Pentest e Recomendações para Sistemas Web da Administração PúblicaAutor: Suene Bezerra Leite

Orientador: Primeiro-Tenente Bruno Juventino

[redacted]

RESUMO

Esse artigo tem como finalidade demonstrar como a Controladoria-Geral da União, CGU, pode auxiliar outros órgãos da Administração Pública Federal, APF, na Guerra Cibernética, tanto destacando vulnerabilidades de Sistemas Web por meio de um Pentest guiado por ferramentas gratuitas, quanto com recomendações para que as aplicações já sejam desenvolvidas com boas práticas de segurança.

Palavras-chave: Pentest Web, Guerra Cibernética APF

ABSTRACT

This article aims to demonstrate how the Controladoria-Geral da União, CGU, can assist other agencies of the Federal Public Administration, APF, in the Cyber War, both highlighting vulnerabilities of Web Systems through a Pentest guided by free tools, as well as recommendations so that applications are already developed with good security practices.

Keyword: Pentest Web, APF Cyber War

1. INTRODUÇÃO

Grande parte da Administração Pública Federal, APF, é responsável por armazenar, manipular e custodiar informações críticas, por meio de seus Sistemas de Informação, que apesar de serem de interesse público, devem ser salvaguardadas de acessos maliciosos, os quais podem resultar, por exemplo, em fraudes, espionagem, vazamento de informações, indisponibilidade e danos à imagem da APF.

O Quadro 1, demonstrado abaixo, exemplifica informações críticas custodiadas, por dois órgãos federais da APF, a título de exemplo.

Quadro 1 - Exemplos de informações críticas custodiadas pela APF

Órgão da APF	Informações Críticas sob a guarda do Órgão da APF
Controladoria-Geral da União	Informações relacionadas a acordos de leniência com empresas investigadas por atos lesivos para a APF.
Ministério da Ciência, Tecnologia e Inovações	Registros referentes à coordenação de ações de controle de transferências, importação e exportação, de bens

	sensíveis e serviços diretamente vinculados a tais bens nas áreas nuclear, química, biológica e de mísseis; Informações sobre resultados financeiros das empresas que se utilizam das políticas de incentivos fiscais para a inovação, o desenvolvimento e a capacitação tecnológica no setor de tecnologias da informação e comunicação.

Fonte: Próprio autor(2020).

A obrigação de proteger informações sensíveis de posse do poder público é pautada em determinações normativas, tal como no Decreto nº 9.637, de 26 de dezembro de 2018, que institui a Política Nacional de Segurança da Informação. Essa norma estabelece responsabilidades a diversos órgãos da APF, como para a Controladoria-Geral da União, em seu art. 14, reproduzido a seguir:

> Art. 14. Ao Ministério da Transparência e Controladoria-Geral da União compete auditar a execução das ações da Política Nacional de Segurança da Informação de responsabilidade dos órgãos e das entidades da administração pública federal.

Importante, ainda, destacar que o Decreto nº 10.222, de 5 de fevereiro de 2020, o qual aprova a Estratégia Nacional de Segurança Cibernética, determina entre os objetivos Estratégicos da APF, o fortalecimento da segurança cibernética, tal como transcrito de seu anexo, a seguir:

> 2.2. OBJETIVOS ESTRATÉGICOS
>
> No intuito de atender à visão proposta, na concepção dos objetivos estratégicos foram considerados os parâmetros estabelecidos na Política Nacional de Segurança da Informação: o estágio de maturidade e as necessidades do País em segurança cibernética e os aspectos relativos ao ecossistema digital, no âmbito nacional e internacional.
>
> Desse modo, estes objetivos estratégicos visam a nortear as ações estratégicas do País em segurança cibernética, e representam macrodiretrizes basilares para que o setor público, o setor produtivo e a sociedade possam usufruir de um espaço cibernético resiliente, confiável, inclusivo e seguro. São os objetivos estratégicos:
>
> 1. Tornar o Brasil mais próspero e confiável no ambiente digital;
>
> 2. Aumentar a resiliência brasileira às ameaças cibernéticas; e
>
> 3. Fortalecer a atuação brasileira em segurança cibernética no cenário internacional.

