Aplicação de Pentest em Sistemas Computacionais para Análise de Vulnerabilidades: Um Estudo de Caso

Primesh Parmar, Alex Feleol

Fundação Centro de Análise, Pesquisa e Inovação Tecnológica - FUCAPI

CEP – 69075-351 – Manaus – AM – Brasil

{primesh,alex}@sec4all.com.br

Abstract: *In this case study have the goal to demonstrate the procedures and methods of a pentest using the PTES methodology, which represents the steps of how a pentest is made, the types of information that can acquire, and do a vulnerability analyses in the servers in production of a multinational corporation. Besides, demonstrate the proof of concept exploring one of the vulnerabilities found, proving the existence of the security flaw and fixing it to avoid a post exploration from an attacker in the future. The Institute of technology that develop projects of informatics which promotes technology innovation in their company clients, had the interest to include and sponsor in their project`s database the pentest to apply in the production environment.*

Resumo: *Este estudo de caso tem como objetivo demonstrar os procedimentos e métodos de pentest utilizando a metodologia PTES, da qual apresenta as etapas de execução do processo de pentest em servidores em produção de uma empresa multinacional. Além de demonstrar a prova de conceito explorando uma das vulnerabilidades encontradas provando a existência da falha de segurança e corrigindo-a a fim de evitar atacantes mal intencionados futuramente. O instituto de tecnologia, que desenvolve projetos de bens de informática que promovam inovação tecnológica em suas empresas clientes, corroborou e patrocinou ao incluir em seu quadro de projetos o pentest a fim de aplicá-lo no ambiente de produção.*

1. Introdução

Tendo em vista as notícias sobre invasões, falhas em sistemas computacionais e até espionagem industrial como em notícia publicada na 1ª edição de Junho 2011, pág. 26, da revista Segurança Digital, se faz necessário repensar a segurança da informação das organizações. São poucas as empresas que abordam a área de segurança da informação com a devida atenção. Geralmente as empresas apenas começam a se importar no momento em que ocorre algum sinistro envolvendo um, ou mais, pilares da segurança da informação, das quais são: disponibilidade, integridade e confidencialidade. Justamente quando ocorrem eventos negativos tais como, roubo de dados sigilosos, exposição ruim da imagem da empresa na mídia ou até mesmo perda financeira significativa que as corporações começam a investir e se importar com a segurança da informação.

Este estudo de caso foi desenvolvido em uma empresa de grande porte na sua área de atuação e tem como objetivo demonstrar a necessidade de implementar políticas de segurança da informação e assim evitar sinistros relacionados na área, pois, a empresa deseja investir em segurança e proteger seu ativo mais valioso, ou seja, a

informação. Com isso, foi realizado o *pentest* (*penetration test* ou teste de intrusão) na empresa, com o objetivo buscar vulnerabilidades em servidores críticos e encontrar soluções de melhorias de segurança em seu ambiente de rede externo.

Para realizar o *pentest* na empresa deste estudo de caso, foi utilizada uma metodologia chamada *PTES* (*Penetration Test Execution Standard*), pois segundo Engebretson (2011), a razão pelo uso de uma metodologia em um *pentest* vem do fato de que vários atacantes (indivíduos maliciosos) seguirem uma linha em comum de abordagem com o objetivo de comprometer um sistema, conforme visto em revisões de relatórios de resposta de incidentes ou divulgação de brechas.

O presente artigo está organizado em quatro sessões. Na sessão 2, intitulada "Metodologia", é descrito sobre a metodologia *PTES* e a descrição de cinco fases do mesmo: interações iniciais, *footprint* [1] rastreamento de portas, análise de vulnerabilidades e exploração. Na sessão 3, "Estudo de caso", ocorre o estudo de caso que mostra as fases da metodologia sendo aplicado em uma empresa real. Na sessão 4, a solução da falha encontrada e por fim, na sessão 5 as principais conclusões absorvidos deste estudo.

2. Metodologia

O *pentest* é efetuado contra redes locais onde se interage com o mundo exterior. Isto pode ser feito por conexões à *Internet*, sem fio, sistemas telefônicos e outras localizações de acesso remoto. A intenção deste tipo de teste de segurança é determinar onde e como a sua rede é vulnerável a ataques [Thomas, 2011].

