Análise da Vulnerabilidade de Sites Web

Débora Lúcia Silva Cordeiro[1], Sandro Roberto Fernandes[2]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas – Campus Juiz de Fora

debora_lsc@hotmail.com, sandro.fernandes@ifsudestemg.edu.br

Abstract. *This article provides an overview of vulnerabilities that can be found on Web Sites through intrusion testing. We used the Nmap and Armitage tools available in Kali Linux. Armitage is an interface that aggregates Metasploit functions in a way that makes it easier to use. Providing access and possible corrections to the security that was used in the site which was tested. This framework makes it possible to learn several security techniques that allow us to find and test vulnerabilities, and can even help developers create software with greater security, since it is possible to determine security vulnerabilities in the program before reaching the final phase. Nmap brings the scanned targets according to the options used, such as the port table that displays the state of the port, details of the software version, possible operating systems, and types of devices.*

Resumo. *Este artigo apresenta uma visão geral das vulnerabilidades que podem ser encontradas em Web Sites por meio de testes de intrusão. Foram utilizadas as ferramentas Nmap e Armitage disponíveis no Kali Linux. Onde o Armitage trata-se de uma interface que agrega funções do Metasploit de forma que facilita a utilização da ferramenta. Disponibilizando acesso e possíveis correções na segurança que foi empregada no site a qual foi testado. Esse framework possibilita aprendizagem de diversas técnicas sobre segurança que nos permite encontrar e testar as vulnerabilidades, podendo auxiliar até mesmo desenvolvedores a criar softwares com maior segurança, uma vez que é possível determinar falhas de segurança no programa antes de chegar à fase final, corrigindo-as previamente. O Nmap traz os alvos escaneados de acordo com as opções utilizadas como, por exemplo, a tabela de portas que exibe o estado dela, detalhes da versão do software, possíveis sistemas operacionais e tipos de dispositivos.*

1. Introdução

As aplicações *web* estão cada vez mais presentes em nosso cotidiano, pois viabilizam o acesso a informação, sendo um dos principais meios de comunicação, relacionamento e também negócios. O que se pode notar é que a *Web* foi criada sem grandes restrições de segurança, como por exemplo, uma política de segurança, que se trata de um instrumento importante para proteger a sua organização contra ameaças à segurança da informação que a ela pertence ou está sob sua responsabilidade [TAIT 2007].

Segundo Caruso e Steffen (1999) política de segurança é:

> *"Um conjunto de diretrizes gerais destinadas a governar a proteção a ser dada a ativos da companhia. As consequências de uma política de segurança implementada corretamente podem ser resumidas em três aspectos: Redução da probabilidade de ocorrência; Redução dos danos provocados por eventuais ocorrências; criação de procedimentos para se recuperar de eventuais danos."*

Uma ameaça a segurança é compreendida neste contexto como a quebra de uma ou mais de suas três propriedades fundamentais: confidencialidade, integralidade e disponibilidade. [GOODRICH e TAMASSIA 2013].

A Internet abrange uma larga escala de sites e servidores que uma vez invadidos podem causar grandes danos a empresas, instituições; como perda de informações e também perda de dados pessoais. É importante avaliar como a segurança deve ser empregada para que as informações contidas não sejam comprometidas e usadas de forma indevida.

A definição de vulnerabilidade dada pelo NIST (*National Institute of Science and Technology*) é: "um ponto fraco em um sistema de informação, nos procedimentos de segurança de um sistema, nos controles internos de uma implementação, e que pode ser explorado por uma fonte de ameaças". Vale destacar que pressões e prazos podem estar ligados a processos falhos de desenvolvimento resultando em maior taxa de falhas nas aplicações. A qualificação técnica dos desenvolvedores contando com revisões constantes para alcançar melhorias também devem ser verificadas.

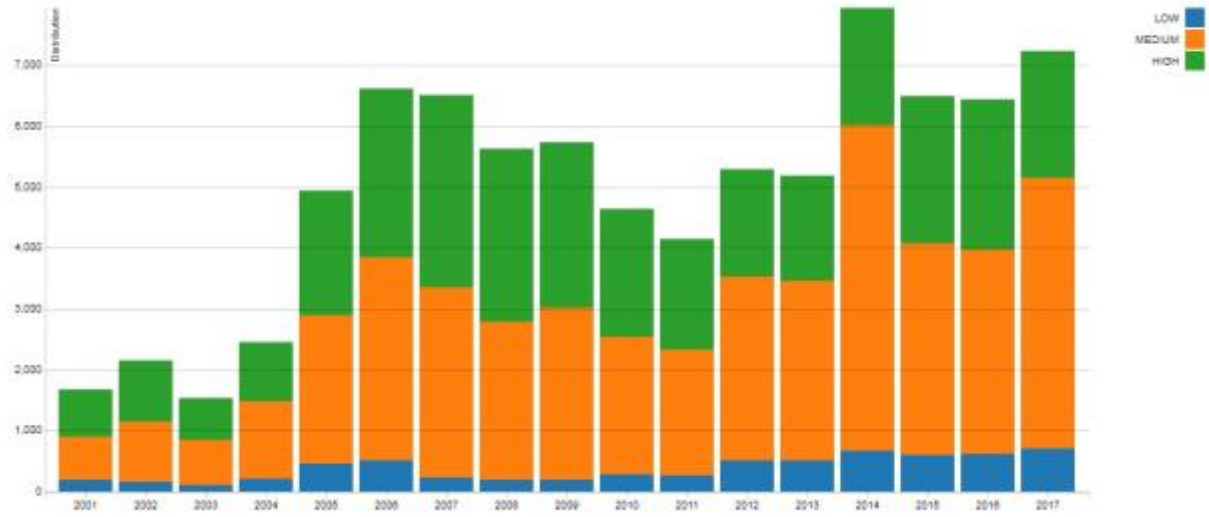

Figura 01: Tela de Vulnerabilidades distribuída ao longo dos anos. Fonte: https://nvd.nist.gov/ncp/visualizations/ncp-distribution-over-time

De acordo com a Figura 01 podemos notar vulnerabilidades ocorridas ao longo de um período de tempo, e como evoluíram. Foram distribuídas em baixas, médias e altas.

