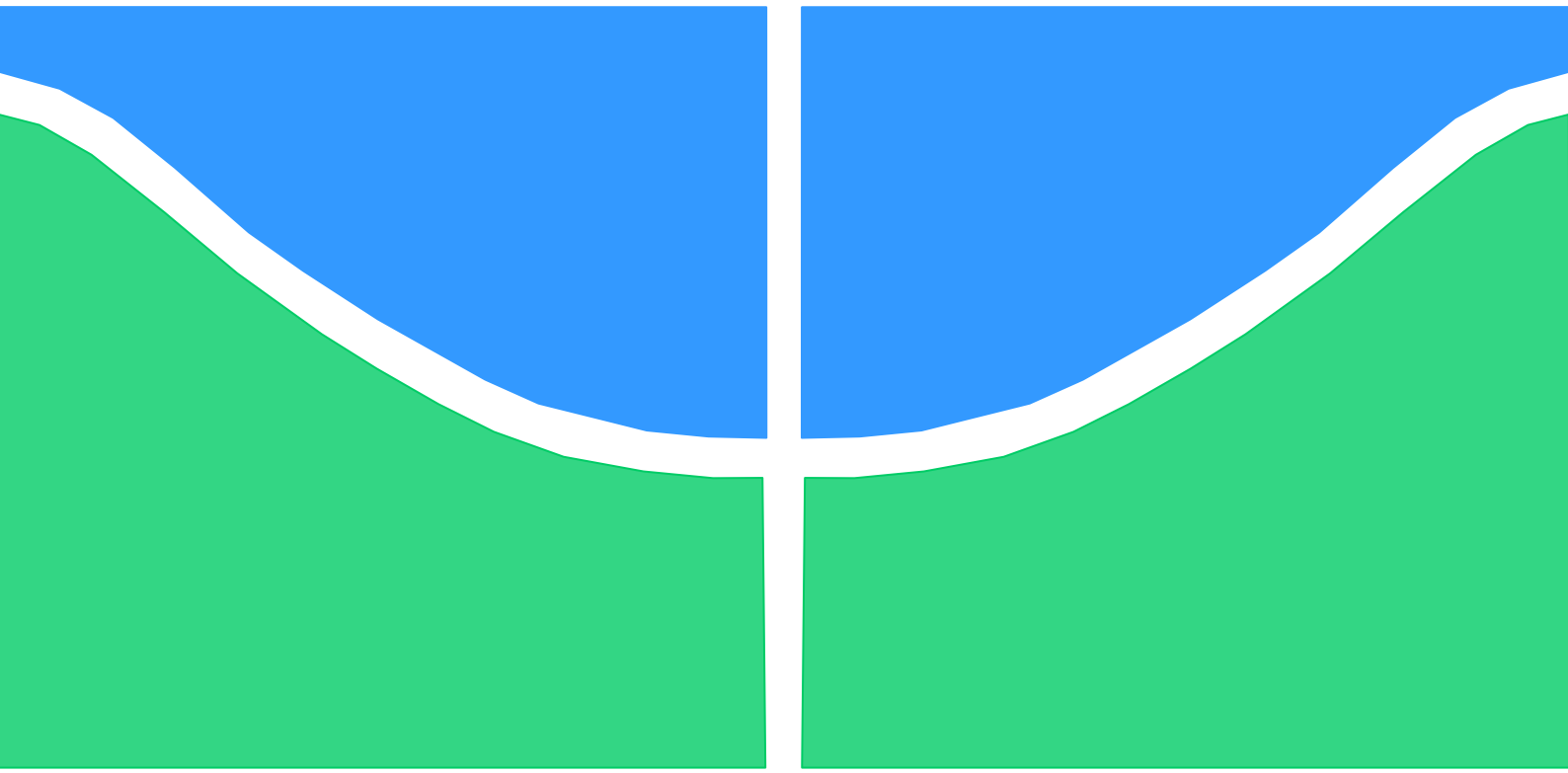


TRABALHO DE GRADUAÇÃO

PENTEST, ANÁLISE E MITIGAÇÃO DE VULNERABILIDADES

Alexandre Mendes Alvim Lepesqueur
Italo Diego Rodrigues Oliveira

Brasília, março de 2006

UNIVERSIDADE DE BRASILIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PENTEST, ANÁLISE E MITIGAÇÃO DE VULNERABILIDADES

ALEXANDRE MENDES ALVIM LEPESQUEUR
ITALO DIEGO RODRIGUES OLIVEIRA

ORIENTADOR: LAERTE PEOTTA DE MELO

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE REDES DE COMUNICAÇÃO

PUBLICAÇÃO: 2/2012
BRASÍLIA/DF: NOVEMBRO/2012

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PENTEST, ANÁLISE E MITIGAÇÃO DE VULNERABILIDADES

.

ALEXANDRE MENDES ALVIM LEPESQUEUR

ITALO DIEGO RODRIGUES OLIVEIRA

MONOGRAFIA DE GRADUAÇÃO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHARIA.

APROVADA POR:

Prof. Laerte Peotta de Melo, Doutor, UnB
(ORIENTADOR)

Profª. Edna Dias Canedo, Doutora, UnB
(EXAMINADOR INTERNO)

Prof. Giovanni Almeida Santos, Mestre, UnB
(EXAMINADOR INTERNO)

BRASÍLIA/DF, 13 DE NOVEMBRO DE 2012

FICHA CATALOGRÁFICA

LEPESQUEUR, ALEXANDRE MENDES ALVIM; OLIVEIRA, ITALO DIEGO RODRIGUES
Pentest, Análise e Mitigação de Vulnerabilidades [Distrito Federal] 2012.
xiv, 74p., 297 mm (ENE/FT/UnB, Grau Obtido, Engenharia de Redes de Comunicação, 2012).

Monografia de Graduação – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Pentests 2. Vulnerabilidades
3. Ataques
I. ENE/FT/UnB. II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

Lepesqueur, A. M. A.; Oliveira, I. D. R. (2012). Pentest, Análise e Mitigação de Vulnerabilidades. Monografia de Graduação em Engenharia de Redes de Comunicação. Publicação 11/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 85p.

CESSÃO DE DIREITOS

Alexandre Mendes Alvim Lepesqueur
CRS 516 Bl. C Ent. 10 Ed. Jardim Tropical Ap 304, Asa Sul
CEP 70.381-550 – Brasília – DF - Brasil

Italo Diego Rodrigues Oliveira
QNP 20 Conjunto K Casa 39, Ceilândia Sul
CEP 72.233-011 – Ceilândia – DF - Brasil

AGRADECIMENTOS

Agradeço a Deus por se fazer tão presente em minha vida. Aos meus familiares por toda força que me deram ao longo de todos esses anos de estudo. Ao meu orientador Prof. Dr. Laerte Peotta de Melo por todo o apoio fornecido. Aos meus amigos pessoais que sempre me forneceram suporte quando necessário. Ao colega Italo Diego por toda a ajuda prestada no desenvolvimento do trabalho.

Alexandre Mendes

Agradeço sobretudo a Deus por ter me abençoado ao longo de todos estes anos. Aos meus pais por todo suporte que me deram durante toda a vida acadêmica. Aos meus colegas de graduação que me ajudaram em diversos momentos de dificuldade. Ao meu orientador Prof. Dr. Laerte Peotta de Melo por ter aceitado orientar o projeto e por ajudar no desenvolvimento do tema. Ao colega Alexandre Mendes, que me auxiliou e ajudou no desenvolvimento e coordenação deste projeto final de graduação.

Italo Diego

RESUMO

PENTEST, ANÁLISE E MITIGAÇÃO DE VULNERABILIDADES

Autores: Alexandre Mendes Alvim Lepesquer e Italo Diego Rodrigues Oliveira
Orientador: Laerte Peotta de Melo
Monografia de graduação em Engenharia de Redes de Comunicação
Brasília, novembro de 2012

Penetrations Tests - PenTests, são basicamente um conjunto de diversos testes de vulnerabilidades realizados em uma rede ou em um sistema, onde hackers ou crackers procuram por vulnerabilidades que lhes forneçam informações que possibilitem a realização de ataques que lhes garantam acesso ao alvo almejado. Existem diversas ferramentas e scripts que automatizam e facilitam a vida do invasor. Uma dessas ferramentas é o BackTrack, um Sistema Operacional Linux próprio para a realização dos referidos testes.

O trabalho descrito nesta monografia objetiva, de uma forma prática, demonstrar todas as etapas necessárias para a realização de testes de vulnerabilidades. Com tal metodologia, pretende-se expor algumas das vulnerabilidades, as quais redes e sistemas possam vir a estar submetidos, os ataques que podem ser realizados em virtudes de tais falhas e por fim o estabelecimento de um conjunto de ações que possibilitem a mitigação de tais vulnerabilidades.

ABSTRACT

PENTEST, VULNERABILITY ANALYSIS AND MITIGATION

Authors: Alexandre Mendes Alvim Lepesquer e Italo Diego Rodrigues Oliveira
Supervisor: Laerte Peotta de Melo
Monografia de graduação em Engenharia de Redes de Comunicação
Brasília, november of 2012

Penetrations Tests - Pentests are basically a set of several tests performed on a network or a system, where hackers or crackers looking for vulnerabilities to provide information to enable the realization of attacks that guarantee access to desired target. There are various tools and scripts that automate and facilitate the life of the attacker. One such tool is the Backtrack, a Linux Operating System to perform this tests.

The work described in this monograph aims in a practical way to demonstrate all the steps necessary to perform vulnerabilities tests. This methodology is intended to expose some vulnerabilities, which networks and systems are likely to be subjected to attacks that could be performed in virtue of such failures and finally the establishment of a set of actions that enable mitigation of such vulnerabilities.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIAÇÕES

NAT	Network Address Translation
DMZ	Demilitarized Zone
NVD	National Vulnerability Database
CVE	Common Vulnerabilities and Exposures
MSB	Microsoft Security Bulletins
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
MAC	Media Access Control
NAT	Network Address Translation
IP	Internet Protocol
SMTP	Simple Mail Transfer Protocol
RADIUS	Remote Authentication Dial In User Service
ARP	Address Resolution Protocol
DLL	Dynamic Link Library
TCP	Transmission Control Protocol
SSL	Secure Sockets Layer
FTP	File Transfer Protocol

1 - INTRODUÇÃO

Com a crescente globalização e a expansão da Internet e da inclusão digital, a necessidade de implementação de medidas de segurança da informação tem crescido e se tornado cada vez mais expressiva. No mundo contemporâneo, o valor da informação é inestimável. Empresas, que grande parte das vezes tem o seu ramo de negócio e diferencial competitivo baseado unicamente em informações, e o próprio Governo, que contém documentos, dados e registros sigilosos, precisam adotar práticas que promovam a preservação e segurança de tais informações.

Infelizmente, boa parte das empresas só deixa para agir quando os problemas decorrentes de ataques e invasões já estão instaurados, não percebendo a importância de se manter uma defesa proativa. Tal defesa é baseada em testes de segurança regulares que identificam e solucionam possíveis vulnerabilidades, tornando, assim os sistemas de segurança mais confiáveis e robustos.

Dentre esses testes de segurança, destacam-se os pentests (acrônimo para testes de penetração), um conjunto de testes de vulnerabilidades que representam o tópico de pesquisa desse trabalho.

1.1 - OBJETIVOS

O principal objetivo do projeto é descrever as etapas necessárias para a realização de um teste de vulnerabilidades e como tais etapas são direcionadas para a descoberta, análise e mitigação de falhas e eventuais riscos a que o sistema pode estar submetido, favorecendo, assim ao desenvolvimento de uma sólida e consistente defesa proativa. A estrutura dos testes segue um modelo cuidadosamente estruturado em passos bem definidos, basicamente em 3 etapas: planejamento, execução e pós execução, as quais são bem descritas no artigo NIST Special Publication 800-115 (Scarfone, et al., 2008).

O desenvolvimento do projeto será feito com o auxílio do Sistema Operacional Linux BackTrack, um sistema desenvolvido sob o modelo *open source*. Tal sistema é focado em testes de segurança e vulnerabilidades, apresentado um conjunto de ferramentas que possibilitam a realização de procedimentos práticos para identificação de riscos e falhas em redes, sistemas e aplicações (Ali e Heriyanto,2011).

Tendo como base tais informações, serão gerados relatórios com os dados obtidos, explicitando quais métodos de intrusão foram contidos pelos mecanismos de defesa e aqueles que obtiveram êxito na intrusão, mostrando o potencial crítico de cada um e as possíveis soluções para contenção de tais vulnerabilidades.

1.2 - JUSTIFICATIVA

Teste de vulnerabilidades, como os pentests, representam a simulação de ataques reais através do uso de ferramentas comumente usadas por invasores, com o intuito de se identificar vulnerabilidades e métodos que possam vir a contornar o perímetro de segurança de redes, sistemas ou aplicações, representando, dentre todos os testes, aqueles que fornecem um cenário mais real de um possível ataque ou invasão.

Como o projeto é centrado em ataques, análise e mitigação de riscos e vulnerabilidades, tais testes representam uma ferramenta imprescindível para o desenvolvimento do projeto, justamente por representar tanto um mecanismo interessante para a validação de segurança do sistema quanto um meio de quantificar o grau de problemas em uma determinada corporação. Dessa forma, funciona como uma verdadeira auditoria de segurança da informação, fornecendo a oportunidade de se saber quais são as reais falhas de segurança que possibilitarão a invasão ao sistema, de forma a corrigi-las antes que sejam exploradas, provendo, dessa forma, a obtenção de um sistema certificado, mais confiável e seguro.

1.3 - METODOLOGIA

O início do projeto consiste na montagem do laboratório de testes. Nesta montagem, deve-se fornecer cenários que se aproximem ao máximo da realidade, simulando diferentes situações, partindo-se da hipótese de que invasões e ataques podem ocorrer em todas as direções, mesmo num âmbito interno.

O laboratório a ser montado será baseado em virtualização, que representa a melhor alternativa quando se tem uma limitação de recursos disponíveis e quando se deseja emular diferentes servidores, sistemas operacionais e aplicações, fornecendo dessa maneira uma riqueza de possibilidades.

Partindo-se de tais pressupostos, será simulado um ambiente de rede empresarial, o qual possuirá toda a estrutura central e periférica típicas de uma rede coorporativa, tais como *firewalls*, internet, servidores DHCP, DNS, de Arquivos, de E-mail, rede DMZ, entre outros, sendo definidas faixas específicas de ip para os diferentes serviços. Os testes serão realizados em dois âmbitos: interno (usuário dentro da rede) e externo (usuário fora da rede).

Como descrito na subseção 1.1 os testes de vulnerabilidades seguem um modelo bem estruturado e definido nas seguintes etapas: planejamento, execução e pós execução. Desta forma, tais processos atuarão como os norteadores da estrutura do projeto, onde serão obtidos resultados teóricos e práticos de cada uma das referidas etapas.

1.4 - ORGANIZAÇÃO DO TRABALHO

O capítulo 1 apresenta uma visão geral do que são os testes de vulnerabilidades, como os pentests, e sua importância ao se comportar como uma verdadeira forma de auditoria de segurança da informação. Os objetivos, bem como a justificativa, também são apresentados neste capítulo e destacam a importância do estabelecimento de uma defesa proativa ao gerar sistemas com uma maior robustez e confiabilidade. Tendo como base os passos essenciais da realização de um teste de vulnerabilidade, a metodologia abordada no capítulo aponta alguns detalhes da parte prática do projeto, além de dar uma visão geral do escopo do trabalho.

No capítulo 2 são apresentados conceitos e fundamentos sobre as principais formas de ataques que podem ser exploradas em virtude de falhas peculiares. Serão dados detalhes teóricos e técnicos sobre os principais métodos de ataque realizados na atualidade.

O capítulo 3 cita todo o referencial teórico que serviu como forma de consulta para o trabalho. São citados livros, artigos, além de websites que serviram como base para a constituição do acervo no qual o projeto foi sustentado. De forma adicional, são descritas com um maior nível de detalhes as etapas de um teste de vulnerabilidade. Além disso, são dados detalhes técnicos do sistema operacional BackTrack, o qual foi a plataforma escolhida para o desenvolvimento de todas as etapas do teste.

No capítulo 4 são descritos todos os procedimentos utilizados durante a realização do trabalho, tais como a montagem do laboratório e os tipos de testes realizados. Em seguida, todos os resultados obtidos durante esta parte prática são analisados, sendo traçados paralelos com a parte teórica de forma a se enriquecer a análise.

O capítulo 5 apresenta a conclusão e uma análise final do trabalho como um todo, considerando se os resultados alcançados foram condizentes com os objetivos propostos, apontando também os pontos de sucesso e os pontos de falha durante o decorrer do trabalho. São também propostas algumas sugestões para realização de trabalhos futuros na área.

2. CONCEITOS E FUNDAMENTOS

Conforme citado anteriormente, testes de vulnerabilidade são basicamente um conjunto de diversos testes realizados para apontar vulnerabilidades em redes ou sistemas. A partir destas falhas de segurança, é possível realizar um mapeamento de potencial crítico, onde são apontados os ataques, os quais o sistema analisado possa vir a estar suscetível. Tendo em mãos os resultados destes testes, é possível se estabelecer uma estratégia de mitigação de vulnerabilidades de forma a se obter uma defesa proativa, gerando um sistema mais robusto, seguro e confiável.

Neste capítulo serão apresentados detalhes técnicos e teóricos, bem como a estrutura dos principais tipos de ataques que serão realizados no projeto.

2.1- TIPOS DE ATAQUES

De uma maneira geral, um ataque é definido como qualquer ação que possa vir a comprometer a segurança da informação de um determinado sistema. Dentro do contexto de testes de vulnerabilidades, os ataques representam a maneira mais objetiva e clara de se avaliar o potencial crítico de uma determinada falha de segurança. Os ataques podem ser classificados basicamente de acordo com três parâmetros: quanto aos resultados que produzem, quanto à forma em que são praticados e quanto ao ponto de vista da rede.

Quanto aos resultados que produzem, podem ser classificados em 4 grandes categorias: ataques de interrupção, de interceptação, de modificação e de fabricação. Os ataques de interrupção são aqueles que geralmente desativam um ou mais serviços, submetendo-os à reinicialização. Nos ataques de interceptação, o atacante se coloca entre dois computadores da rede e se faz passar por um desses computadores originais, obtendo assim uma conexão funcional que permite leitura ou modificação dos dados que são trocados entre os *hosts* originais. Os ataques de modificação são aqueles onde um terceiro agente passa a ter acesso não autorizado a um determinado recurso do sistema, podendo vir a modificar informações ou configurações. Os ataques de fabricação são caracterizados pelo comprometimento da autenticidade do sistema, ou seja, compromete-o no sentido de burlar o processo de confirmação ao receptor que a mensagem é realmente procedente da origem informada em seu conteúdo.

Quanto à forma em que são praticados são classificados em 2 categorias: passivo e ativo. Ataques passivos são aqueles que não alteram a informação, nem o fluxo normal do canal sob escuta, ou seja, o invasor apenas monitora as mensagens que circulam na rede, colhendo informações. As formas de ataque passivo são baseadas em interceptação, monitoramento e análise de pacotes. Por não causar danos impactantes ao sistema, são formas de ataque mais difíceis de se detectar. Ataques ativos são os que promovem intervenção no fluxo normal de informações, quer alterando o seu conteúdo, quer produzindo informações falsas, com intuito de atentar contra a segurança de um sistema. Por serem mais agressivos, são mais facilmente rastreados que os ataques passivos. Ataques ativos, de uma maneira geral, envolvem adulteração, fraude, reprodução e bloqueio.

Quanto ao ponto de vista da rede são classificados em 2 categorias: ataques internos e ataques externos. Os ataques internos consideram que o invasor já esteja dentro da rede. São caracterizados por uma constante elevação de privilégios, de forma a se obter o maior nível de acesso e privilégios possíveis. Os ataques externos são conhecidos como "ataques em profundidade", já que objetivam superar toda a infraestrutura externa (tais como *firewalls*), que isola a rede interna almejada do mundo exterior. Desta forma, superam defesas e contornam obstáculos de forma a ingressar na rede interna do sistema alvo, ou ainda atingem um serviço em específico sem necessariamente promover a entrada na rede.

São inúmeros os métodos de ataque que podem ser realizados. A seguir serão tratados conceitos sobre as formas de ataque que foram praticadas durante o desenvolvimento do projeto.

2.1.1- Captura e análise de pacotes

A captura e análise de pacotes, técnica conhecida como *sniffing*, é uma forma de ataque passivo, onde um host, diferente do destino pretendido, consegue obter acesso à informação que circunda a rede, realizando desta forma uma atividade de escuta da comunicação alheia. Trata-se de um ataque praticamente impossível de se detectar, além do fato de que a ação de *sniffing* de uma rede pode não ser exatamente um ataque, podendo ser realizada no intuito de por exemplo diagnosticar eventuais problemas. O *sniffing* é estabelecido explorando a capacidade que a maioria das interfaces de rede possuem de serem colocadas em um modo promíscuo, onde ao receber um pacote, independente do destino de

origem, a interface o intercepta. O software mais famoso para a prática de *sniffing* é o Wireshark.

As principais ameaças que podem ser provocadas pelo *sniffing* são: interceptação de senhas e logins, informações sobre o tráfego de rede (exemplo: servidores mais utilizados), além da obtenção de informações sobre endereços IP e MAC dos *hosts* da rede.

2.1.2- Falsificação de pacotes

De forma a se garantir maiores níveis de segurança aos sistemas, grande parte dos servidores só libera a utilização de serviços a um número restrito e autenticado de usuários. Para burlar tal processo foi desenvolvido um mecanismo baseado em falsificação de pacotes, técnica conhecida como *spoofing*. Nesta técnica, o invasor fabrica um pacote de dados contendo um falso endereço de origem, fazendo com que o host atacado acredite que a conexão está vindo de um outro local, geralmente se passando por um host que tem permissão para estabelecer esta conexão. Assim, é possível interceptar e capturar uma transmissão legítima de dados entre dois sistemas. Concluída a captura, os dados obtidos são reencaminhados para o destino original, de forma a não se levantar suspeitas. Os tipos de ataque mais conhecidos que utilizam a técnica de *spoofing* são: DNS *Spoofing*, IP *Spoofing* e ARP *Spoofing.*

- DNS *Spoofing*: Consiste basicamente na interceptação do tráfego DNS e posterior substituição por pacotes DNS forjados, tarefa esta não muito difícil, já que o tráfego DNS não é criptografado e nem assinado digitalmente. Após esta interceptação, são praticadas alterações em tabelas que relacionam ip a *hostname.* Assim, ao se realizar uma consulta a um servidor DNS sobre um *hostname* qualquer, o servidor informa um IP errado, o qual é o endereço do *host* que está aplicando o ataque. Desta forma, consegue-se desviar o tráfego da vítima para um servidor falso, o qual pode ser um *site* falsificado, por exemplo.
- IP *Spoofing*: Em poucas palavras, consiste na fabricação de pacotes IP com informação falsa, onde são alterados dados referentes ao endereço IP de origem, substituindo o endereço de remetente original por um endereço falso, que corresponde ao endereço do *host* invasor. É possível graças a falta de verificação de remetente, não havendo validação do endereço IP nem relação deste com o roteador anterior responsável pelo encaminhamento. Assim, com uma simples manipulação de cabeçalho é possível praticar o *spoofing*. Dentre as principais consequências, um

atacante consegue, por exemplo, mascarar a verdadeira fonte de um ataque ou conseguir acesso à sistemas que são baseados em autenticação pelo endereço IP.

- ARP *Spoofing*: Representa uma variação do IP *Spoofing*, que se aproveita do mesmo tipo de vulnerabilidade, ou seja, falta de autenticação de remetente. Caracteriza-se pela interceptação do tráfego e alteração de dados na tabela ARP, responsável por associar endereços MAC a endereços IP. Desta forma, o endereço MAC do *host* original é substituído pelo endereço MAC do host invasor, fazendo com que todo o tráfego destinado a um determinado endereço IP seja encaminhado para o host que pratica o ataque.

2.1.3- Envenenamento de cache DNS

Envenenamento do cache de um servidor DNS, técnica conhecida como *Cache Poisoning* ou DNS *pollution*, é um tipo de ataque que compromete a integridade dos dados em um servidor DNS. O cache atua como um servidor recursivo, que tem a função de armazenar dados e respostas à consultas, de forma a otimizar o desempenho do servidor DNS com autoridade real, conhecido como servidor autoritativo. Em um ataque de *cache poisoning*, são fornecidas falsas respostas ao servidor recursivo, que passa a armazenar resoluções DNS falsas, resolvendo endereços para um servidor *fake*. Assim, quando a vítima realiza uma consulta, caso o cache responda antes do servidor autoritativo, será enviada uma resposta falsa, que consequentemente irá redirecionar, de forma indireta, a vítima ao estabelecimento de uma conexão com o *host* invasor.

A principal diferença entre um ataque DNS *Spoofing* e *Cache Poisoning*, é que no segundo caso, o atacante envia uma falsa resposta a um servidor recursivo (no caso o cache) e não diretamente ao usuário, como ocorre no primeiro caso.