Para uma visão completa do *pentest* o mesmo pode ser dividido em passos e fases, pois segundo Engebretson (2011), quando são unidos esses passos ou fases podem se tornar em uma metodologia compreensiva capaz de completar o *pentest*. Embora a maioria dos nomes ou números de passos possa variar entre as metodologias, o importante é que o processo permita uma visão completa do *pentest*.

Segundo *PTES Technical Guidelines* (2012), a metodologia *PTES* é um novo padrão projetado para fornecer para as empresas serviços de segurança e de negócio em uma linguagem comum para a realização de *pentests*. Iniciado em 2011 o *PTES* é elaborado por um grupo de profissionais de segurança da informação e é uma comunidade livre para quem puder contribuir.

Segundo (Kennedy et al., 2011), o *PTES* está redefinindo o *pentest* de maneira que afeta os *pentesters*[2] novos e experientes, e está sendo adotado por vários membros líderes da comunidade de segurança. O seu objetivo é definir e levantar a conscientização sobre o que um *pentest* real pode significar e estabelecer uma base de princípios fundamentais requeridos para a sua condução.

2.1. Fases do *PTES*

Segundo (Kennedy et al., 2011), as fases do *PTES* são moldadas para definir e garantir para o cliente um nível padrão de esforço que será gasto no *pentest* por qualquer uma

[1] *Footprint* – Consiste na interação com o alvo com o objetivo de obter informação externas sobre a organização.
[2] *Pentesters* – Profissional de segurança responsável por encontrar vulnerabilidades em sistemas computacionais e redes de computadores.

pessoa que conduzir esse tipo de avaliação. O *PTES* é dividido em categorias com diferentes níveis de esforços para cada tipo ou situação em uma organização, conforme descritas a seguir.

2.1.1. Interações Iniciais

A interação inicial com o cliente tipicamente ocorre quando se discute o escopo e as condições do *pentest* com o mesmo. É nessa fase que serve para conscientizar o cliente sobre a expectativa e a possibilidade de um escopo completo [Kennedy et al., 2011].

2.1.2. Coleta de informações ou *footprint*

O objetivo do *footprint* é realizar um reconhecimento do alvo a fim de juntar o tanto de informação que for possível para ser utilizado no *pentest*. Quanto mais informações adquiridas, durante esta fase, maior o número de vetores de ataque será capaz de ser utilizado. Alguns dos modos de adquirir essas informações é através do *Google Hacking* [3], engenharia social, redes sociais, entre outras ferramentas de compartilhamento de informações [*PTES Tecnical Guidelines,* 2012].

Segundo Assunção (2011), o *footprint* consegue descobrir informações úteis, como e-mails dos usuários, informações de rede, links importantes e até alguns documentos desprotegidos. Um modo de se fazer *footprint* eficiente é utilizar páginas de busca na *Internet*, em especial o *Google*. Através do "*GoogleBot*", um *script* automatizado, ele varre toda a rede, atravessando todos os possíveis links e catalogando as páginas descobertas.

2.1.2.1. Reconhecimento passivo e ativo

Pelo fato do reconhecimento passivo usar a vasta quantidade de informações disponíveis na *Internet,* quando se é conduzido, não se interage diretamente com o alvo, assim o mesmo não tem como saber ou registrar alguma atividade. Em contrapartida, o reconhecimento ativo interage diretamente com o alvo, sendo assim, neste processo, pode-se registrar o endereço de *IP* e as atividades executadas [Engebretson, 2011].

No reconhecimento ativo, segundo Faircloth (2011), usa-se a transferência de zona *DNS*, também conhecida como *AXFR* (*a request for transfer of an entire zone*), que é tipicamente usada para replicar dados e arquivos *DNS* entre os servidores dos mesmos. Se o servidor permitir a transferência de zona, todos os nomes de *DNS* e endereços *IPs* salvos neste servidor poderão retornar em um texto de fácil compreensão.

