

Universidade Paulista - UNIP

Gabriel Balsante Marquesini

HACKER ÉTICO: A IMPORTÂNCIA DA SEGURANÇA DA INFORMAÇÃO

**Limeira
2020**

Universidade Paulista - UNIP

Gabriel Balsante Marquesini

HACKER ÉTICO: A IMPORTÂNCIA DA SEGURANÇA DA INFORMAÇÃO

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em ciência da computação sob a orientação do professor Me. Sérgio Eduardo Nunes.

**Limeira
2020**

Gabriel Balsante Marquesini

Hacker Ético: a Importância da Segurança da Informação

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em ciência da Computação sob a orientação do professor Me. Sérgio Eduardo Nunes.

Aprovada em 26 de novembro de 2020.

BANCA EXAMINADORA

RESUMO

As pessoas e empresas sofrem muitos ataques devido a falhas de segurança, as quais não tem a mínima noção de como corrigir. Então o hacker ético veio para descobrir onde estão as vulnerabilidades de cada caso e mostrar meios de corrigir, dependendo de sua necessidade. A profissão vem crescendo de alguns anos para cá, trazendo, assim, cada vez mais profissionais qualificados. O roubo de dados tem tido grande atenção nos últimos anos, principalmente em grandes empresas. Há vários tipos de ataques que um pentester pode realizar depois de feita a análise de caso em que este irá adquirir informações já entregues pelo contratante ou não. Para evitarmos estes ataques, tanto de hackers quanto de crackers, precisamos manter o máximo de critérios possíveis. As vulnerabilidades hoje se encontram em softwares e hardwares, desde os mais simples até os mais robustos, sendo possível que haja um grande dano dependendo das intenções do atacante e do preparo do alvo para estas eventualidades, por isso é tão importante a manutenção e atualização dos firewalls e outras ferramentas. Devido à facilidade de uso das ferramentas que veio ao passar dos anos, a quantia de ataques tem crescido muito, antes precisava-se de um vasto conhecimento em linguagens específicas, hoje este requisito é quase nulo.

Palavra-Chave: Hacker; pentester; ataques; análise; vulnerabilidades;

ABSTRACT

People and companies suffer many attacks due to security flaws, which they have no idea on how to correct it. So, the ethical hacker came to find out where the vulnerabilities are in each case and show ways to fix it, depending on their need. The profession has been growing for some years, thus bringing more and more qualified professionals. Data theft has received a lot of attention in recent years, especially in large companies. There are several types of attacks that a pentester can perform after analyzing the case in which he will acquire information already delivered by the contractor or not. To avoid these attacks both by hackers and crackers, we need to maintain as many criteria as possible. The vulnerabilities today are found out in software and hardware, from the simplest to the most robust, with the possibility of a great deal of damage depending on the intentions of the attacker and the preparation of the target for these eventualities. It is important to maintain and update firewalls and other tools. Due to the ease of use of the tools that came over the years, the amount of attacks has grown a lot, A vast knowledge in specific languages was needed before, today this requirement is almost nil.

Key Words: Hacker; Pentester; Attacks; Analyze; Vulnerabilities;

LISTA DE QUADROS

Quadro 1 - Ambiente e informações críticas que invasores podem identificar .	30
Quadro 2 - Recursos avançados de escaneamento NMAP	39
Quadro 3 - Por que não estamos seguros	68
Quadro 4 - Vulnerabilidades e soluções.....	72

LISTA DE FIGURAS

Figura 1 - ReadNotify (captação de roda de e-mail enviado)	33
Figura 2 - Grabify (Redirecionamento de link para captura de informações) ...	34
Figura 3 - Execução do Dnsenum	36
Figura 4 - Escaneamento com o software Maltego	37
Figura 5 – Execução do nmap.....	38
Figura 6 - Nmap em rede local	42
Figura 7 - Nmap em uma máquina virtual	43
Figura 8 - Captura de banner de máquina virtual	44
Figura 9 - Captura de banner com fingerprint em máquina virtual	45
Figura 10 - Serviços rodando por trás das portas descobertas	45
Figura 11 - Configuração do ProxyChains	48
Figura 12 - Hardwares e senhas padrões	55
Figura 13 - Hardwares e senhas padrões	56
Figura 14 - Hardwares e senhas padrões	57
Figura 15 - Hardwares e senhas padrões	58
Figura 16 - ARP Poisoning	61
Figura 17 - Entradas do tipo dinâmico e estático	75

LISTA DE ABREVIATURAS

ACLs	Acess Control Lists ou Lista de controle de acesso
AES	Advanced Encryption Standard ou Padrão de criptografia avançado
AP	Acess Point ou Ponto de Acesso
ARP	Address Resolution Protocol ou Protocolo de Resolução de Endereços
ARPA	Advanced Research Projects Agency ou Agência de Projetos de Pesquisa Avançada
ARPAnet	Advanced Research Projects Agency Network ou Rede de Agências para Projetos de Pesquisas Avançadas
ArpOn	Arp Handler Inspection ou Inspeção manipuladora Arp
ASCII	American Standard Code for Information Interchange ou Código Padrão Americano Para Intercâmbio de Informações
ASP	Active Server Pages
BSD	Berkeley Software Distribution ou Distribuição de Software Berkeley
CMD	Command ou Comando
CSRF	Cross-site Request Forgery ou Falsificação de Solicitação Entre Sites
DAI	Dynamic Arp Inspection ou Inspeção Arp Dinâmica
DCA	Defense Communications Agency ou Agência de Comunicações de Defesa

DDoS	Distributed Denial Of Service ou Ataque Distribuído Negação de Serviço
DHCP	Dynamic Host Configuration Protocol ou Protocolo de Configuração Dinâmica de Host
DLL	Dynamic-link Library ou Biblioteca de Vínculo Dinâmico
DNS	The Domain Name System ou Sistema de Nomes de Domínio
DNSSEC	Domain Name System Security Extensions ou Extensão de Segurança do Sistema de Nomes de Domínio
ECSP	EC-Council Certified Secure Programmer ou Programador Certificado pelo EC-Council
ENUM	Electronic Number Mapping ou Mapeamento de Número de Telefone
EXIN	Training and Examination ou Treinamento e Exame
FOCA	Fingerprinting Organizations with Collected Archives ou Organizações de Impressão Digital com Arquivos Coletados
FTP	File Transfer Protocol ou Protocolo de Transferência de Arquivos
HIDS	Host-based Intrusion Detection System ou Sistema de Detecção de Intrusão Baseado em Host
HIPS	Host Intrusion Prevention System ou Sistema de Prevenção de Intrução de Host
HTTP	Hypertext Transfer Protocol ou Protolo de Transferência de Hipertexto

HTTPS	Hypertext Transfer Protocol Secure ou Protocolo de Transferência de Hipertexto Seguro
IANA	Internet Assigned Numbers Authority ou Autoridade para Atribuição de Números de Internet
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System ou Sistema de Detecção de Intrusos
IEEE	Institute of Electrical and Electronics Engineers ou Instituto de Engenheiros Eletricistas e Eletrônicos
IP	Internet Protocol ou Protocolo de Internet
IPC	Interprocess Communication ou Comunicação entre Processos
IPSEC	IP Security Protocol ou Protocolo de Segurança IP
IPX	Internetwork Packet Exchange
ISO	International Organization for Standardization ou Organização Internacional de Normalização
JSP	Java Server Page
LAN	Local Area Network ou Rede de Área Local
LOG	Logfile ou Arquivo de Log (Log de dados)
MAC	Media Access Control ou Controle de Acesso ao Meio
MIL STD	Military Standards ou Padrões Militares
MITM	Man-in-the-middle
MX	Mail Exchanger

