A importância do Teste de Intrusão: um estudo de caso em um sistema acadêmico

Luciano Pedreira de Souza - lpedreira@gmail.com
Computação Forense e Perícia Digital
Instituto de Pós-Graduação - IPOG
Salvador, BA, 21 de Agosto de 2019

Resumo
Com o advento da globalização e a expansão da internet, as organizações tiveram de adequar os seus negócios migrando serviços para essa nova realidade, e consequentemente disponibilizando aplicações ou sistemas para acesso do seu público. Este artigo tem como objetivo explanar a importância da realização do Teste de Intrusão, tipicamente conhecido como PenTest, em aplicações e sistemas antes que os mesmos sejam submetidos a uso do público, evidenciando através de um estudo de caso algumas técnicas utilizadas visando explorar vulnerabilidades existentes na codificação de páginas, resultando em acessos não previstos e obtenção de informações muitas vezes sigilosas.

***Palavras-chave**: Teste de Intrusão. Pentest. Segurança da Informação. Vulnerabilidade. Segurança Ofensiva.*

Abstract
With the advent of globalization and the expansion of the internet, organizations had to adapt their business by migrating services to this new reality, and consequently making available applications or systems for access by their public. This article aims to explain the importance of performing the Penetration Test, typically known as PenTest, in applications and systems before they are subject to public use, highlighting through a case study some techniques used to exploit existing vulnerabilities in the Internet. coding pages, resulting in unanticipated access and often obtaining sensitive information.

***Keywords:** Penetration Test. Pentest. Security Information. Vulnerability. Ofensive Security.*

1. Introdução

Desde o advento da *World Wide Web*, mais conhecida pelos seus usuários como *www* ou simplesmente *web*, inúmeras aplicações e sistemas são cotidianamente publicadas para acesso ao público. A *web*, em seu projeto original visava apenas o compartilhamento de dados por meio de hipertextos, onde sua concepção não garantia a segurança no meio digital. Ao longo do tempo, novas funcionalidades tiveram de ser implementadas objetivando que os dados fossem trafegados de forma mais segura. Entretanto, através de fragilidades existentes nos códigos fontes das páginas ou sistemas, a qual denominamos de vulnerabilidades, permitem que usuários mal intencionados consigam acessos ou desvios de fluxos de controle, obtendo informações as quais não deveriam, podendo levar ao comprometimento de alguns dos aspectos da tríade que se baseia a segurança da informação, que são: a confidencialidade, a integridade e a disponibilidade.

Quanto a esse entendimento, é abordado por CABRAL e CAPRINO (2015:139) que o desenvolvimento de software envolve uma série de atividades que exigem um método estruturado e um processo bem definido de validação e acompanhamento, onde com o

aumento das vulnerabilidades de segurança, se faz necessário capacitar e conscientizar os desenvolvedores e todos os envolvidos em práticas de segurança de software.

Visando validar se as funcionalidades as quais a aplicação ou sistema fora projetada se encontram de acordo com um nível de segurança, testes de verificação de conformidade com os requisitos de negócio são feitos antes da concessão do acesso ao seu ambiente de produção. Porém esses testes apenas não são suficientes para prever todas as possibilidades e variáveis existentes em uma determinada aplicação ou sistema, quer seja a parte relacionada a codificação, quer seja os componentes que assistem ao servidor e serviços de hospedagem.

Nesse sentido, discorre Weidman, senão vejamos:

> Testes de intrusão, teste de invasão ou *pentesting* (não confundir com testes de caneta esferográfica ou de canetas-tinteiro) envolvem a simulação de ataques reais para avaliar os riscos associados a potenciais brechas de segurança. Em um teste de invasão (em oposição a uma avaliação de vulnerabilidades) os *Pentesters* não só identificam vulnerabilidades que podeiam ser usadas pelos invasores, mas também exploram essas vulnerabilidades, sempre que possível, para avaliar o que os invasores poderiam obter após uma exploração bem sucedida das falhas. (WEIDMAN, 2014:30).

De acordo com Seginfo (2018), problemas na qualidade do código encabeçam as 10 vulnerabilidades de segurança mais comuns. Estudos realizados pela Veracode[1], apontam que pelo menos metade dos aplicativos, de todas as industrias que participaram, já foram prejudicados por conta de códigos mal elaborados. Chamando a atenção das empresas para adotarem um maior rigor no que tange práticas adequadas de programação.

Não obstante, foi sancionada em 14 de Agosto de 2018 a Lei Geral de Proteção de Dados - LGPD, lei essa que tem o objetivo de aumentar a privacidade dos dados pessoais, a qual cria regras sobre os processos de coleta, compartilhamento e principalmente do armazenamento de informação, inclusive nos meios digitais. Segundo Proof (2018), a definição de "dado pessoal" recai sobre qualquer informação relacionada a pessoa natural identificada ou identificável, ou seja, qualquer dado pelo qual se consiga identificar uma pessoa ou que com a união de outro dado possibilite essa identificação. O descumprimento dessa lei, a qual está sob regulação da Autoridade Nacional de Proteção de Dados (ANPD), pode acarretar à entidade multas altíssimas que podem corresponder até 2% do faturamento da empresa ou conglomerado limitado até R$ 50 milhões por infração cometida. Uma infração pode ser interpretada, no caso de um vazamento de dados, como cada dado pessoal vazado, havendo a possibilidade de multas diárias para compelir a entidade a cessar as violações.

