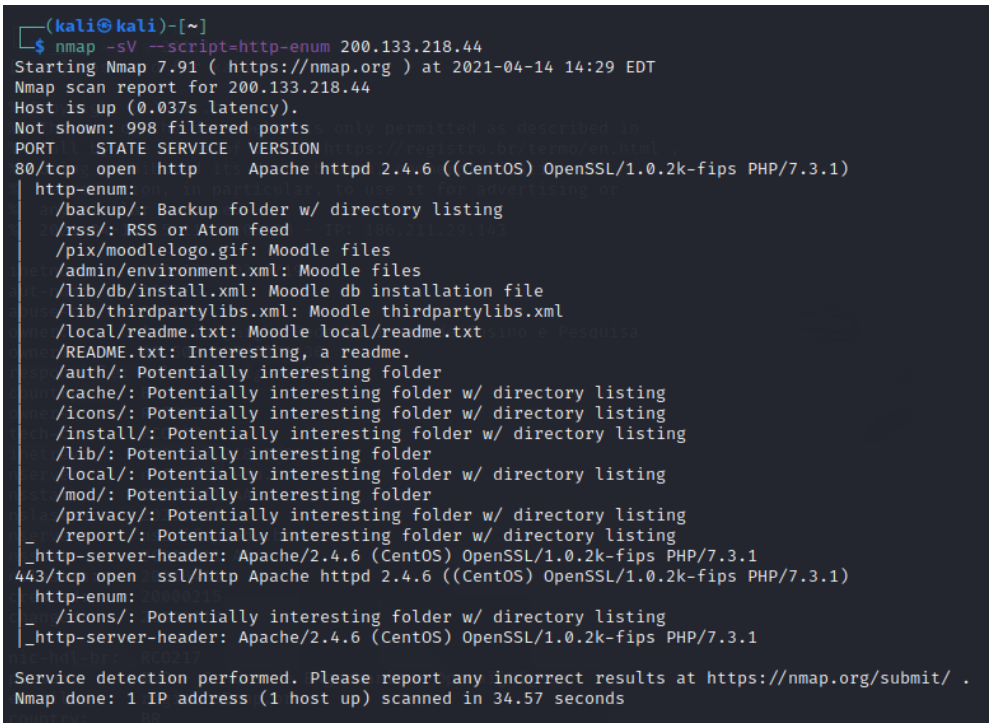
Fonte: Autor (2021)

A ferramenta teve como retorno a informação que a porta 80, conhecida como serviço web está aberta, além de 998 portas filtradas. Softwares de

segurança e determinadas configurações permitem que algumas portas sejam filtradas para que as informações não fiquem tão amostra para o público.

O http-enum é responsável por enumerar diretórios usados por aplicativos e servidores da Web populares. Semelhante ao scanner de aplicações web “Nikto”, porém essa ferramenta também identifica versões específicas.

Figura 5 – Nmap executando script http-enum.

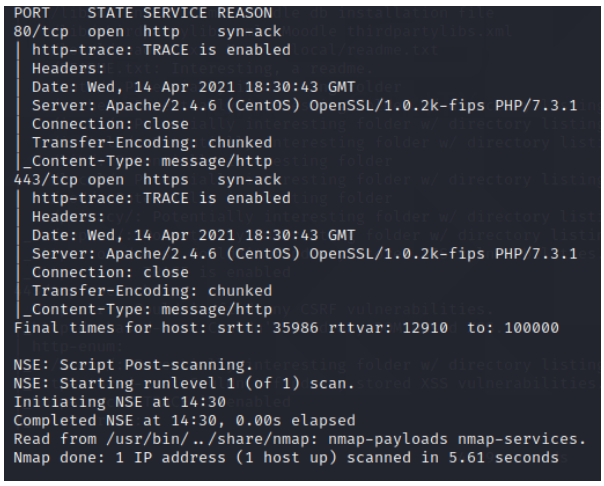


Fonte: Autor (2021)

A execução da ferramenta trouxe como resultado informações de possíveis diretórios do site alvo. Também é possível analisar as versões do Apache e do sistema operacional desse servidor.

Envia uma solicitação e mostra se o método trace está habilitado. Se a depuração estiver ativada, ele retornará os campos de cabeçalho que foram modificados na resposta.

Figura 6 – Nmap executando script http-trace.