NDA	Non Disclosure Agreement ou Acordo de Não-Divulgação
NIST	National Institute of Standards ou Instituto de Padrões e Tecnologia
NMAP	Network Mapper ou Mapeador de Rede
OSI	Open Systems Interconnection ou Interconexão de Sistemas Abertos
PDF	Portable Document Format ou Formato de Documento Portátil
PE	Password Element ou Elemento de Senha
Pentest	Penetration Test ou Teste de Penetração
Pentester	Penetration Tester ou Testador de Penetração
PHP	Hypertext Preprocessor ou Pré-Processador de Hipertexto
PMK	Pairwise Master Key ou Chave Mestre Pareada
POP3	Post Office Protocol 3
PPTP	Point-to-Point Tunneling Protocol ou Protocolo de Túnel Ponto a Ponto
RAR	Roshal Archive
RC4	Rivest Cipher 4
RCF	Request for Comments ou Pedido de Comentários
RST	Reset ou Restabelecer
SAE	Simultaneous Authentication of Equals

SFTP	Secure File Transfer Protocol ou Protocolo de Transferência Segura de Arquivos
SNMP	Simple Network Management Protocol ou Protocolo Padrão da Internet para Gerência de Rede
SO	Operating System ou Sistema Operacional
SOP	Same Origin Policy ou Política de Mesma Origem
SPARC	Scalable Processor Architecture ou Arquitetura Escalável de Processador
SQL	Structured Query Language ou Linguagem de Consulta Estruturada
SSH	Secure Shell
SYN	Synchronisation ou Sincronização
TCP	Transmission Control Protocol ou Protocolo de Controle de Transmissão
TI	Tecnologia da informação
TOR	The Onion Router
TTL	Time to Live
UDP	User Datagram Protocol ou Protocolo de Datagrama do Usuário
URL	Uniform Resource Locator ou Localizador Padrão de Recursos
VLAN	Virtual Local Area Network ou Rede Local Virtual
VM	Virtual Machine ou Máquina Virtual

VPN	Virtual Private Network ou Rede Privada Virtual
WEP	Wired Equivalent Privacy
WI-FI	Wireless Fidelity ou Fidelidade Sem Fio
WIPS	Wireless Intrusion Prevention System ou Sistema de Prevenção Contra Intrusões Sem Fio
WLAN	Wireless Local Area Network ou Rede Local Sem Fio
WPA	Wi-Fi Protected Access ou Acesso Protegido por Wi-Fi
WPA2	Wi-Fi Protected Access 2 ou Acesso Protegido por Wi-Fi 2
WPA3	Wi-Fi Protected Access 3 ou Acesso Protegido por Wi-Fi 3
WPSCAN	WordPress Security Scanner
XSS	Cross-site Scripting

Sumário

1 Introdução	16
1.1 Objetivo	17
1.2 Justificativa.....	17
1.3 Metodologia.....	18
2. Níveis de um Pentester e Legislação	19
2.1 Tipos de Teste de Penetração	20
3. Fases do Teste de Penetração	22
3.1 TCP/IP.....	24
3.1.1 DNS.....	26
3.1.2 HTTP e HTTPS	27
3.1.3 Camada de transporte (TCP e UDP).....	27
3.1.4 <i>Unicast, Broadcast e Multicast</i>	28
3.1.5 Protocolo ARP.....	29
3.2 Reconhecimento (<i>FootPrinting</i>) e Ferramentas de Aquisição de Dados	30
3.3 Varredura	37
3.3.1 Enumeração	43
3.3.2 Vulnerabilidades de um serviço.....	46
3.4 Privacidade (mascarando seu IP)	47
3.5 Exploração	48
3.5.1 Metasploit.....	50
3.5.2 Vulnerabilidades Web.....	52
3.5.3 Senhas	53
3.5.4 Engenharia Social	59
3.5.5 Rede Local	60
3.5.6 Wireless.....	62
3.5.6.1 Quebra de chaves	62

3.5.6.2 <i>Evil Twin</i>	64
4. <i>Honeypot</i>	65
5. Precauções e Contramedidas	68
6 Conclusão	77
Referências Bibliográficas	78

1 Introdução

“*Ethical Hacker*” ou “Hacker Ético” é o contrário do que conhecemos como um “*cracker*”, pois ao invés de atacar para prejudicar, sua função é descobrir vulnerabilidades para corrigi-las. O conceito de um *pentester* é fazer com que um ambiente empresarial seja o mais seguro possível, assim evitando ataques externos ou até mesmo internos. O teste de invasão ou *pentest* é a principal ferramenta deste profissional e consiste em realizar técnicas de hackers em testes de estresse na segurança de sistemas e redes, invadir e proteger.

Tanto pequenas quanto grandes empresas não dão o devido valor a segurança da informação, o que as torna vulneráveis na maior parte dos casos. Obviamente, um hacker ético precisa ser contratado para realizar este tipo de serviço, onde por meio de contrato são estabelecidas as possibilidades e limitações à atuação do profissional.

É importante que haja treinamento para as equipes de TI e funcionários de uma empresa, para o desenvolvimento seguro e engenharia social, ou seja, para prevenir falhas no desenvolvimento de um projeto ou possibilitar que estas sejam corrigidas mesmo depois de estar em funcionamento, no que tange a engenharia social, o objetivo do treinamento é evitar que funcionários de sua empresa, que não tem noção deste fato, sejam facilmente enganados.

Com todo o processo realizado a probabilidade de acontecer algum problema tanto físico quanto financeiro é reduzida, mas nunca nula e isso se dá ao fato de as vulnerabilidades estarem sempre crescendo em sistemas operacionais, softwares de terceiros, redes falhas e vários outros pontos esquecidos. Uma empresa bem preparada não sofrerá com *crackers*.

Por ser uma profissão pouco conhecida falta profissionais qualificados para realizar os testes que serão citados ao decorrer do documento, considerando que estes são para “*pentesters*” iniciantes que estão começando sua carreira e irão se desenvolver ao decorrer dos anos. Um “*pentester*” sênior terá conhecimento suficiente para desenvolver suas próprias ferramentas ou *metasploits*, além de saber quais pontos serão mais importantes nas fases de reconhecimento e varredura. A falta de pessoas qualificadas está diretamente

relacionada ao desconhecimento das bases como redes e algumas linguagens de programações que irão o ajudar.

“Teste de penetração pode ser definido como uma tentativa legal e autorizada de localizar e explorar com sucesso sistemas de computadores com o propósito de fazer estes sistemas seguros.” (The Basics of Hacking and Penetration Testing, 2013, p. 1).