Na figura 2.1, é possível observar um esquemático que ilustra o passo a passo para a realização de um ataque baseado em envenenamento de cache DNS.

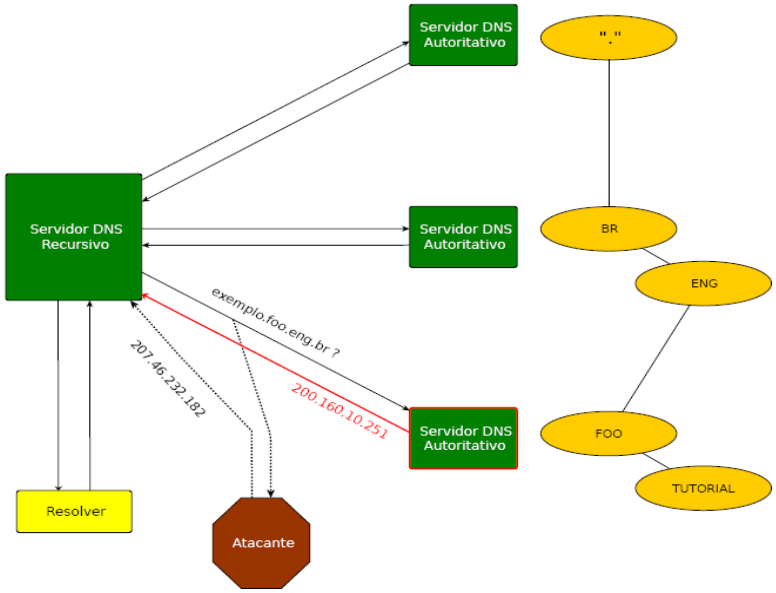


Figura 2.1 – Esquemático de um ataque de envenenamento de cache
Fonte: (De Campos e Justo, 2009)

2.1.4- Negação de serviço

Um ataque do tipo negação de serviço, também conhecido como DOS (*Deny of Service*), objetiva tornar indisponível um servidor ou um serviço. Consiste em "inundar" um determinado servidor com inúmeras requisições, de tal forma que este não consiga responder as consultas que lhe são feitas, muitas das quais legítimas dos usuários. Desta forma, devido à alta carga de processamento, o servidor acaba entrando em um estado inoperante e inacessível. Além de interromper o fornecimento de importantes serviços, após derrubar um servidor, o atacante pode, por exemplo, substituir um servidor autêntico por um servidor *fake*, fazendo com que este assuma o lugar do servidor real e responda as requisições destinadas a ele, obtendo assim importantes informações daqueles com os quais estabelece conexão.

Uma variante do ataque de DOS é conhecida como DDOS (negação distribuída de serviço), onde o ataque é realizado não só por um, mas por vários computadores, até mesmo distantes geograficamente entre si, dificultando ainda mais o rastreio e combate ao ataque praticado.

2.1.5- Buffer Overflow

Buffer é uma região temporária da memória onde são armazenados dados para posteriormente serem transportados de um lugar para outro. De uma maneira sucinta, *buffer*

overflow se refere ao processo de armazenar, em um *buffer* de tamanho fixo, dados maiores que o seu tamanho. São gerados na maioria das vezes devido a erros de programação. As principais consequências são: execução errônea do *software*, acesso indevido às áreas de memórias, interrupção do programa e possíveis falhas de segurança.

Desta forma, um ataque de *buffer overflow* visa explorar uma eventual falha em um código, tentando estourar o *buffer* e sobrescrever parte da pilha (locais estáticos do espaço de endereços de memória) através do envio de muitas informações para uma determinada variável, promovendo sobrecarga na memória.

De uma maneira mais formal, durante o ataque podem ser alterados valores de variáveis locais, valores de parâmetros e endereços de retorno. Ao se alterar estes últimos, pode-se estabelecer um novo endereço que aponte para a área em que o código malicioso que se deseja executar se encontra (podendo estar localizado no próprio *buffer* estourado ou em um trecho de código presente no *software* vulnerável). Dependendo do código utilizado, os mais variados efeitos podem ser produzidos, como um simples DOS ou até mesmo a garantia de uma elevação de privilégios ao invasor.

2.1.6- Injeção de DLL

DLL (*Dynamic Link Library*) é uma implementação que foi desenvolvida pela Microsoft. São códigos prontos que estão espalhados por diversas pastas do sistema e que podem ser usados por programas para executar uma determinada tarefa, dando continuidade a execução de seu respectivo código. Tal procedimento evita a escrita de várias funções em um programa, o que tornaria o desenvolvimento mais demorado e geraria softwares de tamanhos bem maiores.

Partindo-se de tais pressupostos, o ataque baseado em injeção de DLL, tem como objetivo injetar um código malicioso em uma DLL, de modo que quando o software invocar a DLL infectada, o código maléfico também será executado. Dependendo do código, diversos ataques posteriores podem ser gerado, como *Buffer Overflow*, DOS, além de por exemplo, elevação de privilégios e quebra de acessos restritos.

2.1.7- Sequestro de sessão

Sequestro de sessão é uma forma de ataque onde se explora uma sessão válida de um computador de forma a se quebrar acessos restritos à informações ou serviços em um determinado sistema. É uma técnica usada para controlar uma sessão atual depois que o usuário tenha estabelecido uma sessão autenticada, diferente por exemplo de *spoofing*, onde a sessão é criada pelo próprio invasor. Desta forma, é possível haver uma monitoração das atividades da vítima.

Em aplicações Web, é uma forma de ataque frequentemente usada para roubar a chamada *Magic Cookie*, usada para autenticar um usuário em um servidor remoto. Ao roubar este parâmetro, o invasor está apto a fazer requisições como se fosse o usuário legítimo, ganhando acesso às informações ou até mesmo modificando dados. Não é uma forma de ataque limitada só para a Web, podendo ser extendida a qualquer aplicação que apresente um protocolo cujo o estado é mantido através de uma chave que é checada por dois lados comunicantes, principalmente se tal chave não for criptografada.

2.1.8- Quebra de senhas

Na maioria dos sistemas atuais, a forma mais prática de se garantir autenticação de usuários é por meio da utilização de *logins* e senhas. Tendo em vista tal fator, foram desenvolvidos programas conhecidos como crackers, que objetivam a quebra e descoberta de senhas criptografadas e funções de *hash*(estrutura que relaciona longas mensagens a curtas mensagens criptografadas). Existem basicamente duas formas de ataque para se promover a quebra de uma senha: ataques *off-line* e ataques *online* (Ali e Heryianto, 2011).

Nos ataques *off-line*, o invasor obtém o arquivo que contém a senha criptografada e realiza o processo de quebra em sua própria máquina. A principal vantagem deste método é a de que o atacante não precisa se preocupar com nenhum eventual mecanismo de bloqueio de senha por parte do sistema invadido ao tentar realizar a quebra da senha. Além disso, o processo de captura da senha criptografada em si, quando realizada por *sniffing*, é praticamente impossível de se detectar. Uma das técnicas utilizadas neste tipo de ataque para realizar a quebra de senha são as *rainbow tables*, que são tabelas enormes com milhares de *hashs* e suas respectivas senhas, as quais já foram quebradas .

Nos ataques *online*, o invasor tenta "adivinhar" a senha, o que pode provocar um bloqueio por parte do sistema após sucessivas falhas. Uma técnica utilizada neste tipo de ataque é conhecida como ataque de força bruta. Neste tipo de ataque, são geradas milhares de combinações de senhas possíveis, e se tenta adivinhar o conjunto correto por meio de tentativa e erro. O processo é iniciado inicialmente em cima de *logins* padrão, como admin, administrator e root.

Durante o desenvolvimento do projeto, foi utilizado um software conhecido como Cain & Abel para realizar alguns processos de quebra de senhas. O software utiliza vários métodos, dentre os quais estão inclusos uso de *rainbow tables* e ataques de força bruta.

2.1.9- Engenharia Social

Engenharia Social é um termo que se refere ao processo de persuadir pessoas a realizarem determinadas ações ou fornecer informações de cunho confidencial. Do ponto de vista de segurança da informação, é definida como uma coleção de técnicas e ferramentas que podem englobar negociações, psicologia, além de técnicas para enganar, objetivando a utilização do fator humano para burlar mecanismos de segurança de sistemas.

Engenharia social é algo presente no dia-a-dia das pessoas, podendo ser utilizada desde um *hacker* até uma criança quando faz manha para conseguir o que quer. É uma técnica amplamente utilizada durante os ataques, pois não existe "*patches* de correção para a estupidez humana", ou dizendo de outra forma, as maiores vulnerabilidades de qualquer sistema são seus próprios usuários (Reis,2012). Por tais motivos, a engenharia social constitui uma das formas de ataque mais perigosas e bem sucedidas na atualidade.

A ferramenta de engenharia social utilizada no desenvolvimento do projeto foi o SET, o qual é bem referenciado em 3.2.3, contendo inclusive exemplos de alguns ataques.

3- REFERENCIAL TEÓRICO

O desenvolvimento do trabalho foi baseado em artigos e trabalhos técnicos centrados em torno da temática de pentests, muitos dos quais podendo ser encontrados no website http://pentestmag.com o qual constitui um importante acervo contendo uma série de pesquisas e referências sobre o assunto. O diferencial do projeto desenvolvido neste trabalho em relação a maioria dos outros é o fornecimento de uma visão global com um nível adequado de detalhes no que se referem às etapas e tópicos próprios de um teste de vulnerabilidades. Boa parte dos trabalhos na área acaba focando em uma parte em específico, tratando assuntos pertinentes a outras questões e etapas de uma forma secundária, muitas vezes com um nível de detalhes simplista.

Como material de apoio, foram utilizados alguns livros, cujo foco é mais voltado para a descrição da parte técnica de algumas ferramentas frequentemente utilizadas para realização de ataques, identificação e posterior mitigação de vulnerabilidades em sistemas. Por fim, o desenvolvimento do trabalho, teve como um alicerce fundamental, a constante consulta aos chamados Bancos de Dados de Vulnerabilidades, os quais serão descritos com mais detalhes posteriormente.

A seguir, serão apresentados detalhes técnicos e estruturais, além de uma interligação com o referencial teórico utilizado, no que se refere às etapas de um teste de vulnerabilidades, bem como as principais ferramentas do BackTrack utilizadas durante o projeto.

3.1- ETAPAS DE UM TESTE DE VULNERABILIDADES

Em suma, os testes de vulnerabilidades são divididos em 3 fases: planejamento, execução e pós execução. A seguir, será apresentada de uma forma sucinta, as principais características de cada uma destas etapas. Tais etapas são descritas com uma maior riqueza de detalhes na maioria das referências bibliográficas citadas no final do documento. Partindo-se de tais pressupostos, as etapas podem ser divididas em:

- Planejamento*:* Corresponde a um processo de levantamento inicial das informações que serão utilizadas para a modelagem dos testes, sendo levantados detalhes de infraestrutura, bem como recursos e equipamentos necessários para a realização dos testes, além de uma reunião de informações para posteriores avaliações, definição de ameaças de interesse, controles de segurança, planos de gerenciamento, requisitos, responsabilidades técnicas, metas, objetivos, fatores de sucesso, suposições, recursos

disponíveis, entre outros. São definidos também os tipos de testes que serão realizados, como testes internos e externos e se os testes seguirão a filosofia da "caixa preta" (com nenhum conhecimento do sistema a ser avaliado) ou da caixa branca(com conhecimento parcial ou irrestrito de qualquer informação relevante para a execução do teste).

Os testes internos consideram que o invasor já está dentro da rede. Tais testes simulam diferentes níveis de acesso a rede, tanto o de um simples usuário como o de um administrador do sistema, que representa o pior cenário possível. A partir disso são estudadas maneiras de se ganhar privilégios e acessos adicionais. A seguir, procura-se a descoberta de vulnerabilidades que podem ser exploradas e são expostos potenciais danos que podem ser causados, com uma conjunta análise de diferentes níveis de segurança, envolvendo controle de acesso, serviços e configurações.

Os testes externos seguem a filosofia de "Defesa em Profundidade". Isso significa que a corporação deve apresentar toda uma infraestrutura (como a presença de *firewalls*) que isolem sua rede interna de acessos não autorizados. Dessa forma, o objetivo de invasão externa é ver se é possível superar as defesas e obstáculos que isolam a rede interna das empresas. Tal cenário é bem mais difícil de se representar do que os testes de âmbito interno, pois a princípio não se tem nenhuma informação da rede a ser invadida e diversas defesas devem ser superadas, como *firewalls*, proxys, DMZ, NAT, Sistemas de Detecção e Intrusão, entre outros.

- Execução: Corresponde a fase onde os testes são executados. Nesta fase há a identificação dos riscos e vulnerabilidades. É subdivida nas sub etapas obtenção de informação, *scanning* e mapeamento, identificação de vulnerabilidades, e ataques.
- Obtenção de Informação: Procedimento indispensável para modelar os ataques e determinar os caminhos que podem ser explorados com maior facilidade. Destacam-se aqui a coleta passiva (representada principalmente pelo uso de *sniffers*) e a coleta ativa (técnicas como WHOIS, DNS, Engenharia Social, *Dumpster Diving*, etc).
- Scanning e Mapeamento: Processo de varredura e mapeamento da rede a partir das informações obtidas na etapa anterior. As principais atividades são: identificação de atividade de *hosts*, mapeamento da rede bem como de portas e serviços em funcionamento e identificação de sistemas operacionais e rotas. Tais informações ajudam a corroborar ou descartar determinadas hipóteses sobre os alvos (Hurley,2007).

- Identificação de Vulnerabilidades: Após mapeamento dos sistemas e serviços pertencentes à rede, deve-se detectar vulnerabilidades conhecidas ou determinados caminhos que possam ser explorados para promover a invasão, estimando-se o impacto que cada uma delas pode causar. O NVD (*National Vulnerability Database*), CVE (*Common Vulnerabilities and Exposures*) e o MSB (*Microsoft Security Bulletin*) correspondem a importantes bancos de dados de vulnerabilidades que pode ser usados como referência.
- Ataques: São realizados logo após a identificação das vulnerabilidades, através da obtenção de acesso não autorizado com o maior nível de privilégio possível. Diversas formas de ataque serão analisadas no projeto como *Buffer Overflow*, Quebra de Senha, Negação de Serviços, entre outros.
- Pós-Execução: Corresponde a última fase dos testes de vulnerabilidades. Nesta etapa é realizada uma análise de todas as vulnerabilidades identificadas na fase de execução, identificando causas e estabelecendo recomendações de mitigação de tais vulnerabilidades e riscos. Deve ser gerada uma documentação durante a execução do teste com o intuito de se manter registro de todas as atividades de forma transparente. A figura 3.1 ilustra com um maior nível de detalhes as etapas que serão descritas, levando em conta o ponto de vista de quem usa o BackTrack para a realização dos testes.

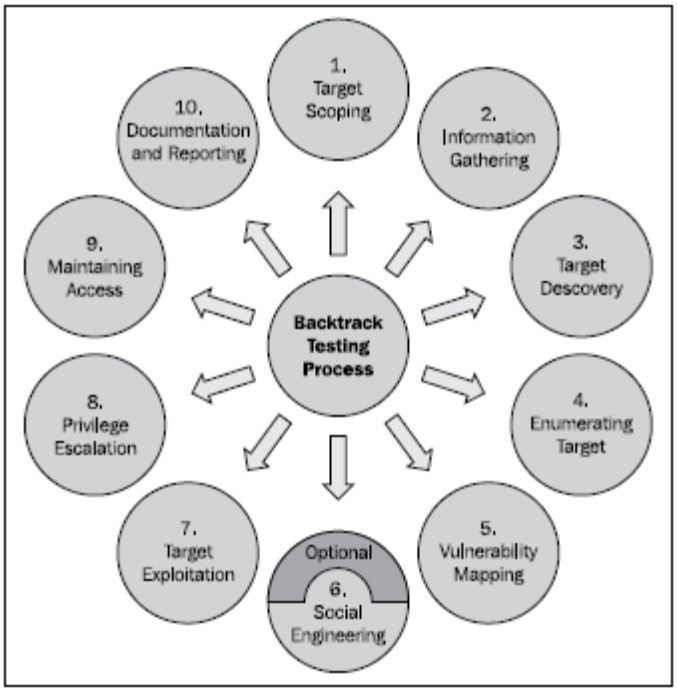


Figura 3.1 – Esquemático das etapas de um teste de vulnerabilidades
Fonte:(Ali e Heriyanto, 2011)

3.2- BACKTRACK

BackTrack é um Sistema Operacional Linux baseado na distribuição Ubuntu. Basicamente é uma plataforma para realização de testes de penetração e auditorias de segurança, através de avançadas ferramentas que possibilitam a identificação, detecção e exploração de vulnerabilidades descobertas no sistema alvo, possibilitando assim, a emulação de diversos tipos de ataques. Representa desta forma uma verdadeira "máquina de guerra" (Ali e Heriyanto, 2011). Durante o trabalho foi utilizado a versão BackTrack 5 R2.

Dentre as classes de ferramentas que o BackTrack possui, destacam-se: coleta de informações, mapeamento de rede, identificação de vulnerabilidades, invasão, elevação de privilégio, manutenção de acesso, eliminação de rastros, analisadores de redes via rádio, análise de voip e telefonia, perícia forense digital e ferramentas para engenharia reversa. No desenvolvimento do projeto, diversas dessas ferramentas foram utilizadas, com destaque para o OpenVAS, Metasploit Framework e o SET (*Social Engineering Toolkit*), as quais por serem utilizadas exaustivamente durante as fases de identificação de vulnerabilidades e realização de ataques, serão explicadas com um maior nível de detalhes a seguir.

3.2.1- OpenVAS

O Sistema de Avaliação de Vulnerabilidade Aberto – OpenVAS, é uma coleção de vários serviços e ferramentas que constituem uma robusta plataforma para gerenciamento de vulnerabilidades. Possui uma estrutura baseada na arquitetura cliente-servidor, onde o cliente seleciona um alvo em um segmento específico da rede e realiza testes de varreduras de vulnerabilidades a partir do servidor OpenVAS.

Apresenta uma estrutura multitarefa, com a possibilidade de realização de testes simultâneos, estando disponível para diferentes sistemas operacionais, como Linux e Windows. De uma forma simplificada, os principais componentes do OpenVAS são:

- *OpenVAS Scanner*: Responsável por gerenciar a execução dos Testes de Vulnerabilidade de Rede – NVTs, que constituem uma base de dados atualizada periodicamente sobre plug-ins, scripts, feedbacks e logs dos principais testes a serem realizados.

• *OpenVAS Client*: Sua principal função é controlar a execução da varredura realizada no sistema alvo, fazendo uso do Protocolo de Transferência OpenVAS (OPT), que age como um protocolo de comunicação com o OpenVAS Scanner.

• *OpenVAS Manager*: Representa o ponto central de análise das vulnerabilidades. Responsável por armazenar os resultados das varreduras realizadas, além de executar funções como agendamento de tarefas, geração de relatórios e atividades de filtragem de dados.

• *Greenbone Security Assistant (GSA):* É uma interface *web*, onde os usuários podem configurar, gerenciar e administrar o processo de varredura das vulnerabilidades. É uma interface amigável e de simples uso, onde seleciona-se um alvo, realiza-se o scanner e se observa os relatórios gerados após execução dos testes. Apresenta também uma variação de interface própria para desktops, conhecida como GSA *Desktop*.

• *OpenVAS Administrator*: Responsável pela administração e logs dos usuários.

A figura 3.2 ilustra um esquemático de todos os componentes do OpenVAS citados anteriormente. A figura 3.3 apresenta uma visão da interface *web* GSA

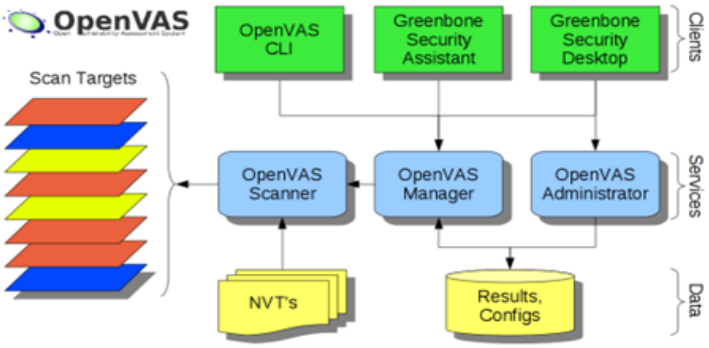


Figura 3.2 – Estrutura do OpenVAS
Fonte: www.openvas.org

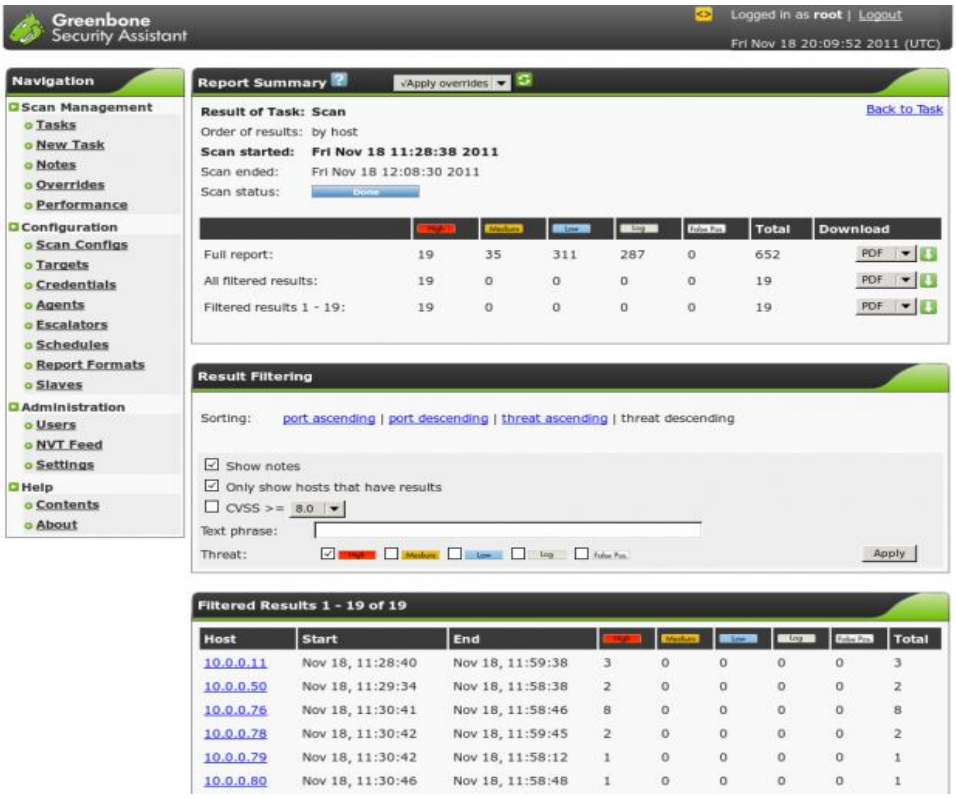


Figura 3.3 – Interface Web GSA
Fonte: www.openvas.org

3.2.2- Metasploit Framework

O Metasploit Framework é uma poderosa plataforma *open source*, desenvolvida com o intuito de aprimorar e acelerar o desenvolvimento, ensaio e exploração de vulnerabilidades de s*oftwares*, através da utilização de exploits.

Exploits são basicamente softwares que tiram vantagem de qualquer *bug* ou vulnerabilidade presente em um sistema. Em outras palavras, é a exploração da falha, a qual permite ao explorador realizar algum tipo de ação, como por exemplo burlar acessos restritos.

O Metasploit Framework é equipado com centenas de exploits, payloads e ferramentas muito avançadas, as quais permitem testar vulnerabilidades em diferentes plataformas de desenvolvimento, sistemas operacionais e servidores.

Payloads são basicamente códigos ou *scripts* maliciosos concebidos como parte de um exploit ou desenvolvidos de uma forma independente, os quais permitirão, dentre outras ações, o controle integral ou parcial do sistema explorado. Estabelecendo um paralelo entre exploit e payload, o primeiro é a exploração da falha em si, a qual permitirá ao invasor realizar alguma ação, já o segundo refere-se à ação que será realizada durante a exploração.

Além de exploits e payloads, outras ferramentas de destaque que constituem o Metasploit Framework são os auxiliares, que constituem um conjunto de ferramentas baseadas na busca de informações dos sistemas alvo, além dos *Encoders*, os quais evitam, durante a invasão, a detecção de antivírus, *firewalls*, Sistemas de Detecção de Intrusão (IDS) e outras defesas similares.

O Metasploit Framework apresenta basicamente três interfaces de uso: uma baseada em linha de comando, conhecida como MSFCli, uma interface web conhecida como MSFWeb, e por fim uma interface console, conhecida como MSFConsole, a qual devido ao fácil uso, velocidade de funcionamento e flexibilidade foi utilizada durante o projeto.

O MSFConsole é um dos meios mais eficientes e poderosos para que durante um teste de vulnerabilidade, se faça um melhor uso do potencial quadro de exploração. As ações realizadas durante um ataque utilizando o MSFConsole se resume basicamente a: seleção do exploit, seleção do alvo, seleção do payload, configuração das opções e desenvolvimento da fase de exploração.