Fonte: Autor (2021)

3.4 Nikto

É uma ferramenta de escaneamento de vulnerabilidades de software livre. Comumente utilizado para escanear servidores web, procurando programas desatualizados e arquivos perigosos. Tal software pode ser encontrado já instalado no Kali Linux.

Figura 7 – Retorno da ferramenta Nikto.

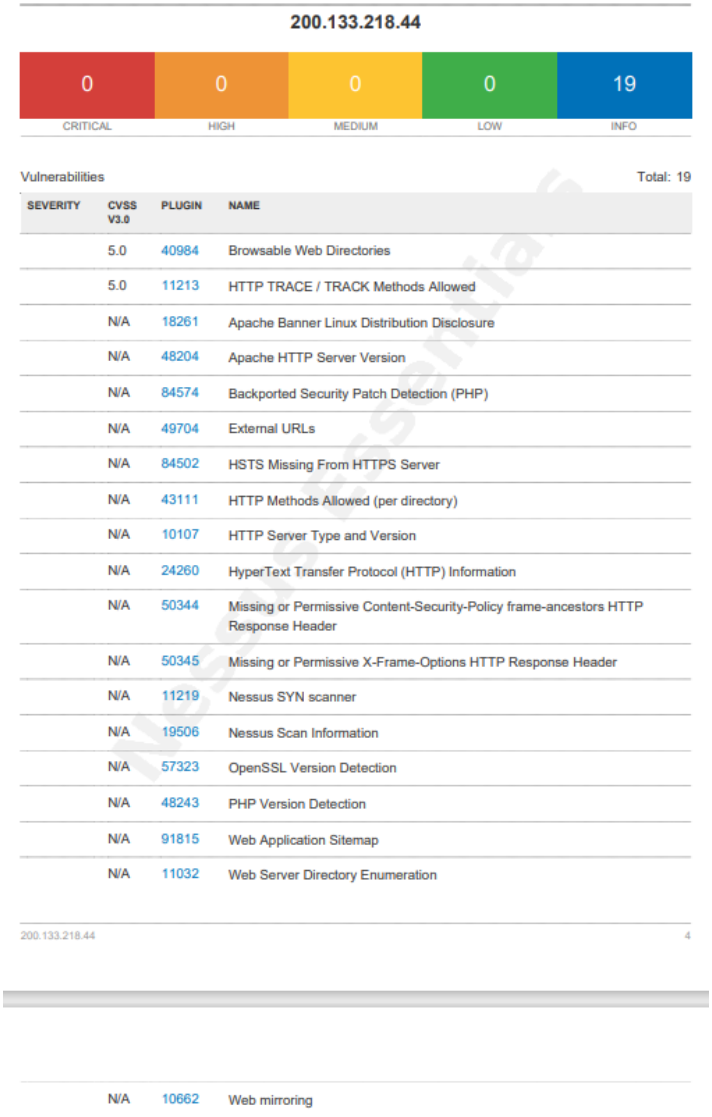
Fonte: Autor (2021)

A ferramenta foi capaz de retornar informações sobre a versão do Apache, OpenSSL e PHP apenas passando o endereço da url.

3.5 Nessus

Software de auditoria comumente utilizado para análise de vulnerabilidades, capaz de efetuar varredura de portas e detecção de servidores ativos. Esse programa conta com uma versão gratuita com algumas restrições ao usuário e uma versão paga, a qual não tem restrição das funcionalidades.

Figura 8 – Escaneamento com Nessus.



200.133.218.44

0	0	0	0	19
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 19

SEVERITY	CVSS V3.0	PLUGIN	NAME
	5.0	40984	Browsable Web Directories
	5.0	11213	HTTP TRACE / TRACK Methods Allowed
	N/A	18261	Apache Banner Linux Distribution Disclosure
	N/A	48204	Apache HTTP Server Version
	N/A	84574	Backported Security Patch Detection (PHP)
	N/A	49704	External URLs
	N/A	84502	HSTS Missing From HTTPS Server
	N/A	43111	HTTP Methods Allowed (per directory)
	N/A	10107	HTTP Server Type and Version
	N/A	24260	HyperText Transfer Protocol (HTTP) Information
	N/A	50344	Missing or Permissive Content-Security-Policy frame-ancestors HTTP Response Header
	N/A	50345	Missing or Permissive X-Frame-Options HTTP Response Header
	N/A	11219	Nessus SYN scanner
	N/A	19506	Nessus Scan Information
	N/A	57323	OpenSSL Version Detection
	N/A	48243	PHP Version Detection
	N/A	91815	Web Application Sitemap
	N/A	11032	Web Server Directory Enumeration

200.133.218.44 4

	N/A	10662	Web mirroring

Fonte: Autor (2021)

Analisado o retorno da ferramenta, vulnerabilidades consideradas como crítica, alta, média e baixa não foram encontradas no escaneamento. Foram

apresentadas apenas 19 informações as quais podem servir de base para estudo e posteriormente encontrar uma possível falha.