Pode-se observar, de forma geral para o período apresentado, que todos os tipos de vulnerabilidades, tiveram um aumento.

Portanto, uma forma para compreender melhor essas vulnerabilidades é estudar e analisar principais erros cometidos desde a fase de desenvolvimento das aplicações e configuração dos serviços e servidores Web. Assim minimizando custos futuros com manutenção relacionadas com a segurança, uma vez que os sistemas Web são alvos de frequentes tentativas de invasões. Este trabalho apresenta testes feitos Sites Web e servidores, utilizando ferramentas pré-selecionadas para identificar as falhas presentes e assim proporcionar sugestões de correções.

2. Fundamentos Teóricos

A definição de Segurança da Informação refere à proteção de determinados dados com a intenção de preservar valores para uma organização ou um indivíduo.

A norma ABNT NBR ISO/IEC 1 7799:2005 define Segurança da Informação como "Preservação da confidencialidade, da integridade e da disponibilidade da informação; adicionalmente, outras propriedades, tais como autenticidade, responsabilidade, não repúdio e confiabilidade, podem também estar envolvidas".

A confidencialidade evita a divulgação não autorizada de informações. Envolve a proteção de dados, fornecendo acesso para aqueles que estão autorizados a vê-lo. Não permitindo que pessoas não autorizadas tenham acesso ao seu conteúdo. A integridade caracteriza-se pela propriedade de que a informação não tenha sido alterada de uma forma não autorizada. Já a disponibilidade é a propriedade de que a informação é acessível e modificável em tempo hábil por pessoas autorizadas a fazê-lo.

É relevante destacar a importância de se utilizar mecanismos de segurança e de armazenamento das informações como forma de sobrevivência dos sistemas. Deve-se avaliar o nível de Segurança da Informação existente nos sistemas, identificando mecanismos e ferramentas utilizadas e realizando os necessários testes de vulnerabilidades, dimensionamento assim o grau de risco ao qual está exposto o sistema, considerando o nível de Segurança da Informação constatado e os respectivos recursos envolvidos, tais como ambientes *hardware*, *software*, dados, pessoas, documentação e materiais.

Quando tratamos de mecanismos de segurança referimo-nos aos recursos disponíveis e que podem ser utilizados para oferecer os serviços.

2.1 *Kali Linux*

O *Kali Linux* é o *live disk* mais recente de uma distribuição de segurança disponibilizada pela *Offensive Security*. Atualmente é um dos sistemas mais famosos no mundo na área de Segurança da Informação. A versão utilizada no trabalho, Kali 2.0 (lançada em 11 de agosto de 2015), contém mais de trezentas ferramentas de segurança e de testes de invasão incluídas, classificadas em grupos úteis, mais frequentemente usadas por *pentests* (testes de intrusão) e outras pessoas que efetuam avaliações de sistemas de informação. Usa como base a distribuição *Debian 7.0*. [BROADE BINDER 2014]. Apesar de ser

distribuído como um Live CD, é possível sua instalação com o Sistema Operacional principal ou em um *dual boot* em um computador ou notebook; fazer a instalação em um *pendrive* ou instalar em uma máquina virtual.

Offensive Security oferece serviços avançados de testes de intrusão para redes de empresas. Oferece testes avaliados de acordo com a necessidade dos clientes. Tornou-se líder em treinamento reconhecido de Segurança da Informação baseada no seu desempenho, sendo autor de múltiplas explorações e várias ferramentas.

Com o *Kali Linux* é possível fazer *SQL Inject*, *Exploits*, *Sniffers*, *Scanner*, *Cracking*, quebra de senhas e ataques em geral. Bem como invasão de redes sem fio, sites e banco de dados. O Kali Linux tem as seguintes seções: *Information Gathering, Vulnerability Analysis, Wireless Attacks, Web Applications, Exploitation Tools, Forensics Tools, Stress Testing, Sniffing & Spoofing, Password Attacks, Maintaining Access, Reverse Engineering, Hardware Hacking*; onde cada uma dessas seções apresenta uma lista de ferramentas especificas. As duas utilizadas estão em *Vulnerability Analysis e Exploitation Tool;* Análise de Vulnerabilidade e Ferramentas de Exploração.

2.2 *Metasploit*

O *Metasploit* é umas das ferramentas mais eficientes do kit de ferramentas do *pentests*; ele carrega consigo recursos associados a anos de conhecimentos e de experimentos meticulosos efetuados por *hackers*, *pentests*, governos e pesquisadores de todo o mundo, que incluem diferentes partes da comunidade da comunidade de segurança de computadores. Para os profissionais que atuam ativamente na área, o *Metasploit* também oferece *templates* de relatórios e inclui verificações de aderência nos níveis exigidos pelo governo [BROAD E BINDER 2014].

O *Metasploit framework* é um conjunto das melhores plataformas de aprendizagem e investigação para o profissional de segurança ou do *hacker*. Ele possui centenas de *exploits, payloads* e ferramentas muito avançadas que nos permite testar vulnerabilidades em muitas plataformas, sistemas operacionais, e servidores.

Ferramentas como o *Metasploit* podem ajudar aos desenvolvedores de software produzirem software com maior qualidade, do ponto de vista da segurança, levando e incentivando os programadores a pensarem a respeito de como algumas técnicas de programação levam as falhas de segurança.

2.3 *Armitage*

O *Armitage* é uma ferramenta para o *Metasploit* disponível no *Kali Linux* que visualiza alvos, recomenda *exploits* e expõe os recursos avançados de pós-exploração no *framework*. Ele organiza as capacidades do *Metasploit* em torno do processo de *hacking*. Existem recursos para descoberta, acesso, pós-exploração e manobra. Também lança varredura e importa dados de muitos *scanners* de segurança, visualiza seu destino atual para que você saiba os hosts com os quais está trabalhando. O *Armitage* recomenda explorações e executa verificações ativas para indicar quais *exploits* funcionará.