Segundo *PTES Tecnical Guidelines* (2012), após identificar todas as informações associadas ao(s) domínio(s) do cliente, inicia-se a consulta *DNS*. Uma das vulnerabilidades mais sérias envolvendo *DNS* é permitir que usuários não autorizados, principalmente provenientes da *Internet*, possam efetuar a transferência de zona *DNS*.

2.1.3. Rastreamento de Portas

O objetivo do rastreamento de portas é identificar quais portas estão abertas e determinar quais serviços que estão rodando no sistema alvo. Esses serviços são tarefas especificas que o computador executa como e-mail, *FTP*, impressão, entre outros. O rastreamento é realizado para observar quais portas estão em uso ou abertas, isso trás

[3] *Google Hacking* – Utilizam-se filtros (dorks) específicos para restringir as buscas pelo Google.

uma visão melhor da proposta do alvo, que em troca, nos trás uma idéia de como atacá-lo [Engebretson, 2011].

Segundo (Simpson et al., 2013), deve-se saber quais portas um atacante procura para que as mesmas sejam fechadas e protegidas. Ao executar rastreamento de portas, não se deve procurar apenas pelas portas mais conhecidas (as de 1 a 1023) já que, muitos programas utilizam portas fora deste alcance. Ao usar uma ferramenta para o rastreamento, o atacante pode rapidamente identificar uma porta vulnerável e executar um *exploit*[4] para atacar um sistema

2.1.4. Análise de vulnerabilidades

Uma vulnerabilidade é uma falha em um software ou configuração de sistema que pode ser explorado. Existem vulnerabilidades de várias formas, mas o mais comum é a falta da aplicação das atualizações que corrigem as falhas. Os fornecedores constantemente publicam as correções de vulnerabilidades conhecidas nos próprios sites da empresa ou portais e fóruns especializados [Engebretson, 2011].

Segundo Boyle e Panko (2012), a análise de vulnerabilidades tenta encontrar falhas rodando um programa que busca por vulnerabilidades em servidores que estão no escopo do *pentest*. Esses programas realizam uma bateria de ataques contra esses servidores e geram relatórios detalhados sobre as vulnerabilidades encontradas

2.1.4.1. Escaneamento de Vulnerabilidades

Para realizar o escaneamento de vulnerabilidades pode-se utilizar uma ferramenta que executa esta atividade de maneira automatizada como o *Nessus*[5], pois segundo Assunção (2011), scanners possuem um banco de dados de falhas e checam os serviços descobertos no *host*-alvo a fim de descobrir alguma vulnerabilidade. É um processo completamente automatizado e extremamente simples, que não requer nenhum tipo de pesquisa manual.

Segundo (Kennedy et al., 2011), vários sistemas operacionais tendem a responder diferentemente quando é feita a sondagem da rede por causa das diferentes implementações de rede em uso. Essas respostas servem como um identificador que o *scanner* de vulnerabilidade usa para determinar a versão do sistema operacional, o nível de sua atualização e enumerar softwares e serviços para determinar quais deles estão atualizados. Com os resultados obtidos, o *scanner* apresenta um relatório com qualquer vulnerabilidade detectada no sistema.

Segundo *PTES Technical Guidelines* (2012), assim que a vulnerabilidade for encontrada no sistema do alvo, é necessário determinar a exatidão do problema identificado e pesquisar um *exploit* em potencial da vulnerabilidade dentro do escopo do *pentest*. Esse resultado da fase de identificação das vulnerabilidades deve ser validado principalmente se códigos de exploração estiverem disponíveis. O identificador CVE (*Common Vulnerabilities and Exposures*) identifica a vulnerabilidade dada, o qual pode ser usado para acessar o sumario de informações e *links* de outras fontes de banco de dados do CVE.

---

[4] *Exploit* - é um programa criado para testar uma falha de segurança, geralmente como prova de conceito
[5] *Nessus* – Ferramenta de análise de vulnerabilidades

2.1.5. Exploração

A exploração tenta conseguir acesso não-autorizado ao sistema-alvo e aos seus comandos internos, se possível, com acesso de super usuário (ou administrador) [Assunção, 2011].