1.1 Objetivo

Este projeto tem como objetivo ser um documento instrutivo, apresentar técnicas de invasão para que se possa entender como estas são realizadas e por onde começar a se defender em alguns casos específicos, assim demonstrando as vulnerabilidades para entender o processo de um *pentester* e apresentando a profissão a todos os tipos de profissionais e empresas. Vale lembrar que cada vulnerabilidade possui uma solução diferente então esta será apenas uma base no entendimento do conceito de segurança.

Para alcançar este objetivo serão realizados testes em máquinas virtuais e uma rede de testes com intuito acadêmico, sem afetar nenhum terceiro. Estes demonstraram o processo de reconhecimento, varredura, ganho de acesso, como manter o acesso e por fim cobrimento dos rastros.

Os exemplos citarão também possibilidades do que fazer dentro dos sistemas acessados, bem como diferentes formas de acessá-los e outras vulnerabilidades de softwares de terceiros, as quais empresas não costumam dar a devida atenção, até mesmo para a engenharia social, que em muitos casos acaba sendo uma das formas mais fáceis de acesso a estes ataques, pelo fato de muitos funcionários das empresas não estarem preparados para estas situações.

1.2 Justificativa

Precisamos mostrar para as empresas qual a importância da segurança da informação, pois muitas delas ainda vivem no passado com tecnologias defasadas e em constante risco de ataques realizados por *crackers* (atacantes maliciosos), que poderão roubar informações valiosas ou até mesmo dinheiro dependendo do caso. Isso vale também para os jovens que estão estudando programação ou a área de tecnologia e necessitam entender qual a importância destes fatores e como evitar problemas futuros antes mesmo de acontecerem. Conhecendo algumas vulnerabilidades é possível se preparar, minimizando os casos de ataques que empresas vem sofrendo ao decorrer dos anos, sendo válido até para suas redes locais e entendendo qual o risco de softwares de terceiros, redes e infraestruturas mal configuradas.

Com mais profissionais qualificados e empresas cientes do fator de segurança é possível gerar mais empregos e preparar funcionários de outras áreas preparados para situações que possam vir a ocorrer.

1.3 Metodologia

Por meio de VMs (*Virtual Machines* ou Máquinas Virtuais) serão realizados testes de invasão (pentest), onde teremos as máquinas Kali Linux e *Metasploitable* como principais e as vítimas serão os Windows XP, 7. Para cada caso teremos pelo menos um tipo de vulnerabilidade demonstrado após serem os reconhecimentos e varreduras das redes, dos endereços de IPs individuais (algumas das vulnerabilidades irão se aplicar para mais de um alvo) e de softwares instalados nestas máquinas.

Os processos de reconhecimento serão realizados por buscas em sites específicos e por meio de comandos no prompt de comando do Kali Linux. As varreduras serão realizadas por comandos e softwares instalados para nos mostrar se é possível atacar o alvo e, se sim, quais serão as formas que poderemos utilizar de acordo com a vulnerabilidade exposta.

2. Níveis de um Pentester e Legislação

Podemos considerar que há três níveis de um *pentester*, o júnior que é um profissional com pouca experiência, o pleno, com um domínio maior das ferramentas e o sênior, que já tem grande domínio e experiência.

Os de nível júnior geralmente conhecem as ferramentas básicas para realizar o teste de penetração, mas ele não tem o conhecimento aprofundado de programação e identificação de falhas, portanto identifica falhas já existentes e não falhas novas.

Os de nível pleno, além de saber usar as técnicas e ferramentas, consegue desenvolver suas próprias ferramentas ou melhorar as existentes, como por exemplo, melhorar um *metasploit*, fazer melhorias no *Burp*.

Já os de nível sênior são aqueles que conseguem identificar e descobrir vulnerabilidades novas, *debugar* códigos e fazer vários tipos de identificação mais avançadas em sistemas.

Temos que levar em consideração os aspectos legais de uma invasão, não podemos simplesmente realizar testes de invasão sem contrato. Com base no artigo da lei “Carolina Dieckmann”, Art.154 “Invadir dispositivo informático alheio conectado ou não a rede de computadores, mediante violação indevida de mecanismo de segurança e com o fim de obter, adulterar ou destruir dados ou informações sem autorização expressa ou tácita do titular do dispositivo ou instalar vulnerabilidades para obter vantagem ilícita: pena: detenção, de 3 (três) meses a 1 (um) ano, e multa”. (Lei Nº 12.737, 2012).

Entretanto ao analisar a lei, caso não haja mecanismo de segurança como senha por exemplo, não há proteção, o que não caracteriza a lei. Além disso, caso não haja obtenção ou destruição de dados, apenas entrar e olhar as informações, não se caracteriza crime. Mesmo assim é necessária uma autorização escrita para ser realizado o *pentest*.

Também é importante o artigo da lei “Carolina Dieckmann” – DDoS, Art.266 “Incorre na mesma pena quem interrompe serviço telemático ou de

informação de utilidade pública, ou impede ou dificulta-lhe o restabelecimento”. (Lei Nº 12.737, 2012).

2.1 Tipos de Teste de Penetração

Conforme Mohd. Ehmer Khan e Farmeena Khan (2012) existem três tipos de testes de penetração (*penetration test*), caixa preta (*Back Box*), caixa cinza (*Gray Box*) e caixa branca (*White Box*). Cada um se aplica a uma situação diferente dependendo da situação de seu alvo.

Caixa preta é um teste realizado sem conhecimento prévio referente ao funcionamento interno, conseqüentemente, sua pesquisa é feita do zero, onde não haverá acesso ao código-fonte. Por estes fatores tende a ser o teste mais caro para se implementar. De acordo com Khan e Khan (2012) algumas das vantagens são a eficiência para grandes segmentos de código, a percepção do testador é mais simples, a perspectiva dos usuários está claramente separada da dos desenvolvedores e o desenvolvimento mais rápido de casos de teste. Já as desvantagens são que apenas uma quantia de cenários é realmente testada, o que torna limitado, não há especificações claras em alguns testes e poderá ser ineficiente.

Para o caixa cinza a ideia é mostrar uma visão entre departamentos, havendo conhecimento de estruturas de dados e algoritmo, com propósito de projetar casos de teste. De acordo com Khan e Khan (2012), este fornece técnicas de teste do caixa branca e preta, o testador depende da interface e especificação funcional ao invés do código fonte, pode projetar excelentes cenários de teste, é feito do ponto de vista do usuário e pode haver teste imparcial. As desvantagens incluem dificuldade de cobertura do teste limitada por falta de acesso ao código-fonte, associar a identificação de defeitos em aplicativos atribuídos, muitos caminhos não são testados e se o desenvolvedor de software já executou algum teste poderá torná-lo redundante.

Já o caixa branca requer que o testador tenha total conhecimento do código fonte, este mostrará se um administrador de rede da empresa tentar causar algum dano. Neste caso é feita uma busca detalhada no código para

encontrar brechas. Segundo Khan e Khan (2012) pode-se considerar como vantagens o fato de revelar erros escondidos no código removendo linhas extras, os efeitos colaterais são benéficos e a cobertura máxima é alcançada durante os testes. Como desvantagens podemos citar o fato de ser muito caro pelos requisitos do testador serem muito altos, muitos caminhos não serão testados e alguns códigos poderão não ser vistos.

“O teste pode descobrir erros de implementação como má gestão de chaves criptográficas analisando funcionamento e estruturas internas de um software. É aplicável nos níveis de integração, unidade e sistema do processo de teste de software” (Khan e Khan, 2012, v. 3, p. 12).