2.Teste de Intrusão

O termo PenTest é derivado de *Penetration Test*, cuja melhor tradução seria Teste de Intrusão ou Teste de Invasão. Segundo ProfissãoHacker (2018) e CoreSecurity (2018) o PenTest atende como um conjunto de técnicas e uso de ferramentas objetivando identificar falhas de segurança em sistemas, redes corporativas e dispositivos. Através de técnicas particulares o profissional, denominado de *Pentester*, irá identificar as vulnerabilidades existentes, explorá-las e entregar um relatório à organização, a qual deverá então tomar as devidas ações para corrigir as eventuais falhas de segurança.

[1] Empresa de segurança de aplicativos com sede em Burlington, Massachusetts. Fundada em 2006, a empresa fornece um serviço automatizado baseado em nuvem para proteger aplicativos corporativos da Web, móveis e de terceiros.

3. Etapa de Reconhecimento

A execução de um PenTest se faz através da realização de algumas etapas, onde inicialmente temos a etapa de Reconhecimento, cuja ação visa coletar o maior número de informações possíveis através do uso de ferramentas, nas quais se inclui o maior mecanismo de busca da internet, o *Google*. O *Google* fornece um mecanismo de buscas avançadas, chamado de *Google Hacking*, através do uso de operadores especiais, denominados *dorks*, no campo de busca. Como exemplo desses *dorks* temos as palavras "*inurl:*" e "*site:*". A *dork* "*inurl:*" apresenta como resultado todas as páginas indexadas em uma URL - *Uniform Resource Locator* que contém um determinado termo ou expressão, já a *dork* "*site:*" é responsável por delimitar o domínio ao qual a busca será realizada.

A Figura 1 apresenta uma consulta através do *Google Hacking* usando as *dorks* "*inurl:?arquivo=luciano*" e "*site:.edu.br*". Algumas informações e domínio foram mascarados por questão de sigilo.

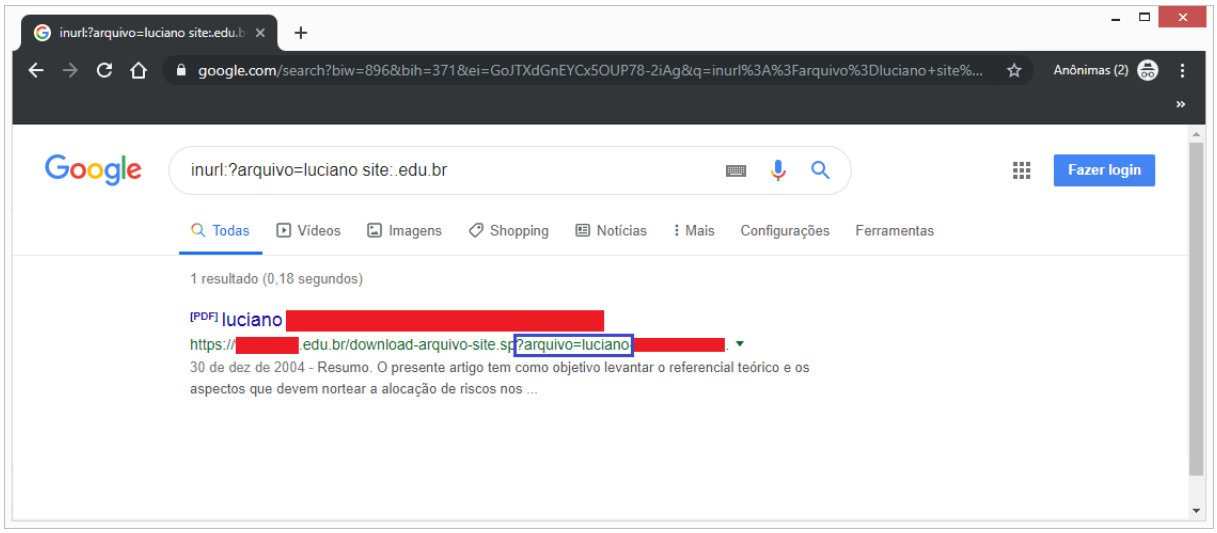

Figura 1 – Consulta através do *Google Hacking* usando *dorks*

Fonte: Produzido pelo o autor (2018)

Segundo GIAVAROTO et al. (2013), Reconhecimento ou *Footprinting*, é o método utilizado para a obtenção de informações a respeito de um determinado alvo, ou organização. Consiste no processo de obtenção de informações a respeito de versões, as quais podem ser realizadas a partir da detecção de *banners* sobre o sistema operacional ativo e serviços em execução, tais como: *Secure Shell* (SSH), *Telnet*, *Apache*, dentre outros, a respeito do alvo.