Segundo (Kennedy et al., 2011), usa-se um *exploit* ou força bruta para explorar uma ou mais falhas encontradas nas fases anteriores. Porém, não se pode utilizar um *exploit* indevidamente, pois, deve-se utilizar o *exploit* adequado para a falha encontrada.

2.1.6. Trabalhos relacionados

Durante o levantamento bibliográfico, ressaltou-se a procura por estudos e pesquisas desempenhadas no campo de atuação deste trabalho, notou-se que a gama de trabalhos relacionados proporcionam uma variedade de informações em diversas ramificações deste assunto que somente enriquecem e produzem embasamento teórico a este estudo de caso como os dois trabalhos relacionados a seguir:

O primeiro artigo publicado por Borges e Helena (2011), intitulado "Estudo comparativo de metodologias de *pentests*.", faz uma comparação com outras metodologias de *pentest* e também possui uma pesquisa sobre a aplicação das metodologias em empresas do Brasil.

O segundo artigo publicado por (Ferreira et al., 2012), intitulado "Análise de vulnerabilidades em Sistemas Computacionais Modernos: Conceitos, Exploits e Proteções.", descreve técnicas de desenvolvimento de *exploits* como estratégia para prevenção de ataques.

3. Estudo de caso

Conforme visto na seção 2, a metodologia *PTES* provê técnicas e ferramentas que possam encontrar vulnerabilidades antes que um atacante mal intencionado a explore. Este estudo de caso aplica os conceitos e métodos do *PTES* a fim de executar o *pentest* em um cenário real.

3.1 Cenário

O estudo de caso foi realizado em uma empresa que é líder de mercado em sua área de atuação. O escopo definido para este cenário possui mais de 10 servidores que estão visíveis na *Internet* e estão alocados em Manaus. Estes servidores possuem *IPs* públicos e diversos serviços como e-mail, *VPN* (*Virtual Private Network*), *DNS* entre outras.

3.2 Execução das fases do *PTES*

Iniciou-se a conversa com o gerente de TI da empresa, nesse momento foram relatados as expectativas, os objetivos e a definição do escopo. O escopo foi definido contendo os servidores que possuem *IPs* válidos ou públicos, haja vista, a visibilidade através da rede externa da empresa, incluindo a rede de perímetro e roteador de borda. Ressalta-se que a rede se estende até São Paulo, onde existe outra unidade fabril, mas no momento o mesmo não entra no escopo devido à descentralização da administração tecnológica do parque computacional.

Para o reconhecimento passivo, a primeira técnica utilizada foi o *Google Hacking,* sendo que, inicialmente não foram encontrados arquivos sigilosos ou comprometedores, mas foi possível detectar alguns dos domínios que a empresa possui e usar o serviço do *whois*[6].Voltando para o *Google* e buscando por informações descobertas pelo *whois*, foi possível encontrar o nome de um diretor, o nome do presidente da empresa e até mesmo, o CPF e RG dos mesmos.

O passo seguinte foi juntar os domínios encontrados e utilizar a ferramenta *Maltego*[7], para descobrir outros domínios pertencentes à empresa e também revelar os *IPs* de rede e de servidores da infraestrutura tanto de São Paulo e Manaus.

Para o reconhecimento ativo utilizando a transferência de zona, puderam-se enumerar vários subdomínios e consequentemente *IPs* de vários servidores inclusive outros serviços como *FTP*, *SMTP*, webmail entre outros tipos de serviços. Conforme mostra a Figura 1 a ferramenta conseguiu encontrar vários subdomínios (o nome da empresa foi substituído por x):