O MSFConsole possui uma extensa lista de exploits, que continua a crescer periodicamente. A seleção do exploit deve ser feita tendo em mãos algumas informações do sistema alvo, como por exemplo identificação de portas abertas, de serviços em execução, hosts existentes,etc. Através de um scanner de vulnerabilidades, como o OpenVAS, é possível identificar as potenciais e vulnerabilidades e escolher exploits desenvolvidos em cima de tais falhas. Outra forma também é olhar diretamente o banco de dados do Metasploit e ver se tem algum exploit disponível para o serviço que se está alvejando. A figura 3.4 ilustra a tela inicial do Metasploit Framework.

.

Figura 3.4- Tela inicial do Metasploit Framework

Após escolha do exploit, deve-se selecionar o alvo, que representa basicamente o sistema operacional a ser invadido. Em muitos casos, um mesmo exploit pode ser aplicável a diferentes sistemas operacionais com diferentes níveis de *patches* e versões.

Uma vez escolhidos o exploit e o alvo, seleciona-se o payload que se deseja executar. Cada exploit apresenta um conjunto de payloads específicos baseados na vulnerabilidade a ser explorada. Com exploit, alvo e payloads selecionados, configuram-se as opções típicas que um determinado payload possui, como, por exemplo, seleção de portas, endereços ou eventuais threads a serem utilizados.

Por fim, chega-se a fase de exploração, na qual o exploit será executado. O resultado dependerá do tipo de ataque que é próprio do exploit selecionado. Em alguns casos, tem-se a fase de pós-exploração, a qual é típica de payloads cujo ataque em si na fase de exploração consiste apenas no estabelecimento de conexão com o alvo, o qual gera uma sessão que possibilita a realização de diversas formas de ataques. Um payload típico desta fase é o Meterpreter.

O Meterpreter não é simplesmente um payload, mas sim uma plataforma de exploração que apresenta um *shell* de comandos, o qual fornece ao invasor uma variedade de atividades possíveis de serem executadas no sistema explorado. Sua forma de execução é por meio de injeção de DLL na memória do sistema alvo. Através do Meterpreter, *scripts* e *plugins* podem ser carregados dinamicamente com o objetivo de se estender a atividade de pós-exploração. Dentre as atividades que o Meterpreter fornece estão: elevação de privilégios, *keylogging*, criação de um *backdoor* persistente, permitindo acesse remoto estável ao alvo, além de outras extensões. Outra opção de suma importância oferecida pelo Meterpreter é a migração de processos, o que permite "camuflar" a invasão, dificultando assim a detecção da invasão por *firewalls*, antivírus e sistemas de detecção de intrusão.

3.2.3- Ferramenta de engenharia social

O BackTrack apresenta uma ferramenta de engenharia social conhecida como Social Engineer Toolkit (SET) que consiste em um conjunto de ferramentas designadas para ataques contra fraquezas de fator humano, tais como curiosidade, falta de conhecimento ou simples estupidez. Desta forma, tal tipo de ataque torna-se muito útil quando o alvo não apresenta determinadas vulnerabilidades e a forma de se conseguir uma brecha para um ataque é por meio da "concessão de acesso" do usuário do sistema que se pretende atacar.

O SET caracteriza-se por ataques que utilizam estruturas conhecidas como *vectors*, as quais fornecem acesso ou informação de um sistema. Os principais tipos de ataques oferecidos pelo SET são:

- *SphearPhishing Attack Vector*: Caracterizado pela criação de exploits em forma de ficheiros (como por exemplo .pdf) e pelo envio de um ataque por e-mail à vitima contendo um anexo, o qual quando aberto compromete o sistema alvo. A figura 3.5 ilustra a tela inicial desta forma de exploit.

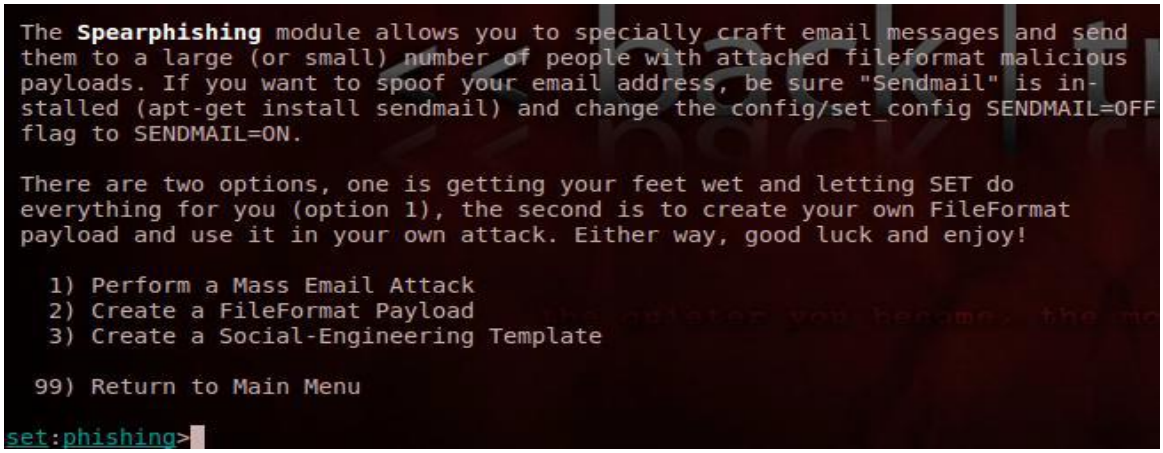


Figura 3.5 – Tela do exploit Sphearphising

- *Website Attack Vectors*: Os *vectors* de ataque *web* são os mais avançados e completos do SET, porque são criados para parecerem autênticos. Dentre as formas de ataque deste tipo, destacam-se:

- *Java Applet*: Representa um dos tipos de ataque mais bem sucedidos no SET. Neste ataque, um Java Applet adapta-se ao *browser* da vítima e entrega um payload na forma de um *applet* assinado com nomes como Google, Microsoft , nomes de bancos,etc. Quando a vítima acessa o *applet*, o ataque é realizado com sucesso.
- *Metasploit Browser Exploit*: Ataque realizado em cima de certas vulnerabilidades de alguns browsers. Para ter êxito é necessário que a vítima seja suscetível a uma vulnerabilidade específica própria do *browser* que utiliza.
- *Credentials Harvesting:* O objetivo deste ataque é obter as credenciais (*logins* e senhas) dos usuários dos sistemas alvos. Isto é possível usando a opção Site Clonner, onde um *website* é clonado, como por exemplo Gmail. Quando o usuário escreve l*ogin* e senha, por exemplo, o invasor captura estas informações e logo em seguida, para não levantar nenhum tipo de suspeita ou alarme, o exploit redireciona as credenciais fornecidas pelo usuário para o *website* original. A figura 3.6 ilustra a tela inicial desta opção.

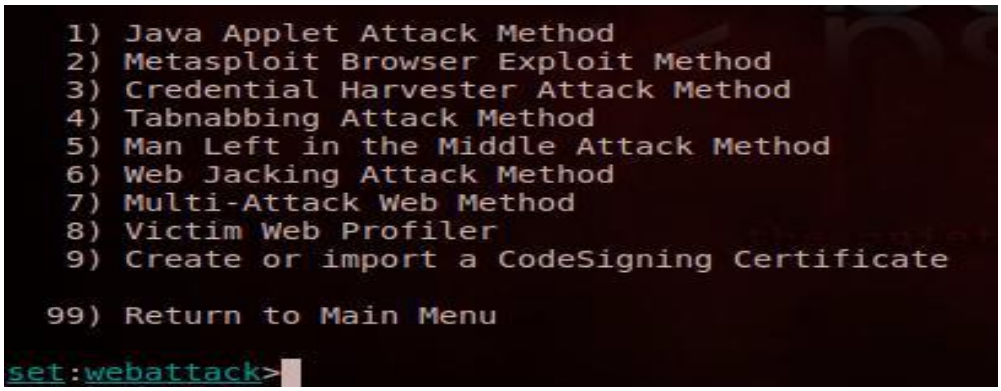


Figura 3.6 – Tela do exploit Website Attack Vectors

3.3- REALIZAÇÃO DE UM TESTE DE VULNERABILIDADE

A série especial de publicações 800 do NIST (National Institute of Standards and Technology) é voltada para a segurança de sistemas de informação. Tal série é constituída de relatórios sobre pesquisas, divulgação de esforços e resultados, além da orientação de práticas adequadas a se adotarem para o estabelecimento de sistemas com alto nível de segurança.

No desenvolvimento do projeto, foram utilizados dois artigos em especial desta série de publicações: NIST Special Publication 800-42: Guidelines on Network Security Testing (Wack, et al., 2003) e NIST Special Publication 800-115: Technical Guide to Information Security Testing and Assessment (Scarfone, et al., 2008), os quais apresentam seus respectivos conteúdos relacionados com a realização de testes de vulnerabilidades. Tais artigos defendem a realização de testes de vulnerabilidades de acordo com um modelo cuidadosamente estruturado e esquematizado em etapas, de forma a se otimizar a realização do experimento e posteriormente a análise dos resultados, obtendo um quadro de exploração com ótimos níveis de excelência. Adicionalmente, descrevem técnicas e ferramentas específicas no âmbito de segurança da informação, além de dar uma visão geral sobre as principais vulnerabilidades que sistemas podem vir a estar submetidos, bem como orientação para a solução de tais falhas, de forma a se aumentar a robustez das práticas de segurança adotadas.

Como referência bibliográfica adicional para orientação da realização de testes de vulnerabilidades, foi consultada a obra de (Hurley, 2007). Tal livro aborda, dentre outros aspectos, estudos de caso que servem como referencial para a montagem de laboratórios de análise, além de apresentar uma descrição de ferramentas open source que podem ser úteis no

desenvolvimento de tais testes. Outra referência importante é (Lyon, 2009), a qual é voltada para a descrição do NMAP, ferramenta para mapeamento de portas, útil na fase de *scanner*.

3.4- BANCOS DE DADOS DE VULNERABILIDADES

Após a fase de *Scanning* e Mapeamento, os softwares utilizados para realizar o processo de scanner, geram relatórios com informações específicas sobre o alvo analisado, tipos de vulnerabilidades encontradas, além de associar cada vulnerabilidade a um número específico que serve para referenciá-las a bancos de dados externos. Tais estruturas são conhecidas como Bancos de Dados de Vulnerabilidades, os quais são estruturados de forma a apresentar uma visão geral da vulnerabilidade, formas como explorar brechas na segurança de sistemas em cima da referenciada vulnerabilidade, bem como a proposição de eventuais soluções para a correção de tais falhas. Os principais bancos de dados consultados durante o projeto foram: CVE (*Common Vulnerabilities and Exposures*), NVD (*National Vulnerability Database*) e MSB (*Microsoft Security Bulletin).*

O NVD apresenta um referencial para diversos outros bancos de dados, incluindo o CVE, sendo de responsabilidade do governo dos EUA. Apresenta nos seus boletins de segurança informações sobre uma grande lista de vulnerabilidades, tais como: os erros que as geram (falhas de software, erros de configuração, métricas inadequadas, etc), medidas e práticas de segurança a serem adotadas para sanar cada falha, maneiras de se obter acesso a um alvo em cima destes pontos fracos e impactos que podem ser causados. A figura 3.7 ilustra a descrição de uma vulnerabilidade no NVD.

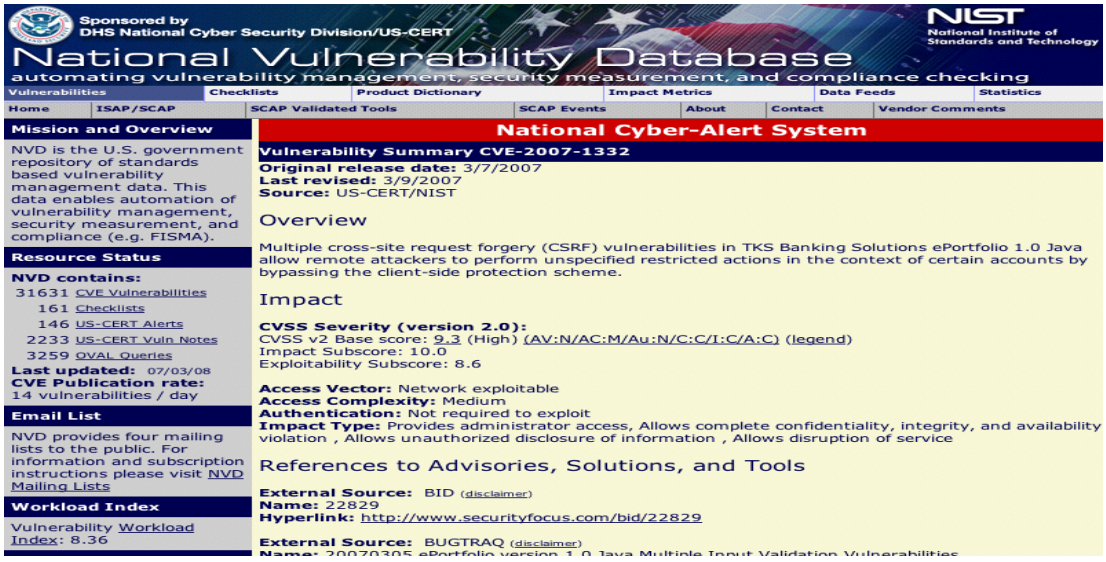
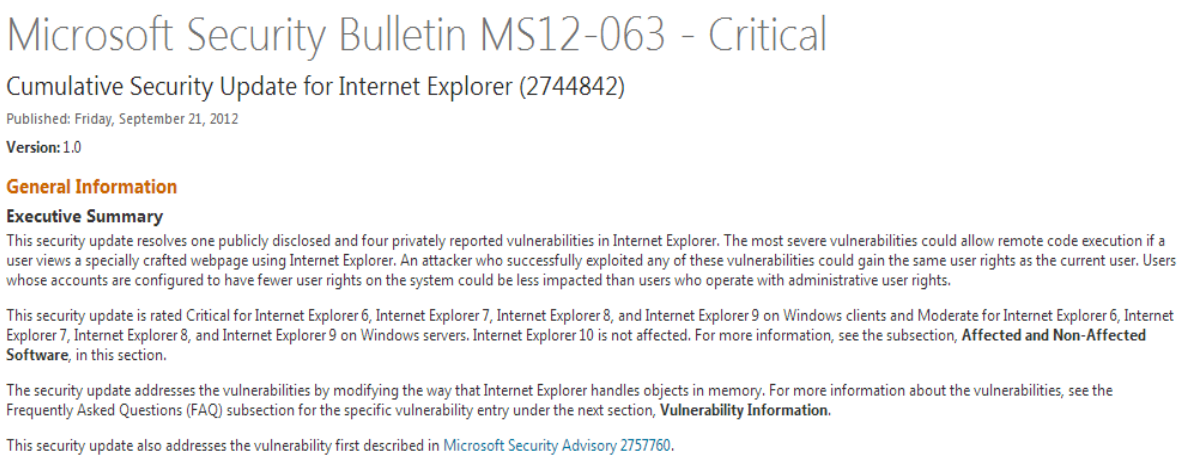


Figura 3.7 – Descrição de vulnerabilidade no NVD

O MSB são boletins de segurança desenvolvidos pela Microsoft, para apontar vulnerabilidades nos sistemas operacionais Windows. De uma maneira geral, apresentam uma sinopse da vulnerabilidade, ataques que podem ser explorados, recomendações de segurança, além de uma lista de *softwares* e sistemas operacionais afetados e não afetados pela falha analisada. São disponibilizados também, um conjunto de patches que ao serem instalados mitigam integralmente ou parcialmente a falha existente. A figura 3.8 ilustra a descrição de uma vulnerabilidade no MSB.



Microsoft Security Bulletin MS12-063 - Critical

Cumulative Security Update for Internet Explorer (2744842)

Published: Friday, September 21, 2012

Version: 1.0

General Information

Executive Summary

This security update resolves one publicly disclosed and four privately reported vulnerabilities in Internet Explorer. The most severe vulnerabilities could allow remote code execution if a user views a specially crafted webpage using Internet Explorer. An attacker who successfully exploited any of these vulnerabilities could gain the same user rights as the current user. Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights.

This security update is rated Critical for Internet Explorer 6, Internet Explorer 7, Internet Explorer 8, and Internet Explorer 9 on Windows clients and Moderate for Internet Explorer 6, Internet Explorer 7, Internet Explorer 8, and Internet Explorer 9 on Windows servers. Internet Explorer 10 is not affected. For more information, see the subsection, **Affected and Non-Affected Software**, in this section.

The security update addresses the vulnerabilities by modifying the way that Internet Explorer handles objects in memory. For more information about the vulnerabilities, see the Frequently Asked Questions (FAQ) subsection for the specific vulnerability entry under the next section, **Vulnerability Information**.

This security update also addresses the vulnerability first described in Microsoft Security Advisory 2757760.

Figura 3.8 – Descrição de vulnerabilidade no MSB

O NVD, CVE e MSB, podem ser acessados respectivamente através dos websites: nvd.nist.org , cve.mitre.org , http://technet.microsoft.com/en-us/security/bulletin.

3.5- FASE DE ATAQUES

Para a fase de realização de ataques, após a identificação de vulnerabilidades, as principais formas de referência utilizadas para consulta foram livros técnicos, com abordagem voltada para descrição de formas de práticas de ataque através de ferramentas peculiares.

A obra (Ali e Heriyanto, 2011) é focada na descrição do sistema operacional BackTrack, descrevendo as principais ferramentas do sistema e relacionando-as aos diferentes tipos de ataque que podem ser praticados em virtude de algumas propriedades que possuem. Tal obra foi de suma importância para o desenvolvimento do trabalho, já que o BacTrack foi a plataforma escolhida para a realização de todas as etapas do pentest.

Dentre as diversas ferramentas que o BackTrack possui, a principal delas e a que foi utilizada com uma maior frequência durante o projeto é o Metasploit Framework, plataforma

que conta com exploits e payloads, os quais, como dito anteriormente, são softwares de exploração que aceleram e automatizam a realização dos ataques. A obra (Foster, et al., 2007) apresenta como tema a descrição dos diferentes tipos de ações que podem ser realizadas através da utilização do Metasploit Framework. São apresentados também estudos de caso que ilustram alguns tipos de ataques praticados a partir de ferramentas do Metasploit Framework, além de uma descrição dos principais payloads e exploits pertencentes à plataforma.

Uma importante fonte de consultas para exploits que podem ser praticados a partir de determinadas vulnerabilidades é o website http://www.metasploit.com. O website em questão é referenciado a alguns bancos de dados de vulnerabilidades, tais como o CVE e o MSB. Tendo em mãos o número de CVE ou MSB que descreve a vulnerabilidade, é possível fazer consultas que retornam como resposta uma lista de comandos próprias do Metasploit Framework que são necessárias para a realização de um ataque que explora a vulnerabilidade em questão. Além disso, descreve também, todos os sistemas operacionais e versões de softwares que podem ser atingidos pelo exploit consultado. Desta forma, o trabalho para se realizar o ataque é reduzido drasticamente. A figura 3.9 ilustra a descrição de um exploit bem como os comandos para executá-lo no metasploit.com.

Exploit Usage Information

```
$ msfconsole

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show payloads
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > set LHOST [MY IP ADDRESS]
msf exploit(ms08_067_netapi) > set RHOST [TARGET IP]
msf exploit(ms08_067_netapi) > exploit
```

Exploit Module Options

RHOST	The target address
RPORT	Set the SMB service port (default: 445)
SMBPIPE	The pipe name to use (BROWSER, SRVSVC) (default: BROWSER)
CHOST	The local client address
CPORT	The local client port
ConnectTimeout	Maximum number of seconds to establish a TCP connection
ContextInformationFile	The information file that contains context information
DCERPC::ReadTimeout	The number of seconds to wait for DCERPC responses
DisablePayloadHandler	Disable the handler code for the selected payload

Figura 3.9 – Descrição de exploit no metasploit.com para uma dada vulnerabilidade

4- DESENVOLVIMENTO E RESULTADOS

Neste capítulo serão abordados todos os procedimentos que foram utilizados para a realização do trabalho, bem como será fornecida uma análise de todos os resultados gerados após a prática experimental. Em um primeiro momento, serão dados detalhes técnicos que explanam sobre a montagem do laboratório de experimentos. A seguir, será descrito todo o quadro de exploração que foi realizado, englobando com o devido nível de detalhes todas as etapas realizadas durante o teste de vulnerabilidades. Por fim, será feita uma minuciosa análise dos dados obtidos, estabelecendo-se um paralelo entre estes e os objetivos propostos no início do trabalho.

4.1- DESCRIÇÃO DO LABORATÓRIO

Definiu-se inicialmente 2 cenários para desenvolvimento dos testes.

Cenário 1 – O cenário em questão é centrado na realização de ataques externos. Ele é caracterizado pela construção de um ambiente comum nos dias de hoje, tendo em vista a presença cada vez maior de dispositivos que se utilizam do meio sem fio para obterem acesso à Internet, tais como *smartphones*, *tablets*, *notebooks*, *netbook*s. Atualmente é comum que as empresas, além da rede interna cabeada, ofereçam também o serviço de acesso sem fio, que proporciona mobilidade, praticidade e conforto para aqueles que se utilizam constantemente de dispositivos móveis no desenvolvimento de seus trabalhos.

Em cima desta configuração, é realizado um ataque no qual o invasor está fora da rede sem fio e explora formas que lhe permitam ingressar nesta. Em suma, objetiva-se quebrar a senha que garante acesso à rede sem fio, de forma a se infiltrar e explorar possíveis vulnerabilidades que estiverem ao alcance, podendo desta forma, afetar alguns serviços de rede ou ainda abstrair todas as informações confidenciais que forem de interesse.

Observando a figura 4.1, que mostra a topologia de rede que caracteriza o cenário 1, percebe-se que o *Wireless Router* representa o ponto de acesso, o qual é responsável por promover a conexão entre os dispositivos móveis e a rede sem fio. Tal equipamento está conectado diretamente ao fwinterno.pjf.unb.br, o qual é o *Firewall/Router*, que é responsável por fazer a interligação do mesmo com o provedor de internet(ISP), fornecendo assim internet para todos os usuários da rede sem fio. A rede sem fio é baseada no protocolo 802.11 i, está

configurada no canal 11, e utiliza-se do protocolo de comunicação sem fio WPA/WPA2 PSK e padrão de criptografia AES. Autentica-se à rede por meio de uma senha de acesso, cuja responsabilidade é do administrador da rede. Para a realização dos testes, assumiu-se a configuração de um sistema de segurança *default*, sem elevados níveis de defesas em profundidade.

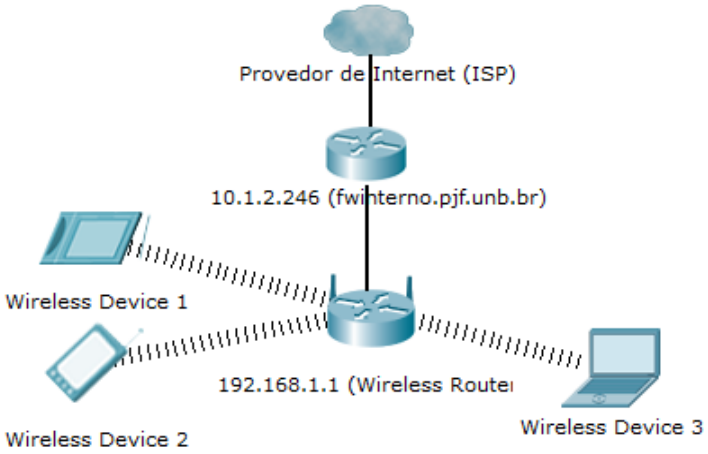


Figura 4.1 – Topologia de rede do Cenário 1

Cenário 2 – O cenário em questão é centrado na realização de ataques do tipo interno, ou seja, considerando-se que o invasor é um usuário da rede interna. Neste tipo de configuração, os usuários utilizam-se de computadores fixos (*Desktops*) e são conectados diretamente a um ponto de rede, ligado diretamente a um *Switch*, o qual interliga-se a um roteador central. O roteador central é responsável pela conexão da rede local com roteador do provedor de internet (ISP). Os usuários realizam a maioria dos trabalhos nestes desktops, o que os leva a armazenarem nestas máquinas importantes documentos relacionados com as atividades da empresa. De uma forma geral, um usuário interno apresenta permissão de acesso à intranet, e-mails corporativos, internet, servidores de arquivos, além de outros serviços.

Tendo em mente tais pressupostos, é simulado neste cenário um atacante como sendo um usuário da rede interna, apresentando um endereço IP da rede usuários, fornecido via DHCP e apresentando acesso à serviços e aplicativos da rede na qual está inserido. Desta forma, o atacante pode explorar possíveis falhas de segurança na rede interna, e com sucesso, pode causar danos a serviços, abstrair informações confidenciais, e afetar outros usuários e consequentemente as atividades da empresa. Para a realização dos testes, assumiu-se a configuração de um sistema de segurança *default*, sem elevados níveis de defesas internas.