4. Teste de Intrusão em um sistema acadêmico

Como parte para a conclusão desse trabalho, foi realizado um estudo de caso em um sistema web acadêmico, listado através de uma *dork* do *Google Hacking*, onde na oportunidade o Teste de Intrusão identificou vulnerabilidades críticas de segurança que podem comprometer o ambiente.

Em análise ao comportamento das requisições de uma determinada página desse sistema acadêmico, pode se verificar que certa variável passada pela a URL através do método "*GET*", não está contida em um processo de sanitização. A ausência de controle em

questão, permite que um usuário mal intencionado consiga manipular o caminho ou *path* e realizar downloads de arquivos confidenciais ou de configurações do servidor, sejam eles do sistema operacional, do serviço WEB ou do serviço de gerenciamento de banco de dados, assim como quaisquer páginas que faça parte e compõem o sistema. Tal vulnerabilidade é conhecida como LFD (*Local File Disclosure* / *Local File Download*) e será utilizada também na etapa de Exploração.

A Figura 2 apresenta a requisição da página para realizar o download de um determinado arquivo. O domínio e algumas informações da URL foram mascarados por questão de sigilo.

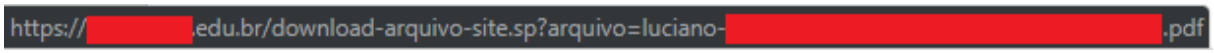

Figura 2 – Requisição para realização de download de um arquivo de extensão .pdf

Fonte: Produzido pelo o autor (2018)

Através dessa vulnerabilidade é possível ter acesso às informações de arquivos que não deveriam estarem acessíveis por um usuário qualquer, tais como: o "*etc/**shadow"***, o "*etc/**passwd***", o "*etc/**issue***", o "*var/log/apache2/**access.log***", o "*logs/**tomcat.log***", o "*home/ilion/**.bash_history***", o "*root/**.bash_history***", o "*home/ilion/certificado-2016/**cert.crt***", dentre outros.

O arquivo "*cert.crt*", refere-se a um arquivo de chave privada de um certificado digital que assiste ao site do sistema da instituição. É responsável por prover uma camada de segurança (SSL/TLS) ao protocolo HTTP (*Hyper Text Transfer Protocol*), chamado de *Hyper Text Transfer Protocol Secure* - HTTPS, para que os dados sejam transmitidos por meio de uma conexão criptografada sendo possível verificar a autenticidade do servidor e do cliente. Uma chave privada não deve ser deixada em locais públicos, ou estar acessível a qualquer usuário tendo em vista que ela funciona como uma espécie de senha pessoal e intransferível. Se outro alguém obtiver a chave privada, essa pessoa poderá se passar pelo verdadeiro dono do certificado em eventuais transações na internet. Nesse caso o Certificado Digital, seria como uma identidade, perfeitamente legítima, se passando por alguém ou instituição, quando, na verdade, não é o caso.

O arquivo *.bash_history* armazena a lista dos comandos digitados pelo usuário em uma última sessão. De posse desse arquivo, um usuário qualquer, consegue obter informações sensíveis do que foi executado no servidor.

O arquivo *issue* armazena informações da distribuição e versão do sistema operacional em uso pelo servidor. No servidor em questão, foi possível identificar que trata-se de uma distribuição *Linux Ubuntu 14.04.1 LTS*, datada de 24 de julho de 2014. Essa versão encontra-se desatualizada, com diversas vulnerabilidades publicadas e passível de exploração, principalmente a *Local Privilege Escalation* (Escalonamento de Privilégio Local), conforme publicações no *Common Vulnerabilities and Exposures*[2] - CVE (CVE-2015-8660 e CVE-2019-7304).

Através das informações contidas no arquivo *tomcat.log*, foi possível obter inúmeras informações pessoais, as quais continham dados como: nome completo, telefone, e-mail,

[2] É uma lista de ameaças de segurança conhecidas às quais são designados números de identificação. Está acessível através do site: https://cve.mitre.org/

unidade de ensino, curso, cidade, estado, etc. O possível vazamento de tais informações poderia, conforme consta na LGPD, acarretar em prejuízos financeiros altíssimos para a instituição.

A Figura 3 apresenta parte do arquivo contendo as informações pessoais. Algumas informações foram mascaradas por questão de sigilo.

Figura 3 – Conteúdo do arquivo contendo informações pessoais

Fonte: Produzido pelo o autor (2018)

Uma vez munido do conteúdo dos arquivos *shadow* e *passwd*, um usuário poderia utilizar uma ferramenta denominada *UnShadow*, ferramenta essa que faz a combinação da relação das senhas criptografadas apresentadas no arquivo *shadow* e de usuários apresentadas no arquivo *passwd*, onde através de um grande dicionário e auxílio da ferramenta *John the Rippe*[3], se consegue "quebrar" a criptografia SHA-512[4] e identificar em texto plano o(s) usuário(s) e senha(s) de acesso ao servidor.