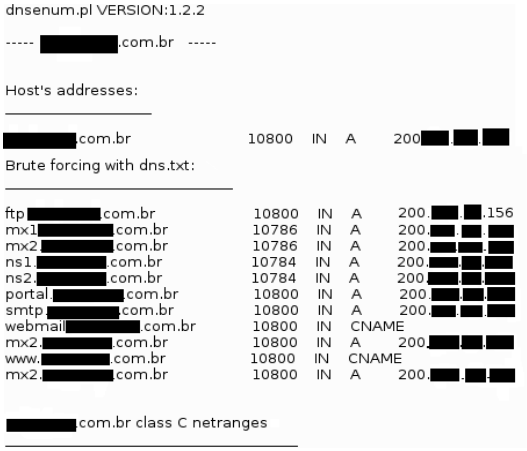

Figura 1. Transferência de zona com a ferramenta *dnsenum*

A partir da transferência de zona nos domínios conhecidos, foi possível a obtenção de nomes de outros domínio assim como ftp.xxxxxxxxxxx.com.br e também de endereços *IPs* públicos como 200.xxx.xxx.156. Devido ao escopo do *pentest* ser apenas a rede de Manaus, separando os *IPs* de cada rede obteve-se um número considerável de *IPs* ativos que serão analisados nas etapas seguintes.

Para realizar o rastreamento de portas, foi utilizado o *Nmap*[8] (*Network Mapper*). Em todos os servidores analisados foram encontrados portas abertas e algum serviço ativo nestas portas. Utilizando a opção –sV é possível identificar o serviço e a versão dos mesmos. Conforme a Figura 2 observa-se que se trata de um servidor de e-mail e o *Nmap* consegue descobrir também o sistema operacional deste servidor.

[6]*Whois* - Ferramenta que possibilita acessar informações públicas especificas sobre o domínio como endereços de *IP*, servidores de *DNS*, e nomes e telefones e endereços de pessoas da empresa (Engebretson, 2011).
[7]*Maltego* - Ferramenta que realize mineração de dados de coleta de informação e cria um mapa com essas informações.
[8]*Nmap* - Ferramenta de escaneamentos de portas.

Figura 2. Exemplo de rastreamento de portas com o *Nmap*

Após realizar a análise de vulnerabilidade dos servidores, foram descobertas falhas com vários tipos de criticidade conforme elencado na Figura 3.

Figura 3. Resultado de uma análise de vulnerabilidade em um servidor

Para comprovar e validar a veracidade da vulnerabilidade encontrada buscou-se por um banco de dados onde os mesmos são conhecidos e são registradas oficialmente.

A realização da prova de conceito pode ser explorada com uma das vulnerabilidades encontradas. Neste caso a vulnerabilidade era de criticidade média de um serviço chamado *Webmin*[9] que está em execução em um dos servidores de e-mail na porta 10000. A sua referência pode ser visualizada pelo CVE-2006-3392.

Na fase de exploração, onde é realizada a prova de conceito, foi encontrada a vulnerabilidade relatada no CVE-2006-3392 que a descreve informando que o *Webmin* possui um *script* chamado "miniserv.pl" que provê serviços web básicos e a sua versão instalada no computador remoto contém uma falha que permite que um atacante não autenticado possa ler arquivos arbitrários no computador afetado, sujeitos aos privilégios da identificação do usuário do servidor web.

Outra informação importante que o CVE-2006-3392 afirma é que esta vulnerabilidade possui um *exploit*. Com esta informação pode-se buscar pelo *exploit* em alguns dos sites que possuem banco de dados de *exploits* públicos para baixar e ser utilizado em ferramentas especializadas como o *Metasploit Framework* ou diretamente por um compilador de linguagem.

Para a utilização do *exploit* referente à vulnerabilidade, é dado o caminho do arquivo e o *exploit* faz busca e mostra na tela o conteúdo deste arquivo solicitado. O endereço do arquivo solicitado é o */etc/passwd*, onde o mesmo mostra os usuários existentes naquele computador remoto. Em seguida foi solicitado o arquivo */etc/shadow*, que possui as senhas codificadas dos usuários. Na Figura 4, mostra o resultado que o *exploit* encontrou.

[9] *Webmin* - Serviço de gerenciamento de comunicação (ssh, FTP, entre outros meios) com o servidor.

Figura 4. Demonstração do *exploit* em execução

Com o nome do usuário e a senha codificada do servidor, é possível utilizar uma ferramenta que tenha a capacidade de quebrar a codificação utilizando força bruta e descobrir a senha do usuário conforme é mostrado na Figura 5.