Com *Armitage*, é possível localizar as máquinas que funcionam em uma rede, incluindo a versão do sistema operacional em execução em cada dispositivo. Com essa informação, *Armitage* fornece uma lista completa dos métodos de ataque com potencial de sucesso para cada versão do sistema operacional. Oferece ainda a capacidade de tirar fotos de webcam e teclas de registro nas máquinas das vítimas, permitindo uma avaliação mais completa da superfície de ataque de uma organização. A figura 02 mostra um teste feito pelo *Armitage* que apresenta um alvo vulnerável. Que desta forma se encontra de forma a ser invadido.

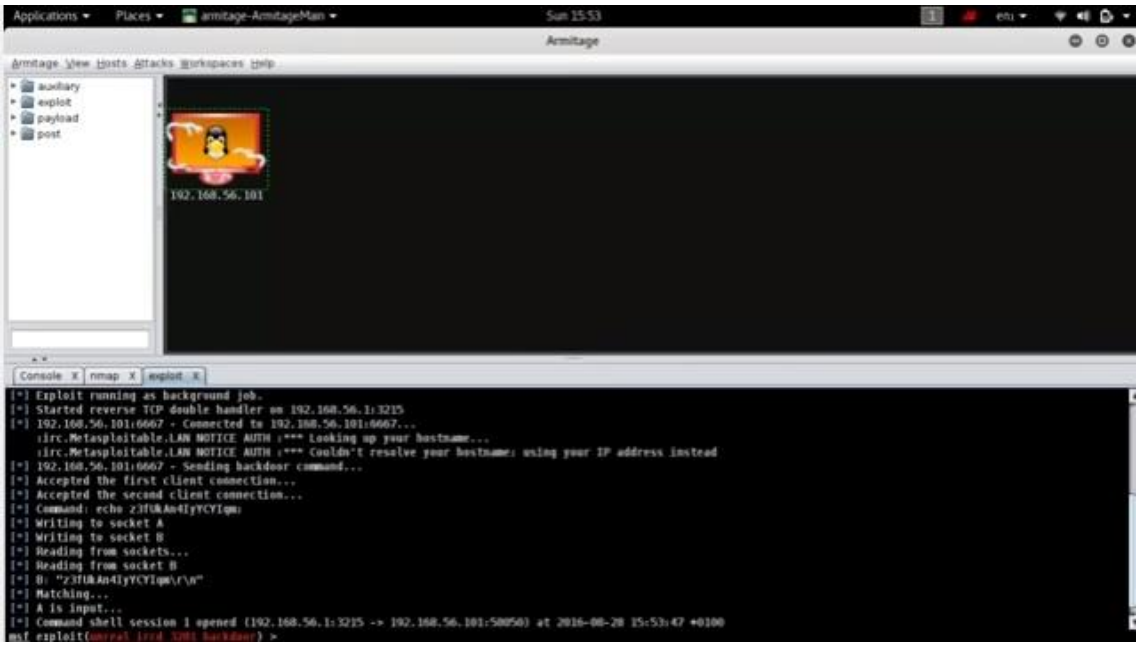

Figura 02 – Tela do Armitage apresentando um computador na rede e/ou Internet com vulnerabilidades encontradas. Fonte: Kali Linux - Armitage Tutorial

2.4 *Nmap*

Foi criado para descoberta de rede e auditoria de segurança. Também utilizado para gerenciamento de andamentos de atualização de serviços e atualização e monitoração de tempo de atividade de host ou serviço. Ele usa pacotes de IP para determinar quais hosts estão disponíveis na rede, quais serviços estão oferecendo, quais sistemas operacionais estão sendo executados e sua versão, tipos de filtros, pacotes estão em uso e uma série de outras características. Ele foi projetado para *escanear* rapidamente grandes redes, podendo ser executados em todos os principais sistemas operacionais.

O Nmap foi primeiramente publicado em setembro de 1997, em um artigo na revista *Phrack* com o código fonte incluso. Com a ajuda e contribuições da comunidade de segurança de computadores, o desenvolvimento continuou. Atualizações do programa incluem detecção do sistema operacional, detecção de serviço, código reescrito de C para C++, tipos adicionais de *scanning*, suporte a novos protocolos e novos programas que complementam o núcleo do *Nmap*.

O *Nmap* é uma ferramenta que possui técnicas avançadas para mapear redes com filtros IP, *firewalls* e roteadores. Permite a varredura de portas TCP e UDP. Pode chegar a escanear uma rede com milhares de computadores, podendo perfeitamente ser utilizado na Internet. Suporta a maioria dos sistemas operacionais incluindo *Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS, e etc.* Oferece um rico conjunto de recursos avançados para usuários mais experientes e conta também com versão mais simples. Tanto as versões tradicionais como gráficas estão disponíveis para se adequar as preferências. Conta ainda com tutorias disponíveis em páginas manuais e abrangentes, além de oferecer um vasto suporte, disponibilizando canais para solucionar relatórios de erros através de inscrição dos usuários no mesmo.