A Figura 4 e a Figura 5 apresentam o conteúdo dos arquivos *shadow* e *passwd* respectivamente, obtidos do servidor. Algumas partes dos *hashs* foram mascarados por questão de sigilo.

[3] Ferramenta de código aberto, desenvolvida por Alexander Peslyak, eficaz para a quebra de senhas escondidas através de métodos de encriptação.

[4] Função hash criptográfica de 64 bytes projetada pela NSA (Agência de Segurança Nacional dos EUA). SHA significa: *Secure Hash Algorithm* (Algoritmo de Hash Seguro).

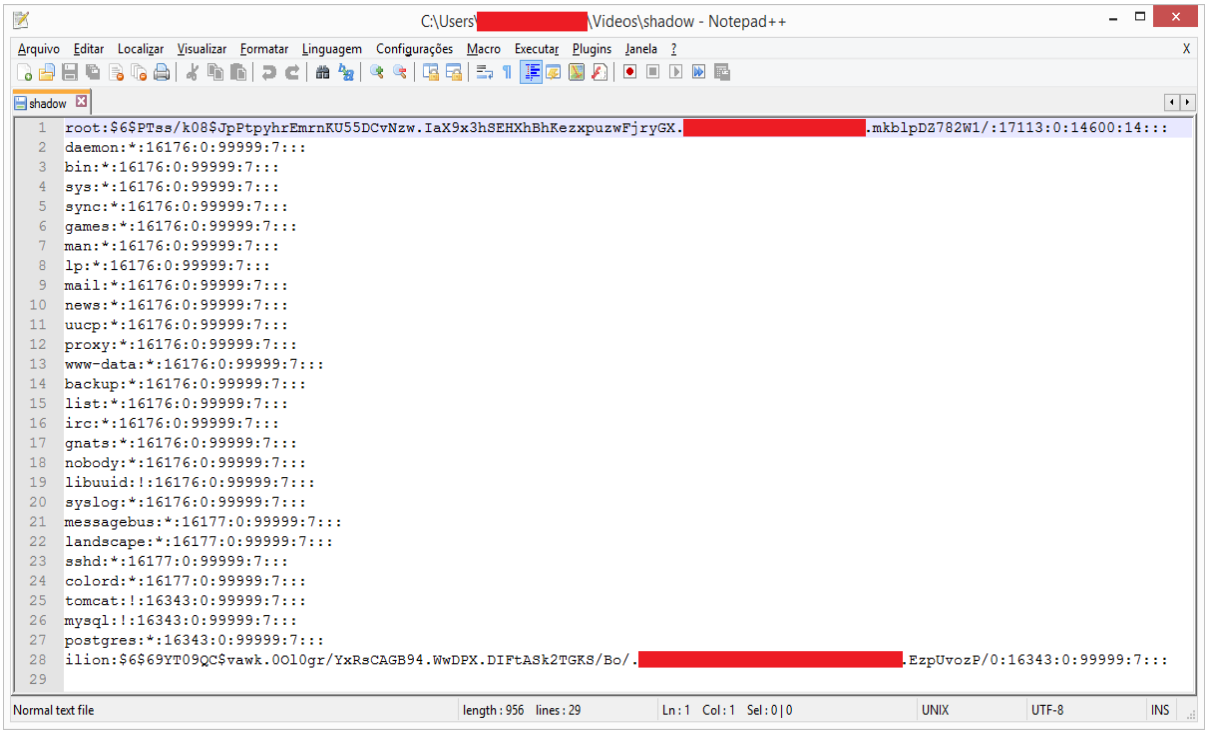

Figura 4 – Conteúdo do arquivo *shadow*

Fonte: Produzido pelo o autor (2018)