Juntando todas informações encontradas no escaneamento do nmap, nikto e nessus, as ferramentas não conseguiram retornar nenhuma vulnerabilidade crítica, porém isso não significa que a aplicação esteja isenta de falhas. Como não foram encontradas vulnerabilidades críticas no sistema real, foi configurado um ambiente de testes (DVWA) para desenvolver as técnicas de invasão.

3.6 DVWA

O Damn Vulnerable Web Application (DVWA) é um aplicativo web desenvolvido em PHP e MySQL com diversas vulnerabilidades propositais em seu código para que entusiastas da segurança da informação possam treinar e desenvolver suas técnicas de invasões. Visto o cenário em que as ferramentas de escaneamento não identificaram vulnerabilidades presentes na aplicação, algumas específicas serão demonstradas no DVWA para fins didáticos em ambiente seguro de testes.

3.7 Cross-Site Scripting (XSS)

Vulnerabilidade citada entre as 10 mais famosas em aplicações web, ocorre quando um invasor envia código malicioso para um aplicativo da web, geralmente na forma de um script para chegar a um usuário final diferente. O atacante conta com diversas possibilidades como, redirecionar um usuário para um site malicioso, sequestro de sessão ou até mesmo ativar um key logger, técnica que monitora todas as teclas que o alvo digitou. Existem diversos tipos de vulnerabilidades XSS, aqui será apresentada três das mais comuns, DOM, Reflected, Stored.

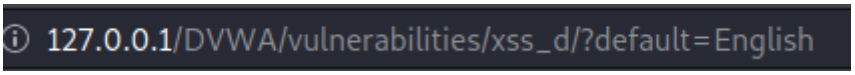
O DOM Cross-Site Scripting tem como objetivo inserir algum valor na página, injetando informação na camada view da aplicação. Como exemplo na ilustração abaixo que possui uma lista de opções de idiomas para selecionar.

Figura 9 – Opções de idioma na aplicação DVWA.

Fonte: Autor (2021)

Selecionado a opção de idioma “English”, a URL da aplicação utiliza da informação:

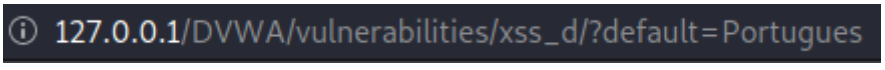
Figura 10 – URL da aplicação após seleção de idioma.



Fonte: Autor (2021)

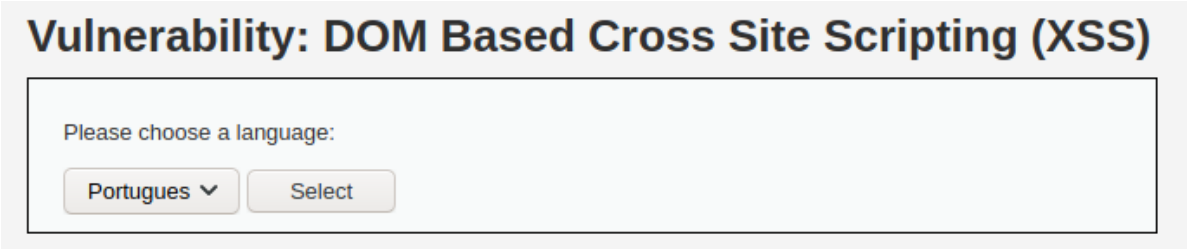
Analisando as opções, português não é um idioma que pertence a lista. Porém ao inserir manualmente essa opção na URL no lugar do “English” mais uma opção será apresentada.

Figura 11 – Alteração da URL usando “Português”



Fonte: Autor (2021)

Figura 12 – Opção “Portugues” disponível.