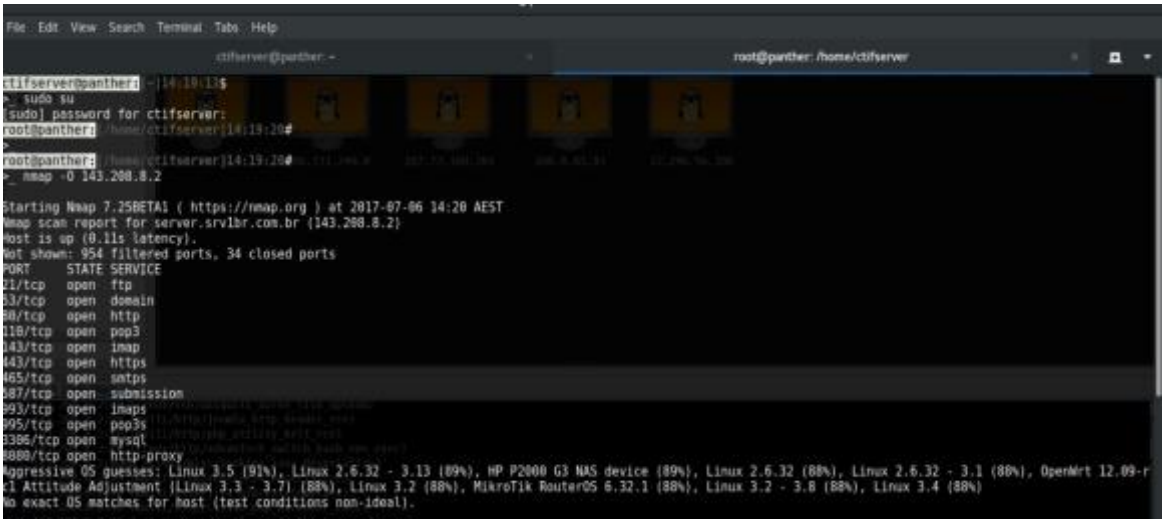

Figura 03 – Tela de Scan do Nmap utilizada na fase de escanear em um dos sites escolhidos. Fonte: o autor.

3. Metodologia

A abordagem utilizada para explorar falhas utiliza as fases de Reconhecimento, *Scanning*, Exploração de Falhas e Preservação do Acesso. Estas etapas foram concebidas por Patrick Engebretson [BROAD e BINDER, 2014].

- Reconhecimento: obter informações sobre os sites que serão alvos do processo;
- *Scanning*: Usa-se as informações obtidas no passo anterior e possibilita uma melhor definição da rede e da infraestrutura do sistema;
- Exploração de Falhas: O propósito desta fase é determinar como o sistema alvo poderia ser invadido. Nesta etapa não será utilizada nenhuma ação que seja considerada ilegal;
- Preservação de Acesso: Uma vez que as falhas foram exploradas, ferramentas são deixadas no sistema para garantir o acesso. Esta etapa não será utilizada neste trabalho;

A exploração de falhas foi testada em Web Sites e servidores Web e em uma simulação. Foram utilizados para teste um total de cinco sites. Dentre estes foram sites regionais e nacionais; governamentais e particulares. A fim de se estabelecer uma comparação por isso optamos por perfis de sites variados.

3.1 Utilizando NMAP

Inicialmente foi feito um *scan* utilizando a ferramenta *Nmap*, onde foi filtrado opções para exploração dos alvos, dentre as principais estão:

- Porta 80 e 8080, para uso de servidores web;
- Serviços de *HTTP,HTTPS*;
- Ferramenta de gerenciamento de conteúdo, Drupal;
- *Firewall-bypass*, que funciona por um pacote do servidor de destino pedindo abertura de uma conexão relacionada a uma porta de destino que será cumprida pelo *firewall* através do protocolo adequado;
- *Apache Server Status*, que permite que um administrador do servidor descubra como seu servidor está funcionando;
- *Wordpress-brute*, que se refere a ataque de força bruta, baseando em tentativas;
- *http-exif-spider*, procura dados interessantes em sites incorporados em arquivos jpg;
- *http-xssed* que pesquisa o banco de dados;
- *Whois-domain* que faz tentativas de recuperar informações sobre o nome de domínio do alvo.

A lista completa dos parâmetros utilizados no *Nmap* é apresentada na Figura 02 abaixo:

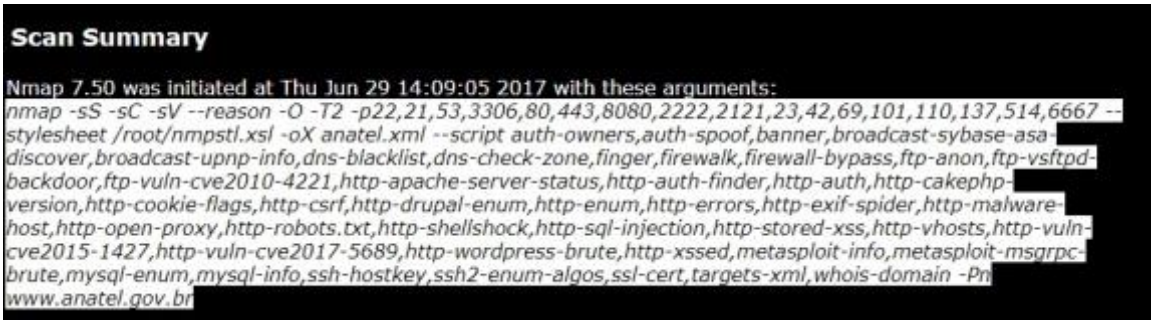

Figura 04 – Tela dos filtros utilizados para Scan no Nmap. Fonte: o autor.

O resultado Nmap é apresentando sem uma interface gráfica no Kali Linux. Para termos um resultado mais eficiente visualmente, foi criada uma interface utilizando a linguagem HTML. Assim foi separado os resultados em seções que facilitam sua compreensão: *Address, Hostnames, Ports, Remote Operating System Detection, Host Script Output* e *MiscMetrics*.

Dos cinco sites analisados, foi escolhido, arbitrariamente, um para apresentar em detalhes o resultado. Estes resultados estão nas imagens abaixo e a explicação de cada tela é feita em seguida a sua apresentação.

Figura 05: Primeira Tela de Resultado Nmap. Fonte: o autor

De acordo com a Figura 05 inicialmente temos o endereço IP do site escaneado e em seguida como o site é exibido para navegação.

Desta forma temos uma lista dos serviços que estão ativos neste servidor. Também temos a informação de qual serviço/porta que está aberto e/ou fechado e as portas que estão filtradas.

Onde open (aberto) é uma aplicação que está ativamente aceitando conexões TCP ou pacotes UDP nesta porta. Encontrar esse estado é frequentemente o objetivo principal de um escaneamento de portas. Dessa maneira fica mais suscetíveis a ataques. Invasores e profissionais de avaliação de segurança tentam explorar as portas abertas, enquanto os administradores tentam fechar ou proteger com firewalls sem bloquear usuários legítimos. Portas abertas são também interessantes para scans não-relacionados à segurança, pois mostram os serviços disponíveis para utilização na rede.