Para execução dos testes decidiu-se inicialmente quais serviços a rede ofereceria, e em quais sistemas operacionais esses serviços seriam instalados. Como resultado, a rede foi

estruturada de forma a ter, em sua base, os computadores de usuários instalados com Sistemas Operacionais Windows XP e Windows 7, que foram sistemas amplamente utilizados nos últimos anos. Os computadores de usuários seriam virtualmente interligados ao gerenciador de máquinas virtuais, que faz o papel de *switch* na topologia de rede, como pode ser observado na figura 4.2, que descreve a topologia de rede do cenário 2. Ainda analisando-se a figura, percebe-se que o endereçamento da rede local é 10.1.2.0/24.

Os principais serviços de rede a serem oferecidos foram instalados em uma máquina virtual rodando o sistema operacional Windows Server 2003. O Windows Server 2003 foi escolhido por ter sido amplamente utilizado como controlador de domínio de redes, e provavelmente permanece sendo utilizado em muitas empresas, e sendo mais antigo, crê-se que possua mais vulnerabilidades que o atual sistema da Microsoft, o Windows Server 2008. Alguns serviços instalados foram: *Active Directory,* SMTP, FTP, DHCP,DNS, INTRANET, etc. As máquinas de usuário e o controlador de domínio da rede estão interligados pela função de *switch*, exercido pelo gerenciador de máquinas virtuais. Contudo, ainda não se têm o controle de tráfego e roteamento entre diferentes redes. Desta forma, logo acima do *switch* foi inserido uma máquina virtual com a função de *firewall/router* da rede. Para tal, foi instalado o Sistema Operacional CentOS 6. O mesmo foi escolhido por ser uma versão *open source* do sistema pago Linux Red Hat e por ser conhecido por sua estabilidade, segurança e flexibilidade para implementação geral de segurança, além de outros serviços de rede. Foram instalados no CentoS 6 os serviços Iptables e Squid, com os quais é possível aplicar reencaminhamento de pacotes, filtro de pacotes e filtro de conteúdo respectivamente.

Determinou-se que o *firewall/router* seria o responsável pela conexão direta com o provedor de internet (ISP) e faria então o compartilhamento e distribuição de acesso à Internet para os demais usuários e servidores da rede.

Em suma, os servidores e usuários, já conseguiriam se comunicar via camada 2, visto que estão todos interligados pelo *switch*. É importante ressaltar que, para que haja sucesso no desenvolvimento do trabalho, deve haver algum mecanismo de filtro, sobre o tráfego da rede. Um administrador de rede deve ter o conhecimento de todo o tráfego que circula em sua rede, e para tanto, todo o tráfego deve ter um ponto em comum em sua rota, no qual possa ser implementando esse controle. Como o gerenciador de máquinas virtuais interligava por padrão todos os usuários, o que simula um *switch* de camada 2, configurou-se em todas as máquinas, um mecanismo de camada 3, uma rota estática, em que todos os pacotes devem, necessariamente, seguir o caminho: Máquina Remetente → Switch → Router/Firewall → Caminho do Destinatário.

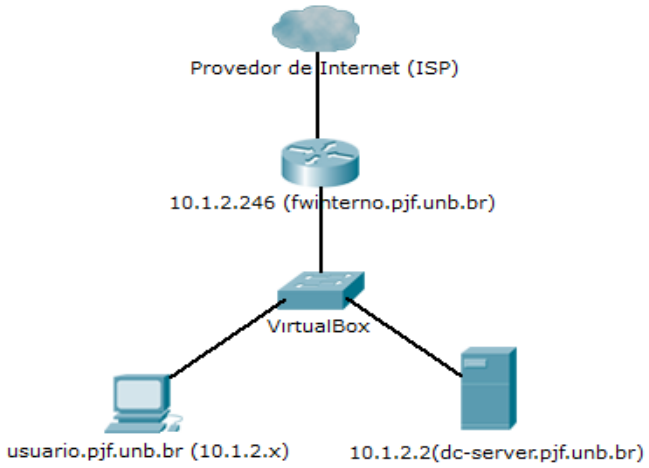


Figura 4.2 – Topologia de rede do Cenário **2**

4.1.1 – Detalhes da configuração dos hosts

A seguir, serão dados detalhes técnicos de configuração dos principais hosts e serviços que caracterizam as duas topologias utilizadas como referência durante o desenvolvimento do projeto.

4.1.1.1 - Windows Server 2003

No Windows Server 2003, instalou-se o serviço DHCP para atender a faixa 10.1.2.0/24.

No DNS foram inseridos os nomes do *router/firewall* e do controlador de domínio, onde o primeiro recebeu o IP fixo: 10.1.2.246 e o nome de domínio fwinterno.pjf.unb.br. Já o controlador de domínio da rede, a máquina virtual com Windows Server 2003, recebeu os seguintes ip fixo e nome: 10.1.2.2 e servidor-dc.pjf.unb.br.

Foram instalados também o protocolo POP3, necessário para configuração e funcionamento do serviço de e-mail, o SMTP. Posteriormente foi feita a configuração do Outlook, software proprietário Microsoft, usado no envio e recebimento de e-mails. Todos os usuários que desejem ingressar-se à rede e usufruir-se dos serviços, são criados no Windows Server 2003, por meio de um usuário com permissão de administrador. Permitiu-se ainda o serviço de acesso remoto, via *Remote Desktop* ,*software* proprietário Microsoft, para que um usuário com permissão de administrador pudesse fazer configurações remotamente.

4.1.1.2 - CentOS 6

No CentOS 6, foram instalados o Firewall Iptables e o filtro de conteúdo Squid. No Iptables, foi feita apenas uma configuração básica, visando o reencaminhamento de pacotes, função exercida do Linux como o roteador da rede, assim como permissão de pacotes ICMP e porta 22 para serviço SSH, para que o acesso remoto ao *firewall/router* possa ser feito por um administrador que precise executar tarefas remotamente. Foi configurado também pelo Iptables o compartilhamento de internet, que dessa forma permitia aos usuários e servidores da rede terem acesso à internet.

Com a instalação do Squid, permitiu-se a realização de algumas configurações, tais como limitação de banda por usuário, filtro de conteúdo e bloqueio ou liberação de acesso à endereços conforme o código padrão de permissões da empresa.

4.1.1.3 – Windows XP e Windows 7

Nas máquinas virtuais utilizadas por usuários comuns, empregou-se a configuração *default* fornecida pela Microsoft ao instalar os referidos sistemas operacionais e instalou-se os principais serviços utilizados pelos usuários nas empresas, como aplicativos de redes sociais e de *webmail.*

No caso do Windows XP, foram simulados ataques a *hosts* contendo diferentes versões de Service Pack, que é constituído por um pacote de correções para determinado programa ou sistema operacional. Desta forma, foram praticados ataques às diferentes versões (1,2 e 3) de forma a se ilustrar os ataques que cada versão pode vir a estar submetida.

Através da mitigação das vulnerabilidades, serão mostradas as configurações adicionais que devem ser empregadas em cada sistema operacional para correção das falhas.

4.2. ATAQUES EXTERNOS

Como já mencionado, tal classe de ataque objetiva maneiras de se ingressar em uma rede em específico e a partir daí realizar outras formas de ataque que possibilitem obtenção de informações de cunho confidencial ou comprometimento do funcionamento de determinados serviços. Em outra escala de análise, consiste também em promover tipos de ataque sem a necessidade de ingressar na rede, consistindo apenas na superação de defesas e contorno de obstáculos de forma a se atingir serviços em específico.

No projeto, foram realizadas duas formas de ataque: um ataque a uma rede sem fio para a obtenção de senha e um ataque a um servidor web que contém um *website* hospedado.

4.2.1 – Atacando a rede sem fio

O ataque objetiva o ingresso na rede sem fio. Para que o ataque seja executado, é necessário um computador móvel (*notebook* ou *netbook*) com uma memória RAM mínima de 1 GB, com pelo menos 130 GB livres de espaço no disco rígido. Como já mencionado, será utilizado o Sistema Operacional BackTrack para a realização dos ataques.

Assumiu-se a configuração de uma senha relativamente pequena, até 7 caracteres na rede sem fio configurada para o teste.

O objetivo do ataque foi a captura da autenticação, *handshake*, WPA/WPA2 entre um cliente e o ponto de acesso sem fio e então usar um software específico para craquear a chave pré-compartilhada. Senhas de 8 a 63 caracteres apresentam um grau de trabalho e dificuldade bem maior, dificultando a realização do ataque.

O ataque realizado é um ataque de força bruta no qual se faz necessário ter de antemão, um dicionário com o máximo possível de combinações de palavras para que via *software* e uma sequência de passos, possa ser quebrada a senha da rede. O software utilizado para o ataque é o aircrack-ng, nativo no BackTrack. A tabela 4.1 apresenta a explicação de alguns comandos utilizados para a quebra de senha.

Tabela 4.1 – Comandos utilizados no ataque à rede sem fio

aircrack-ng	Quebra chaves WEP ,WPA e WPA2
airmon-ng	Coloca placas diferentes em modo de monitor.
aireplay-ng	Injeção de pacotes.
airodump-ng	Coloca tráfego do ar em um arquivo .cap e mostra informação das redes.

Fez-se o monitoramento da rede sem fio por meio do airmon-ng. Praticou-se *sniffing* na rede wireless por meio da função airodump-ng. Foi possível assim enxergar dados como BSSID(que identifica o roteador wireless), MAC do ponto de acesso sem fio do alvo, canal de transmissão, tipo de encriptação configurada no mesmo, e ainda o MAC da placa de rede sem fio do atacante. Todo o tráfego “escutado” foi armazenado em um arquivo. Posteriormente, desconectou-se um cliente interligado ao ponto de acesso e capturou-se o *handshake*, quando o mesmo se reconectou. Aplicou-se então a ferramenta aircrack-ng, utilizando-se de um vasto dicionário de 33GB que através do *handshake* e do tráfego capturado realizou a quebra da senha.

Com a senha em mãos é possível então infiltrar-se na rede sem fio e então executar vários tipos de ataques objetivando-se uma escalada de privilégios, a partir da já inclusão do invasor dentro da rede interna.

4.2.2 – Atacando um website

O ataque ilustrado nesta seção consiste em uma forma de ataque DOS que como descrito na seção 2.1.4 se baseia na inundação de um determinado servidor com inúmeras requisições, fazendo com que este acabe entrando em um estado inoperante devido à alta carga de processamento. O alvo será o *website* testepentest.cjb.net, o qual foi criado apenas para fins de teste. O *website* está hospedado em um servidor Web da rede interna criada, estando registrado no domínio cjb.net.

Seguindo as etapas básicas de um teste de vulnerabilidade, deve-se inicialmente obter informações do alvo. Nesta etapa, poderia ser realizada uma consulta WHOIS, a qual consiste em retornar informações sobre o domínio alvo, ou mesmo um simples ping ao *website* que se deseja atacar, de forma a se obter o endereço IP do alvo. Para o ataque que será realizado a seguir, o endereço IP da vítima é a única informação necessária. A figura 4.3 ilustra a obtenção do IP do alvo.

```
root@bt:~# ping testepentest.cjb.net
PING testepentestet.cjb.net (177.133.88.5) 56(84) bytes of data.
64 bytes from 177.133.88.5: icmp_seq=1 ttl=247 time=30.1 ms
64 bytes from 177.133.88.5: icmp_seq=2 ttl=247 time=29.3 ms
64 bytes from 177.133.88.5: icmp_seq=3 ttl=247 time=32.8 ms
64 bytes from 177.133.88.5: icmp_seq=4 ttl=247 time=29.7 ms
64 bytes from 177.133.88.5: icmp_seq=5 ttl=247 time=29.0 ms
64 bytes from 177.133.88.5: icmp_seq=6 ttl=247 time=28.7 ms
```

Figura 4.3- Obtenção do IP do website alvo

Antes de se realizar o ataque, é preciso verificar se o website atacado apresenta *load balancing* (balanceamento de carga). *Load Balancing* é basicamente o processo de distribuição de solicitações de serviços de um servidor para um grupo de servidores. Assim, promove-se um aumento de escalabilidade e desempenho, além da capacidade de recuperação de falhas. Tal técnica minimiza a probabilidade de sobrecarga em servidores, além de promover uma otimização da largura de banda disponível. Desta forma, é um ótimo meio de proteção contra ataques que se baseiam em sobrecarga, como é o caso de ataques DOS.

Para o efetivo sucesso de um ataque DOS, é necessário a ausência de balanceadores de carga no servidor web a ser atacado. O script lbd.sh é uma ferramenta nativa do Bactrack 5 R2 própria para detecção de balanceadores de carga em servidores web. A figura 4.4 ilustra a

execução deste script em cima do IP 177.133.88.5, que corresponde ao servidor web que hospeda o website testepentest.cjb.net. Analisando a figura, é possível identificar a versão do servidor web analisado, o qual corresponde a um servidor Apache/2.2.20 (Unix), além da constatação de que este não apresenta balanceamento de carga.

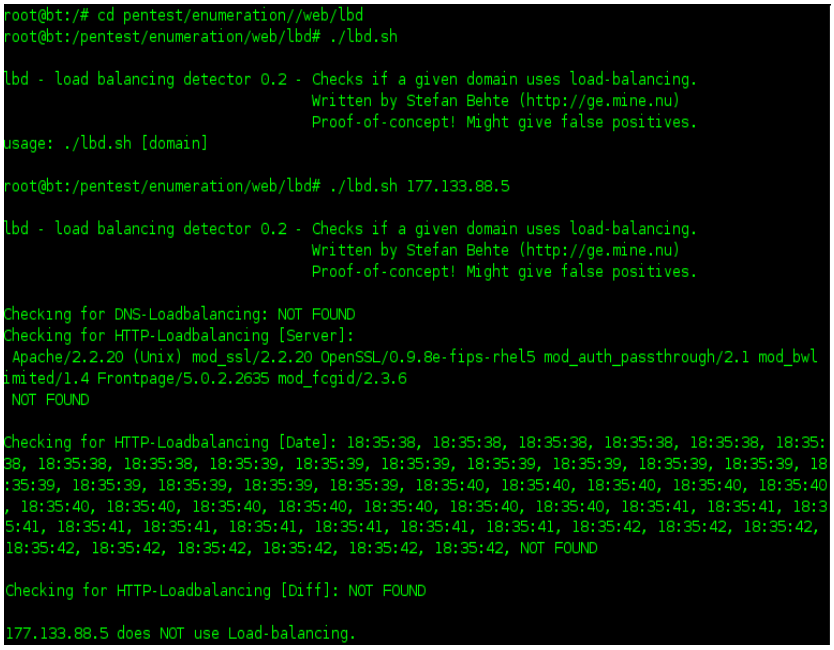


Figura 4.4- Execução do script lbd.sh

Verificada a ausência de balanceadores de carga, pode-se prosseguir com o ataque. Para a realização do DOS, utilizou-se o script slowloris.pl, o qual é o mais tradicional e eficiente script de conhecimento público para este propósito. O link para visualização do referido script pode ser acessado em: http://ha.ckers.org/slowloris/slowloris.pl.

O script slowloris.pl funciona enviando através de um processo *multithread* várias requisições parciais ao servidor Web alvo, as quais nunca são completadas. Servidores Web como o Apache mantem por um determinado período de tempo as conexões tcp. Tal característica, aliada ao envio de inúmeras requisições acarretam sobrecarga no servidor e a posterior queda de seus serviços. Outro problema proveniente do estabelecimento dessas requisições parciais é a falta de registro imediato nos logs dos servidores, o que faz com que a detecção do ataque por parte dos administradores seja dificultada.

A figura 4.5 ilustra a execução do slowloris.pl. A linha de comando **./**slowloris.pl –dns 177.133.88.5 –port 80 –timeout 50 –num 7000 –tcpto 5 –httpready significa a abertura de 7000 conexões simultâneas na porta 80 com 50 segundo de timeout de janela TCP e incremento de 5 segundos para o timeout TCP. A figura 4.6 ilustra a queda do website tespentest.cjb.net após um período de tempo de realização do ataque.

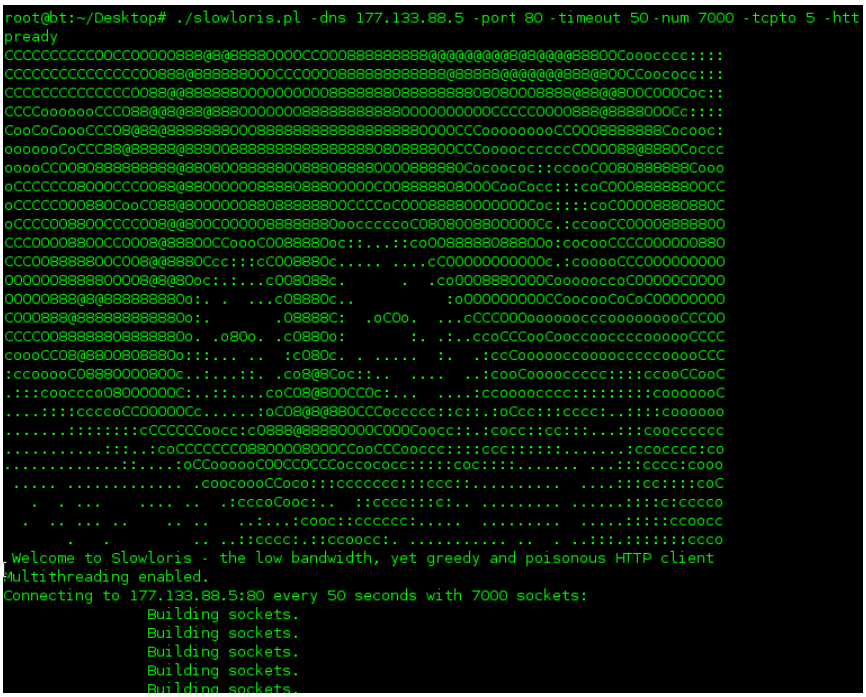


Figura 4.5- Execução do script slowloris.pl

Figura 4.6- Queda do site testepentest.cjb.net

4.3. ATAQUES INTERNOS

Como já mencionado anteriormente, nesta forma de ataque o usuário já se encontra inserido na rede interna e desta maneira, tendo um conhecimento diferenciado, pode causar danos a determinados serviços. Diversas formas de ataque foram desenvolvidas no desenvolvimento do projeto, em especial os tipos descritos no capítulo 2.

4.3.1 – Descoberta dos hosts ativos na rede

Seguindo o passo a passo das etapas de um teste de vulnerabilidades, antes de qualquer coisa é preciso obter informações dos possíveis alvos em potencial através da descoberta de hosts ativos. Uma das formas de se obter tal informação, é através da ferramenta NMAP (Lyon, 2009).

Inicialmente obteve-se informações sobre possíveis hosts no range 10.1.2.0/24, que representa a rede interna. Através do método básico de pings NMAP, foi possível obter informações das máquinas que estavam ativas, tais como: IP, MAC, nome no DNS dos *hosts* encontrados e ainda a latência na comunicação com os mesmos. A figura 4.7 ilustra o resultado da busca inicial.

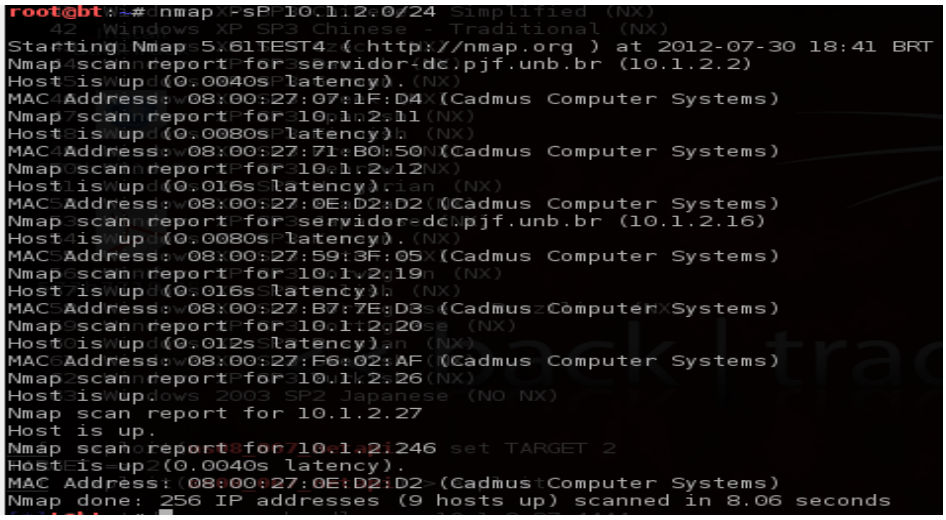


Figura 4.7 – Resultado do comando: nmap –sP 10.1.2.0/24

Com as informações adquiridas, o atacante pode aprofundar sua base de informações sobre cada *host* e montar aos poucos o seu quadro de ataques. Sabendo quais hosts estão ativos, é possível promover um scanner de vulnerabilidades no alvo que se deseja atacar.

4.3.2– Ataques ao firewall/router

Através da ferramenta NMAP, foi possível descobrir que o host 10.1.2.246 era o *gateway* da rede, bem como possuía o *hostname* fwinterno.pjf.unb.br, o que leva a concluir que tal host é o *router/firewall* da rede a ser atacada. A seguir, serão dados detalhes das ações praticadas para a realização dos ataques em cima de tal host.

4.3.2.1. – Informações adicionais com scanner de portas

Efetuada a etapa de descoberta por meio de pings, utilizou-se o método de scanner SYN, que verifica rapidamente milhares de portas por segundo e não é prejudicado por *firewalls* restritivos. Buscou-se também informações adicionais, tais como estado e tipos de serviços rodando em cada porta, versão do sistema operacional ativo, etc. A figura 4.8 ilustra o resultado obtido. Nela, pode se observar que a porta 22 encontra-se aberta e que está associada ao serviço OpenSSH. Além disso, pode-se observar que o Sistema Operacional é Linux com versão de *kernel* 2.6.39 e que o invasor está a 1 *hop* da vítima. Com uma pesquisa posterior, pode-se chegar a conclusão de que se trata da versão Linux CentOS 6.

```
root@bt: # nmap -Pn -sS -sV -O  10.1.2.246

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-07-30 18:28 BRT
Nmap scan report for 10.1.2.246
Host is up (0.0027s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.3 (protocol 2.0)
MAC Address: 08:00:27:0E:D2:D2 (Cadmus Computer Systems)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|specialized|WAP|media device
Running (JUST GUESSING): Linux 2.6.X (92%), Crestron 2-Series (90%), Netgear embedded (90%), Western Digital em
bedded (90%)
OS CPE: cpe:/o:linux:kernel:2.6 cpe:/o:crestron:2_series
Aggressive OS guesses: Linux 2.6.38 - 2.6.39 (92%), Linux 2.6.38 (91%), Linux 2.6.22 - 2.6.36 (91%), Linux 2.6.
37 (91%), Linux 2.6.39 (90%), Crestron XPanel control system (90%), Netgear DG834G WAP or Western Digital WD TV
 media player (90%), Linux 2.6.24 - 2.6.36 (86%), Linux 2.6.22 (85%), Linux 2.6.23 - 2.6.38 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.66 seconds
```

Figura 4.8 – Scanner detalhado do host 10.1.2.246

4.3.2.2. – Scanner de vulnerabilidades do firewall interno

Após obter diversas informações importantes a partir do uso do NMAP, tais como IP, MAC, nome DNS, sistemas operacionais instalados, distância em saltos e portas e serviços abertos, o próximo passo consiste na realização do scanner de vulnerabilidades. Para tal tarefa foi usado o software OpenVAS, que é bem descrito em 3.2.1.

Após a varredura das vulnerabilidades, foi gerado um relatório contendo as vulnerabilidades encontradas, a descrição de cada uma delas, bem como o nível de impacto que cada uma apresenta. É gerado também, um quadro resumo que relaciona o impacto de cada vulnerabilidade aos serviços que rodam nas portas com estado aberto, como pode ser observado na Tabela 4.2.

Tabela 4.2- Resumo das vulnerabilidades encontradas no firewall interno

Port Summary for Host 10.1.2.246

Service (Port)	Threat Level
ssh (22/tcp)	Medium
general/icmp	Low
general/tcp	Low
general/CPE-T	Log
general/HOST-T	Log

A figura 4.9 expõe duas das vulnerabilidades presentes no *firewall*, uma de potencial médio, baseada no protocolo SSH, que pode vir a estar submetido a ataques de DOS e outra de potencial baixo, destacando apenas a presença da rota estabelecida entre os hosts 10.1.2.246 e 10.1.2.15, o que ajuda a corroborar a hipótese de que 10.1.2.246 é o *gateway* da rede.