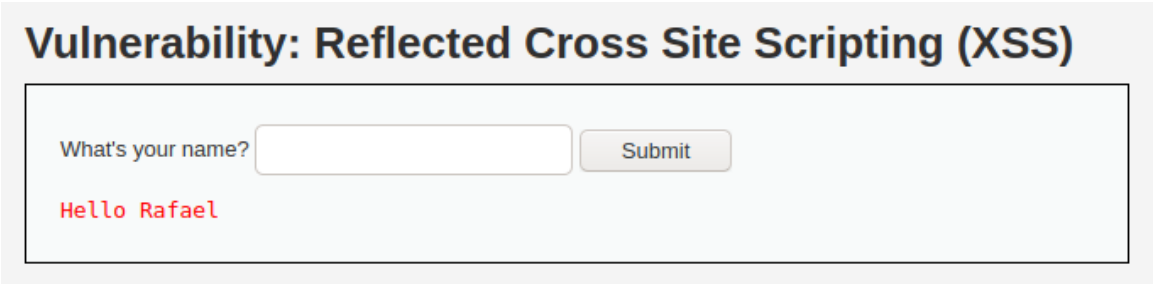


Fonte: Autor (2021)

Como resultado, a opção português foi inserida como um idioma o qual não existia nas opções originais da aplicação.

Reflected Cross-Site Scripting é um outro tipo de XSS, com objetivo de modificar a camada view porém com restrições, ao atualizar a página, as modificações serão desfeitas na aplicação. Este ataque para funcionar consiste na vítima estar disposta a interargir com a aplicação, clicando em um link. É possível fazer a inserção de código em Javascript no aplicação de modo que o atacante consiga efetuar um ataque denominado roubo de sessão para obter acesso a informações ou serviços de um sistema de forma não autorizada. Considere uma aplicação simples a qual pergunta o nome do usuário e diz olá demonstrada abaixo:

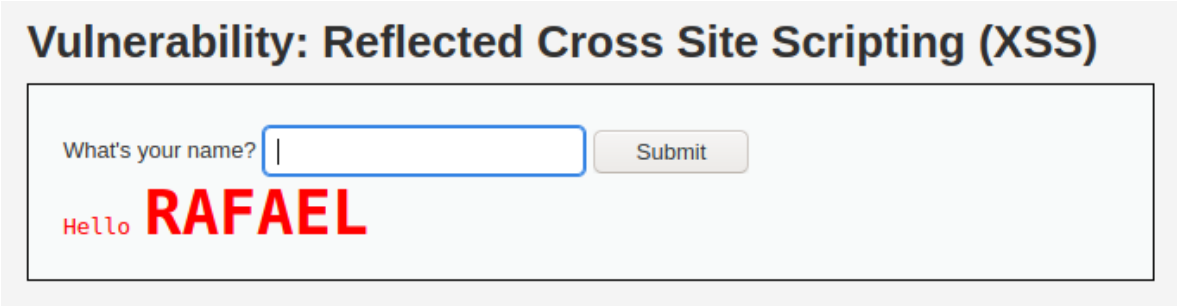
Figura 13 – Inserção de “Rafael” no campo de entrada.



Fonte: Autor (2021)

O ataque ocorre com a inserção de código malicioso no campo de entrada. Como exemplo a utilização de HTML para aumentar o tamanho da fonte e letras em caixa alta.

Figura 14 – Execução do Reflected Cross-Site Scripting.