Para que se cumpram as normas que determinam à APF a vigilância sobre a sua segurança cibernética, as vulnerabilidades de sistemas de informação da APF devem ser constantemente avaliadas. A exposição

controlada dessas eventuais vulnerabilidades tem por objetivo conhecer e mitigar os riscos, viabilizando a prevalência do interesse público.

Atualmente, os sistemas web do setor público são testados para validar seu atendimento a pontos de função pré estabelecidos. Tais testes verificam entradas, saídas, mensagens e redirecionamentos. A segurança de um sistema é um requisito não funcional importantíssimo, porém muitas vezes negligenciado por falta de conhecimento do gestor público.

É competêcia da CGU, não só auditar, mas também prover orientação normativa e supervisão técnica às unidades de auditoria interna governamental. Esse artigo visa a demonstrar como a CGU pode auxiliar os órgãos da APF a proteger seus sistemas Web na guerra cibernética, de duas formas. A primeira, consiste em guiar o órgão em um pestest por meio de passos pré-definidos e ferramentas gratuitas. A segunda forma possui formato de recomendações e visa a mitigar falhas de segurança já durante a confecção do sistema. Para viabilizar a demonstração, foi utilizado um sistema real Web em fase de implmentação, da APF, e aplicados os conhecimentos adquiridos na especialização de Guerra Cibernética, ofertada pelo Exército Brasileiro no ano de 2020. Por não haver necessidade de exposição, o nome da instituição, dona do sistema Web, bem como a URL e o IP do sistema serão preservados.

1. METODOLOGIA E ESCOPO

Penetration Testing Execution Standard, PTES, é a metodologia escolhida como guia para o Pentest em questão, por ser didático, com passos pré-definidos e bem separados. O Quadro 2, manifesta as fases da metodologia seguida, bem como uma breve descrição sobre cada uma delas.

Quadro 2 – Fases do PTES

Fase	Ações
Coleta de Informações	Entender o alvo ao máximo e coletar informações sobre todo o ambiente que afeta o sistema Web.
Varredura e Enumeração	Identificar portas e serviços ativos nos servidores envolvidos na aplicação; Identificar tecnologias utilizadas;
Análise de Vulnerabilidades	Elencar pontos de entrada no ambiente e tipos de ataques mais viáveis.
Exploração e Ataque	Utilizar técnicas que explorem as vulnerabilidades.
Pós-exploração	Valorar o nível de comprometimento do sistema , bem como analisar causas e consequências para a existência das vulnerabilidades exploradas.
Documentação	Documentar e sugerir recomendações.

Fonte: Próprio autor(2020).

O **OWASP** (Open Web Application Security Project), ou Projeto Aberto de Segurança em Aplicações Web, é uma comunidade, sem fins lucrativos, a qual visa a disponibilizar de forma gratuita, documentação, e tecnologias relacionadas à segurança de aplicações Web.

Dois dos principais documentos do OWASP utilizados foram:

- Manual para testes de segurança em sistemas Web, o qual expõe informações relevantes a serem observadas durante o desenvolvimento de um sistema Web;
- Lista de dos dez maiores riscos em sistemas Web , a fim de esolher os tipos de ataques contra a aplicação.
- Em relação ao escopo do Teste de Invasão destaca-se O resultado do Pentest em questão limita-se à applicações Web;
- A execução do Pentest foi feito na forma Caixa Preta, porém deve e pode evoluir para um Teste de Caixa Cinza, pois os sistemas da APF não são aleatórios e totalmente desconhecidos, tendo, muitas vezes, suas configurações replicadas em vários órgãos e entidades da APF.
- O Pentest em questão foi executado, externamente, a partir da Internet. Porém em uma situação real, testes executados dentro do Órgão, a partir de sua rede interna aumentam a precisão dos testes.

2. DESENVOLVIMENTO

3.1 - Coleta de Informações

3.1.1 - Shodan

Shodan é um site que busca informações sobre quaisquer dispositivos conectados à Internet. Como é necessário estar logado para fazer pesquisas no Shodan, foi criado um e-mail temporário para os testes.

A única informação conhecida do sistema Web é sua URL. Ao pesquisar a URL no Shodan foram encontrados dois IPs e a localização em Brasília/DF. Não foi encontrada nenhuma informação a respeito do servidor Web.