3. Fases do Teste de Penetração

A metodologia varia conforme a atuação de cada profissional ou de alguns métodos de certificações utilizados, mas na realidade a sequência é a mesma o que muda é a forma de agrupar, podendo ser dividido em fases.

Na fase de reconhecimento é feita uma varredura ou pesquisa sobre o sistema, para identificar por exemplo: informações sobre o domínio, dados importantes na página da empresa, sistemas básicos e possíveis usuários para contato. “Técnica utilizada para pesquisar e descobrir informações sobre o alvo. Incluímos recursos de mapeamento de redes e consulta a banco de dados *Whois*, além de sites como o Google”. (Segredos do Hacker Ético, 2014, p. 63).

A varredura pega as informações da fase de reconhecimento e usa técnicas de varreduras de endereços IPs e portas para identificar serviços como quantos computadores ou servidores tem na empresa, qual o nome e IP de cada, bem como através de enumeração e *fingerprint* descobrir o sistema operacional destes e quais serviços estão rodando. “Descoberta de computadores ativos na rede através de ICMP (*Ping*) e de portas de serviços abertas nesses sistemas.” (Segredos do Hacker Ético, 2014, p. 63).

“Identificação dos serviços que estão rodando nas portas dos sistemas descobertos, além da descoberta do sistema operacional utilizado. Compilação de dados sobre recursos disponíveis, tais como compartilhamentos e usuários existentes no sistema” (Segredos do Hacker Ético, 2014, v. 5, p. 63).

No ganho de acesso pega-se todas as informações colhidas anteriormente, por tanto as anteriores deverão ser muito bem feitas, pois sem isso não terá informações suficientes para esta fase. É nesta fase que o pentester irá decidir qual caminho e medidas tomar.

“Aqui entra a pesquisa para se identificar falhas nos serviços, o que fica mais fácil após a enumeração, onde descobrimos quais os serviços que estavam rodando. Também utilizamos *scanners* de vulnerabilidade para adiantar a pesquisa” (Segredos do Hacker Ético, 2014, v. 5, p. 63).

Concluídos os passos anteriores é preciso manter o acesso, neste ponto o hacker ético já tem o acesso e irá instalar algo no sistema para manter o

acesso, mais conhecido como “*backdoor*” (porta dos fundos), bem como instalar um serviço ou agendar algum processo e tomar alguma medida caso precise voltar aquele sistema posteriormente.

Para que não sejam descobertos, é imprescindível realizar a fase de cobertura de rastros e essa etapa envolve vários aspectos, desde esconder o endereço IP de origem, usando servidores proxy ou rede TOR, como apagar *logs* de acesso, utilizar *root kits* e várias outras técnicas.

Todas as etapas anteriores baseiam-se em um termo de responsabilidade, que requer extrema atenção do profissional que irá realizar os testes, pois, quando este termo é feito, existem pontos que devem ser levados em consideração para realizar um *penetration test* em um cliente, posto que irá invadir a rede dele tentando quebrar senhas e realizar procedimentos de ataque.

Devido a isso utiliza-se o “NDA” ou “*non disclosure agreement*”, termo o qual irá definir até onde o profissional pode chegar e que estas informações não serão utilizadas para outros procedimentos, ou seja, manter sigilo. Witman (2005) consta que é um acordo que restringe o uso de informações ao proibir o contratante de divulgar os dados. A intenção é restringir o uso e divulgação de informações confidenciais pela outra parte, o que exige a definição de quais informações e como usá-las.

O escopo define as seguintes situações:

- O que será testado
- Quais os prazos
- Se terá engenharia social

Limites devem ser estabelecidos, como o que poderá fazer caso consiga uma conta de diretor por exemplo, pois se não forem definidos, e a empresa não estiver preparada, o profissional pode causar danos irreparáveis caso não haja um *backup* adequado.

A pessoa que contrata um hacker ético deve ter a autorização da empresa para lhe permitir específicas tentativas de acessos como a conta de um diretor,

pois um responsável pela área de TI não pode lhe dar acesso total sem permissão de alguém como diretor ou gerente.

Depois de todos os testes realizados é feito um relatório, que é parte essencial para entregar ao cliente, onde irá relatar de forma detalhada todas as vulnerabilidades. Há dois tipos de relatórios para *penetration test*, o sumário executivo e o relatório técnico. Vale ressaltar que o *pentester* não resolve os problemas encontrados, apenas os detecta e dá sugestões à equipe de TI da empresa de como resolver. Nestes relatórios deve ser feita uma análise do grau de risco de determinada vulnerabilidade.

O sumário executivo é um resumo preparado para os executivos da empresa e não para um técnico, no qual deve ser demonstrado, em um gráfico por exemplo, quais são os problemas e o prazo médio para resolução, pois alguns prazos devem ser passados à equipe de segurança.

Já o relatório técnico é destinado à equipe de TI e profissionais de segurança da empresa, deve mostrar passo a passo, detalhando-as sugestões de como resolver os problemas e os acessos obtidos.

Existem algumas ferramentas que geram estes relatórios por padrão. Como o *Bryter*, *NonDisclosureAgreement*, *eForms* e etc.

3.1 TCP/IP

De acordo com Hunt (2002) em 1969 a ARPA (*Advanced Research Projects Agency* ou Agência de Projetos de Pesquisa Avançada) fundou um projeto de pesquisa e desenvolvimento para criar uma rede de comutação de pacotes experimental. Essa rede, chamada ARPAnet (*Advanced Research Projects Agency Network* ou Agência de Projetos de Pesquisa Avançada de Rede) foi desenvolvida com o intuito de estudar técnicas para fornecer comunicações de dados robustas, confiáveis e independentes. A rede experimental foi um sucesso, tanto que muitas organizações optaram por usá-la desde o começo. Em 1975 a ARPAnet foi convertida de uma rede experimental para operacional e a responsabilidade da administração foi dada à DCA

(*Defense Communications Agency* ou Agência de Comunicação de Defesa). Entretanto, o desenvolvimento da ARPAnet não parou por causa disso, passou a ser utilizado como rede operacional, os protocolos TCP/IP foram desenvolvidos depois disso e adotados como MIL STD (*Military Standards* ou Padrões Militares) em 1983, quando todos os *hosts* conectados à rede foram obrigados a converter para os novos protocolos.

Hunt (2002) menciona que a popularidade do TCP/IP não cresceu rapidamente por causa dos protocolos e pelo fato de necessitar de uma conexão com a internet. Eles atenderam uma necessidade importante da época, que foi a comunicação de dados mundial. Vários recursos lhes permitiram atender tal necessidade como, padrões de protocolo abertos, independência de *hardware* de rede física específico, um esquema de endereçamento comum que permite a qualquer dispositivo TCP/IP endereçar de maneira exclusiva qualquer outro em toda rede e protocolos padronizados de alto nível para serviços de usuário consistente e amplamente disponíveis.

“O TCP/IP cria uma rede heterogênea com protocolos abertos que são independentes de sistema operacional e das diferenças de arquitetura, estão disponíveis para todos e são desenvolvidos e alterados por consenso de todos, não por decreto de algum fabricante”. (TCP/IP Network Administration, 2002, v. 3, p. 4).