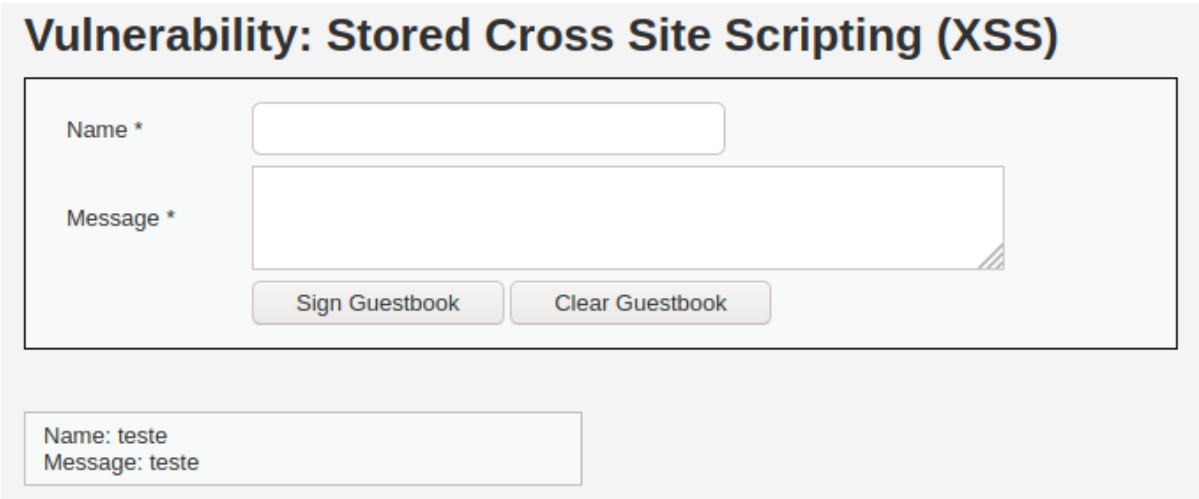


Fonte: Autor (2021)

A aplicação permitiu que o código inserido fosse executado, modificando o retorno padrão. Notada essa vulnerabilidade, é possível efetuar diversos ataques utilizando da inserção de código malicioso.

O Stored Cross-Site Scripting possui a mesma base do Reflected, porém com duração permanente, as alterações ficarão presentes mesmo ao atualizar a página. Nesse ataque, não é necessário a aplicar engenharia social no usuário, pois a aplicação será executada contra qualquer pessoa que acesse a página. Considere uma aplicação que pede o nome do usuário e uma mensagem qualquer como demonstrado abaixo:

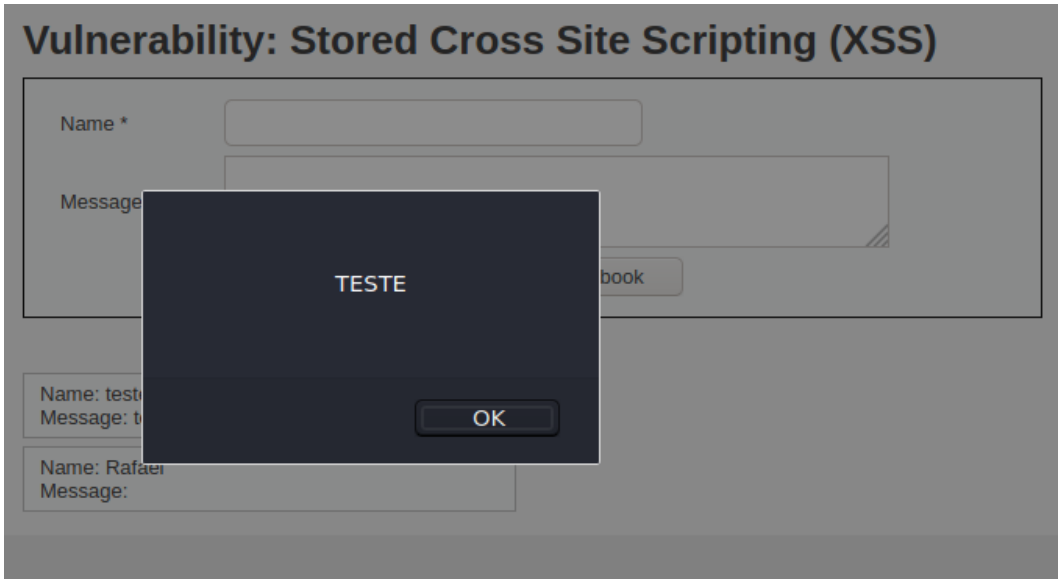
Figura 15 – Campos de entrada.



Fonte: Autor (2021)

Ao inserir os valores e enviar, a aplicação salva o nome, a mensagem e exibe as informações. Aproveitando da execução, scripts podem ser enviados, como exemplo um alerta em Javascript para que apareça na tela do usuário a palavra “teste”.

Figura 16 – Execução do Stored Cross-Site Scripting.