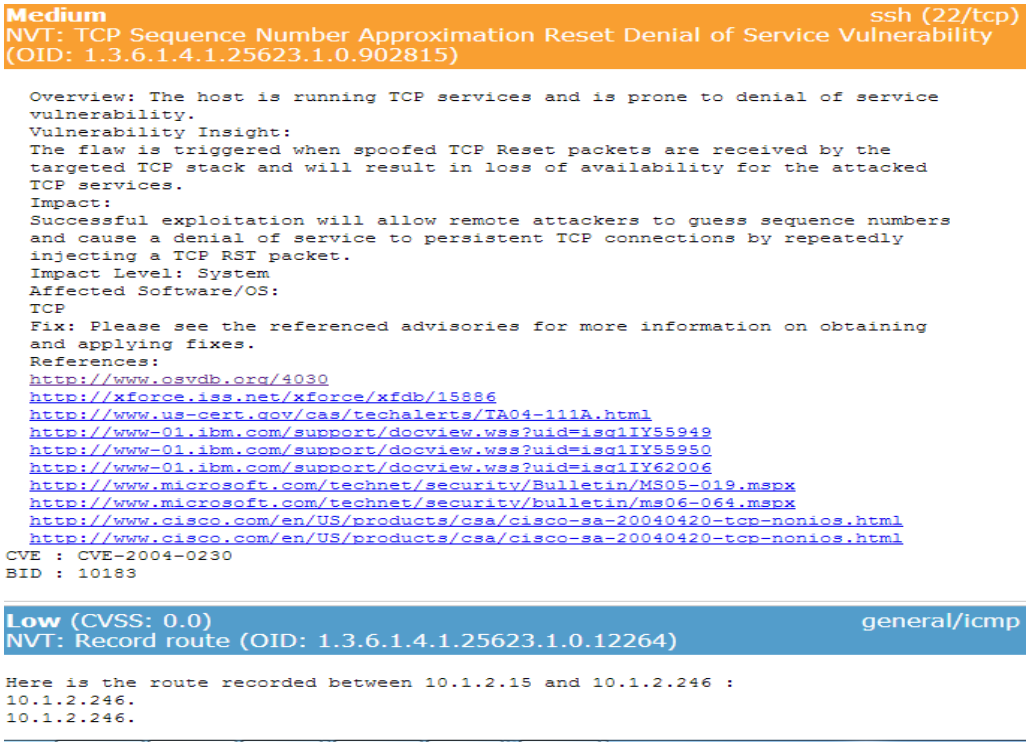


Figura 4.9- Descrição de falhas de potencial médio e baixo presentes no CentOS 6

4.3.2.3. – Ataque de inundação de pacotes do tipo SYN

O ataque realizado nesta seção explora a vulnerabilidade de potencial médio apresentada na figura 4.9. A vulnerabilidade em questão se refere a execução de serviços TCP propensos a ataques DOS. Sabendo que uma conexão está ativa, é possível enviar um pacote TCP forjado com a flag RST (responsável por encerrar uma conexão) setada, com o objetivo de derrubar conexões legítimas. Posteriormente, pode-se realizar um tipo de ataque DOS

basadeado na inundação de pacotes SYN, conhecido como SYN *Flood*. Esta forma de ataque consiste em enviar um grande volume de pacotes SYN(flag para iniciar uma conexão), o que faz com que estoure o limite de conexões e como consequência, o servidor deixe de responder a novas conexões, mesmo que exista banda disponível.

Para a realização deste tipo de ataque, foi utilizada uma ferramenta conhecida como HPING na versão 3**,** ferramenta nativa do BackTrack própria para ataques de SYN *Flood*. A figura 4.10 ilustra a execução do ataque. O comando apresentado na figura significa que 10000 pacotes SYN serão enviados na porta 80 para o endereço 10.1.2.246, que corresponde ao endereço IP do host fwinterno.pjf.unb.br. Outro ponto de destaque pode ser observado ao se utilizar a opção *spoof*, que substitui o verdadeiro IP do host do atacante por um IP *fake*, camuflando desta forma a identidade do invasor.

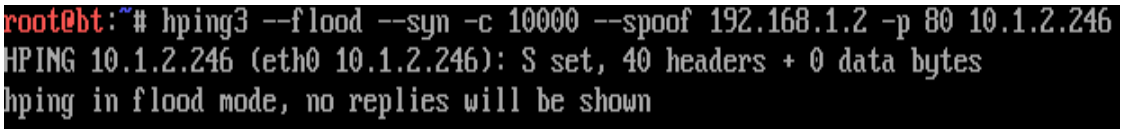


Figura 4.10- Execução do hping

A figura 4.11 ilustra uma tela de captura do Wireshark, onde pode ser visto a enorme quantidade de pacotes SYN enviados para o host 10.1.2.246. A figura 4.12 mostra a falta de conexão com a internet (fato demonstrando pela falta de comunicação por ping com o domínio www.google.com) em consequência do ataque realizado.

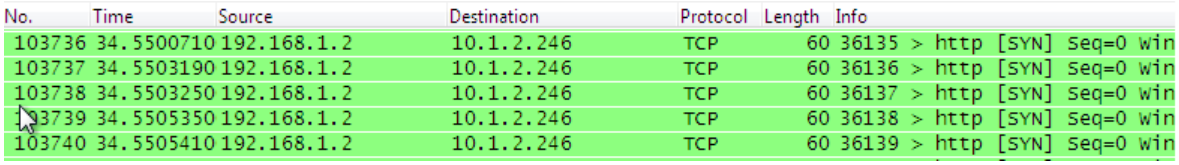


Figura 4.11- Ilustração do ataque SYN Flood

```
root@fwinterno:/# ping www.google.com
ping: unknown host www.google.com
root@fwinterno:/#
```

Figura 4.12- DOS da internet do host alvo

Ataque SYN *Flood* é apenas uma das opções que o hping pode realizar. Na sua versão 3, foi desenvolvida uma opção que permite, por exemplo, uma transferência de informações através do protocolo ICMP(protocolo do ping). Desta forma, é possível se extrair a partir do

hping as informações de um determinado arquivo presente no host alvo. No caso do host 10.1.2.246, por ser um sistema Linux, poderia se extrair, por exemplo, as informações do diretório /etc/passwd, o qual contém informações de nomes e senhas de autenticação dos usuários do sistema.

4.3.3 – Ataque ao controlador de domíno

Através do NMAP, foi possível identificar que o *host* 10.1.2.2 encontrava-se ativo. Tal host apresentava como nome servidor-dc.pjf.unb.br. , o que por intuição sugere que seja o controlador de domínio da rede. A seguir, serão dados detalhes das ações praticadas para a realização dos ataques em cima de tal host.

4.3.3.1 – Informações adicionais com scanner de porta

Realiza-se agora o método de scanner NMAP SYN para descoberta de estado de portas, serviços, versões de sistema operacional, entre outras informações. O resultado pode ser observado na figura 4.13.

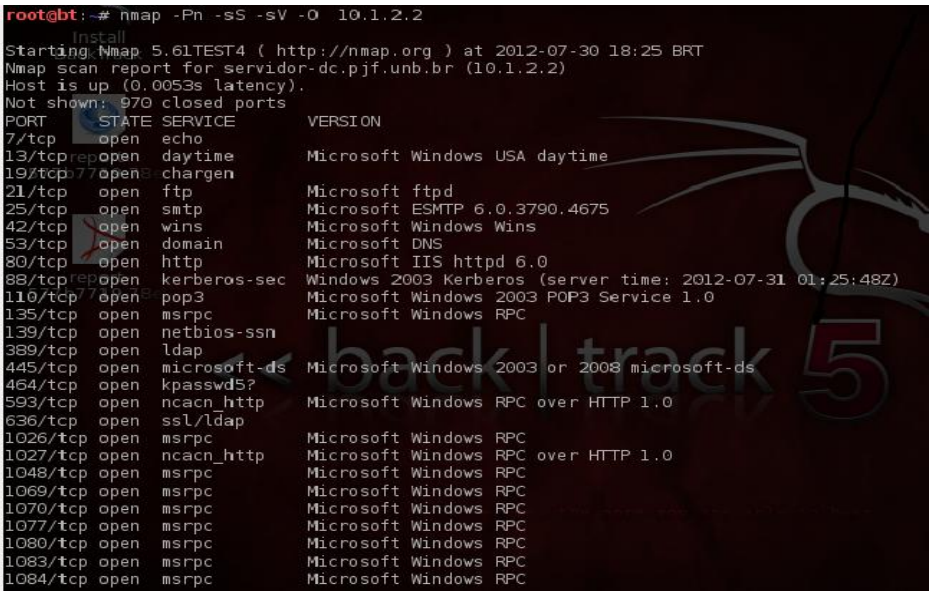


Figura 4.13 – Scanner parcial detalhado do IP 10.1.2.2

Através do scanner, foi possível identificar várias portas em estado *open*, dentre elas, portas de serviços comuns, como: 21/tcp, 53/tcp, 80/tcp, 110/tcp, que se referem respectivamente aos serviços ftp, dns, http e pop3. Além disso, foi possível identificar que o Sistema Operacional do alvo era o Windows Server 2003. Outras informações adicionais que puderam ser obtidas foram o nome de serviços rodando nas portas abertas, como o

kerberosec, o qual é um protocolo de segurança. Além disso, foi possível também identificar que o alvo encontrava-se a um salto de distância (*hop*) do invasor.

4.3.3.2. – Scanner de vulnerabilidades do controlador de domínio

O objetivo é realizar a varredura de vulnerabilidades para geração de relatórios detalhados. O quadro resumo que relaciona o impacto de cada vulnerabilidade aos serviços que rodam nas portas com estado aberto, como pode ser observado na Tabela 4.3.

Tabela 4.3 – Resumo de algumas vulnerabilidades encontradas no Windows Server 2003

Port Summary for Host 10.1.2.2

Service (Port)	Threat Level
discard (9/tcp)	High
microsoft-ds (445/tcp)	High
ms-wbt-server (3389/tcp)	High
name (42/tcp)	High
smtp (25/tcp)	High
ftp (21/tcp)	Medium
qotd (17/tcp)	Medium
domain (53/tcp)	Low
domain (53/udp)	Low

High (CVSS: 9.3) ms-wbt-server (3389/tcp)
NVT: Microsoft Remote Desktop Protocol Remote Code Execution Vulnerabilities (267138... (OID: 1.3.6.1.4.1.25623.1.0.902818)

Overview: This host has critical security update missing according to Microsoft Bulletin MS12-020.
Vulnerability Insight:
The flaws are caused due to the way Remote Desktop Protocol accesses an object in memory that has been improperly initialized or has been deleted and the way RDP service processes the packets.
Impact:
Successful exploitation could allow remote attackers to execute arbitrary code as the logged-on user or cause a denial of service condition.
Impact Level: System/Application
Affected Software/OS:
Microsoft Windows 7 Service Pack 1 and prior
Microsoft Windows XP Service Pack 3 and prior
Microsoft Windows 2K3 Service Pack 2 and prior
Microsoft Windows Vista Service Pack 2 and prior
Microsoft Windows Server 2008 Service Pack 2 and prior
Fix:
Run Windows Update and update the listed hotfixes or download and update mentioned hotfixes in the advisory from the below link,
http://technet.microsoft.com/en-us/security/bulletin/ms12-020
References:
http://blog.binaryninjas.org/?p=58
http://secunia.com/advisories/48395
http://support.microsoft.com/kb/2671387
http://www.securitytracker.com/id/1026790
http://technet.microsoft.com/en-us/security/bulletin/ms12-020
CVE : CVE-2012-0002, CVE-2012-0152
BID : 52353, 52354

Figura 4.14 – Vulnerabilidade no Windows Server 2003 - ms-wbt-server (3389/tcp)

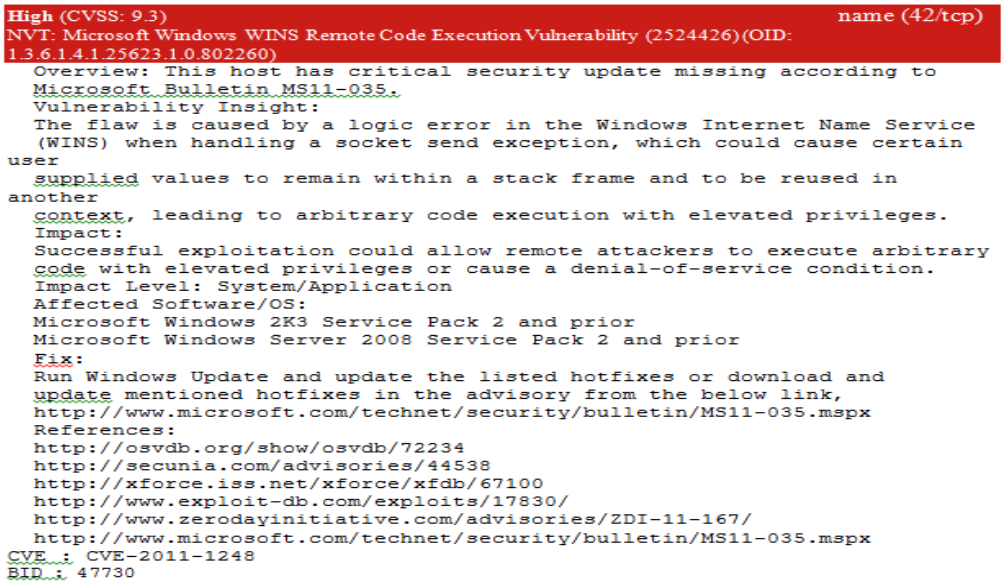


Figura 4.15 – Vulnerabilidade no Windows Server 2003 - name (42/tcp)

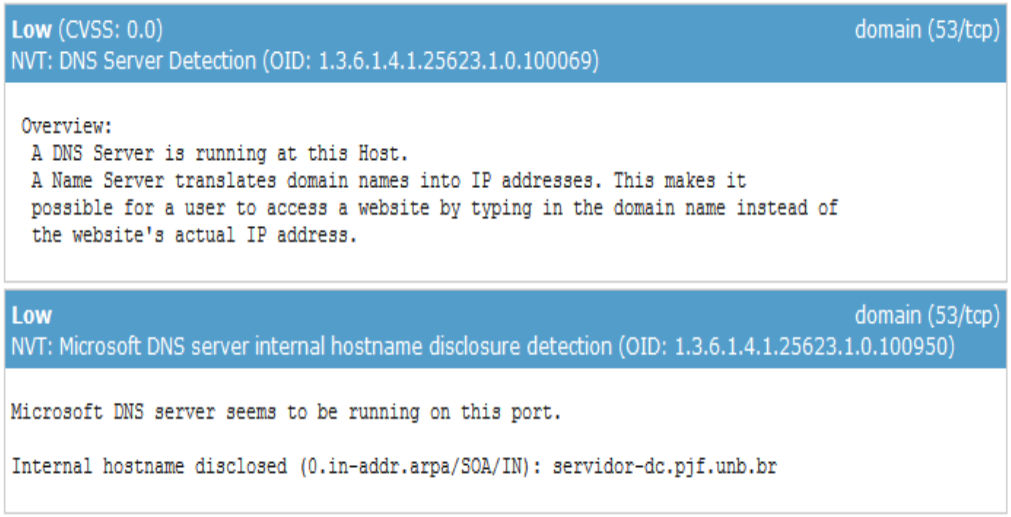


Figura 4.16 – Vulnerabilidade no Windows Server 2003 - domain (53/tcp)

As figuras 4.14, 4.15 e 4.16 são exemplos de como as vulnerabilidades são referenciadas no relatórios gerados pelo OpenVAS e ilustram as vulnerabilidades de alto e baixo impacto que foram encontrados após o scanner no Windows Server 2003. Analisando tais informações, é possível ter uma visão completa sobre as vulnerabilidades, tais como: boletins de ocorrências, tipos de ataques que podem ser realizados, maneiras de se praticar ataques, referências a bancos de dados de vulnerabilidades, como o CVE e o MSB (como podem ser observados nas tabelas), além de sugestões para a mitigação das vulnerabilidades. Tendo em mãos, tais informações, a seguir serão ilustradas algumas formas de ataque praticadas em cima das referenciadas falhas.

4.3.3.3. – Ataque DOS

Tendo em mãos os relatórios de vulnerabilidades obtidos, configurou-se alguns exploits para promoção de ataques no host 10.1.2.2 (Windows Server).

A primeira falha de segurança explorada foi a mostrada na figura 4.14, cujo boletim de ocorrência liberado pela Microsoft tem o seguinte identificador: MS12-020. Trata-se de um ataque do tipo *Deny of Service*. Através da execução do exploit ms12_020_maxchannelids no Metasploit Framework, o resultado será o desligamento do host alvo, através da famosa "tela azul". Desta forma, todos os serviços fornecidos pelo host serão paralisados temporariamente devido ao desligamento. A figura 4.17 ilustra a execução do referido exploit.

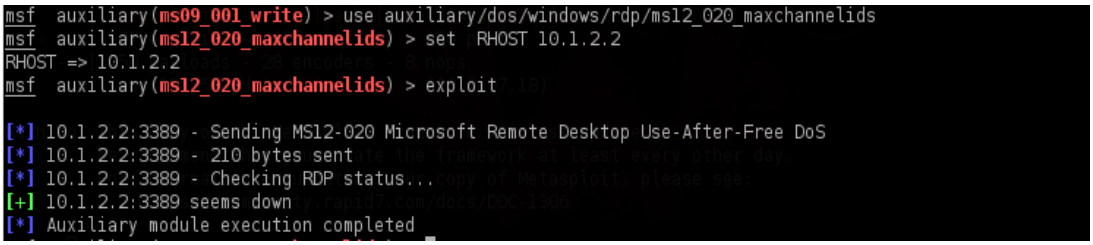


Figura 4.17 – Execução do exploit ms12_020_maxchannelids

Uma variante desta forma de ataque é a execução de um código avulso que não faz uso do Metasploit Framework e tem como base a vulnerabilidade apresenta na figura 4.15, descrita pelo boletim de segurança MS11-035. O código pode ser acessado através de referências externas listadas nesta mesma tabela, mas especificamente no acervo: http://www.exploit-db.com/exploits/17830/. Tal código explora a referida falha de por meio de *buffer overflow.*

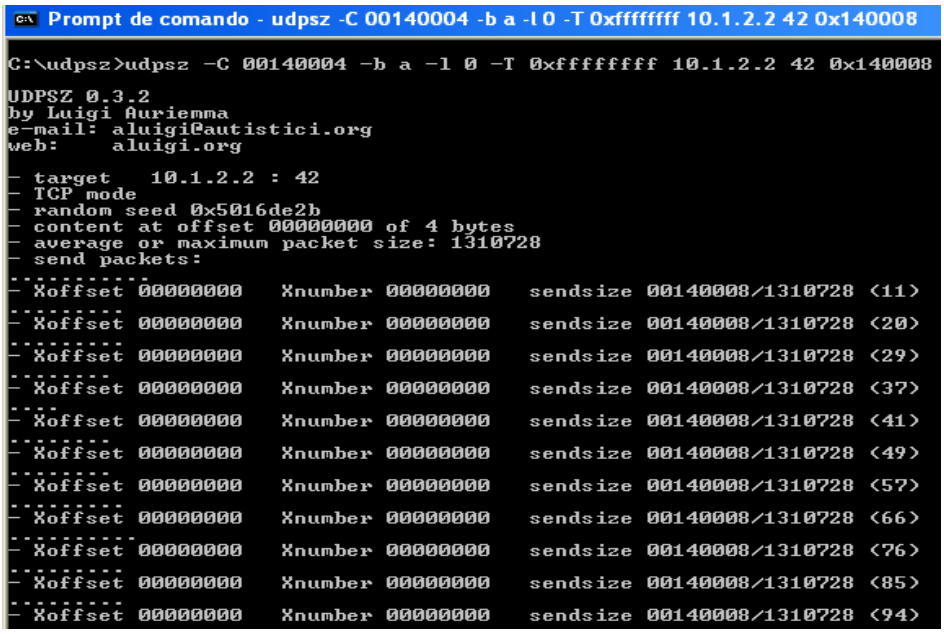


Figura 4.18 – Exploit que explora técnica de buffer overflow, 94 pacotes enviados

Figura 4.19 – Buffer Overflow após o envio de 751 pacotes

As figuras 4.18 e 4.19 ilustram a execução do código, sendo claro observar que o objetivo é o estouro do espaço livre de memória RAM pela quantidade de pacotes enviados. A figura 4.20 exibe a "tela azul da morte" que foi observada no sistema operacional das máquinas alvo durante a realização dos dois ataques DOS anteriores. O resultado final foi o desligamento forçado do *host* alvo, o que no laboratório estudado provocaria sérios danos, já que o host 10.1.2.2 roda serviços importantes, como o DNS e o DHCP.

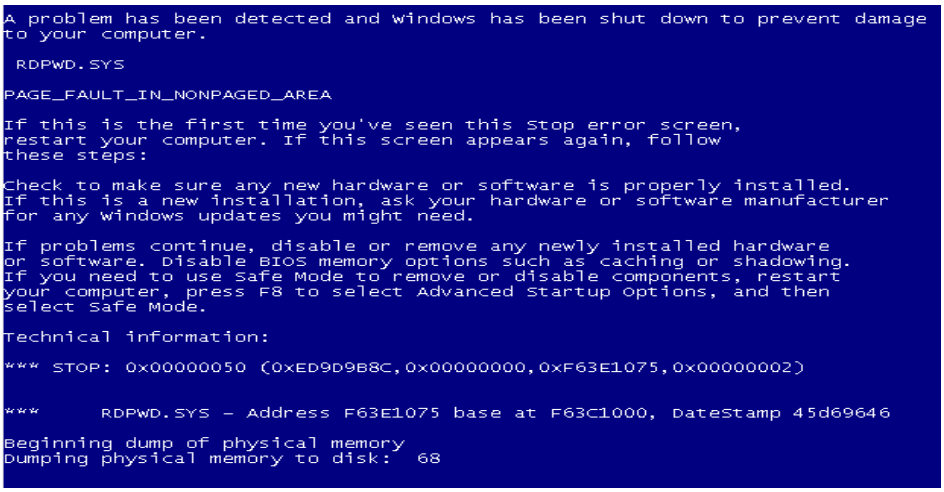


Figura 4.20 – Host alvo após os ataques de DOS

4.3.3.4 – Ataque de falsificação de pacotes na camada 2

Ainda tendo como o alvo o host 10.1.2.2, foi realizado um ataque de falsificação de pacotes na camada 2, conhecido como ARP *Spoofing*, que como descrito na seção 2.1.2 é um tipo de ataque onde o invasor passa a ficar localizado entre dois computadores que estabelecem uma comunicação através de alterações na tabela ARP e consequente substituição do MAC do host original pelo MAC do host invasor, promovendo desta forma um desvio de tráfego.

Tal ataque pode ser explorado graças a vulnerabilidades como as presentes nos serviços discard (9/tcp) e ms-wbt-server (3389/tcp), os quais não verificam a identidade do

servidor a cada nova conexão, o que já não ocorre em serviços como o SSH, por exemplo, o qual a cada nova conexão, pede confirmação de identidade e aborta a conexão caso haja suspeita de pacote adulterado. Em uma escala maior, a referida forma de ataque acaba atingindo sistemas operacionais que em determinados segmentos não trabalham com dados criptografados na camada 2, o que facilita a realização de formas de ataque deste cunho.

Para melhor entendimento, o ataque ocorreu da seguinte maneira: O host invasor passou a responder na rede como se tivesse o MAC do *firewall/router* 10.1.2.246, que como dito anteriormente, representa o *gateway* do host 10.1.2.2. Desta maneira, todo o tráfego da rede, antes direcionado ao host 10.1.2.246 verdadeiro, passou a ser encaminhado primeiramente para o falso destinatário, ou seja, a máquina do atacante. Assim, o invasor passou a enxergar todo o tráfego da rede, podendo conseguir uma quantidade enorme de informações restritas. É importante lembrar que para que a comunicação ocorra naturalmente, o ARP Spoofing tem ainda um mecanismo, pelo qual a máquina do atacante, após receber pacotes falsamente direcionados a ela, reencaminha os mesmos para o destinatário verdadeiro.

Explorou-se o ARP *Spoofing* de duas formas: pelo método 1 e pelo método 2.

No método 1, o ataque foi feito através da ferramenta arpspoof nativa do Backtrack e a posterior interceptação dos pacotes via Wireshark.

No método 2, utilizou-se a ferramenta Cain & Abel, que explora de forma automática o ARP *Spoofing* e a captura de informações de *login*, dos mais diversos serviços, através dos pacotes interceptados.

No método 1, ao tentar se logar em um dado servidor, o administrador responsável pelo host 10.1.2.2 tem os dados de autenticação interceptados pelo atacante. O ataque é exibido nas figuras 4.18, 4.19 e 4.20 a seguir.