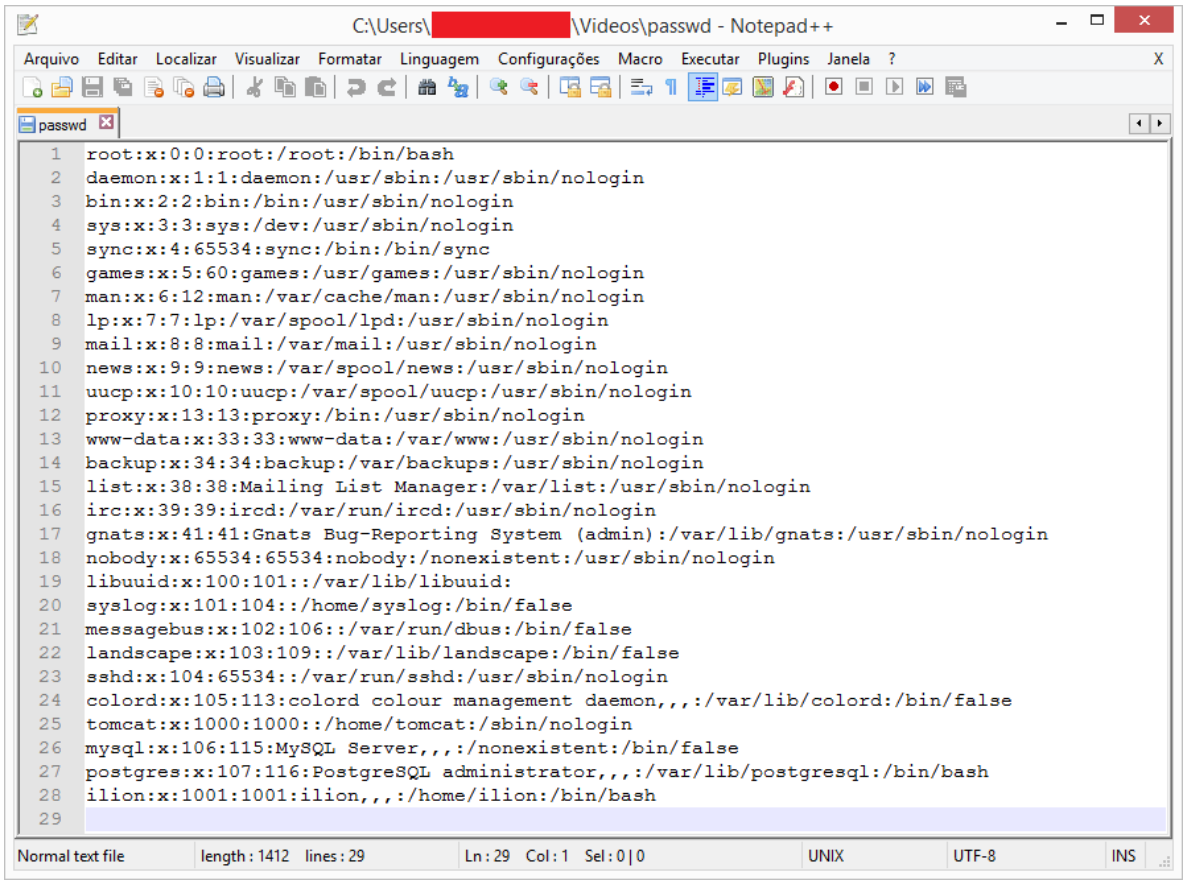

Figura 5 – Conteúdo do arquivo *passwd*

Fonte: Produzido pelo o autor (2018)

Já a Figura 6 apresenta o uso do comando para "quebrar" a criptografia através da ferramenta *John the Rippe*.

```
root@kali:/usr/sbin# ./john --wordlist=/usr/share/wordlists/passhash.lst  /ro
ot/unshadow.txt
```

Figura 6 – Comando para uso da ferramenta *John the Rippe*

Fonte: Produzido pelo o autor (2018)

5. Etapa de Varredura

Uma vez feito o reconhecimento do alvo, passamos para a etapa de Varredura ou *Scanning*, a qual consiste em obter dados mais específicos tais como: portas abertas, sistema operacional, serviços e suas respectivas versões. Tais informações ajudam no direcionamento a ser tomado para a elaboração do plano de ações.

Nessa etapa foi utilizado a ferramenta *Zenmap*, que é a versão gráfica da poderosa ferramenta NMAP, a qual de maneira versátil, permite diversos tipos de varreduras e enumeração do sistema, apresentando informações do estado das portas. O *Zenmap* possui suporte para uso de scripts, divididos em categorias, os quais melhoram o refinamento das varreduras.

A Figura 7 apresenta o resultado do parâmetro "*--open*", o qual verifica se as portas mais comumente usadas encontram-se abertas, ou seja, acessíveis. O endereço IP do servidor e horário foram mascarados por questão de sigilo.