3.1.2 - Listagem de diretórios

Utilizando os IPs encontrados, foram executadas as ferramentas Dirb e Gobuster, as quais buscam por diretórios de nomes convencionais. Em ambas as ferramentas, a wordlist de diretórios usada foi a dirb/big.txt.

O Gobuster foi constantemente desconectado. Seu parâmetro user-agent foi configurado para algo desconhecido. Já o Dirb não encontrou

nenhum diretório significativo. O arquivo robots.txt foi editado e constatado que está vazio.

3.1.3- WhatWeb

A ferramenta foi utilizada para encontrar informações sobre o que está sendo executado no servidor Web.

Foi retornado o IP, que corroborou com o mesmo encontrado na fase de Coleta de informações, além da informação de suporte do servidor ao HTML 5. Há ainda a indicação de que o servidor web comunica-se com clientes apenas via HTTPs.

3.1.4 - Curl

Curl é a abreviação de Client URL e consiste em uma feramenta de interação com servidores HTTP.

Ao executar o comando "curl –k Url", a aplicação respondeu com uma página contendo nomes de rotinas Javascript, entre outras informações irrelevantes.

3.2 - Varredura e Enumeração

3.2.1 - SQLMap

O SQLMap permite descobrir informações sobre a base de dados de uma aplicação Web. Seus parâmetros foram ajustados para um teste às cegas. A ferramenta não retornou nenhuma informação útil.

Os seguintes comandos foram utilizados: sqlmap -u IP --current-–forms ; sqlmap -u IP --tables –forms e sqlmap -u IP --forms --dump –T

3.2.2 - Nmap

O Nmap é uma ferramenta de escaneamento, utilizada, principalmente, para descobrir portas, serviços e servidores em uma rede

A execução da ferramenta Nmap com o IP encontrado, trouxe duas portas: 80 e 443. Não foi encontrado o banner do servidor web, ou seja, sua versão e nome. Muito provavelmente devido à existência do proxy relatado no parágrafo a seguir.

Descobriu-se que há o proxy HAProxy, uma solução gratuita, fornecido com distribuições Linux, que oferece alta disponibilidade e balanceamento de carga. A versão do HAProxy informada pelo Nmap foi a 1.3.1.

O CVE-2019-11323 aponta uma falha nessa versão de serviço que basicamente consiste em ao se tentar substituir uma chave criptográfica existente por uma nova, ocorre uma retomada para a chave de criptografia anterior. A solução é atualizar para a versão 1.9, a qual não possui o erro.

Outros achados úteis do Nmap foram relacionados ao Sistema Operacional, Linux 4.0, o que vai ao encontro do HAProxy integrar distribuições Linux, além dos métodos suportados: Get,Head, Post e Options.

A fim de tentar obter mais informações, apesar do proxy, foram executados diversos scripts Nmap, conforme o Quadro 3.

Quadro 3 – Comandos Nmap utilizados no pentest

Objetivo	Comando
Busca por possíveis usuários.	nmap -p 80 --script http-userdir-enum IPAlvo
Busca pelo banner.	nmap -sV -p 80,443 2 IPAlvo --script banner
Busca por páginas de autenticação.	nmap IPAlvo -p80,443 --script http-auth-finder
Enumeração de diretórios.	nmap IPAlvo -p 80,443 --script http-enum
Busca por arquivos de backup no site.	nmap IPAlvo -p 80,443 --script http-backup-finder
Busca por portas filtradas por firewall.	nmap IPAlvo --script firewalk --traceroute
Busca pelo framework da aplicação.	nmap IPAlvo -p 80,443 --script http-devframework
Verifica se o site está protegido com algum tipo de WAF (Web Application.	nmap IPAlvo -p 80,443 --script http-waf-detect
Verifica falhas em servidores utilizando o método de saída do tipo verbose –v.	nmap -sS -v -Pn -A --open --script=vuln IPAlvo
Tentativa de burlar o firewall por meio de fragmentação de pacotes que serão enviados para se conectar ao alvo.	nmap -f -sV -A IPAlvo
Tentativa de burlar o firewall por meio de varreduras do tipo SYN na rede alvo.	nmap -sS -sV -A IPAlvo
Tentativa de burlar o firewall sem enviar pacotes ICMP para o alvo.	nmap -Pn -sV -A IPAlvo
Tentativa de encontrar vulnerabilidades na rede do IPAlvo.	nmap -sS -v -Pn -A --open --script=vuln IPAlvo/24
Busca por falhas de DDos	nmap -sU -A -PN -n -pU:19,53,123,161 –script=ntp-monlist,dns-recursion,snmp-sysdescr IPAlvo

Fonte: Próprio autor(2020).