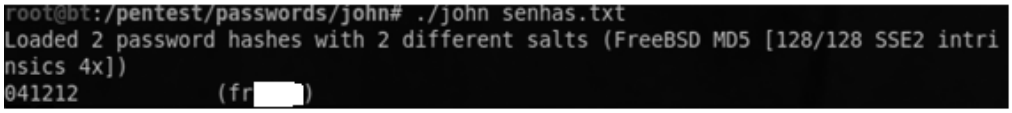

Figura 5. Descodificação da senha do usuário

Com a senha do usuário decodificada é utilizado o serviço de SSH (descoberto em fases anteriores) e tentar o acessar ao servidor conforme a Figura 6.

Figura 6. Acesso ao servidor

Ao acessar o servidor com o usuário demonstrado, o acesso ao computador já possui privilégios irrestritos logo que o usuário possui privilégios de *root*[10]. Nota-se que na fase anterior esta vulnerabilidade foi detectada com a criticidade média, ou seja, não é pelo fato de que a vulnerabilidade não seja alta ou crítica que não possa ser usado como a arma principal de um atacante.

4. Análise dos resultados obtidos

No servidor analisado foram detectadas várias vulnerabilidades, sendo que a falha relatada no CVE-2006-3392 foi escolhida para este estudo de caso, pois se trata de uma falha que permitiu o acesso com um usuário com privilégios de *root*.

Segundo o CVE-2006-3392 e o relatório técnico da ferramenta *Nessus*, para corrigir a falha explorada através do *Webmin*, é preciso atualizar este serviço para uma versão que corrige esta falha. A informação sobre a falha, assim como corrigi-la, foi submetida a gerencia para a devida autorização da correção. Após uma reunião com o Gerente de TI e em seguida com a equipe de suporte técnico, foi obtida a autorização da correção da falha encontrada, pois se trata de algo crítico para a empresa.

A equipe de suporte técnico informou que o serviço *Webmin* não era mais utilizado há certo tempo e o serviço poderia ser desativado permanentemente. Outra correção possível foi a remoção do usuário, que possuía a senha fraca, pois o usuário não acessava mais o servidor analisado.

Após a correção da vulnerabilidade uma nova tentativa de obter as informações foi realizada através do *exploit*. Sendo que desta vez não poderia mais obter êxito na leitura de nenhum arquivo solicitado. Conforme a Figura 7 uma a nova tentativa de exploração realizada sem sucesso.

[10] *Root* - Usuário com privilégios irrestritos.

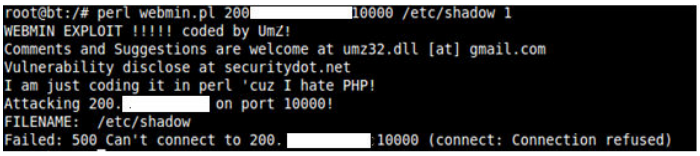

Figura 7. Falha ao explorar a vulnerabilidade

Foi observado que o *exploit* não consegue mais explorar a vulnerabilidade, sendo assim, verificou-se que a falha foi corrigida com sucesso.

5. Conclusão

Este trabalho teve como objetivo principal demonstrar a utilização do *pentest* em um estudo de caso em uma empresa. Podemos observar que a utilização de uma metodologia pode ser fundamental para realizar o processo de *pentest* de maneira organizada, compreensiva e obter resultados significativos para a segurança da informação.

Destaca-se que o *pentest* pode ser fundamental para conhecer a organização, entender sua atividade fim e encontrar dados sensíveis que estão disponíveis publicamente na *Internet* e que é possível encontrar esses dados com uma simples pesquisa sobre uma pessoa. Outras informações importantes que podem ser coletadas são sobre o funcionamento e a infraestrutura computacional de uma organização.

Foi observado durante o *pentest* realizado que com apenas uma única ferramenta de detecção automatizada de falhas e vulnerabilidades é possível coletar inúmeras informações significativas que podem revelar brechas na segurança a serem exploradas e possibilitar o roubo de informações críticas.