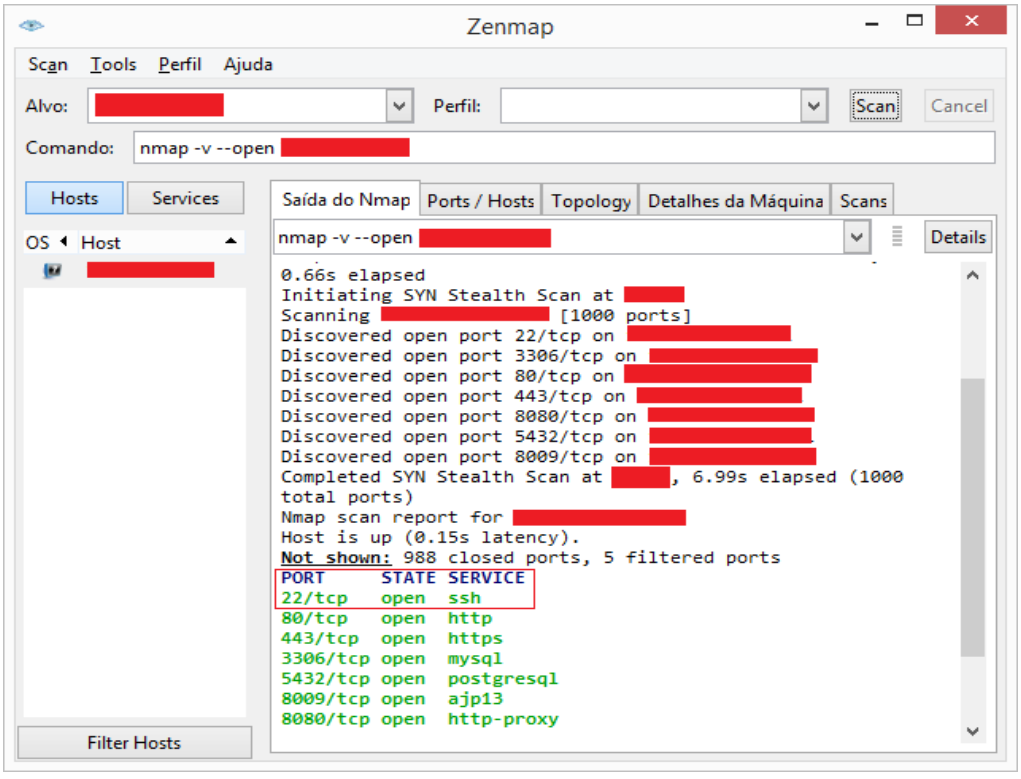

Figura 7 – Varredura de portas abertas através da ferramenta Zenmap

Fonte: Produzido pelo o autor (2018)

No exemplo mostrado é possível verificar que diversas portas, incluindo a porta 22, a qual é utilizada pelo serviço de SSH, a porta 8009 utilizada pelo serviço *Apache JServ Protocol v13* (AJP13) e a porta 3306 utilizada para acesso e administração do sistema de gerenciamento de banco de dados *MySQL*, se encontram abertas (*open*).

Utilizando o *script* "*AJP-Brute.nse*" do *Zenmap*, o qual apresenta os parâmetros de configuração do serviço AJP13 em execução, é possível checar que o mesmo encontra-se sem a necessidade de autenticação para acesso.

A Figura 8 apresenta o resultado da execução do *script* "*AJP-Brute.nse*", o qual informa que o referido serviço está acessível sem a necessidade de autenticação. O endereço IP do servidor e horário foram mascarados por questão de sigilo.

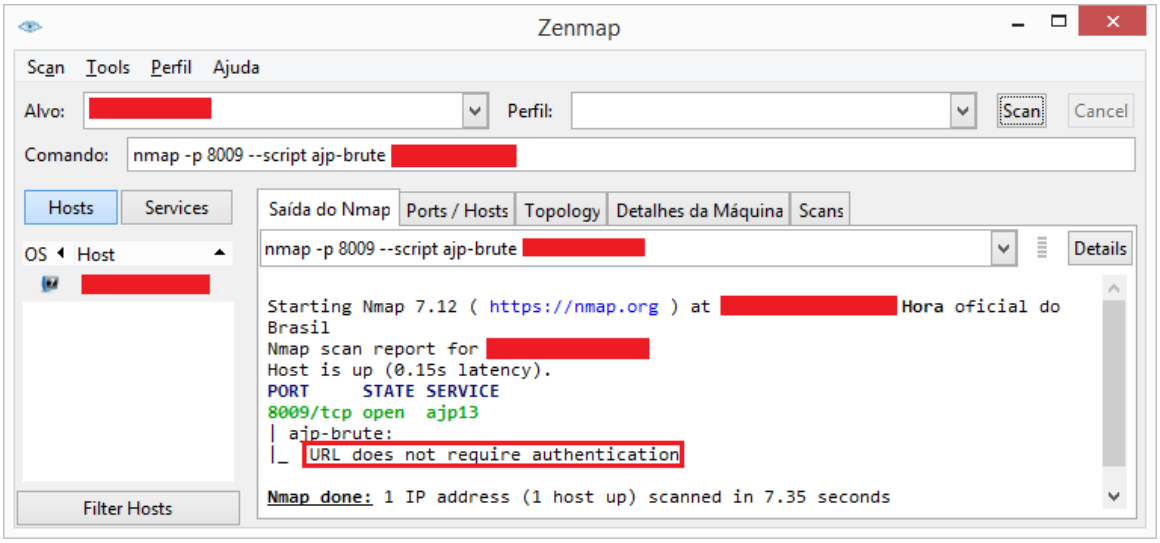

Figura 8 – Resultado da execução do script "*AJP-Brute.nse*" através da ferramenta *Zenmap*

Fonte: Produzido pelo o autor (2018)

6. Etapa de Exploração

Uma vez feito o Reconhecimento e de posse dos resultados obtidos na etapa de Varredura, é chegada a etapa de Exploração ou *Exploitation*, a qual o *pentester* irá explorar cada item de forma separada.