Briscoe (2000) relata que o modelo OSI é genérico e se aplica a todos os tipos de rede, não apenas TCP/IP e todos os tipos de mídia, não apenas a *Ethernet*. O OSI era um grupo de trabalho dentro da ISO (*International Standards Organization* ou Organização de padrões internacionais).

Quando se lida com teste de penetração deve-se utilizar vários níveis de camadas, desde a de rede até a de aplicação. Briscoe (2000) também diz que o modelo OSI constitui-se em sete camadas, sendo elas:

- Aplicação: funções a nível de aplicação. Onde se encontram os protocolos de usuário e aplicação finais, como telnet, ftp e *mail*.
- Apresentação: formatação de dados. Os dados da aplicação são tanto compactados quanto descompactados, prontos para uso da

aplicação em execução. As conversões de protocolo, criptografia, decriptografia e expansão de gráfico acontecem aqui.

- Sessão: estabelecer a conexão. Esta permite a comunicação de duas entidades de apresentação para a troca de dados. A camada é muito importante no campo do E-commerce (comércio eletrônico), pois, uma vez que o usuário começa a comprar produtos em um servidor *Web* é muito importante que estes servidores não tenham balanceamento de carga em diferentes servidores em um conjunto de servidores.
- Transporte: método de entrega de dados. É onde o TCP reside. O padrão diz que a camada de transporte alivia a de sessão da carga de garantir a confiabilidade e integridade dos dados.
- Rede: roteamento de pacotes. Fornece um meio de comunicação de sistemas abertos para estabelecer, manter e encerrar conexões de rede. O protocolo IP reside nesta camada, assim como alguns protocolos de roteamento.
- Enlace: correção de erros. Essa camada define a estratégia de acesso para compartilhar o meio físico, incluindo *link* de dados e problemas de acesso à mídia.
- Física: define as características físicas e elétricas da rede.

3.1.1 DNS

Assunção (2014) diz que o DNS (*Domain Name System* ou Sistema de Nomes de Domínios) localiza e traduz os endereços de sites para IPs. Por padrão se utiliza o DNS oferecido pelo provedor ou empresa que mantém sua conexão. “É como se o sistema DNS tivesse sua própria rede. Se um servidor não consegue traduzir um nome de domínio, ele pergunta a outro que, se não souber pergunta a um terceiro servidor e, assim, sucessivamente” (Segredos do Hacker Ético, 2014, v. 5, p. 39) .É viável utilizar serviços que oferecem uma

melhor performance e/ou mais segura como por exemplo *OpenDNS*, *SecureDNS*.

3.1.2 HTTP e HTTPS

Val e Klemets (2000) mencionam que HTTP (*Hypertext Transfer Protocol* ou Protocolo de Transferência de Hipertexto) é um método para transmitir dados de mídia digital de um servidor. O servidor é configurado para conversar com um computador cliente por meio de uma rede de computadores. O método inclui receber no servidor do cliente uma solicitação HTTP *POST*. A solicitação *POST* solicita uma primeira parte dos dados de mídia digital e inclui um cabeçalho (*Header*) de solicitação e um corpo de entidade de solicitação. O corpo da entidade de solicitação inclui um comando de mídia para fazer com que a primeira parte dos dados de mídia digital sejam enviados do servidor para o cliente. O método inclui ainda o envio de uma resposta HTTP ao cliente a partir do servidor. A resposta HTTP inclui um cabeçalho de resposta e um corpo de entidade de resposta. O corpo da entidade de resposta inclui pelo menos uma primeira parte dos dados de mídia digital.

De acordo com Gourley et al. (2002) HTTPS (*Hypertext Transfer Protocol Security* ou Protocolo de Transferência de Hipertexto Seguro) é a versão segura mais popular do HTTP. É amplamente implementado e está disponível em todos os principais navegadores e servidores comerciais. HTTPS combina o protocolo HTTP com um poderoso conjunto de técnicas criptográficas simétricas, assimétricas e baseadas em certificados, tornando-o muito mais seguro, flexível e fácil de administrar.

“O HTTPS acelerou o crescimento dos aplicativos da internet e tem sido uma grande força no rápido crescimento do comércio baseado na *web*. Também tem sido crítico na administração segura e de área ampla de aplicativos *web* distribuídos”. (HTTP: The Definitive Guide, 2002, v. 1, p. 322).

3.1.3 Camada de transporte (TCP e UDP)

Dostálek e Kabelová (2006) defendem que O TCP (*Transmission Control Protocol* ou Protocolo de Controle de Transmissão) transporta dados usando segmentos TCP que são endereçados a aplicativos individuais. O UDP (*User Datagram Protocol* ou Protocolo de Datagrama de Usuário) transporta dados usando datagramas UDP. Os dois estabelecem uma conexão entre aplicativos executados em computadores remotos. Também podem facilitar a comunicação entre processos em execução no mesmo computador. A diferença entre eles é que o TCP é um serviço orientado a conexão, o destino confirma os dados recebidos. Se alguns dados forem perdidos, o destino solicitará uma retransmissão dos mesmos. O UDP transporta dados usando datagramas (a entrega não é garantida), então não há preocupação se a entrega de dados foi feita ou não. O UDP é um serviço orientado a conexões, a porta é usada como endereço. Para melhor entendimento, o IP corresponde ao endereço de uma casa, enquanto a porta informa o nome da pessoa que deve receber uma correspondência.

Numa mesma máquina é possível ter várias portas, cada porta é associada a um serviço por exemplo POP3 (110), FTP (20 e 21).

“O número de portas conhecidos são considerados portas privilegiadas que não devem ser vinculadas a um processo do usuário. As portas numeradas de 1024 a 49151 são portas registradas. A IANA (*Internet Assigned Numbers Authority* ou Autoridade para Atribuição de Números de Internet) tenta manter um registro dos serviços que usam essas portas, mas não atribui oficialmente números de porta nesta faixa. Os números de porta de 49152 a 65535 são portas privadas. Números de portas privadas estão disponíveis para qualquer uso.” (TCP/IP Network Administration, 2002, v. 3, pg. 46).

3.1.4 Unicast, Broadcast e Multicast

A obra de Assunção (2014) menciona que a quantidade de tráfego gerada em uma rede pode ser definida em três tipos: *unicast*, *multicast* e *broadcast*

Na transmissão *Unicast*, uma cópia dos dados é enviada da origem para cada computador (cliente) que os solicite, então nenhum outro computador na

rede os receberá. No entanto, nem sempre este será eficiente, pois numa rede com muitos computadores terá que transmitir múltiplas cópias dos dados.

No tipo *Broadcast* os dados serão enviados apenas uma vez, porém para a rede toda, o que torna o processo não muito eficiente devido ao fato de cair a velocidade de transmissão já que todos clientes receberão os dados (mesmo os que não fizeram pedido receberão). O protocolo ARP é um dos que utiliza este tipo.

O *Multicast* é uma mistura dos dois, com apenas uma cópia dos dados e enviados somente aos clientes que fizeram o pedido, evitando o tráfego muito intenso e o congestionamento na rede.