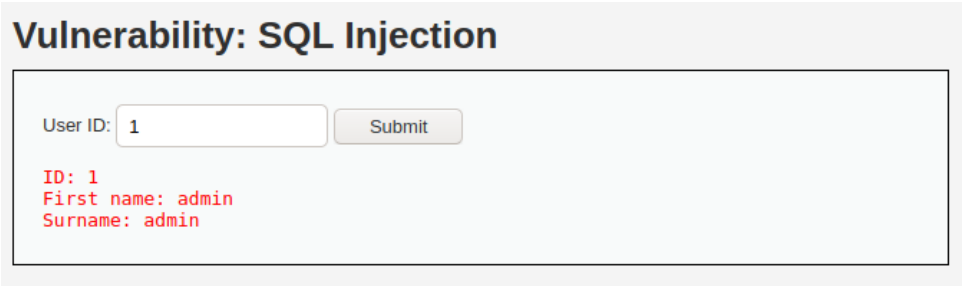
Fonte: Autor (2021)

Ao enviar o código malicioso e atualizar a página, a mensagem será exibida na tela de qualquer usuário que acesse a aplicação. São diversas as possibilidades de ataques reconhecendo que a aplicação aceita a injeção de códigos maliciosos em seu sistema.

3.8 SQL Injection

O SQL Injection é um dos ataques mais conhecidos, trata-se da exploração de falhas entre o sistema e o banco de dados com o qual ele está conectado. Esse tipo de ataque acontece quando o atacante insere códigos maliciosos em formato SQL dentro de uma query (consulta) através de algum campo de entrada de dados na aplicação.

Figura 17 - Parâmetro de entrada da aplicação.

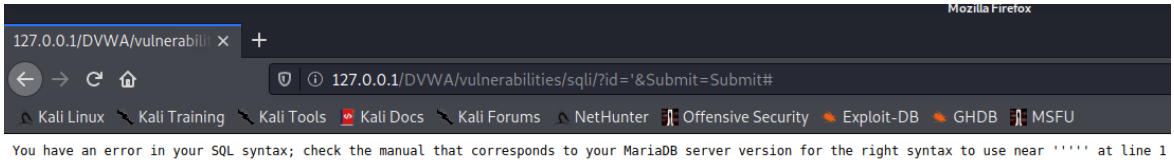


Fonte: Autor (2021)

Trata-se de uma aplicação que solicita ao usuário um ID. Após a inserção ele retorna informações como o nome e sobrenome da pessoa cadastrada com o ID específico.

Uma técnica comumente utilizada para verificar a possível existência desta vulnerabilidade, é inserir um caractere de aspas simples no final da URL do site.

Figura 18 – URL com aspa simples no final.



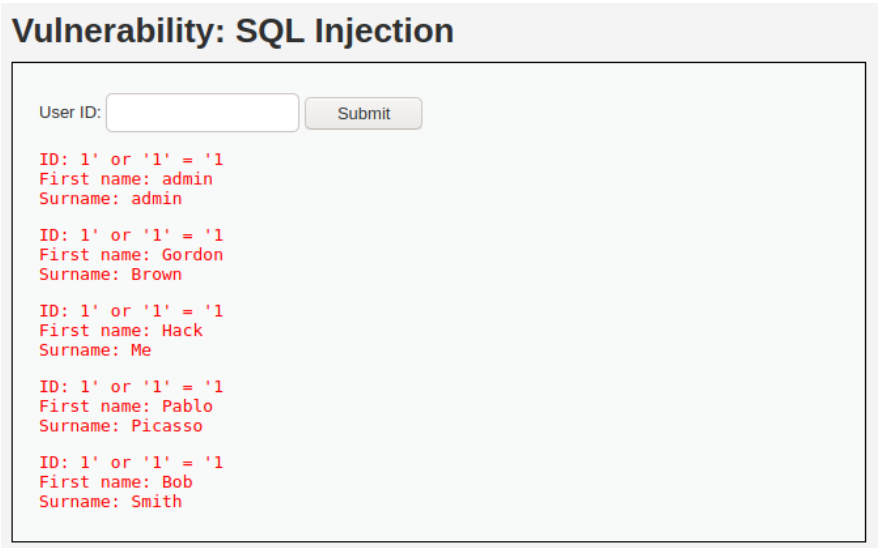
Fonte: Autor (2021)

Analisada a mensagem de erro informada, é possível que o sistema esteja vulnerável a ataques de SQL Injection. A partir dessa informação o atacante pode começar inserir códigos SQL no campo de entrada e verificar o retorno. Muitos usuários optam pela string:

1’ or ‘1’ = ‘1

Tal string é muito conhecida neste ataque, pois faz com que as condições de verificação da consulta retornem um valor verdadeiro, retornando informações sensíveis ao usuário.