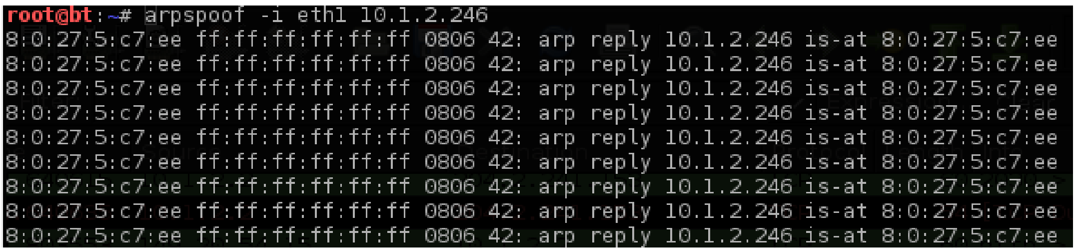


Figura 4.21 – Host invasor responde como se fosse o host 10.1.2.246

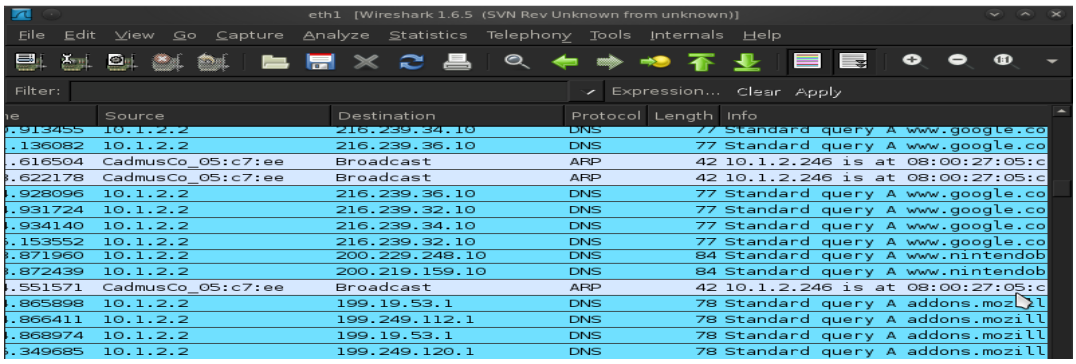


Figura 4.22 – Atacante inicia interceptação de dados do host 10.1.2.2 via Wireshark

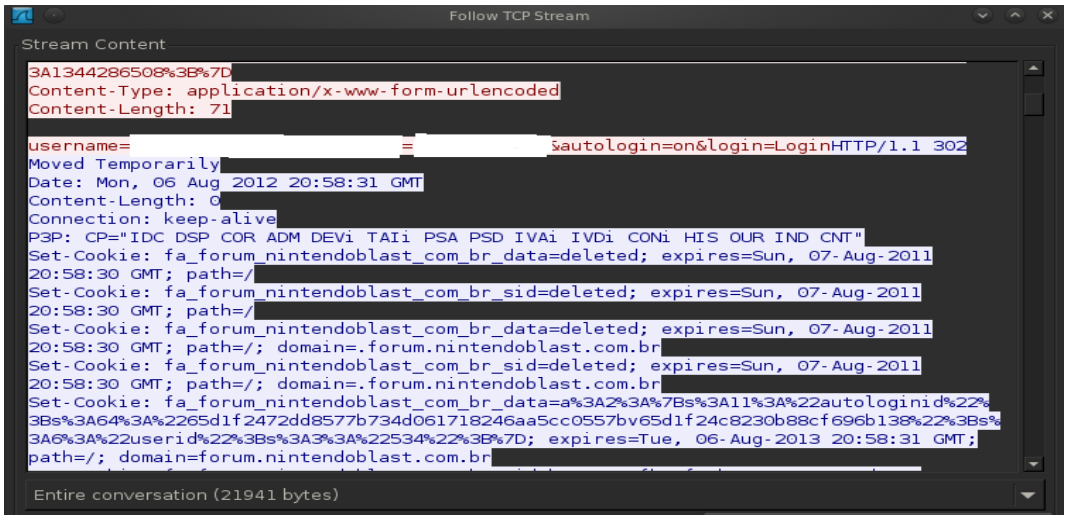


Figura 4.23 – Username e Password capturados com sucesso

A figura 4.21 mostra o ataque ARP *Spoofing* em ação, falsificando respostas. O host do atacante passa a responder falsamente pelo host 10.1.2.246. O atacante situou-se entre o host 10.1.2.2 e o *firewall/router* 10.1.2.246, podendo enxergar desta forma todo o tráfego trocados entre eles.

A figura 4.22, mostra a interceptação dos dados feita através do *sniffer* Wireshark. Posteriormente, a figura 4.23 mostra a interceptação do pacote que contém informações de *login*, exibindo-se o fluxo TCP completo do pacote interceptado, através do qual é possível ver as informações confidenciais capturadas nos campos *username* e *login* do fluxo analisado.

No método 2, é apresentada uma variação do ataque de ARP *Spoofing*, o qual é realizado de forma automática pela ferramenta Cain & Abel.

Cain & Abel é uma ferramenta que em um primeiro momento foi criada para auxiliar usuários que perderam suas senhas e deseja recuperá-los. Posteriormente também acabou sendo utilizada para realização de ataques. Ela permite a quebra de senhas criptografadas utilizando ataques de criptoanálise como *rainbow tables* e ataques de força bruta. Além disso,

também pode interceptar informações de autenticação como *logins* e senhas através de ARP *Spoofing*. As figuras 4.24 e 4.25 ilustram respectivamente a interceptação de *logins* e senhas em um servidor FTP e em um servidor de *e-mail* que utiliza o protocolo POP3. As referidas interceptações foram realizadas com Cain & Abel por meio de ARP *Spoofing*.

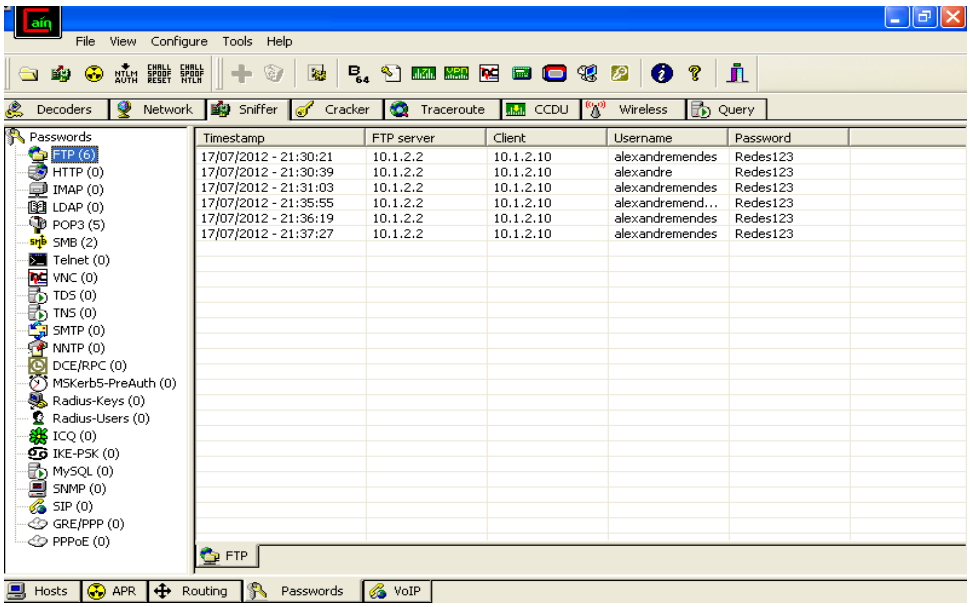


Figura 4.24- Interceptação de login e senha de servidor FTP

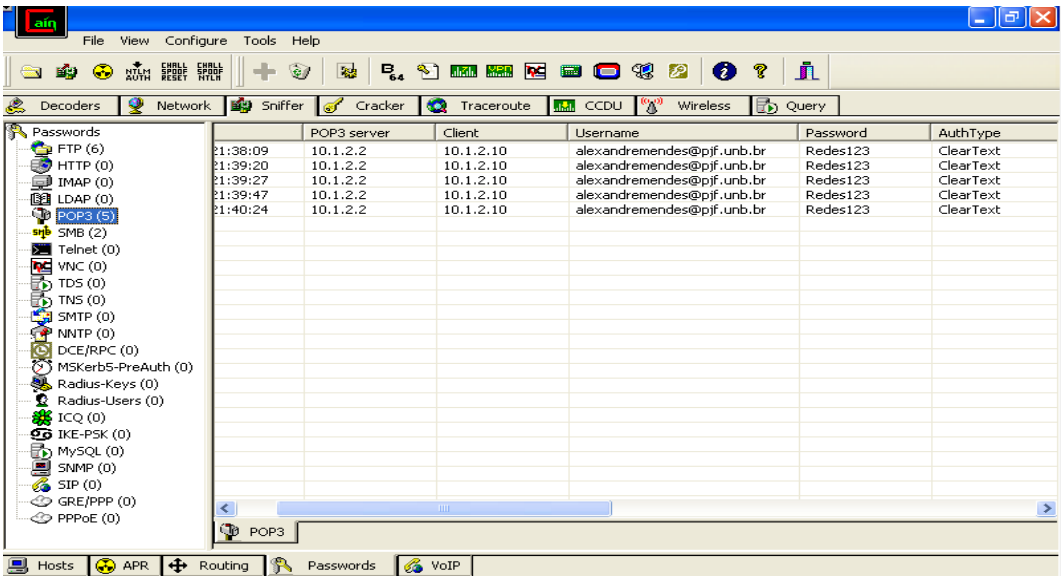


Figura 4.25- Interceptação de login e senha de servidor de e-mail

4.3.3.5 – Ataque de envenenamento de cache DNS

Como referenciado em 2.1.3, *Cache Poisoning* (Envenenamento de Cache) é um tipo de ataque onde são fornecidas respostas falsas a um servidor de cache DNS, o qual passar a armazenar resoluções DNS falsas, redirecionando as consultas para o *host* do invasor. Desta

forma, estabelece-se de forma indireta uma conexão entre um *host* que faz uma consulta ao *host* do atacante. Assim, o invasor consegue interceptar o tráfego entre o *host* alvo e o servidor DNS real.

O ataque foi em cima de uma condição de porta aberta descrita na figura 4.16 e ilustra como uma vulnerabilidade de baixo impacto pode se agravar e causar danos impactantes ao sistema. O ataque foi possível graças a uma vulnerabilidade própria do Microsoft DNS no Windows Server 2003 conhecida como "*The Kaminsky Bug*", a qual permite execução de códigos remotos que conduzem ao envenenamento do cache. Tal vulnerabilidade, aliada ao fato da porta 53 estar aberta, possibilitou a realização do ataque.

Utilizou-se o exploit bailiwicked_host do Metasploit Framework, que explora a vulnerabilidade citada anteriormente. Neste exploit, há uma alteração entre *hostnames* e endereços IP. Tal fato pode ser claramente observado na figura 4.26, onde o *hostname* pjf.unb.br (que corresponde ao Windows Server 2003, onde está hospedado o DNS) será associado ao novo IP 10.1.2.11, que corresponde ao host do invasor.

Figura 4.26 – Execução do exploit bailiwicked_host.

Na figura 4.27 é possível observar o prosseguimento da execução do ataque. Observa-se o envio de pacotes envenenados ao host 10.1.2.2, que originalmente apresenta o *hostname* pjf.unb.br. Terminado a execução do ataque, através do comando nslookup (que mostra o IP associado a um dado *hostname*) é possível verificar que o *hostname* pjf.unb.br agora está associado ao endereço 10.1.2.11, que corresponde ao *host* invasor. Com isso, o envenenamento do cache é concluído. Ao realizar uma consulta ao cache DNS, um host alvo será redirecionado de forma indireta a estabelecer uma conexão com o *host* do atacante devido à falsa resolução de nomes que comprometeu o cache DNS.

Figura 4.27 – Ilustração do envenenamento do cache

4.3.4 – Ataques ao Windows XP

Na fase de descoberta de hosts ativos, foi possível se identificar alguns endereços, provavelmente *desktops* de usuários da rede interna. Tais *hosts* não apresentavam nenhum nome em específico, o que por intuição sugere que sejam *hosts* de usuários. A seguir, serão dados detalhes das ações praticadas para a realização dos ataques em cima de hosts com Sistema Operacional Windows XP, com diferentes níveis de segurança, desde configurações com Service Pack 1, até configurações com Service Pack 3.

4.3.4.1 – Scanner de vulnerabilidades

Depois de se identificar informações a respeito de MAC, IP e versões do sistema operacional, procede-se agora com a varredura de vulnerabilidades dos *hosts* escolhidos como alvo. O *software* utilizado aqui foi o Nessus, que basicamente apresenta a mesma estrutura de funcionamento e análise de dados que o OpenVAS. A principal diferença consiste na licença, já que o Nessus é um *software* de distribuição paga, diferentemente do OpenVAS que é *open source.*

Foram gerados relatórios com descrição em detalhes da vulnerabilidade encontrada e referências a bancos de dados externos de vulnerabilidade, sobretudo aos boletins de segurança da Microsoft (MSB) e ao CVE. O resultado do scanner de varredura realizados em um dado host com Sistema Operacional Windows XP é descrito na figura 4.28. A seguir, serão descritos alguns ataques em cima das referenciadas vulnerabilidades. As figuras 4.29, 4.30 e 4.31 apresentam mais detalhes das vulnerabilidades de potencial crítico presentes na figura 4.28.

Summary

Critical	High	Medium	Low	Info	Total
3	1	0	2	19	25

Details

Severity	Plugin Id	Name
Critical (10.0)	34476	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Unspecified Remote Code Execution (958644)
Critical (10.0)	35361	MS09-001: Vulnerabilities in SMB Could Allow Remote Code Execution (958687)
Critical (10.0)	53514	MS11-017: Vulnerabilities in Remote Desktop Connection Could Allow Remote Code Execution (2509553)
High (7.8)	26964	MS07-058: Vulnerability in RPC Could Allow Denial of Service (933729)

Figura 4.28 – Resumo de algumas vulnerabilidades encontradas no Windows XP

MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Unspecified Remote Code Execution (958644)

Synopsis
Arbitrary code can be executed on the remote host due to a flaw in the 'server' service.

Plugin Output

- C:\Windows\system32\Netapi32.dll has not been patched
Remote version : 6.0.6001.18000
Should be : 6.0.6001.18157

Description
The remote host is vulnerable to a buffer overrun in the 'Server' service which may allow an attacker to execute arbitrary code on the remote host with the 'System' privileges.

Solution
Microsoft has released a set of patches for Windows 2000, XP, 2003, Vista and 2008 :

http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx

Risk Factor
Critical/ CVSS Base Score: 10.0
(CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

Figura 4.29 – Descrição da vulnerabilidade MS08-067

MS09-001: Vulnerabilities in SMB Could Allow Remote Code Execution (958687)

Synopsis
It may be possible to execute arbitrary code on the remote host due to a flaw in SMB.

- C:\Windows\system32\drivers\Srv.sys has not been patched
Remote version : 6.0.6001.18000
Should be : 6.0.6001.18185

Description
The remote host is affected by a memory corruption vulnerability in SMB that may allow an attacker to execute arbitrary code or perform a denial of service against the remote host.

Solution
Microsoft has released a set of patches for Windows 2000, XP, 2003, Vista and 2008 :

http://www.microsoft.com/technet/security/bulletin/ms09-001.mspx

Risk Factor
Critical/ CVSS Base Score: 10.0

Figura 4.30 – Descrição da vulnerabilidade MS09-001

Figura 4.31 – Descrição da vulnerabilidade MS11-017

4.3.4.2 – Ataque de sequestro de sessão

Como dito na seção 2.1.7 , tal forma de ataque explora uma sessão válida de um computador de forma a se quebrar acessos restritos. O ataque em questão foi realizado com o intuito de se obter acesso a um serviço do *host* alvo e em seguida promover a execução de um código remoto que promova outras formas de ataque.

A vulnerabilidade explorada para a execução do ataque é descrita no boletim de segurança da Microsoft MS08-067, descrita na figura 4.29. Em poucas palavras, é caracterizada por uma falha de autenticação no serviço *Remote Procedure Call* (RPC). A partir de tal vulnerabilidade, é possível se promover a execução de códigos arbitrários que permitem o controle total ou parcial do *host* alvo.

Para a realização do ataque, foi utilizado o exploit ms08_067_netapi do Metasploit Framework. O alvo foi a máquina 10.1.2.13, configurada com Windows XP Service Pack 2. A figura 4.32 ilustra a execução do referido exploit. Como pode se observar, após a execução do exploit, o resultado é a abertura de uma sessão Meterpreter.

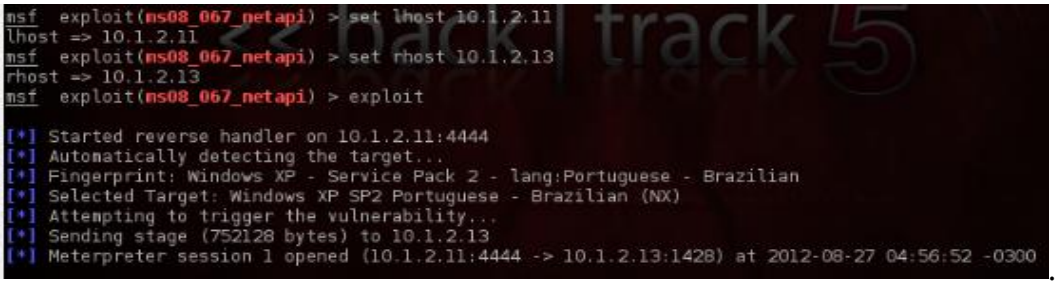


Figura 4.32 – Execução do exploit ms08_067_netapi

Como explanado na seção 3.2.2, o Meterpreter não é simplesmente um payload, mas sim uma plataforma de exploração que apresenta um *shell* de comandos, o qual fornece ao invasor uma variedade de atividades possíveis de serem executadas no sistema explorado. As principais atividades realizadas com o Meterpreter foram:

- Migração de processo: Com a intenção de se camuflar durante a invasão, a sessão Meterpreter estabelecida muda o ID que descreve o seu processo para um ID de um processo comum e bastante utilizado no host alvo. No ataque realizado, o ID foi mudado para o ID 1744, que no caso em específico caracterizava o explorer.exe. A figura 4.33 ilustra esta ação.

Figura 4.33 – Migração de ID de processo da Sessão Meterpreter.

- Screenshot*:* Tal comando permite a captura da imagem de uma determinada região do host alvo, possibilitando ao atacante ter uma visão de arquivos e diretórios em uma dada área do computador invadido. A figura 4.34 ilustra um exemplo.

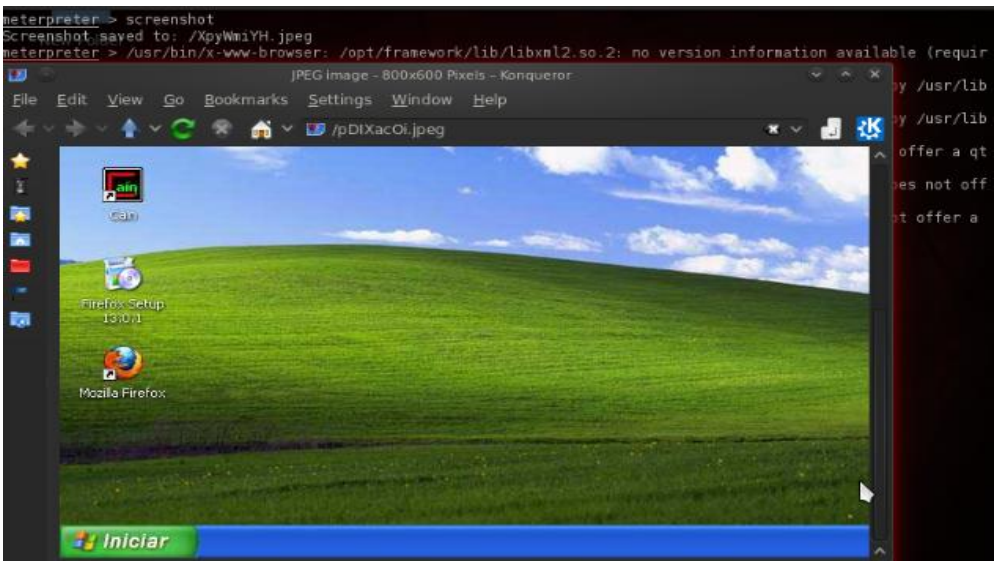


Figura 4.34– Visualização do desktop do host invadido.

- Shell: Tal comando permite acesso a diretórios do host invadido. A partir daí, é possível se promover criação, substituição e exclusão de arquivos, além de diversas outras atividades que podem ser impactantes e trazer sérias consequências ao *host* alvo. A figura 4.35 mostra um exemplo do que foi explicado.

Figura 4.35 – Acesso a diretórios do host alvo.

- Keyscan_Start/Keyscan_Dump*:* Estes dois comandos em conjunto permitem o *sniffing* de todos as palavras que são digitadas no host invadido e a posterior apresentação de tais dados na tela do computador do invasor. É uma técnica que pode ser usada, por exemplo, para captura de logins e senhas, como ilustrado na figura 4.36.

Figura 4.36 – Captura de login e senha com keyscan_start e keyscan_dump

4.3.4.3 – Ataque DOS

O ataque DOS realizado explora a vulnerabilidade listada na figura 4.30. A vulnerabilidade em questão refere-se ao serviço *Server Message Block*(SMB), que basicamente é um aplicativo a nível rede, que atua como um protocolo utilizado principalmente para o acesso aos arquivos compartilhados. Em suma, o problema reside na forma como o SMB trata certas mensagens de rede. O invasor pode então, explorar a vulnerabilidade enviando uma mensagem de rede especialmente criada a um computador que esteja executando SMB. O atacante que explorar essa vulnerabilidade com sucesso poderá

fazer com que o computador afetado pare de responder, concretizando-se assim a conclusão do DOS.

Para a realização do ataque, foi utilizado o exploit ms06_063_trans, o qual é um exploit compatível para a realização de ataque em cima da vulnerabilidade MS09-001. Tal exploit pode afetar sistemas configurados Windows XP com Service Pack 1 e 2. Por isso, o ataque foi promovido em cima do host 10.1.2.11. A figura 4.37 ilustra a execução do exploit, através do envio das mensagens SMB forjadas, bem como a imediata reinicialização da máquina atacada através de tela azul.

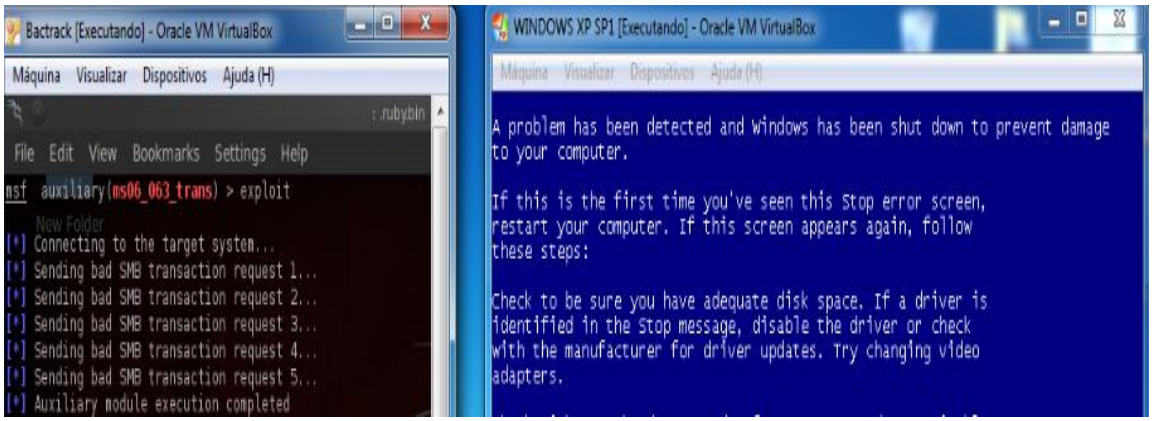


Figura 4.37 – Execução do exploit ms06_063_trans e DOS no host alvo.

4.3.4.4 – Ataque de injeção de DLL

Como descrito na seção 2.1.6, um ataque de injeção de DLL objetiva injetar um código malicioso em uma DLL, de modo que quando se invocar a DLL infectada, o código maléfico que possibilitará a invasão também será executado.

O ataque foi realizado em cima da vulnerabilidade descrita na figura 4.31. Em suma, tal vulnerabilidade está relacionada com uma restrição incorreta do caminho utilizado para carregar bibliotecas externas do cliente *Remote Desktop*. A partir disso, o atacante pode estabelecer no próprio *host* uma DLL maleciosa que ao ser carregada pela vítima, executa códigos arbitrários que possibilitam afetar o host alvo. Tal ataque atinge todas as versões de Service Pack para Windows XP.

Para a realização do ataque, foi o escolhido o exploit webdav_dll_hijacker. Em suma, o exploit funciona criando um arquivo em algum diretório do *host* invasor. Quando a vítima acessa este caminho, o ataque é realizado e o resultado final é a criação de uma sessão Meterpreter no *host* do atacante, o que possibilita a realização de diversas formas de ataque, conforme visto na seção 4.3.3.2. Tal ataque é comum em servidores de arquivos, onde

primeiramente o atacante invade o host servidor e a partir daí carrega o arquivo com a DLL maliciosa nele. Quando a vítima acessar tal arquvio, a DLL é carregada e o ataque é consumado.A figura 4.38 ilustra a execução do ataque. Analisando a figura, observa-se a criação de um arquivo chamado Senhas no diretório documents, hospedado na máquina 10.1.2.11. Vale ressaltar neste exploit, o uso do payload windows/meterpreter/reverse_tcp, o qual é utilizado para camuflar o ataque realizado.