Conforme resultado apresentado através da execução do *script* "*AJP-Brute.nse*", um usuário poderia configurar um serviço de *Proxy Reverso* local e utilizar o protocolo AJP13 para realizar um *bypass*, ou seja, burlar o serviço de Proxy Reverso que antecede o acesso até as consoles internas do servidor de aplicação *Tomcat*. DIABLOHORN (2018) apresenta através do uso do *Metasploit*[5], um módulo que contém um *exploit* denominado "*tomcat_mgr_deploy*" preparado para exploração dessa vulnerabilidade e obtenção de um *Reverse Shell*, que consiste em um usuário obter acesso de maneira remota ao servidor e conseguir executar comandos na *shell* (terminal ou *prompt* de comando).

Através da leitura do arquivo *.bash_history*, foi possível identificar credenciais de acesso referentes ao sistema de gerenciamento de banco de dados *MySQL*.

[5] Projeto cujo objetivo é prover um ambiente de pesquisa e exploração de vulnerabilidades, onde através de *exploits* já existentes, facilite os testes de intrusão (Pentests) e auxilie no desenvolvimento de assinaturas para Sistema de Prevenção de Intrusão (*Intrusion Prevention System* - IPS).

A Figura 9 apresenta a parte do arquivo contendo as informações sobre as credenciais de acesso ao *MySQL*. A referida senha foi mascarada por questão de sigilo.

```
397  cd /etc/apache2/
398  ls -l
399  vim apache2.conf
400  cd sites-enabled/
401  ls
402  vim 000-default.conf
403  service apache2 start
404  mysql -u root -p[redacted]
405  sudo /etc/init.d/mysql stop
406  sudo /usr/sbin/mysqld --skip-grant-tables --skip-networking &
407  mysql -u root
408  cd /etc/mysql/
409  ls -l
410  vim my.cnf
```

Figura 9 – Informações sobre as credenciais de acesso ao *MySQL*

Fonte: Produzido pelo o autor (2018)

De posse das credenciais de acesso ao *MySQL*, e tendo em vista que a porta 3306 referente a esse serviço está aberta no servidor e acessível através da internet, um usuário poderia estabelecer remotamente uma conexão a esse sistema de gerenciamento de banco de dados e consequentemente ter acesso as informações das tabelas e dados da aplicação.

A Figura 10 apresenta a simulação de conexão ao servidor através da console *MySQL*. O endereço IP do servidor foi mascarado por questão de sigilo.

```
root@kali:/# mysql -h [redacted] -u root -p
Enter password:
```

Figura 10 – Simulação de conexão ao servidor via console *MySQL*

Fonte: Produzido pelo o autor (2018)

Conforme credenciais de acesso (usuário e senha) obtidas através do arquivo *passwd* e da "quebra" da criptografia do arquivo *shadow*, além da constatação através da varredura que a porta 22, referente ao serviço SSH encontra-se acessível, é possível realizar a conexão para obter acesso a *shell* do referido servidor e a partir desse momento ter o seu total controle.

A Figura 11 apresenta a simulação de conexão e acesso ao servidor. O endereço IP do servidor foi mascarado por questão de sigilo.

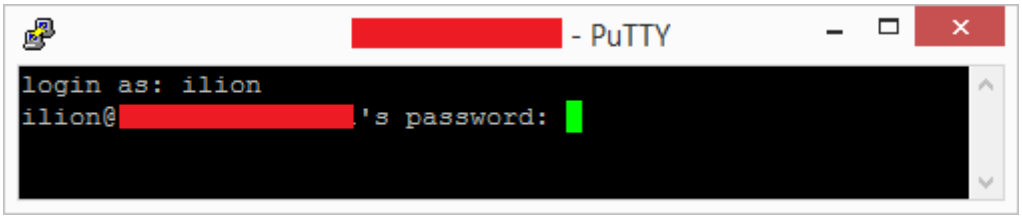

Figura 11 – Simulação de acesso via SSH ao servidor

Fonte: Produzido pelo o autor (2018)

7. Conclusão

Testes de funcionalidades assim como os testes de segurança automatizados apenas, não possuem o mesmo êxito em identificação de vulnerabilidades se comparado ao método de teste de segurança ofensivo, o qual toma como referência o ponto de vista do atacante. Neste contexto foram utilizadas diversas formas de ataques conhecidas por especialistas da comunidade. A inspeção técnica deste contexto foi capaz de oferecer uma visão mais real no que tange os riscos da segurança da informação de uma instituição. As ameaças aos dados corporativos tem evoluído mais rapidamente se comparado aos métodos de defesa. Empresas devem se adaptar, deixando de lado as medidas triviais e buscarem soluções proativas, tal como é a mentalidade ofensiva, a qual pode apresentar os ataque factíveis e os possíveis impactos ao negócio.