Com os comandos listados no Quadro 3, foram encontrados os seguintes resultados:

- HAProxy roda na porta 8080;
- Há realmente balanceamento de carga para esse servidor web, uma vez que a rota até ele ora passa por uma máquina com um nome, ora por outra máquina, com um nome diferente.

3.3 Análise de Vulnerabilidades

Figura 1- Top 10 ataques Web

Fonte:owasp.org/www-pdf-archive//OWASP_TOP_TEN_-_2017_Cincinnati.pdf

A cada três anos o OWASP publica uma lista contendo as 10 vulnerabilidades mais exploradas em aplicações Web no mundo. A lista mais atual, até o momento presente, é de 2017 e se encontra ilustrada na Figura 1. A lista é um parâmetro importante para a escolha dos ataques a serem efetuados, além de características individuais da aplicação.

Apesar de poucas informações sobre vulnerabilidades terem sido coletadas nas fases anteriores, há algumas caracteríticas do sistema que permitiram a escolha de alguns tipos de ataques, alguns contidos na lista top 10 do OWASP:

- A página de login não possui captcha, nem múltiplos fatores, o que propicia uma ataque de força bruta nas credenciais.
- Há parâmetros Get na Url da tela inicial de login, o que permite ataques de SQLInjection e Poluição de Parâmetro HTTP.
- Há uso de javascript na aplicação, o que sugere ataque de Cross Script
- Há um ponto para upload de arquivos, oportuno para um ataque de Inclusão de Arquivo Local.

Todos os ataques relatados acima estão descritos no próximo tópico.

3.4 - Exploração e Ataque

3.4.1 - Força Bruta Credenciais

Foi tentado contra a aplicação a quebra da senha e login com as ferramentas Hydra e Medusa. Foi usada, em ambas as ferramentas, uma wordlist composta por palavras brasileiras, extraída do site GiTHub, na tentativa de quebra de senha.

Para o usuário foram tentadas as palavras sugeridas pelo OWASP, as quais são: "admin", "administrator", "root", "system", "guest", "operator", e "super". As variações em português também foram incluídas nos testes.

Não foi obtido sucesso na tentativas de força bruta. O login da aplicação é um e-mail, o que reduz bastante a eficácia de uma wordlist composta por senhas em forma de palavras.

Ainda que o ataque tenha sido frustrado, a aplicação não ofereceu nenhum tipo de resistência às tentativas.

Segundo a empresa de segurança norte-americana Praetorian, em um estudo realizado em 2015, há estatíscas que podem prever senhas. Por exemplo, caso sejam solicitados a usarem letras maiúsculas e números, a grande maioria dos usuários coloca a letra maiúscula como o primeiro caracter da senha, e ainda, dois dígitos no final da senha, que por sua vez, em grande parte, representa o ano de graduação. Há ainda a grande possibilidade de letras consecutivas formarem uma palavra. Esses padrões previstos em estatístcas permitem que invasores ganhem vantagem no acesso a senhas de usuários.

Há ainda ferramentas como o CeWL que podem destacar palavras com valores significativos em páginas Web pré determinadas, a fim de gerar dicionários específicos para uma empresa,devido a tendências dos funcionários em utilizar senhas que remetam ao tipo de trabalho.

3.4.1.1 Recomendações Preventivas

Fazer, de tempos em tempos, checagem da Seguranca das senhas dos usúarios do sistema. A ferramenta por linha de comando Pypal, escrita em Python, por exemplo, permite a geração de relatórios acerca dos pontos fracos na escolha de senhas dos usuários. A ferramenta detecta, por exemplo, padrões numéricos comuns e senhas compartilhadas.

Implementar um número equilibriado de tentativas de login mal sucedidos antes do bloqueio. O OWASP sugere que de 5 a 10 tentativas permitidas é um número razoável, de forma que usuários válidos não sejam bloqueados com uma grande frequência, nem os invasores tenham uma margem significativa de tentativas em força bruta.