Closed (fechado) é uma porta fechada que está acessível (ela recebe e responde a pacotes de sondagens do *Nmap*), mas não há nenhuma aplicação ouvindo nela. Elas podem ser úteis para mostrar que um host está ativo em um determinado endereço IP (descoberta de hosts, ou *scan* usando *ping*), e como parte de uma detecção de SO.

Filtered (filtrado) é quando o *Nmap* não consegue determinar se a porta está aberta porque uma filtragem de pacotes impede que as sondagens alcancem a porta. A filtragem poderia ser de um dispositivo firewall dedicado, regras de roteador, ou um software de firewall baseado em *host*. Essas portas inibem ataques uma vez que fornecem poucas informações. Podendo às vezes responder com mensagens de erro ICMP (destino

inalcançável: comunicação proibida administrativamente), mas os filtros que simplesmente descartam pacotes sem responder são bem mais comuns. [LYON, 2017]

São exibidas sobre o site testado uma relação de portas que são 21, 22,23, 42,53,69 e 80. Onde 21 o FTP é um dos protocolos de transferência mais antigos e ainda assim um dos mais usados. Uma das observações acerca da segurança que pode ser feita é que as senhas trafegam em texto puro e podem ser capturadas por qualquer um que tenha acesso a transmissão. A porta 22 é utilizada para o serviço de SSH. A porta 23, *Telnet* é um protocolo completamente aberto que transmite *login*, senha e todos os comandos em texto puro, sendo simples capturar a transmissão e assim invadir o servidor. A porta 25, SMTP é o protocolo padrão para envio de e-mails que é usado tanto para o envio de mensagens original quanto para transferir mensagens para outros servidores. A porta 53, UDP DNS são responsáveis por converter nomes de domínios nos endereços IP dos servidores. A porta 69, UDP TFTP utiliza portas UDP para a transferência de arquivos em geral, usada em sistemas de boot remoto. A porta 80é o principal protocolo da internet usado para acesso a páginas web. Havendo um detalhamento sobre a porta ela que se encontra aberta.

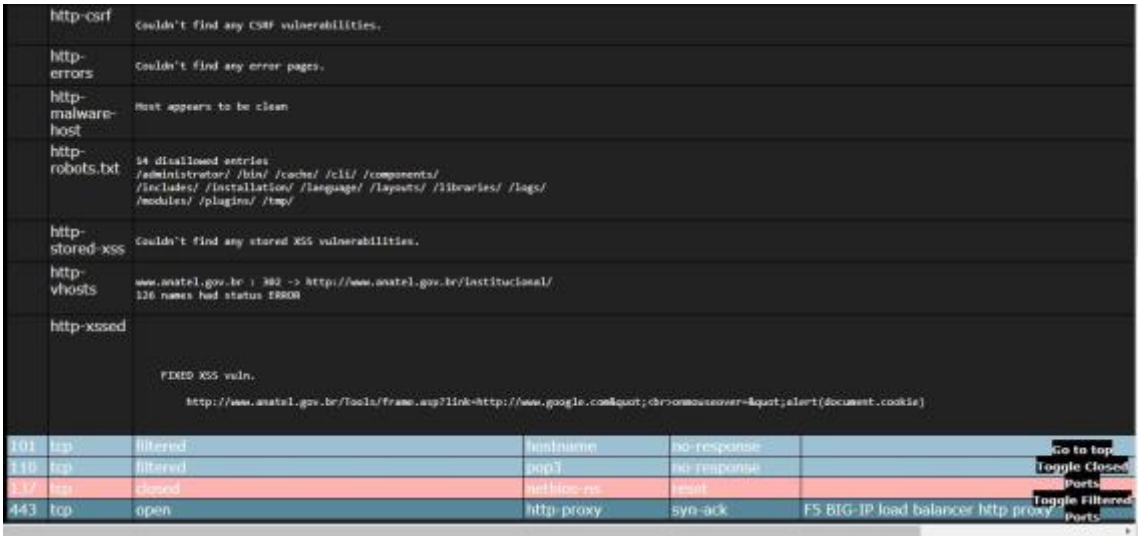

Figura 06: Tela de Resultado de scan Nmap. Fonte: o autor

Na Figura 06 apresenta a continuação da análise da porta 80. Não foram encontradas vulnerabilidades para *o http-crsf* de acordo com o teste. Este é um ataque que obriga um usuário final a executar ações indesejadas em um aplicativo web no qual eles estão atualmente autenticados. Este tipo de ataque visa especificamente os pedidos de mudança de estado, e não roubo de dados. Também não foram encontradas *http-erros,* sobre *http-malware-host* também demonstrou estar limpo. Sobre *http-robots* que funciona como um filtro para sites de busca e controla permissões de acesso, foram encontradas 14 entradas não permitidas como *plugins, includes, layouts*. O *http-stored-xss* também não encontrou vulnerabilidades. *Http-vhosts* foram identificados 126 nomes que mostraram *status* de erro. O *http-xssed* pesquisou no banco de dados e apresentou um link onde pode haver vulnerabilidades.

Também foram apresentadas as portas 101 de *hostname* que está sendo filtrada, a 110 de pop3 que é um protocolo para acesso remoto a caixa de correio também se encontra

filtrado, a porta 137 que se trata netbios se encontra fechada e a 443 que é de *http-proxy*se encontra aberta.

Figura 07: Tela de Resultado de Scan do Nmap. Fonte: o autor.

Para a porta 443, a análise de *Http-csrf* não encontrou vulnerabilidades, *http-erros* não encontrou página de erros. Http-server-header apresentou *BigIp* como vulnerabilidade que se trata de um tipo de suporte de Web Sites. *Http-vhosts* mostrou dados de 125 nomes com erro. *Http-xssed* mostrou mais um *link* vulnerável. *Ssl-Cert* exibe dados sobre a licença do Web Site como tipo de chave pública, algoritmo de assinatura e período de expiração do certificado.