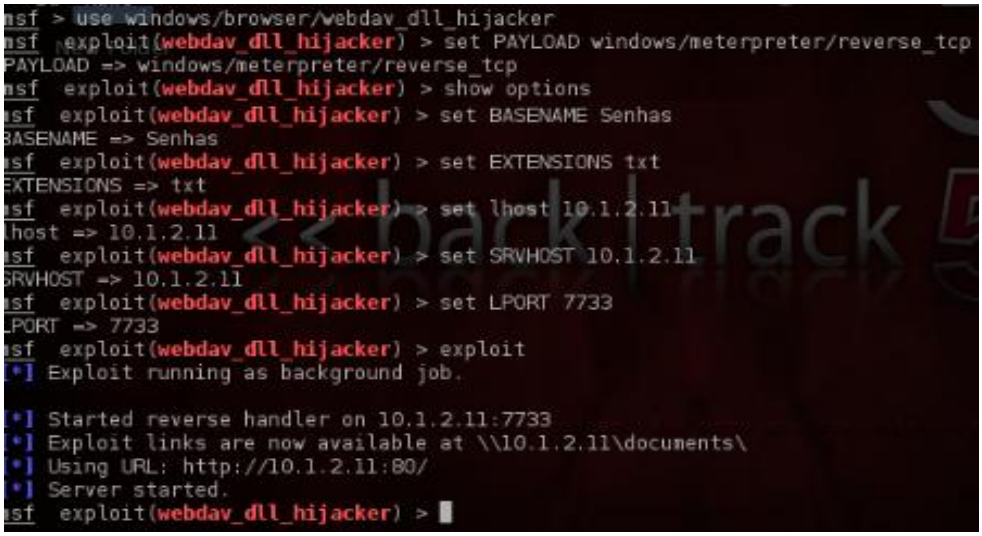


Figura 4.38 – Execução do exploit webdav_dll_hijacker

A figura 4.39 mostra o arquivo Senhas.txt criado pelo exploit e a vítima acessando-o. Quando a vítima abre o arquivo, uma DLL maliciosa é então carregada, executando consigo um código que traz como consequência a criação de uma sessão Meterpreter no *host* do invasor, como pode ser observado na figura 4.40.

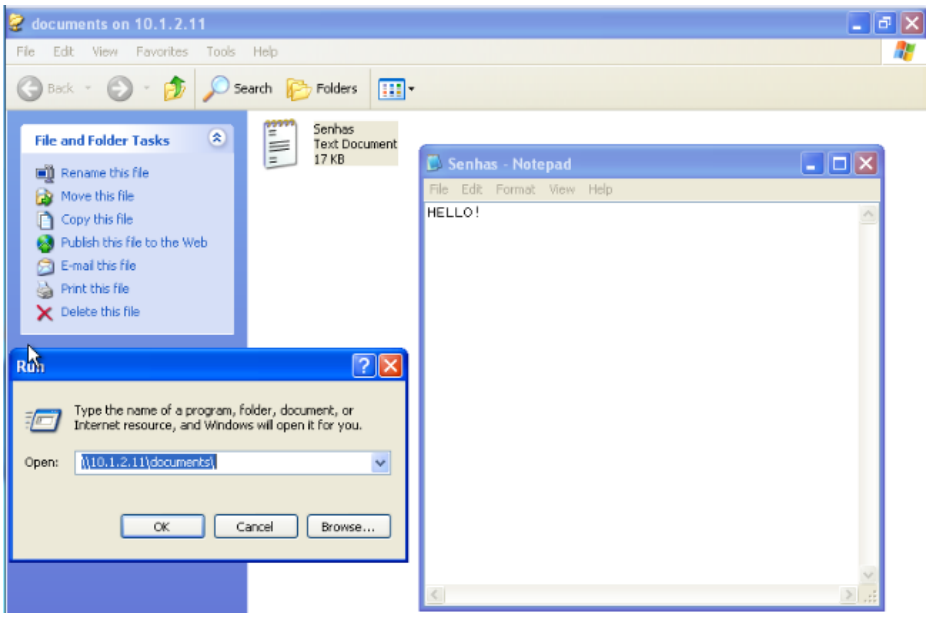


Figura 4.39 – A vítima acessa o arquivo Senhas.txt que contém a DLL maliciosa

Figura 4.40 – Criação da sessão Meterpreter após vítima acessar o documento Senhas.txt

4.3.5 – Ataque ao Windows 7

Seguindo a mesma linha de raciocínio durante a realização dos ataques aos hosts configurados com Sistema Windows XP, a seguir serão apresentados resultados referentes a descoberta de vulnerabilidades e posterior realização de ataques em hosts configurados com Sistema Operacional Windows 7.

4.3.5.1. – Scanner de vulnerabilidades

Mais uma vez, através da utilização do NMAP, foi possível se identificar informações de host ativos na rede configurados com Sistema Operacional Windows 7. Após a obtenção do IP de cada alvo em potencial, promoveu-se o scanner de vulnerabilidades nas máquinas de interesse. Assim como na varredura dos Sistemas com Windows XP, o scanner de vulnerabilidades utilizado foi o Nessus.

A figura 4.41 apresenta um resumo das principais vulnerabilidades encontradas ao se promover um scanner em um Sistema Operacional Windows 7 Professional. A figura 4.42 descreve com maiores detalhes a vulnerabilidade de potencial crítico descrita na figura 4.41.

Summary

Critical	High	Medium	Low	Info	Total
1	0	1	0	20	22

Details

Severity	Plugin Id	Name
Critical (10.0)	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)
Medium (5.0)	57608	SMB Signing Disabled
Info	10150	Windows NetBIOS / SMB Remote Host Information Disclosure
Info	10287	Traceroute Information
Info	10394	Microsoft Windows SMB Log In Possible
Info	10736	DCE Services Enumeration
Info	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure
Info	11011	Microsoft Windows SMB Service Detection
Info	11153	Service Detection (HELP Request)
Info	11219	Nessus SYN scanner
Info	11936	OS Identification

Figura 4.41 – Resumo de algumas vulnerabilidades encontradas no Windows 7 Professional

53514 (1) - MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)

Synopsis

Arbitrary code can be executed on the remote host through the installed Windows DNS client.

Description

A flaw in the way the installed Windows DNS client processes Link- local Multicast Name Resolution (LLMNR) queries can be exploited to execute arbitrary code in the context of the NetworkService account.
Note that Windows XP and 2003 do not support LLMNR and successful exploitation on those platforms requires local access and the ability to run a special application. On Windows Vista, 2008, 7, and 2008 R2, however, the issue can be exploited remotely.

Solution

Microsoft has released a set of patches for Windows XP, 2003, Vista, 2008, 7, and 2008 R2 :
http://technet.microsoft.com/en-us/security/bulletin/ms11-030

Risk Factor

Critical

CVSS Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

STIG Severity

I

References

BID	47242
CVE	CVE-2011-0657

Figura 4.42 – Descrição da Vulnerabilidade MS11-030

4.3.5.2. – Ataque DOS / Buffer Overflow

Tendo como base a vulnerabilidade descrita na figura 4.42, foi realizado um ataque duplo de DOS e *Buffer Overflow*. Em suma, a vulnerabilidade em questão trata de uma falha relacionada à maneira como o cliente do serviço Windows DNS manipula consultas de resolução de nomes. O invasor que explorar tal vulnerabilidade pode promover a execução remota de códigos arbitrários que podem vir a permitir, por exemplo, a instalação de programas maliciosos no host, controle parcial ou total de uma sessão criada, ou ainda exclusão de arquivos e informações.

Para a realização do ataque duplo, foi utilizado o exploit ms11_030_dnsapi. Tal módulo atua enviando uma consulta alterada de resolução de nome, que promove a execução de código remoto malicioso. Tal código leva a comprometimento de um determinado espaço

livre da memória por meio de sobrecarga da pilha, caracterizando-se assim um *buffer overflow*. Sem espaço livre na memória RAM, o host invadido fica pesado e então ingressa em um quadro de DOS em virtude da falta de espaço de memória livre para execução da atividade que o usuário requisita. Os efeitos do DOS podem ser diversos, desde uma simples negação de abertura de programa até o quadro mais grave de tela azul e reinício da máquina. Assim, o ataque duplo é finalizado.

A execução do exploit pode ser observada na figura na figura 4.43, onde observa-se o ataque praticado em cima do host de IP 10.1.15 na porta 5355, *default* do exploit e que se encontrava aberta no host atacado. Na figura 4.44, pode ser observada a elevação do uso da cpu para 100% pouco tempo depois da execução do exploit. Tal quadro é o suficiente para provocar DOS para qualquer novo serviço requisitado.

Figura 4.43 – Execução do exploit ms11_030_netapi

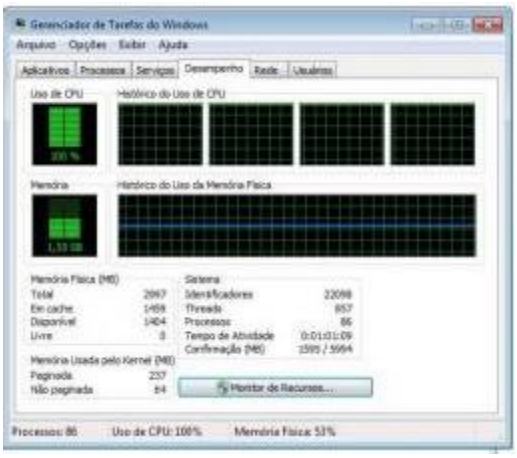


Figura 4.44 – Uso em 100% da CPU, o que pode provocar DOS

Outra forma de se obter um ataque do tipo DOS é explorando a vulnerabilidade de potencial médio SMB *Signing Disable* descrita na figura 4.41. A forma de ataque referida é praticada em cima de vulnerabilidades no serviço SMB.

4.3.5.3. – Ataque de falsificação de pacotes de consulta DNS

Ainda tendo em vista a vulnerabilidade expressa na figura 4.42, é possível realizar uma outra forma de ataque, porém de uma forma indireta. Ao provocar um DOS no cliente Microsoft DNS, o atacante pode assumir o seu lugar momentaneamente e a partir disso efetuar um ataque de DNS *Spoofing*. Como já explicado na seção 2.1.2, este ataque consiste na interceptação e substituição do tráfego legítimo por pacotes DNS forjados, conseguindo com isso o desvio do tráfego da vítima para um servidor fake, o qual pode conter um website falsificado, por exemplo. Tal tipo de caso será explorado a seguir.

Para a realização do ataque de DNS *Spoofing*, foi utilizada a ferramenta SET, descrita na seção 3.2.3. Dentre as várias formas de ataque que a ferramenta apresenta, foi escolhido um ataque do tipo *Website Attack Vectors*, sendo realizado por meio do sub ataque *Credentials Harvesting*, que objetiva a obtenção de credenciais (l*ogins* e senhas) dos usuários do sistema alvo. Para obter tal efeito, utilizou-se a opção *Site Clonner*, que promove a clonagem de *website*. Quando o usuário escrever *login* e senha, por exemplo, o invasor captura estas informações e logo em seguida, para não levantar nenhum tipo de suspeita ou alarme, redireciona-se as credenciais fornecidas pelo usuário para o website original. A figura 4.45 ilustra a realização dessa operação, onde foi escolhido o website www.orkut.com.br para ser clonado.

Figura 4.45- Clonagem do site www.orkut.com.br

Ao acessar o *website*, a vítima será redirecionada para a máquina do invasor, a qual contém um servidor *fake* com uma cópia exata da página original. Quando digitar as

informações de *login* e senha, estas serão capturadas e mostradas no host do atacante. A seguir, a vítima é redirecionada para o website original, não levantando suspeitas. A figura 4.46 ilustra o processo de captura das credenciais e a figura 4.47 mostra o website clonado. Nota-se que acessando o IP do invasor, 10.1.2.11, acessa-se a página inicial clonada do website original, comprovando o ato de *spoofing*, ao associar o endereço 10.1.2.11 ao nome www.orkut.com.br.

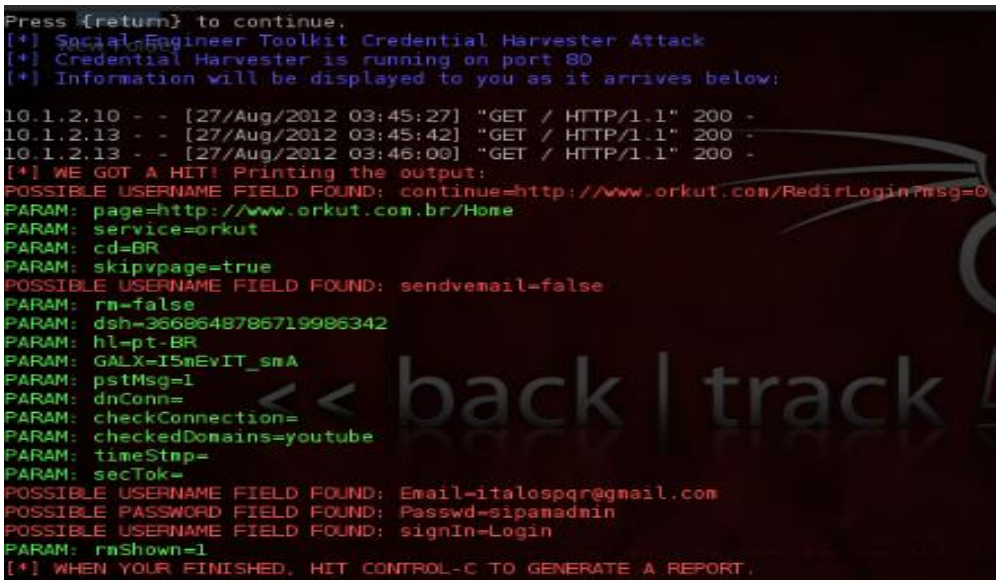


Figura 4.46- Captura de login e senha do website original

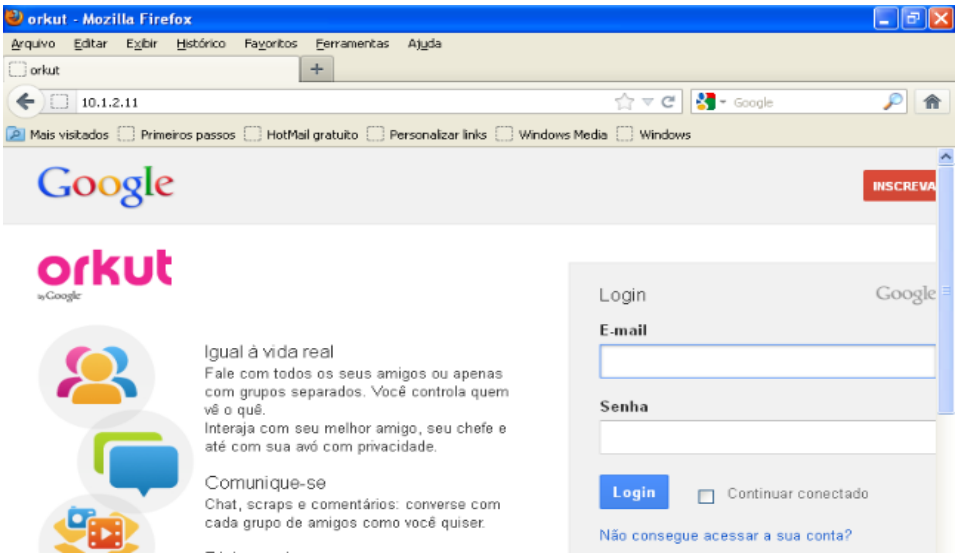


Figura 4.47- Ilustração da página clonada a partir do ip do invasor

4.3.5.4- Ataque combinado com Engenharia Social

Analisando-se a figura 4.41, percebe-se que as vulnerabilidades encontradas foram poucas, apresentando apenas 2 com algum potencial crítico. Em sistemas com níveis de segurança adequados, dificilmente encontram-se potenciais vulnerabilidades a serem exploradas. A forma encontrada então pelos atacantes de superar tais obstáculos é por meio da técnica de engenharia social. Como descrito em 2.1.9, engenharia social é um conjunto de técnicas e ferramentas que podem englobar negociações, psicologia, além de técnicas para enganar, objetivando a utilização do fator humano para burlar mecanismos de segurança de sistemas. Em suma, é um ataque que para se concretizar necessita que o usuário do *host* alvo realize uma ação em específico.

O ataque realizado nesta seção é uma combinação de um ataque de engenharia social, com um ataque de (semelhante ao descrito na seção 4.3.3.5) e com uma vulnerabilidade no *browser* Mozilla Firefox, afetando todas as versões de arquitetura Windows x86 e Linux x86. A vulnerabilidade em questão se refere a uma falha na instalação de *plug-ins* e atinge versões do Mozilla Firefox desatualizadas com os *patches* de segurança.

Para realização do ataque, foi utilizado o exploit firefox_xpi_bootstrapped_addon. A figura 4.48 ilustra a execução do exploit. Em poucas palavras, o exploit funciona relacionando um *host* a um *plug-in.* Caso deseje-se acessar tal *host*, é necessário instalar um *plug-in* malicioso, que ao ser instalado concederá acesso do host da vítima ao invasor.

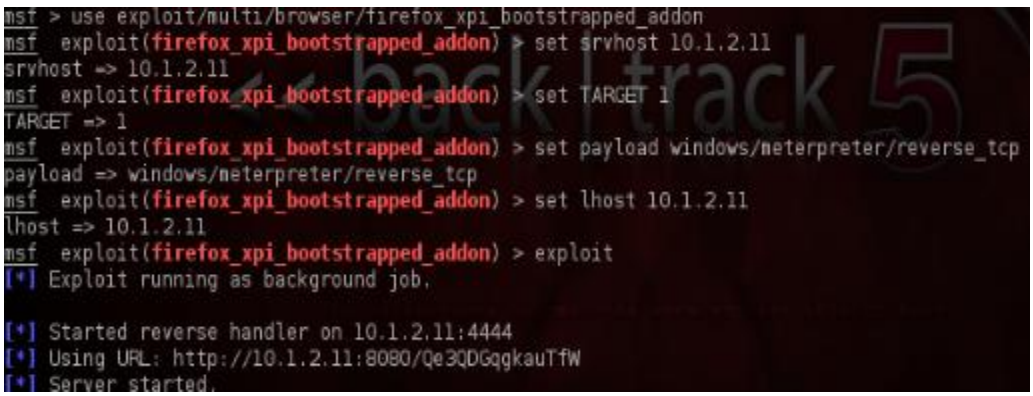


Figura 4.48- Execução do exploit firefox_xpi_bootstrapped_addon

Para eficácia do ataque, o host a ser acessado deve ser relacionado com algum nome de domínio ou website frequentemente utilizado pela vítima. Por tais motivos, é necessário antes de mais nada, realizar um ataque de *cache poisoning* que falsifique o cache DNS, relacionando um domínio com um endereço IP *fake*, o qual pertence ao atacante. Neste ataque

em específico, o domínio pjf.unb.br, originalmente associado ao *host* 10.1.2.2, passou a ser relacionado ao *host* 10.1.2.11, que pertence ao invasor.

A execução do ataque é ilustrada nas figuras 4.49 e 4.50. Ao tentar acessar o domínio pjf.unb.br, o acesso é negado, pedindo então que a vítima permita a ação. Ao permitir, é solicitado a vítima a instalação de um determinado *plug-in*. É ai que entra a parte de engenharia social. Uma vítima que não analisa direito a origem do plug-in, ou se ele realmente é necessário para o acesso, acaba sendo ludibriada e acabada instalando-o para obter acesso ao site desejado.

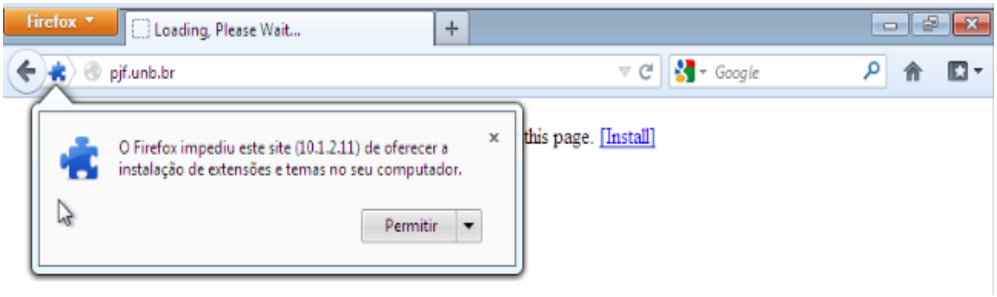


Figura 4.49- Negação de acesso ao domínio pjf.unb.br

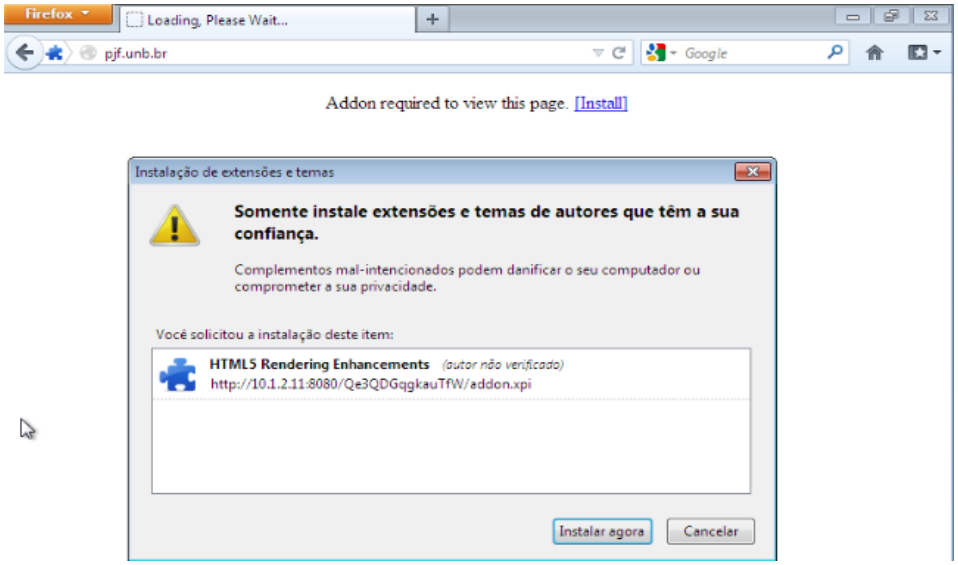


Figura 4.50- Plug-in solicitado para acessar o domínio pjf.unb.br

Após instalar o plug-in, o ataque é finalizado e é criada uma sessão Meterpreter na máquina do invasor. A partir daí, diversos outros tipos de ataque podem ser realizados. A figura 4.51 mostra a criação da sessão Meterpreter após instalação do *plug-in* realizada pela vítima.

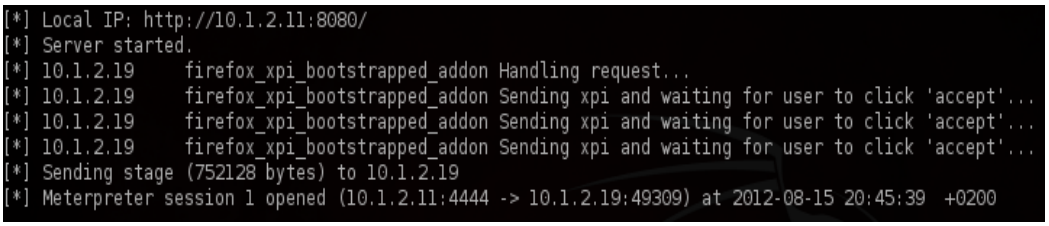


Figura 4.51- Criação da sessão Meterpreter após instalação do plug-in

4.4 MITIGAÇÃO DAS VULNERABILIDADES

Pode-se definir vulnerabilidade como uma falha em um sistema, que permite a um atacante usá-lo de uma forma não prevista pelo projetista (ANLEY, 2007). As vulnerabilidades podem basicamente ser resolvidas por instalações de *patches* de segurança (que corrigem as falhas), atualizações de sistemas, adoção de determinadas políticas e alteração de configuração da infraestrutura relacionada aos serviços afetados. A seguir serão dados exemplos práticos de tais formas de correção através da mitigação de vulnerabilidades nos hosts e sistemas operacionais explorados na fase de ataques.

4.4.1 Mitigação de Vulnerabilidades da rede sem fio

Para que o atacante conseguisse sucesso na intrusão ao cenário 1, foi necessário que o mesmo tivesse em mãos um grande dicionário de palavras e utilizasse ferramentas do BackTrack ,voltadas para a quebra de encriptações mais seguras como WPA/WPA 2 PSK.

Uma possível solução para a falha de segurança em questão seria a configuração de um servidor RADIUS (*Remote Authentication Dial In User Service*) e a integração da rede sem fio com o mesmo.

Em todo sistema de autenticação, a grande preocupação com relação à segurança é a passagem da senha do usuário pela rede. Caso essa passe como um dado comum, sem nenhuma proteção, qualquer escuta ao trafego da rede permite descobri-la. A fim de dar confiabilidade à esses dados, é recomendável a configuração do serviço RADIUS.