O OWASP sugere ainda um tempo de bloqueio de 5 a 30 minutos como um meio-termo para conter ataques de força bruta, e não causar negação de serviço aos usuários legítimos. A forma de desbloqueio também deve ser segura o suficiente para que o invasor não consiga desbloquear contas de forma fácil. A OWASP sugere que o desbloqueio seja por meio de envio de um link único por e-mail ou conjuntos de perguntas e respostas secretas. O meio mais seguro de desbloqueio de senha, porém mais caro, é manualmente pelo administrador. A instituição deve encontrar, portanto, um equilíbrio no custo-benefíco para a forma de desbloqueio de logins.

Sempre que possível, implementar a autenticação multi-fator.

3.4.2 - SQL Injection

No tipo de ataque SQL Injection, ou Injeção de SQL, o atacante visa a extrair dados contidos na base de dados por meio de falhas na forma como as consultas à base de dados são montadas pela aplicação. Em testes às cegas, como é o caso, deve-se comparar os resultados dos valores modificados dos parâmetros com os valores não modificados.

Como a ferramenta SQLMap não acusou qual a Base de Dados da aplicação, usou-se na URL códigos de SQL Injection para quatro bancos diferentes: Oracle, PostgreSQL, SQLServer e MySQL. Os comandos foram usados após um um parâmetro do tipo Get, o qual compõe a URL do login. As injeções foram retiradas dos exemplos de teste recomendados pelo OWASP e estão mostradas no Quadro 4, abaixo.

Quadro 4 – Comandos SQLInjection para diversas bases de dados

Base de dados	SQL Injection
Postgresql	UNION ALL SELECT user,NULL,NULL--
Oracle	%0ASYS.PACKAGE.PROC
MySQL	AND 1=1 UNION SELECT DATABASE()
SQLServer	UNION%20ALL%201,1,'a',1,1,1%20FROM%20users;--

Fonte: Próprio autor(2020).

Em todos os casos, a página de login foi redirecionada para outra página, contendo uma mensagem de erro. Há indícios de que, ao menos na página de login, as consultas não são montadas em tempo de execução.

3.4.2.1 - Recomendações Preventivas

Consultas SQL não devem ser dinâmicas, ou seja, devem ser compostas de templates, de forma que os parâmetros, os quais são variáveis, entrem em posições esperadas e sejam tratados de acordo com seu tipo, como por

exemplo string, inteiro, float, etc., pois, ainda que injeções sejam tentadas, essas não alterarão a consulta.

Codificações de sql bruto devem ser evitadas, sendo recomendado que métodos sejam utilizados para a montagem das Strings SQL. A vantagem é utilizar módulos testados, os quais fazem tratamento de exceções. Exemplos de frameworks open-source para aplicações web são : Ruby on Rails e Django.

3.4.3 - Poluição de Parâmetro HTTP

Outro tipo de injeção tentada contra a aplicação foi a replicação de parâmetros utilizados na URL. Como na enumeração não foi possível descobrir qual o servidor da aplicação, repetiu-se o último parâmetro GET, de forma a descobrir o servidor da aplicação com base no comportamento conhecido de servidores mais utilizados, conforme Quadro 5.

Quadro 5 – Comportamento de servidors diante de Gets repetidos

Servidor	Comportamento
Apache-PHP	Utiliza o valor do último parâmetro Get repetido.
Apache-Tomcat	Utiliza o valor do primeiro parâmetro Get repetido.
IIS Aspx	Utiliza todos os valores dos parâmetros Get repetido.

Fonte: Próprio autor(2020).

Devido ao fato da aplicação não ter tido nenhum tipo de comportamento diferente após a tentativa de execução com um parâmetro Get repetido, e adicionado após o valor do parâmetro Get original, pode-se supor que o servidor da aplicação é um Apache Tomcat.

3.4.3.1 - Recomendação Preventiva

Valores de parâmetros Get repetidos devem ser tratados como erro de execução, e portanto, devem gerar mensagens de erro para o usuário.

3.4.4 - Cross-site Scripting (XSS)

Esse tipo de ataque consiste em fazer com que a aplicação execute código javascript malicioso. O código javascript pode ser injetado tanto na URL, quanto em campos de formulário do tipo texto.

Mais uma vez o comportamento da aplicação foi, após a tentativa da execução do código javascript, redirecionar a página de login para uma página contendo uma mensagem de erro .