O objetivo deste artigo, apresentado através de um estudo de caso e suas reais consequências, foi sensibilizar as pessoas quanto a importância da submissão de aplicações e sistemas, de maneira prévia, à Testes de Intrusão, mitigando assim os riscos de eventual incidente de segurança que possam trazer prejuízos as organizações.

Conforme fora abordado, devido a existência de uma vulnerabilidade no código fonte da aplicação, um usuário mal intencionado conseguiria adquirir arquivos sensíveis do servidor onde o sistema estava hospedado e a partir daí concretizar ataques mais elaborados, obtendo acesso e podendo até mesmo indisponibilizar o serviço. Foi apresentado também vulnerabilidades relacionadas a uma má configuração de serviços no servidor.

Portanto, caso o sistema em tela tivesse passado por um processo de PenTest antes de ser submetido ao ambiente de produção, tais vulnerabilidades poderiam ter sido identificadas, mitigadas e consequentemente tornado o sistema mais seguro.

Referências

CABRAL, Carlos e CAPRINO Willian, organizadores. **Trilhas em segurança da informação: caminhos e ideias para a proteção de dados**. Rio de Janeiro: Brasport, 2015.

Certisign. **Certificado Digital**. Disponível em: <https://www.certisign.com.br/certificado-digital/servidor-ssl> Acesso em: 18 de Novembro de 2018.

CoreSecurity. **What is Penetration Testing?**. Disponível em: <https://www.coresecurity.com/content/penetration-testing> Acesso em: 10 de Novembro de 2018.

DIABLOHORN. **8009, the forgotten Tomcat port**. Disponível em: <https://diablohorn.com/2011/10/19/8009-the-forgotten-tomcat-port/> Acesso em: 15 de Novembro de 2018.

GIAVAROTO, Sílvio César Roxo; SANTOS, Gerson Raimundo dos. **Backtrack Linux - Auditoria e Teste de Invasão em Redes de Computadores**. Rio de Janeiro: Ciência Moderna LTDA., 2013.

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

NMAP. **AJP-brute**. Disponível em: <https://nmap.org/nsedoc/scripts/ajp-brute.html> Acesso em: 12 de Novembro de 2018.

Planalto. **L13.709 - Lei Geral de Proteção de Dados (LGPD)**. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm> Acesso em: 15 de Novembro de 2018.

PMGAcademy. **Como funciona o Metasploit**. Disponível em: <https://www.pmgacademy.com/pt/blog/artigos/como-funciona-o-metasploit> Acesso em: 15 de Novembro de 2018.

ProfissãoHacker. **Pentest - Os Testes de Intrusão**. Disponível em: <http://profissaohacker.com/pentest/> Acesso em: 10 de Novembro de 2018.

Proof. **Como funciona a LGPD**. Disponível em: <https://www.proof.com.br/blog/como-funciona-a-lgpd> Acesso em: 12 de Novembro de 2018.

Seginfo. **As 10 vulnerabilidades de Segurança mais comuns**. Disponível em: <https://seginfo.com.br/2015/12/08/as-10-vulnerabilidades-de-seguranca-mais-comuns-2/> Acesso em: 10 de Novembro de 2018.

Siteblindado. **Por que o Pentest é a melhor maneira de identificar vulnerabilidades de segurança?**. Disponível em: <https://blog.siteblindado.com/pentest-vulnerabilidades-seguranca/ > Acesso em: 10 de Novembro de 2018.

TIinside. **Crimes ciberneticos causaram cerca de us 550 bilhoes em perdas no ano passado segundo a AON**. Disponível em: <https://tiinside.com.br/tiinside/seguranca/mercado-seguranca/02/10/2018/crimes-ciberneticos-causaram-cerca-de-us-550-bilhoes-em-perdas-no-ano-passado-segundo-a-aon/> Acesso em: 14 de Novembro de 2018.

VINES, Russell Dean. **Penetration testing reconnaissance Footprinting, scanning and enumeration**. Disponível em: <http://searchitchannel.techtarget.com/tip/Penetration-testing-reconnaissance-Footprinting-scanning-and-enumerating> Acesso em: 15 Novembro de 2018.

WEIDMAN, Georgia. **Testes de Invasão: Uma introdução prática ao hacking**. São Paulo: Novatec, 2014.

Wikipedia. **Hyper Text Transfer Protocol Secure**. Disponível em: <https://pt.wikipedia.org/wiki/Hyper_Text_Transfer_Protocol_Secure> Acesso em: 16 de Novembro de 2018.

Wikipedia. **SHA-2**. Disponível em: <https://pt.wikipedia.org/wiki/SHA-2> Acesso em: 14 de Novembro de 2018.

Wikipedia. **Teste de Intrusão**. Disponível em: <https://pt.wikipedia.org/wiki/Teste_de_intrus%C3%A3o> Acesso em: 09 de Novembro de 2018.