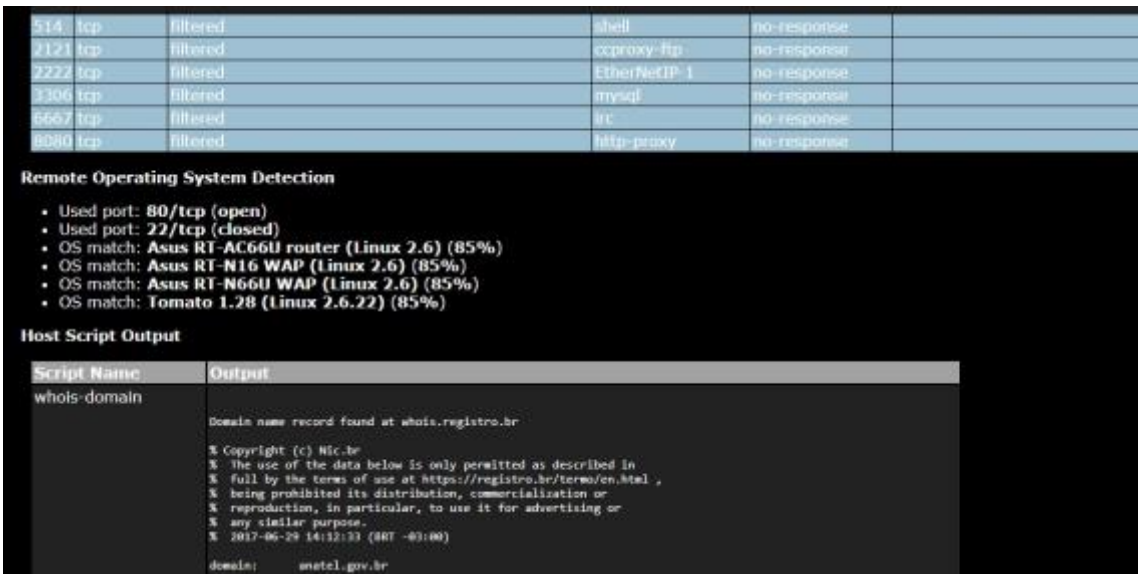

Figura 08: Tela de Resultado de Scan do Nmap. Fonte: o autor.

A figura 08 mostra as portas 514 de *Shell* que se encontra filtrada, a 2121 de *ccproxy* filtrada, a 2222 de *EtherNet* no status de filtrada, 3306 de *mysql* filtrada, 6667 de *IRC* filtrada e 8080 de http-proxy filtrada.

Também informa a porcentagem para que se analise de qual sistema operacional se trata, que como se pode observar há uma porcentagem de 85% de ser uma kernel Linux 2.6. Logo em seguida, inicia-se a descrição sobre o domínio do *Web Site.*

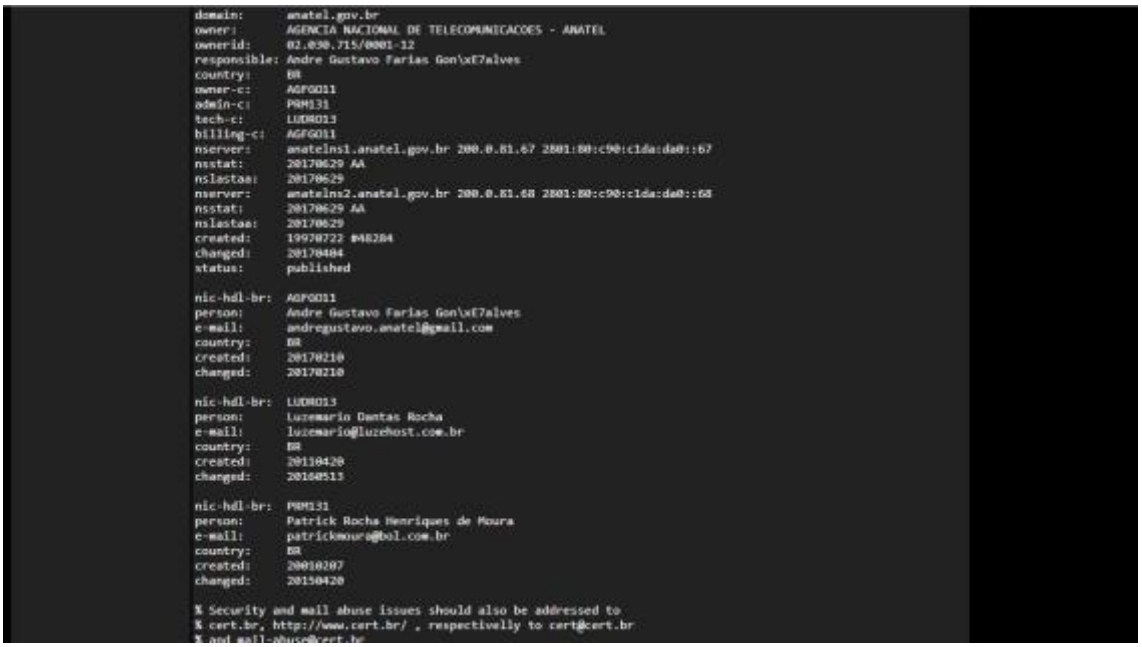

Figura 09: Tela de Resultado de Scan do Nmap. Fonte: o autor.

A figura 09 apresenta informações que mostra que o registro do domínio foi encontrado, por onde é fornecido o serviço de registro, o nome do domínio e informações como contatos de quem registrou esse domínio. Nome de servidores, quando foi criado e quando expira o domínio.

3.2 Utilizando o *Armitage*

Foi iniciado o *Armitage*, foram adicionados os hosts dos alvos as serem testados. É feito um *scan* inicial onde é possível verificar quais portas estão abertas e também o sistema operacional. Como se encontra na figura.