3.1.5 Protocolo ARP

De acordo com Tingley e Walsh (2002) o ARP (*Address Resolution Protocol* ou Protocolo de Resolução de Endereço) 826 é padrão, conforme definido na RCF (*Request for Comments* ou Solicitação de Comentários) em 1982, ele fornece mapeamentos entre os endereços IP e endereços *Ethernet* (MAC) em uma rede (faz a conversão da camada de rede na de enlace). Os endereços IP são endereçados na camada 3 (rede) do modelo OSI.

Os roteadores são normalmente considerados dispositivos da camada 3 e o roteamento de pacotes por meio de uma rede se baseia nos endereços da camada 3 contidos nos pacotes. Quando um roteador deseja enviar dados para outro dispositivo conectado via *Ethernet* ou *Gigabit link* (GbE ou 1 GigE) ele emite uma solicitação ARP contendo o endereço IP desse dispositivo, a solicitação é então transmitida a todos os dispositivos em um *link* físico compartilhado, ao qual o roteador está conectado. O dispositivo de destino, vendo seu próprio endereço IP na solicitação, responde com uma resposta ARP contendo seu próprio endereço. O remetente original pode então armazenar o mapeamento do endereço IP para o endereço *Ethernet* internamente e usá-lo para gerar cabeçalhos do tráfego de dados de saída tendo estes como de destino.

3.2 Reconhecimento (*FootPrinting*) e Ferramentas de Aquisição de Dados

McClure, Scambray e Kurtz (2001) dizem que o reconhecimento de uma organização permite que os invasores criem um perfil completo da postura de segurança. Ao usar uma combinação de ferramentas e técnicas, podem pegar uma quantia desconhecida e reduzi-la a um intervalo específico de nomes de domínio, blocos de rede e endereços IP individuais de sistemas conectados diretamente à internet. Embora existam muitos tipos de técnicas de *footprint*, elas têm como objetivo principal descobrir informações relacionadas aos seguintes ambientes: Internet, intranet, acesso remoto e extranet. O Quadro 1 descreve estes ambientes e as informações críticas que um invasor tentará identificar.

Quadro 1 - Ambiente e informações críticas que invasores podem identificar

Tecnologia	Identificação
Internet	<p>Nome de Domínio.</p> <p>Bloco de rede.</p> <p>Endereços IP Específicos de sistemas acessíveis através da Internet.</p> <p>Serviços TCP e UDP em execução para cada sistema identificado.</p> <p>Arquitetura do sistema (por exemplo, SPARC x X86).</p> <p>Mecanismos de controle de acesso e listas de controle de acesso relacionadas (ACLs).</p> <p>Sistemas de detecção de intrusão (IDSes).</p> <p>Enumeração do sistema (nomes de usuários e grupos, banners do</p>

	sistema, tabelas de roteamento, informações SNMP).
Intranet	<p>Protocolos de rede em uso (por exemplo, IP, IPX, DecNET).</p> <p>Nomes de domínio interno.</p> <p>Blocos de rede.</p> <p>Endereços IP específicos de sistemas acessíveis via internet.</p> <p>Serviços TCP e UDP em execução em cada sistema identificado.</p> <p>Arquitetura do sistema.</p> <p>Mecanismos de controle de acesso e listas de controle de acesso relacionadas (ACLs).</p> <p>Sistemas de detecção de intrusão.</p> <p>Enumeração do sistema.</p>
Acesso Remoto	<p>Número de telefone analógico / digital.</p> <p>Tipo de sistema remoto.</p> <p>Mecanismos de autenticação.</p> <p>VPNs e protocolos relacionados (IPSEC, PPTP)</p>
Extranet	<p>Origem e destino da conexão.</p> <p>Tipo de conexão.</p> <p>Mecanismo de controle de acesso.</p>

Fonte: quadro adaptado do livro Hacking Exposed: Network Security Secrets and Solutions, Third Edition, 2001.

O reconhecimento é necessário para garantir, de forma sistemática e metódica, que todas as informações relacionadas às tecnologias mencionadas sejam identificadas. Sem uma metodologia sólida para realizar esse tipo de *footprinting*, é provável que se perca informações importantes relacionadas a uma tecnologia ou organização específica.

Esta é uma etapa muito importante na realização do *penetration test*, considerando uma *black box* onde se precisa descobrir todas as informações, é possível utilizar sites como www.whatsmyip.org e www.registro.br para descobrir algumas destas informações. Um pouco mais a fundo em *footprinting* pode-se utilizar o site www.archive.org para descobrir informações mais antigas e profundas de um site (alterações efetuadas, data de criação e etc.).

Outra forma de conseguir estas informações é em site de empregos, geralmente empresas de grande porte colocam vagas de TI (tecnologia da informação) com requisitos e sistemas utilizados, o que dá um caminho por onde realizar o *penetration test*.

Para fazer o rastreamento de *e-mails* pode-se utilizar ferramentas como *ReadNotify*, *Grabify*, entre outras.

Depois da abertura do *e-mail* enviado tem as seguintes informações (Figura 1), onde é possível ver dados como localização, navegador, IP etc. Informações importantes que podem gerar parte dos requisitos para realizar uma penetração.

Figura 1 - ReadNotify (captação de roda de e-mail enviado)

ReadNotify

Refresh Di

ReadNotify email tracking history

To	gb.marquesini@uol.com.br
From	gb_marquesini@hotmail.com
Subject	teste readnotify
Sent on	2020/09/12 , 15:25:46pm 'America/Sao_Paulo' time
1st Open	2020/09/12 , 15:26:36pm -3:00

Tracking Details

Opened	
Opened	2020/09/12 , 15:26:36pm (UTC -3:00) - 50sec after sending
Location	Artur Nogueira, Sao Paulo, Brazil (86% likelihood)
Opened on	23.126.255.143.customer.netaki.com.br (143.255.126.23:19267)
Language	of recipient's PC: pt-BR (Portuguese/Brazil), pt;q=0.9 (Portuguese), en-US;q=0.8 (English/United States), en;q=0.7 (English)
Browser	used by recipient: Moz/5.0 (WinNT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36
Accepts	Files browser can open: i/avif,i/webp,i/apng,i/*,*/*;q=0.8

Summary - as at 2020/09/12 , 15:27:10pm (UTC -3:00) - 1min24sec after sending

Total Opened 1 time by 1 reader

Fonte: Elaborada pelo autor, captura de tela

O *Grabify* é um site como muitos outros que permite criar um redirecionamento de um *link* para rastrear informações da pessoa que acessá-lo. Utilizando o endereço unip.br, por exemplo, pode-se criar uma nova URL (*Uniform Resource Locator* ou Localizador Padrão de Recursos) para criar um *link* chamativo como este <https://freegiftcards.co/image.php?id=6E3JSD.jpg> (desativado, por motivos de segurança, não acesse) que irá levar diretamente ao site da Universidade.