Pode-se inferir que a aplicação Web, em questão, aplica tratamento a determinados tipos de caracteres de forma a não permitir a execução de java script indesejado.

3.4.4.1 - Recomendações Preventivas

Recomenda-se não permitir que a aplicação receba, sem nenhum tipo de tratamento, seja em sua URL, seja em seus campos de formulário, os caracteres, ou suas respectivas codificações url, conforme Quadro 6.

Quadro 6 – Caracteres que permitem XSS

Caractere	Codificação URL
" (aspas duplas)	&quot &#34
` (aspa simples)	&apas &#39
< (menor que)	&lt &#60
> (maior que)	&gt &#61

Fonte: Próprio autor(2020).

Recomenda-se incluir a flag **HttpOnly** para os cookies. Tal flag não permite que os cookies sejam acessados, no lado cliente, por scripts.

3.4.5 - Inclusão Local de Arquivos

A Inclusão Local de Arquivos permite que o atacante inclua códigos maliciosos nos uploads que a aplicação permite. A aplicação aceita arquivos PDF. Foi gerado, então, um código malicioso de shell reverso em formato jsp, pressupondo-se que o servidor é um Apache TomCat. Após o upload o arquivo foi renomeado para um nome padrão, seguido da extensão pdf e não foi possível enumerar sua localização. Portanto, a aplicação já executa as boas práticas referentes ao upload do arquivo, não sendo possível concluir o ataque do tópico em questão.

3.4.6 Pós Exploratório e Documentação

Após o término do pentest, uma avaliação sobre a resistência da aplicação à ataques deve ser fornecida, juntamente com recomendações..

O OWASP possui uma metodologia baseada no cálculo do risco por meio de pesos atribuídos a três fatores para cada falha: prevalência, detecção e facilidade de abuso. Os pesos correspondem a fatores de impacto e variam de 1 – Baixo a 3- Alto. A proposta da OWASP é disponiblizar uma tabela genérica, de riscos para cada tipo de ataque, porém cada organização deve atribuir pesos de agentes de ameaça específicos do negócio.

Para o caso analisado, foram testados 3 ataques com impacto técnico grave, segundo o OWASP (SQLInjection, Poluição HTML e Força Bruta) e um

ataque moderado (Cross-Site Scripting). A aplicação se saiu em todos os testes, não se deixando corromper, porém várias recomendações foram feitas. Cabe, então, à instituição ponderar os riscos que deseja assumir.

Os passos efetuados no pentest, bem como as vulnerabilidades encontradas e as recomendações cabíveis, devem ser fornecidas à entidade dona do sistema avaliado em forma de documentação de fácil entendimento. A documentação deve conter essencialmente: o número do processo contendo a formalização da autorização para o pentest, bem como seu escopo, a identificação das partes envolvidas, as Vulnerabilidades encontradas e recomendações.

4 - CONCLUSÃO

A confecção do artigo demonstrou que os conhecimentos adquiridos sobre segurança podem realmente auxiliar a Administração Pública a efetuar sua defesa cibernética. A CGU pode desempenhar um importante papel auxiliando as instituições públicas a perceberem que a segurança de suas aplicações Web merecem tanta importância quando as funcionalidades desempenhadas por tais aplicações.

Podem existir barreiras iniciais relacionadas à exposição de vulnerabilidades das aplicações pelas próprias instituições, pois cada sistema tem uma quantia pecuniária aquisição associada a ele, porém cabe ao pentester demonstrar respeito à institiução, bem como transmitir confiança e respeitar a confidencialidade acerca das informações e resultados oriundos do pentest, pois a finalidade maior é disseminar a percepção de que os sistemas de informação da APF compõem uma importante parcela do patrimonio público, e merecem, portanto, atenção especial.

BIBLIOGRAFIA

MORENO, Daniel. **Pentest em Aplicações Web**. São Paulo: Novatec, 2017

STUTTARD, Dafydd & PINTO, Marcus. **The Web Application Hacker's Handbook**. Wiley: Indianapolis,2001.

WEIDMAN, Georgia. **Testes de Invasao**. São Paulo: Novatec, 2014.

YAWORSKI, Peter. **Real-World Bug Hunting: A Field Guide to Web Hacking**. No Starch Press: San Francisco , 2019.