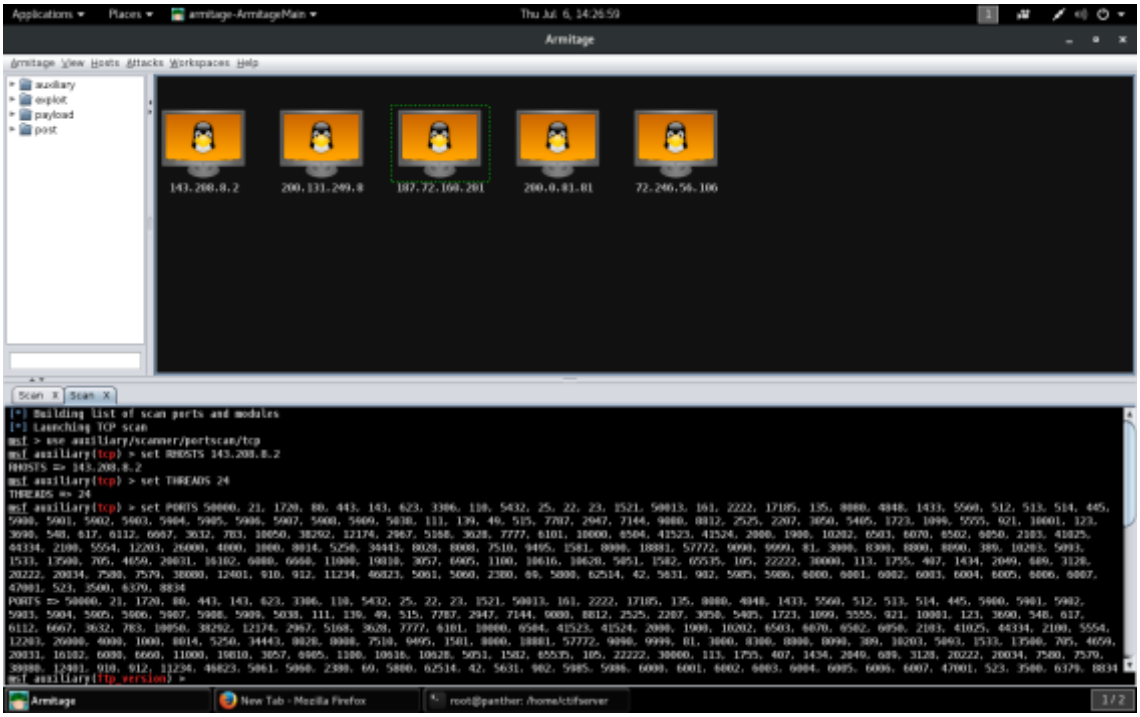

Figura 10: Tela de Resultado Armitage. Fonte: o autor.

Em seguida é utilizada a opção disponível no *Armitage* chamada *Hall Mary* que tenta executar vários *exploits* com a finalidade de encontrar alguma vulnerabilidade, exibindo assim uma lista da quantidade dos *exploits* encontrados que podem vir a ser executado contra esse alvo.

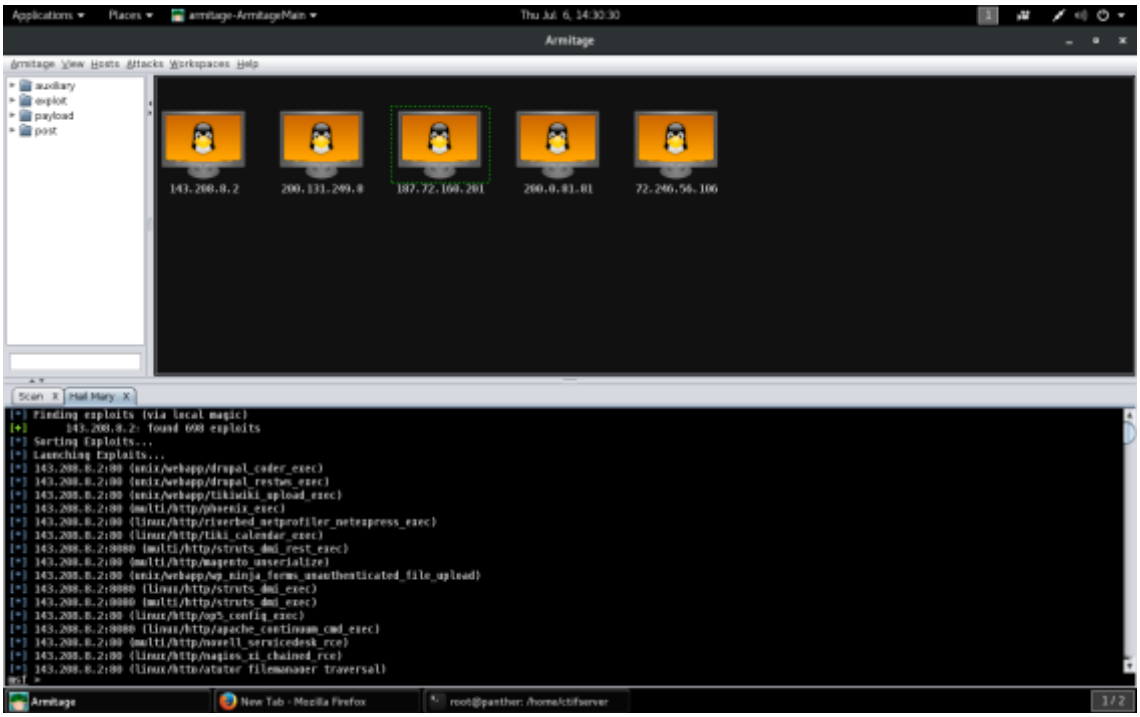

Figura 11: Tela de Resultado de Exploits encontrados pelo Armitage. Fonte: o autor

4. Resultados

Durante este teste utilizando *Armitage* foram encontrados 698 *exploits* para os cincos Web Sites testados, porém nenhum dos apresentados se encontrava vulnerável a ponto de ser feito acesso da máquina.

Sobre os testes feitos no *Nmap* os dados encontrados estão listados abaixo contando com informações do tipo de Site, quantas portas foram filtradas, quais as portas foram encontradas abertas e fechadas e qual porcentagem do sistema operacional.