Figura 2 - *Grabify* (Redirecionamento de link para captura de informações)

ADVANCED LOG	
Date/Time	2020-09-13 06:38:17
IP Address	143.255.126.23
Country	Brazil, Artur Nogueira
Orientation	landscape-primary
Timezone	America/Sao_Paulo GMT-3
User Time	Sat Sep 12 2020 15:38:18 GMT-0300 (Horário Padrão de Brasília)
Language	pt-BR
Incognito/Private Window	No
Ad Blocker	No
Screen Size	1600 x 900
GPU	NVIDIA GeForce GTX 960
Browser	Chrome (85.0.4183.102)
Operating System	Windows 10 x64
Touch Screen	No
User Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36
Platform	Win32
Referring URL	no referrer

Fonte: Elaborada pelo autor, captura de tela

O *link* contém as palavras “*free gift cards*” (cartões de presente grátis) o que o torna chamativo e pode atrair muitos leigos e desatentos ao acessá-lo poderão expor informações relevantes.

O Google também pode ser uma ferramenta para este processo, chamada de Google *Hacking* que por meio de uma tecnologia denominada *spiders* ou *webcrawlers*, que irão indexar as páginas, sabendo disso pode-se utilizá-lo para descobrir URLs e algumas outras informações relevantes para ataques. Long (2008) diz que a interface da *web* do Google é inconfundível, sua aparência e seu comportamento são protegidos por direitos autorais. O que a maioria das pessoas não consegue perceber é que a interface também é extremamente poderosa, pois a ferramenta oferece termos especiais conhecidos como operadores avançados para ajudá-lo a realizar consultas mais específicas. Estes operadores, se usados corretamente, poderão ajudar a obter informações precisas sem demandar muito tempo analisando uma página. Quando estes operadores não são fornecidos em uma consulta, o Google localiza seus termos de pesquisa em qualquer área da página *Web*, incluindo título, texto, URL e etc. Estes são alguns dos operadores avançados:

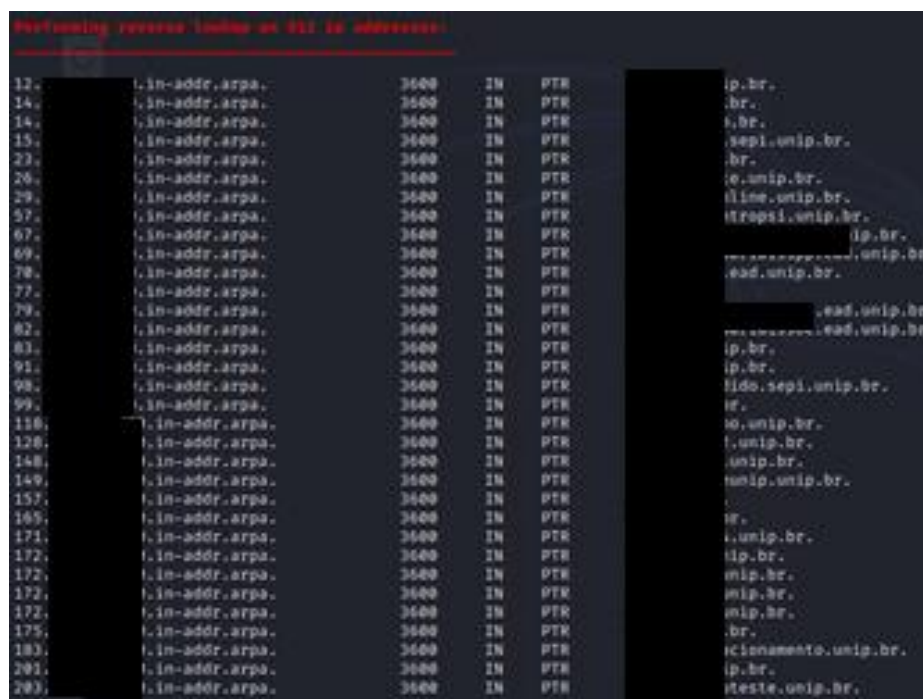
- *Intitle, allintitle*
- *Inurl, allinurl*
- *Filetype*
- *Allintext*
- *Site*
- *Link*
- *Inanchor*
- *Daterange*
- *Cache*
- *Info*
- *Related*
- *Phonebook*
- *Rphonebook*
- *Bphonebook*
- *Author*
- *Group*
- *Msgid*
- *Insubject*
- *Stocks*
- *Define*

Operadores avançados são adições a uma consulta projetada para restringir os resultados da pesquisa, e embora sejam relativamente fáceis de usar, têm uma sintaxe bastante rígida que deve ser seguida.

Uma das melhores opções é utilizar o Dnsenum que irá procurar informações de DNS não bloqueadas, como ips e subdomínios. Dnsenum é uma ferramenta de análise aos servidores DNS, esta irá conseguir o endereço do alojamento, nome do servidor, registro MX, sub domínios, informações de Whois e pesquisa reversa.

“O *Electronic Number Mapping* (Enum) é um protocolo utilizado para mapear números de telefone E.164 em um *Uniform Resource Identifiers* (URLs) utilizados na internet. O ENUM é uma nova tecnologia em fase de pesquisa, desenvolvimento e implementação que utiliza o *Domain Name System* (DNS) como banco de dados e insere novos desafios de desempenho e segurança ao sistema de nomes e domínios atual.” (Mata e Guardieiro, 2011, v. 1, p. 1).

Figura 3 - Execução do Dnsenum



```

Performing reverse lookup on 512 IP addresses:
12. 1.in-addr.arpa. 3600 IN PTR ip.br.
14. 1.in-addr.arpa. 3600 IN PTR .br.
14. 1.in-addr.arpa. 3600 IN PTR .br.
15. 1.in-addr.arpa. 3600 IN PTR sepi.unip.br.
23. 1.in-addr.arpa. 3600 IN PTR .br.
26. 1.in-addr.arpa. 3600 IN PTR e.unip.br.
29. 1.in-addr.arpa. 3600 IN PTR line.unip.br.
57. 1.in-addr.arpa. 3600 IN PTR tropi.unip.br.
67. 1.in-addr.arpa. 3600 IN PTR ip.br.
69. 1.in-addr.arpa. 3600 IN PTR .unip.br.
78. 1.in-addr.arpa. 3600 IN PTR ead.unip.br.
77. 1.in-addr.arpa. 3600 IN PTR .ead.unip.br.
79. 1.in-addr.arpa. 3600 IN PTR .ead.unip.br.
82. 1.in-addr.arpa. 3600 IN PTR .ead.unip.br.
83. 1.in-addr.arpa. 3600 IN PTR ip.br.
91. 1.in-addr.arpa. 3600 IN PTR ip.br.
98. 1.in-addr.arpa. 3600 IN PTR lido.sepi.unip.br.
99. 1.in-addr.arpa. 3600 IN PTR if.
118. 1.in-addr.arpa. 3600 IN PTR io.unip.br.
128. 1.in-addr.arpa. 3600 IN PTR i.unip.br.
148. 1.in-addr.arpa. 3600 IN PTR .unip.br.
149. 1.in-addr.arpa. 3600 IN PTR tunip.unip.br.
157. 1.in-addr.arpa. 3600 IN PTR .
169. 1.in-addr.arpa. 3600 IN PTR if.
171. 1.in-addr.arpa. 3600 IN PTR i.unip.br.
172. 1.in-addr.arpa. 3600 IN PTR ip.br.
172. 1.in-addr.arpa. 3600 IN PTR mip.br.
172. 1.in-addr.arpa. 3600 IN PTR mip.br.
172. 1.in-addr.arpa. 3600 IN PTR mip.br.
175. 1.in-addr.arpa. 3600 IN PTR .br.
183. 1.in-addr.arpa. 3600 IN PTR cionamento.unip.br.
201. 1.in-addr.arpa. 3600 IN PTR ip.br.
293. 1.in-addr.arpa. 3600 IN PTR teste.unip.br.
  