Com o *pentest* também é possível encontrar portas abertas e serviços não utilizados onde, geralmente, são aproveitados por pessoas com atitudes maliciosas para ataques únicos ou como *backdoors* [11] para roubo posterior de informações mais complexas. A falta de atenção ou desconhecimento técnico pode ser também um fator determinante para inferir na segurança da informação da empresa, pois permitem que serviços vulneráveis e passíveis de exploração fiquem visíveis.

6. Referências

ENGEBRETSON, P. **The basics of hacking and penetration testing: ethical hacking and penetration testing made easy**. Waltham: Syngress, 2011.

KENNEDY, O'GORMAN, KEARNS e AHARONI. **METASPLOIT The Penetration Tester's Guide.** São Francisco: No Starch Press, 2011.

ASSUNÇÃO, M. F. **Segredos do Hacker Ético**. Florianópolis: Visual Books, 2011.

THOMAS, T. **Segurança em redes – Primeiros Passos**. Rio de Janeiro: Editora Ciência Moderna, 2007.

[11] *Backdoors* - Ferramenta onde o atacante instala no alvo afetado a fim de retornar a qualquer momento.

SIMPSON, BACKHAM e CORLEY. **Hands-On Ethical Hacking and Network Defence**. Boston: Course Technology PTR, 2013.

FAIRCLOTH, J. **Penetration Tester's Open Source Toolkit**. Waltham, Massachusetts: Syngress, 2011.

BOYLE, R. e PANKO, R. **Corporate Computer Security**. New Jersey: Pearson, 2012.

PTES. ***Technical Guidelines***, 2012. Disponivel em: <http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines>. Acesso em: 10 novembro 2012.

FERREIRA, F.; PEREIRA, J. **Revista Segurança Digital - 1ª Ed, Junho 2011**, 2011. Disponivel em: <http://www.segurancadigital.info/sdinfo_downloads/1_edicao_julho_01_07_2011.php>. Acesso em: 05 novembro 2012.

PATERVA. **Maltego**, 2012. Disponivel em: <http://www.paterva.com/web6/products/download.php>. Acesso em: 03 janeiro 2013.

WAEYTENS, F.; HAROUNI, D. **Dnsenum: enumerating DNS info about domains**, 2011. Disponivel em: <http://dnsenum.googlecode.com/files/dnsenum-1.2.2.tar.gz>. Acesso em: 15 janeiro 2013.

NMAP.ORG. **Nmap Free Security Scanner For Network Exploration & Hacking**, 2012. Disponivel em: <http://nmap.org/>. Acesso em: 20 janeiro 2013.

NESSUS. **Nessus: The Network Vulnerability Scanner**. 2013. Disponivel em: <http://www.nessus.org/nessus/>. Acesso em: 30 janeiro 2013.

ÖZKAN, S. **CVE-2006-3392**. 2006. Disponivel em: <http://www.cvedetails.com/cve-details.php?t=1&cve_id=cve-2006-3392/>. Acesso em: 10 março 2013.

METASPLOIT. **Metasploit – penetration testing resources.** 2013. Diponivel em: <http://www.metasploit.com/ >. Acesso em: 15 janeiro 2013.

EXPLOIT-DB. **Google Hacking Database. 2012.** Disponivel em: <http://www.exploit-db.com/google-dorks>. Acesso em: 05 janeiro 2013.

REGISTRO.BR. **Serviço de diretório whois. 2012.** Disponivel em: <https://registro.br/cgi-bin/whois/>. Acesso em: 03 janeiro 2013.

BORGES, C.; HELENA, E. Estudo comparativo de metodologias de *Pentests*. 2011. Disponivel em: <http://www.cristiano.eti.br/Documentos/Artigo-Pentest-Cristiano.pdf>. Acesso em: 20 Março 2013.

FERREIRA, ROCHA, MARTINS, FEITOSA e SOUTO. Análise de vulnerabilidades em Sistemas Computacionais Modernos: Conceitos, *Exploits* e Proteções. Disponivel em: <http://dainf.ct.utfpr.edu.br/~maziero/lib/exe/fetch.php/ceseg:2012-sbseg-mc1.pdf>. Acesso em: 21 Março 2013.