Sites	Nº Portas filtradas	Portas Abertas	Portas Fechadas	Sistema Operacional
1 – Particular/ Nacional	954	21, 53, 80, 110, 143, 443, 465, 587, 993, 3306, 8080	___	91% - Linux 3.5
2–Público/ Regional	995	___	22,25,139, 445,1723	94% - Linux 2.6
3 –Publico/ Regional	992	80, 443	22,25, 110, 139,445, 1723	92% - Linux 3.19
4 – Particular/Nacional	992	80, 443	22, 137	85% - Linux 2.6
5– Regional/ Particular	993	80, 443	25,25,139,445, 1723	90% - Linux 2.6

Pode-se notar que em relação ao Site 1 é necessária uma correção no Site avaliando questões que podem estar em risco. De acordo com a tabela, há portas abertas onde qualquer pessoa que tenho acesso a transmissão possa capturar algum tipo de dados.

Existem também abertas portas que são responsáveis por armazenar *e-mails*, onde tudo deve ser bem avaliado para não comprometer o andamento da empresa que o utiliza. Também é observado que houve um site particular que apresentou maior vulnerabilidade, um público que não continha nenhuma porta aberta. E outros três que apresentaram apenas as portas 80 e 443 abertas.

Para demonstrar um teste de vulnerabilidade de um sistema, durante nosso trabalho foi criado um cenário de intrusão: um laboratório contendo uma máquina apresentando maiores vulnerabilidades e outra contendo a ferramenta *Armitage* que é o responsável por viabilizar a demonstração da intrusão. O resultado mostra que é possível obter acesso da mesma.

5. Conclusões e Trabalhos Futuros

Os testes de intrusão têm por finalidade verificar de maneira efetiva e preventiva as principais ameaças e pontos fracos de um sistema antes de ser implantado. Desta forma, pretende-se minimizar os riscos e ameaças que o sistema pode sofrer. É importante que haja uma conscientização relacionada as falhas de segurança, tendo em vista que com um conjunto maior de informações relativas ao sistema utilizado, a tendência é que sua segurança seja feita de forma mais segura.

Fica claro que existem fatores que contribuem para que sempre existam falhas, até mesmo aquelas consideradas de média proporção. Dessa forma pode ser de suma importância treinar os administradores de rede, administradores de servidores e desenvolvedores de software para que a infraestrutura e programação seja feita de forma segura pretendendo que as aplicações não apresentem vulnerabilidades, eventualmente antes de ser colocadas em uso.

Os estudos realizados permitiram identificar características importantes em um *scanner* de vulnerabilidades. Tanto o *Armitage* quanto o *Nmap* visam melhorar e tentar manter a qualidade dos softwares.

Pode-se concluir esses estudos que é possível garantir a sistemas mais segurança a usuários, onde o ambiente de produção pode se tornar de maior confiável trazendo um menor risco com problemas futuros. Um dos cuidados a serem tomados são testes de intrusão visando verificas falhas no sistema, para, posteriormente, corrigi-los.

Em atividades futuras pode-se incluir mais sites particulares de empresas visando identificar em quais ciclos de desenvolvimento foram introduzidas essas falhas e propor as melhorias. Esta mesma metodologia também pode ser usada em uma rede local, visando a análise de vulnerabilidades internas.

Referências

ARAUJO, Nonata Silva, (2008) “Segurança da Informação (TI)”, Disponível em: <http://www.administradores.com.br/artigos/tecnologia/seguranca-da-informacao-ti/23933/>. Acesso em: Abril 2017.

BROAD, James; BINDER Andrew. (2014) “Hacking com Kali Linux”. São Paulo, Novatec.

CARUSO, Carlos A. A., STEFFEN, Flavio Deny. “Segurança em Informática e de Informações” São Paulo. Senac/SP, 1999.

FREITAS, Vanderlei Junior; SILVA, Tales do Nascimento; NUNES, Lucyene Lopes da Silva Todesco; LUZ; Gerson Luis. “Tecnologia e Redes de Computadores”, página 148.

GOERING, Richard. “Scan design Called Portal For Hackers”.

GOODRICH, Michael T.; Tamassia, Roberto. “Introdução à Segurança de Computadores”. Bookman Companhia Editora. 2013.

HACKING, RedPill, (2016) “Kali Linux - Armitage Tutorial (part1)”, disponível em: <https://www.youtube.com/watch?v=FHbthOR9q3w>. Acesso em: Abril 2017.

LIMA, Gustavo, (2011), “Armitage + Metasploit = Hackerismo com facilidade”, Disponível em: <http://blog.corujadeti.com.br/armitage-metasploit-hackerismo-com-facilidade/>. Acesso em: Abril 2017.

LYON, Gordon. (2017) “Nmap: The Network Mapper”, https://nmap.org, Junho 2017.

Metasploit e Armitage no Kali Linux - Hackeando sua rede. Disponível em: <https://under-linux.org/entry.php?b=3031> Acesso em: junho 2017.

Armitage. Disponível em: <https://www.offensive-security.com/metasploit-unleashed/armitage> Acesso em: abril 2017.

MONTEVERDE, Wagner Aparecido; CAMPIOLO Rodrigo. “Estudo e Análise de Vulnerabilidades Web”.

MORIMOTO, Carlos E. (2008) “Redes, Guia Prático” 2ª Ed. Disponível em: <http://www.hardware.com.br/livros/redes/portas-tcp-udp.html>. Acesso em: Maio2017.

ROCHA, Douglas; KREUTZ, Diego; TURCHETTI Rogério. “Uma Ferramenta Livre e Extensível Para Detecção de Vulnerabilidades em Sistemas Web”.

TOLEDO, Alex Sanader de Oliveira; SOUZA, Vinícius Augusto Celestino. **“**METASPLOIT: um cenário de intrusão**”.**

“Manual Fastandeasyhacking”. Disponível em: http://www.fastandeasyhacking.com/manual>. Acesso em: Maio 2017.