O RADIUS é um sistema utilizado para prover uma autenticação centralizada em redes *dial-up,* VPNs e redes sem-fio. Através do RADIUS as senhas passam a ser criptografadas entre o servidor e os cliente RADIUS. Para isso, uma senha é trocada entre servidor e cliente, manualmente, para que eles possam conversar utilizando chaves simétricas. Suas funções básicas são: autenticar os usuários ou dispositivos antes de conceder-lhes acesso a uma rede; autorizar os usuários ou dispositivos para determinados serviços de rede;

verificação de seções abertas, tais como tempo de uso, continuidade ou fim de seção, dentre outras informações importantes sobre as seções autenticadas. Com o RADIUS instalado pode-se centralizar os mecanismos de autenticações tais como switches, roteadores, pontos de acesso, outros servidores Windows, Linux, etc. Em suma, o RADIUS é um protocolo que foi idealizado para centralizar as atividades de autenticação, autorização e contabilização, visto que o crescente número de sistemas independentes inviabiliza a administração descentralizada (SILVA,2003).

Os pontos de acesso sem fio devem exigir autenticação e autorização do nó sem fio antes que os dados possam ser enviados e recebidos da rede que está conectada ao ponto de acesso sem fio. Para fornecer sua própria autenticação e autorização, cada ponto de acesso sem fio deve exigir um banco de dados de conta de usuário com credenciais de autenticação de cada usuário e um conjunto de regras pelas quais a autorização é concedida. Como isso é difícil de gerenciar, alguns pontos de acesso sem fio são clientes RADIUS que usam o protocolo RADIUS padrão da indústria para enviar mensagens sobre contabilização e solicitação de conexão para um servidor RADIUS central. O servidor RADIUS tem acesso a um banco de dados de conta de usuário e a um conjunto de regras para conceder autorização. O servidor RADIUS processa a solicitação de conexão do ponto de acesso sem fio e aceita ou rejeita a solicitação de conexão (MICROSOFT, 2005d).

A solução proposta cobriria a falha de segurança, pois para que o atacante consiga ingressar à rede, é necessário que o mesmo possua um usuário e senha individuais, pré-configurados no banco de dados do servidor RADIUS.

Segue a proposta da solução, de forma mais detalhada:

Determinar se há duas SSID(Service Set Identifier) com autenticação diferentes, uma para usuários internos, servidores da empresa e um SSID que conceda acesso às pessoas internas que utilizem-se de dispositivos não homologados pela empresa, como *smartphones, tablets* e pessoas externas à empresa. Para acesso à SSID servidores , uma condição seria exigida aos usuários internos da empresa, utilizarem-se de dispositivos móveis, como notebooks, pré homologados e cedidos pela empresa. Quando o acesso for requisitado, seria necessário de antemão, o usuário e senha individuais, aos quais os funcionários utilizam-se normalmente em seu dia-a-dia para acesso à aos seus serviços na empresa, como *e-mail* corporativo, portal intranet, etc. Com esse usuário e um computador pré-homologado, seria possível o acesso à rede interna com todos os privilégios já comuns ao usuário na rede cabeada, através da integração do *Active Directory* (implementação da Microsoft em ambientes Windows, que consiste em um serviço de diretório no protocolo LDAP

(*Lightweight Directory Access Protocol*) que armazena informações sobre objetos e disponibiliza essas informações a usuários e administradores desta rede) e RADIUS.

Para usuários internos que não utilizam de dispositivos móveis, pré-homologados, como celulares, *tablets*, etc e ainda usuários externos visitantes. O acesso seria liberado pela segunda SSID, seria uma, SSID para dispositivos móveis visitantes, aos quais a autenticação ao ponto de acesso sem fio poderia dar-se em ABERTO diferentemente da WPA, configurada no cenário 1 inicial. Esta condição iria suprir inicialmente, apenas a autenticação do dispositivo móvel ao ponto de acesso sem fio, o que não garante ainda acesso à internet. Com o usuário autenticado ao ponto de acesso, o cliente requisitaria um usuário temporário ou permanente à um administrador de rede. O administrador então configuraria um usuário conforme as necessidades do visitante, se é um servidor da empresa que utiliza-se de acesso a celular, *tabl*et poderia ser criado um usuário permanente para acesso, se é apenas um visitante temporário, o usuário para o mesmo teria validade em quantidade de dias ou horas às quais o usuário teria acesso à rede, etc..

Por essa solução sugerida, os usuários internos, que se autenticassem ao SSID da rede interna, teriam o acesso conforme aos seus perfis de acesso configurados no *Active Directory*, já os usuários com acesso ao SSID dos visitantes teriam seus acessos configurados conforme ao perfil de acesso mais limitado, que seria igual para todos os usuários visitantes.

4.4.2. Mitigação de Vulnerabilidades do website

A principal forma de mitigação de vulnerabilidades para o ataque desenvolvido na seção 4.2.2 seria a utilização de *load balancing* (balanceadores de carga), que como dito anteriormente é um processo de distribuição de solicitações de serviços de um servidor para um grupo de servidores, promovendo aumento de escalabilidade, desempenho, recuperação de falhas e melhor utilização da banda disponível, evitando desta forma ataques que exploram sobrecarga. A solução é muito simples, porém boa parte dos *websites* da atualidade não fazem uso de balanceadores de carga.

Para o Servidor Web Apache atacado, a Apache Foundation emitiu uma série de dicas para solução do problema. Em http://httpd.apache.org/docs/trunk/misc/security_tips.html#dos é possível ter acesso a tais dicas, que consistem basicamente na instalação de módulos de correção.

Em um nível de maior profundidade, poderiam ser adicionadas regras ao *firewall* que bloqueassem flood de conexões na porta 80 onde se encontra o Website. Como o *firewall* do laboratório de testes é o Iptables, linhas de configuração como: iptables -I INPUT -p tcp --

dport 80 -m state --state NEW -m recent –set e iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 60 --hitcount 6 -j DROP evitariam o referido *flood* de conexões na porta 80.

Por fim, o estabelecimento de redundância no Servidor Web (ou seja, um outro servidor pronto para assumir caso o servidor original se torne inoperante) pode ser uma solução provisória adotada para evitar a queda do serviço caso um ataque seja realizado.

4.4.3- Mitigação de Vulnerabilidades do firewall/router

O host 10.1.2.246, tem a função de *firewall/router* da rede. É o ponto central da rede, onde todas as informações necessariamente passam em qualquer comunicação. Para exercício de tal função, o mesmo deve estar sempre que possível atualizado contra possíveis falhas de sistema. Recomenda-se ao administrador de rede estar sempre atento às vulnerabilidades em pacotes instalados no sistema, brechas através de servidores de bancos de dados tais como MYSQL, ou ainda em servidores WEB rodando PHP, dentre outros.

Deve-se definir a política a ser seguida pelo *firewall*, tais como: bloquear todas as conexões e liberar aos poucos, de acordo com a necessidade, ou liberar todas elas e fechar conforme o necessário. Outro serviço essencial que deve ser definido é o filtro de conteúdo, pelo qual, são bloqueados acessos a sites, conforme a política de acesso definida pela empresa. A seguir, serão apresentados detalhes sobre configurações realizadas no *firewall* Iptables e no filtro de conteúdo Squid, as quais possibilitarão a mitigação de vulnerabilidades no *firewall* interno, bem como fornecerão um maior nível de defesas aos hosts que estão por trás do *firewall*.

4.4.3.1- Configuração do firewall

O *firewall* foi configurado de forma a recusar todas as conexões as quais tenham seu IP como destino final, com exceção de serviços nas portas 22, 3128 que são as portas do SSH e Proxy Squid respectivamente. Ao *firewall* ainda foi autorizado retransmitir todos os pacotes provindos da rede interna 10.1.2.0/24, visto que o mesmo é o roteador da rede. Configurou-se ainda o compartilhamento da conexão de internet, através da porta eth0. Ativou-se alguns filtros do sistema quem fazem a proteção contra ataques do tipo *Spoofing.* Desativou-se o suporte a ping *broadcasts*, recurso que tem poucos usos legítimos e ainda pode ser usado de forma que outros servidores participem involuntariamente de ataques DOS. No *kernel*, ativou-se o serviço tcp_syncookies que protege o sistema contra ataques do tipo SYN *Flood.*

As portas DNS e NTP são redirecionadas à porta 3128 do Squid. O protocolo ICMP foi liberado para uso dentro da rede interna, mas o *firewall* foi configurado para responder aos pings em intervalos de 1 segundo, evitando-se o ataque DOS conhecido como pings da morte

Pacotes inválidos, aqueles enviados sem serem precedidos por pacotes SYN e pela abertura de conexão, são rejeitados e são armazenados nos logs. O *firewall* sempre responderá aos pacotes nas interfaces aos quais eles foram originados, o que previne ataques que se aproveitam de conexões já iniciadas na interface *loopback.* Todas as conexões direcionadas ao *firewall* que não tenham sido permitidas anteriormente são rejeitadas.

4.4.3.2- Configuração do filtro de conteúdo

O Proxy Squid foi configurado de forma a ficar ativo somente para a rede interna. Através de acls SSL_Ports, e Safe_ports limitou-se as portas que podem ser utilizadas pela rede local, sendo estas portas essenciais indicadas pela documentação do Squid, tais como: 443, 80, 21, 1025-65535, que respectivamente são as portas dos serviços de certificação digital SSL, HTTP, FTP e portas altas, dentre outras não citadas. É importante destacar ainda que o filtro de conteúdo foi configurado de forma a bloquear o acesso à palavras chave tais como exploit, payloads, hacker, hacking, metasploit, além de outras palavras chaves que podem ser utilizadas por um possível atacante interno que esteja pesquisando por formas de executar um possível ataque.

Foram bloqueados ainda acesso e possíveis downloads de arquivos do tipo .exe, .rar, .zip, .bas, .com, .cpl , .dll, .hta, .ink, .reg, .src, .shs, .sys, .vb, .wsf, .wsh, .divx, .mp2 , .mp3, .mp4, .vcd , .vob, .wmv, .xvid, .ac3, .aiff, .au, .cda, .mid, .ra, .ram, .wav, .wma, .vox, .ogg, .bat, .piff, .vbs, .asf, .avi, .mov, .mpg e .mpeg aos quais não são essenciais ao serviço, o que já pode evitar uma série de ataques do tipo Engenharia Social, por meio do acesso a páginas falsas e *downloads* de arquivos maliciosos.

4.4.4- Mitigação de Vulnerabilidades do controlador de domínio

Uma vez detectadas falhas, o projetista, neste caso a Microsoft, lança em seu site boletins de ocorrência, avisando sobre as mesmas. Em caso de soluções, disponibiliza para download os patches de segurança que possibilitam a mitigação das falhas para todas as versões dos seus sistemas operacionais. Tais boletins de segurança podem ser acessado a partir do website http://technet.microsoft.com/en-us/security/bulletin.

Para sanar as vulnerabilidades descobertas no Windows Server 2003,os boletins de ocorrência específicos contendo os patches de segurança para a correção das falhas, podem

ser acessados através de seus números identificadores: MS12-020, MS11-035, ou ainda seus identificadores BID e CVE, como pôde ser observado nos relatórios analisados anteriormente.

Em relação aos ataques ARP *Spoofing* realizados, para mitigação das vulnerabilidades, pode-se bloquear portas de serviços que não realizam autenticação a cada nova conexão. Adicionalmente, pode-se fazer basicamente o uso de três métodos.

O primeiro método consiste na habilitação da opção MAC *binding* nos *switchs*. Esta é uma opção encontrada em alguns *switchs* e faz com que os endereços MAC associados com uma determinada porta não sejam alterados depois de configurados. Tais mudanças podem ser realizadas apenas pelo administrador da rede.

O segundo método consiste no uso de caminhos estáticos. Basicamente, se realizaria uma alteração no cache ARP de forma a se ter apenas entradas estáticas. Desta forma, respostas ARP falsas seriam ignoradas. Tal abordagem porém, torna-se viável apenas em redes de pequeno porte, essencialmente redes caseiras.

O terceiro método consiste na detecção de ARP *Spoofing*. Para isto utiliza-se a ferramenta Arpwatch, nativa no Backtrack. Tal ferramenta consiste na monitoração da atividade ethernet, mantendo uma base de dados dos pareamentos MAC/IP e reportando alterações via e-mail. Desta forma, o Arpwatch mantem o administrador informado sempre que uma nova máquina adquire um endereço da rede, notificando os respectivos endereços IP e MAC da nova máquina. Além disso, informa se um dado endereço MAC mudou de IP ou ainda se há alterações na configuração da rede, como por exemplo mudanças maliciosas de IP do gateway ou de algum servidor.

Em relação ao ataque ARP *Spoofing* realizado por meio do software Cain & Abel, a melhor forma de mitigação é adoção de um mecanismo de bloqueio do referido programa em qualquer host da rede. Uma forma eficiente de se realizar tal tarefa é, por exemplo, a criação de chaves no registro para a instalação de qualquer software.

Em relação ao ataque *Cache Poisoning*, pode-se fazer uso do DNSSEC (*Domain Name System Security Extensions*), o qual é uma extensão de segurança para o DNS que tem como foco principal oferecer autenticidade e integridade nas consultas DNS. O DNSSEC faz uso de criptografias de chaves públicas e assinaturas, evitando desta forma, ataques que exploram a integridade e autenticidade das transações DNS, como é o caso do *Cache Poisoning.* Mais informações sobre o DNSSEC podem ser obtidas em (Silva, 2009).

Como medidas de segurança adicionais, pode-se proceder com a instalação e configuração do RADIUS, que como descrito na seção 4.4.1, faz uma centralização dos mecanismos de autenticação.

4.4.5- Mitigação de Vulnerabilidades dos Windows XP e 7

Assim como na mitigação das vulnerabilidades do Windows Server 2003, as vulnerabilidades exploradas na fase de ataques realizada no Windows XP e no Windows 7 são solucionadas através da instalação dos patches de correção de segurança, disponíveis nos boletins de segurança apontados nos relatórios das vulnerabilidades.

A análise nesta seção será voltada para o estabelecimento de medidas que garantam o estabelecimento de uma sólida defesa proativa nos hosts dos usuários. A análise pode ser feita sob dois pontos de vistas: um interno e outro externo.

O ponto de vista externo consiste na adoção de medidas de segurança centralizadas no *firewall/router* 10.1.2.246 e no Windows Server 2003 10.1.2.2, adotando basicamente o conjunto de passos exposto na análise de cada conjunto de vulnerabilidades dos respectivos hosts. Como medida adicional poderia se propor a utilização de um sistema automatizado de segurança voltado para detecção de tentativas de intrusão e coleta de dados. Para tal proposição, faz-se necessário a utilização de Sistema de Detecção de Intrusão (IDS).

O uso de IDS permite prevenção a certos tipos de ataques, que envolvem invasão ou coleta de informações, tais como *Sniffers*, *Port Scanners*, *ARP Spoofing*, dentre outros. A prevenção a tais tipos de ataques seria realizada tendo como pressuposto a monitoração constante do tráfego, visando a detecção de anormalidades, como excesso de tráfego vindo de um único host, vários hosts enviando pacotes ICMP para um único host, excesso de pacotes broadcasts enviados pelo mesmo host, etc. Fazer essa monitoração manualmente exigiria demasiada atenção e esforço humano. Isto poderia ser evitado com uma robusta configuração de um IDS, o que agregaria ainda mais proteção à rede em um nível global, juntamente com as soluções propostas anteriormente. Alguns IDSs encontram-se disponíveis em open source, tais como o Snort e o Honeypot.

A análise do ponto de vista interno pode ser resumida nos seguintes pontos: análise e correção de falhas internas, atualizações automáticas de segurança, proteção por meio de antivírus contra *malwares* e *spywares*, criação e organização de políticas de grupos, configuração do protocolo SSL para acesso à internet e capacitação dos usuários.

É dever do administrador da rede estabelecer práticas que forneçam aos host usuários correção de falhas e automatização de atualizações de segurança. Atitudes simples, como o uso de licenças originais (as quais fornecem permissão aos serviços completos dos Sistemas Operacionais tais como atualizações freqüentes contra falhas de seguranças e melhoras no desempenho dos sistemas), habilitação do Firewall Windows, seleção de atualizações

automáticas de segurança, além da frequente observação de logs que permitam identificar e corrigir falhas rapidamente, são algumas das atitudes imprescindíveis para o estabelecimento de uma consiste defesa interna.

A utilização de um bom antivírus, sempre atualizado com as mais novas soluções contra os diferentes tipos de ataque que surgem a cada dia, representa também um importante ponto de defesa, principalmente contra *malwares* e *spywares* (scripts maliciosos que objetivam obtenção de informações corriqueiras ou de cunho confidencial da vítima). Em um nível global, poderia ser configurada a contratação/instalação de um servidor de antivírus, o qual seria sempre atualizado com os últimos níveis de segurança e gerenciaria de forma centralizada, a proteção de todos os host e servidores da empresa, que possuiriam o software do lado cliente do antivírus instalado, possibilitando assim a referida integração.

Outra importante forma de defesa consiste na criação e organização de políticas de grupo. O *Active Directory* presente nos sistemas Windows Server permite a estruturação de políticas de grupo, de forma que sejam liberados, estritamente, somente as ferramentas necessárias aos desempenhos dos trabalhos de cada área da empresa. Desta forma, deve restringir-se tanto quanto possível o acesso de usuários comuns, não administradores de rede, a certos serviços. Desta forma, pode-se promover, por exemplo, o bloqueio de acesso ao *prompt* de comando do Windows, bloqueio à utilização da ferramenta de acesso remoto *Remote Desktop*, negação de acesso à pastas em servidores remotos através da barra de endereços do Windows Explorer, etc. Utilizando tais artefatos, diversas formas de ataque interno podem ser evitadas.

Em relação à confiabilidade de dados trafegados na internet, pode-se fazer uso da configuração de certificados digitais, SSL (Secure Sockets Layer), que são tecnologias de segurança comumente utilizadas para codificar os dados trafegados entre o computador do usuário e um Website, da empresa por exemplo. O protocolo SSL, através de um processo de criptografia dos dados, previne que os dados trafegados possam ser capturados, ou mesmo alterados no seu curso entre o browser do usuário e o site com o qual ele está se relacionando, garantindo desta forma proteção de informações sigilosas como logins, senhas ou outros dados pessoais. Além disso, serviços baseados em FTP também podem ser integrado ao protocolo SSL de forma a gerar transferências de arquivos mais seguras.

Por fim, para se evitar ataques baseados em ludibriar os usuários, tais como *Mail Phishing* e Engenharia Social, recomenda-se a capacitação de todos os usuários da empresa, através de mini cursos focados em boas práticas de segurança, afim de se evitar as referidas

formas de ataque, as quais com o “auxílio” dos usuários podem ser realizadas mesmo com todo o sistema de segurança da rede em funcionamento.

Em suma, tendo como base o conjunto de passos e medidas apresentados no desenvolvimento do trabalho, é possível se estabelecer níveis de defesa adequados para implementação de sistemas de segurança.

5- CONCLUSÃO

O principal objetivo do projeto consistia em descrever as etapas necessárias para a realização de um teste de vulnerabilidades e como tais etapas seriam direcionadas para a descoberta, análise e mitigação de vulnerabilidades e eventuais riscos que os sistemas analisados pudessem vir a estar submetidos.

Primeiramente, foi montado o laboratório de testes. Tal laboratório foi construído por meio de virtualização devido a quantidade limitada de recursos que se dispunha para a realização dos testes. Mesmo com tal empecilho, foi possível simular de forma satisfatória uma rede coorporativa e considerável parte da infraestrutura que lhe é pertinente.

O trabalho foi focado nas etapas de identificação das vulnerabilidades, realização de ataques e mitigação das falhas.

A etapa da identificação de vulnerabilidades foi realizada com o auxílio das ferramentas NMAP e dos scanners de vulnerabilidades OpenVAS e Nessus. Utilizando tais ferramentas, foi possível se obter vários relatórios que apresentavam em seu escopo informações sobre detalhes técnicos das vulnerabilidades, bem como formas de se realizar ataques e de se solucionar as falhas. Além disso, os relatórios apresentavam ligações com bancos de dados de vulnerabilidades externos, como o CVE e o MSB, que apresentavam mais detalhes das falhas apontadas. Tendo em mãos os referidos relatórios, foi possível se fazer um estudo aprofundado das vulnerabilidades encontradas, estabelecendo-se desta forma paralelos com problemas frequentes de segurança que assolam diversos sistemas de segurança.

Durante a realização da fase de ataques, conseguiu-se analisar todo o passo a passo para o estabelecimento de diferentes formas de ataque, bem como foi possível obter uma visão local e global do potencial crítico de cada ataque realizado. Utilizando ferramentas nativas do BackTrack , como o Metasploit Framework, foi possível também a observação de diferentes softwares bastantes úteis para simulação de ataques e exploração de diversas situações. Para o estabelecimento de uma defesa sólida e robusta é preciso entender como o invasor age e estudar as ferramentas frequentes que utilizam para promover os ataques. Tais objetivos foram alcançados durante a execução da fase de ataques.

A etapa de mitigação das vulnerabilidades possibilitou a visualização das principais ações a serem adotadas para sanar diferentes níveis de falhas, tanto de potencial baixo, quanto de potencial alto. Além disso, forneceu como lição o quanto é importante o estabelecimento

de políticas de segurança pré-determinadas para a construção de um sistema de segurança proativo.

De uma maneira geral, mesmo com limitações de fatores materiais, técnicos e operacionais, os principais objetivos pré-estabelecidos para o projeto foram alcançados. Foi possível se observar a importância dos testes de vulnerabilidades para o estabelecimento de melhores níveis de segurança, e como se utilizados de forma correta podem atuar como verdadeiras ferramentas de auditoria de segurança da informação.

Ainda tendo em vista a temática dos testes de vulnerabilidades, trabalhos futuros poderiam ser realizados de forma a serem centrados em determinadas fases de realização dos testes. Desta forma, poderiam ser desenvolvidos trabalhos voltados, por exemplo, para: análise de *scanners* de vulnerabilidades e estudos aprofundados sobre as principais vulnerabilidades encontradas nos sistemas atuais; desenvolvimento de exploits e *scripts* que simulem novas formas de ataque, fornecendo desta maneira o desenvolvimento de eventuais soluções; desenvolvimento de novos *scripts* para mitigação de vulnerabilidades de determinados ataques; estudos de políticas e medidas para se alcançar uma elevada otimização de um determinado sistema de segurança.

REFERÊNCIAS BIBLIOGRÁFICAS

ALI, Shakeel; HERIYANTO, Tedi. BackTrack 4: Assuring Security by Penetration Testing. Birmingham: Packt, 2011. 371p.

CASAGRANDE, Rogério Antônio. Técnicas de Detecção de Sniffers UFRGS, Porto Alegre, 2003. Disponível em: < http://hdl.handle.net/10183/3423>. Acesso em: 30 Agosto, 2012.

FOSTER, J. C., MAYNOR, D. and COLIN. Metasploit Toolkit for Penetration Testing, Exploit Development,and Vulnerability Research. Syngress , 2007. 272p.

GREGG, M., "Certified Ethical Hacker – Study Guide". Sybex, 2007.

HURLEY,Chris.Penetration Tester's Open Source Toolkit,Volume 2. Syngress 2007.568p.

KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet – Uma Nova Abordagem. 3 ed. Rio de Janeiro: Pearson Addison Wesley, 2006.

MALEBRA, César. Vulnerabilidades e Exploits: técnicas, detecção e prevenção. Disponível em: <http://hdl.handle.net/10183/26337>. Acesso em: 20 Junho, 2012.

MELO, Laerte P.; AMARAL, Dino M.; SAKAKIBARA, Flavio; ALMEIDA, André R.; Sousa Jr., Rafael T.; Nascimento, Anderson. Análise de Malware: Investigação de Códigos Maliciosos Através de uma Abordagem Prática. Disponível em: <http://tinyurl.com/93u75ut>. Acesso em: 20 de Maio, 2012.

NIST Special Publication 800-42: Guidelines on Network Security Testing. (Wack, Miles, Souppaya, 2003)

NIST Special Publication 800-115: Technical Guide to Information Security Testing and Assessment (Scarfone, Souppaya, Cody, Orebaugh,2008)

NMAP Networking Scanning. Disponível em: http://nmap.org/man/pt_BR/index.html

SILVA, L. A. F. d. Radius em Redes sem Fio UFRJ, Rio de Janeiro, 2003.

SILVA, Sílvio Lucas da. Segurança em DNS: Investigando o DNSSEC através de experimento prático. - 69p. Monografia (Especialização em Segurança da Informação) – Faculdade de Tecnologia IBRATEC de João Pessoa.

SOUZA, Fabiano. Autenticação de usuários no Active Directory utilizando RADIUS através do serviço de autenticação da internet. Joinville: SOCIESC, 2007/2.

TANENBAUM, Andrew. Redes de Computadores. 4. ed. Rio de Janeiro: Campus Elsevier, 2003.