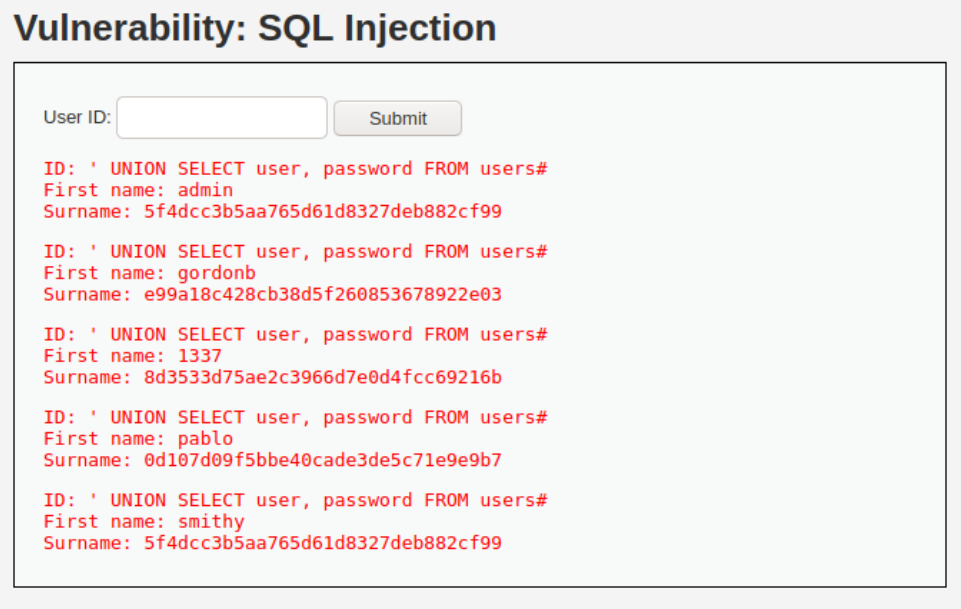
Figura 19 – Envio da string 1’ or ‘1’ = ‘1 ao servidor.



Fonte: Autor (2021)

A aplicação está sujeita a tal falha, com o envio da string foi possível analisar o retorno de múltiplos usuários cadastrados no banco de dados sem precisar saber o ID específico de cada um deles. O atacante tem como objetivo explorar a estrutura do banco de dados, testando comandos SQL e verificando os retornos até que consiga identificar informações sigilosas. Uma ferramenta muito utilizada para detecção e exploração é o SQL Map. Esta ferramenta permite construir um mapeamento da estrutura do banco de dados para que o atacante consiga elaborar comandos de forma assertiva no ataque.

Figura 20 – Acesso a informações sensíveis.



Fonte: Autor (2021)

Após um estudo da estrutura do banco de dados em questão e utilizando comandos SQL, foi possível identificar informações sensíveis, que podem comprometer a aplicação. O atacante obteve acesso ao nome e a senha em formato criptografado dos usuários. As senhas estão criptografadas em formato de hash, porém se o atacante descobrir qual o formato utilizado nessas hashes, existem programas que quebram essa criptografia e retornam o texto limpo, chegando na real senha dos usuários. Desta forma o sistema em questão foi comprometido, permitindo que invasores tenham acesso a informações salvas no banco de dados e possam seguir com diversos ataques.

3.9 Mantendo o acesso

Uma forma de manter acesso é o netcat, conhecido por ser um canivete suíço do protocolo TCP/IP devido suas múltiplas funções. Bruno Fraga (2019, p.220) define que esta ferramenta dá ao usuário permissão para atuar como cliente ou servidor, criando conexão direta ou indireta, funcionando como um backdoor. O bind e reverse shell são técnicas comumente utilizadas para obter e manter o acesso dentro do pentest.

Tiago Souza (2014) define como o bind shell explora a oportunidade de o alvo abrir um shell de comandos e o atacante analisa os dados da porta local com qual se conectou, fazendo com que os dois computadores fiquem conectados por meio da porta que o invasor está ouvindo. Uma porta é uma aplicação ou processo que serve de ponto final de comunicações em um sistema operacional, associada ao endereço de IP do usuário. Nesse ataque, a máquina do atacante atua como um cliente e a máquina do alvo como um servidor, esperando estabelecer uma comunicação entre eles. O invasor após estabelecer a conexão e ouvir a porta, insere comandos que serão remotamente executados na máquina alvo, porém possui restrições como a vítima utilizar de um IP público e com acesso a rede de internet.

Por outro lado, Tiago Souza (2014) também define o reverse shell possui a mesma base, porém a máquina do atacante torna-se o servidor com IP público e acesso à internet, enquanto a máquina alvo torna-se o cliente da conexão. Esse tipo de ataque é muito perigoso, o atacante por meio da execução de comandos remotos pode escalar seus privilégios a nível de sistema, podendo obter permissão para executar qualquer tipo de ação, como por exemplo instalar um malware na máquina alvo e impossibilita o alvo de utilizá-la. O invasor pode solicitar um “resgate”, solicitando para que a vítima contribua monetariamente para que o malware seja removido e a máquina volte a funcionar, um ataque muito famoso e recorrente em grandes empresas.

Caso o atacante consiga estabelecer esse acesso a máquina da vítima, ele se torna um hospedeiro podendo monitorar todas ações e informações salvas no computador sem que o alvo esteja ciente, podendo desenvolver uma série de ataques a uma pessoa específica, buscando por cartões, dados pessoais, etc. Dado o cenário, é extremamente importante que os softwares de antivírus e proteção de ameaças estejam sempre ativos e atualizados para tornar mais difícil a missão de um invasor conseguir acesso a aplicações.

4 RESULTADOS

Foram efetuados alguns testes e a utilização de ferramentas em busca de vulnerabilidades na plataforma Moodle. Não foi possível identificar nenhuma falha crítica, mas isso não significa que o sistema esteja totalmente seguro. Foram aplicados testes comuns em que programadores possuem em mente que devem se preocupar no desenvolvimento para não deixar brechas na aplicação. Para fins didáticos alguns ataques foram reproduzidos em ambiente seguro de testes.

5 CONCLUSÕES

A plataforma Moodle possui uma segurança mínima, baseado nos testes executados com as vulnerabilidades mais comuns em aplicações web da atualidade. É de conhecimento geral na segurança da informação que nenhum sistema está completamente seguro, sempre existe um ponto que pode ser crítico, podendo ser um fator tecnológico, físico ou até mesmo na interação entre pessoas. Existem inúmeros tipos de ataques virtuais, os quais alguns deles podem ser efetivos na plataforma e causar danos. Conclui-se que o sistema possui um determinado nível de segurança, mas não pode ser definido como muito seguro devido outros tipos de ataques não testados nesse projeto. Alguns tipos de ataques e algumas teorias foram apresentadas afim de demonstrar o processo de invasão e como um atacante age durante o processo. São inúmeras possibilidades de ataque, um conhecimento básico sobre redes de computadores e segurança da informação é extremamente importante para a população, afim de minimizar o número de vítimas as quais se tornam alvo de ataques cibernéticos. Muitas informações sobre segurança são divulgadas na rede de internet, desde a orientação de criar senhas complexas e altera-las depois de um período até o entendimento mais afundo da estrutura dos sistemas e como podem ser burlados.

REFERÊNCIAS

DVWA disponível em: https://dvwa.co.uk/

FRAGA, B. **Técnicas de Invasão: Aprenda as técnicas usadas por hackers em invasões reais**, 1ª edição, Labrador, 2019.

Kali Linux disponível em: https://www.virtualbox.org/

MAUÉS, R. **Diferentes tipos de pentests.** Acesso em: 26 de janeiro de 2022. Disponível em: https://blog.convisoappsec.com/diferentes-tipos-de-pentests/

Moodle disponível em: https://www.moodle.org/

Nessus disponível em: https://www.tenable.com/downloads/nessus?loginAttempted=true

Nmap disponível em: https://nmap.org/

OWASP TOP 10 VULNERABILITIES, Acesso em: 20 de abril de 2021. Disponível em https://snyk.io/learn/owasp-top-10-vulnerabilities/

SOUZA, T. **Reverse shell e cheat sheet**. Acesso em: 28 de janeiro de 2022. Disponível em: https://tiagosouza.com/reverse-shell-cheat-sheet-bind-shell/

WALKER, M. **CEH Certified Ethical Hacker All-In-One Exam Guide**, 4.Ed. Local: editora, 2019.

Wappalyzer disponível em: https://chrome.google.com/webstore/detail/wappalyzer/gppongmhjkpfnbhagpmjfkannfbllamg?hl=pt-BR

Whois disponível em: https://registro.br/