```

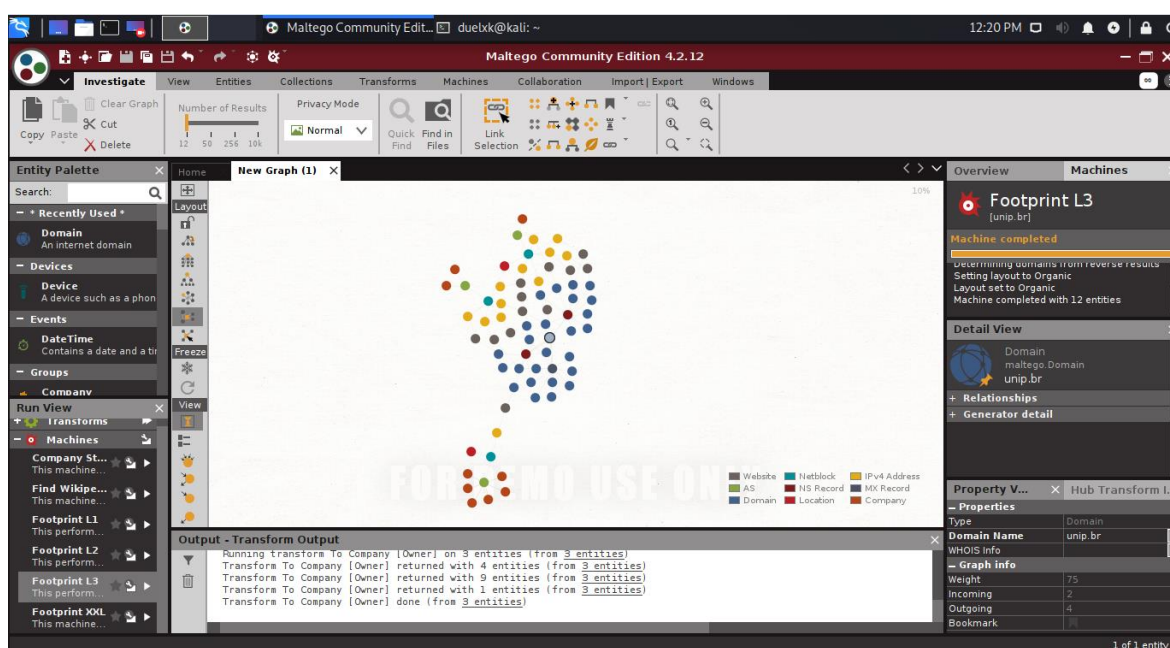
Fonte: Elaborada pelo autor, captura de tela com informações ocultas

Dirb é uma ferramenta de “*crawling*” desenvolvida em *Python* que permite a revelação de aplicações *web*. Esta faz um ataque de força bruta com vários nomes de pastas e trata o retorno identificando, se são acessíveis ou não.

O Sn1per é uma ferramenta que tem a função de todas as citadas anteriormente e que faz um escaneamento completo do domínio como *e-mails*, subdomínios etc.

Há também a possibilidade de ferramentas gráficas para este processo, um grande exemplo é o Maltego, entretanto é um programa pago, onde em sua versão gratuita é possível ter um pouco de seu funcionamento.

Figura 4 - Escaneamento com o software Maltego



Fonte: Elaborada pelo autor, captura de tela

A Figura 4 representa os mesmos resultados obtidos nos outros meios, porém, de forma gráfica, onde cada ponto colorido irá representar uma máquina, IPs, *e-mails* e etc.

Para usuários Windows tem a opção de utilizar o FOCA, este faz uma análise de arquivos possibilitando a extração de informações como usuários, servidores, sistemas operacionais, senhas etc.

3.3 Varredura

Nesta fase é feita uma varredura do que está na rede, como servidores existentes, o range de IPs, SOs, portas e etc. e, de acordo com Assunção (2012),

este é o próximo passo após o *footprinting* (reconhecimento). A varredura é feita para detectar computadores ativos na rede e quais portas os sistemas estão rodando. “Essa é uma etapa importantíssima, pois, dependendo de seu resultado, o ataque posterior pode ser muito bem sucedido ou totalmente fracassado” (Segredos do Hacker Ético, 2014, v. 5, p. 77).

O NMAP (*Network Mapper* ou Mapeamento de Rede) é uma ferramenta de exploração de rede, para escanear redes amplas e *hosts* individuais. Utiliza pacotes IP em estado bruto para determinar os *hosts* disponíveis, quais serviços e sistemas operacionais estão executando, *firewalls* em uso e várias outras características relevantes para um pentester. “Criado por Fyodor, esse programa é o scanner mais conhecido de todos, extremamente poderoso e cheio de recursos. Utilizado por dez entre dez hackers como parte de seus ataques” (Assunção, 2014, v. 5, p. 79).

“Alguns sistemas podem oferecer proteções em nível de rede que fazem com que pacotes ICMP ou conexões a determinadas portas, entre outras restrições, não consigam ser efetuadas. O NMAP possui alguns recursos para tentar burlar esses *firewalls*”. (Segredos do Hacker Ético, 2014, v. 5, p. 79).

Figura 5 – Execução do nmap

```
root@kali:/home/duelxk# nmap unip.br
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-13 13:37 -03
Nmap scan report for unip.br ( )
Host is up (0.010s latency).
Not shown: 955 filtered ports, 42 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
```

Fonte: Elaborada pelo autor, captura de tela

Seu resultado é uma lista com os alvos escaneados e/ou informações adicionais dependendo do solicitado, mostrando por exemplo se as portas encontradas estão abertas ou fechadas.

Assunção (2014) defende que utilizando apenas o comando NMAP pode-se obter os resultados mostrados na Figura 5, os números das portas, seu estado

e, possivelmente, uma descrição do serviço. Dentre as vantagens deste em relação as outras ferramentas de escaneamento estão seus recursos avançados, que permitem varreduras de formas mais precisas, muitas vezes passando por proteções impostas por *firewalls*. O Quadro 2 mostrará alguns recursos avançados de escaneamento NMAP:

Quadro 2 - Recursos avançados de escaneamento NMAP

-P0 No ping	Habilite essa opção para não pingar o <i>host</i> antes de fazer a varredura. Isso é necessário para conseguir escanear alguns sites que bloqueiam ICMP ECHO.
-sT TCP connect() scan	Essa é a forma mais básica de escaneamento. A chamada de sistema <i>connect()</i> , provida pelo sistema operacional, é usada para abrir uma conexão com as portas. Se a porta estiver no estado <i>listening</i> , <i>connect()</i> terá sucesso. Uma grande vantagem dessa opção é que não requer nenhum privilégio especial (<i>root</i>). Qualquer usuário pode usar, mesmo com permissões limitadas. Entretanto, é o escaneamento mais facilmente detectado por sistemas de IDS.
-sS TCP Syn scan	Esta tática é muito conhecida como “ <i>half-open</i> ” <i>scanning</i> , porque não realiza uma conexão TCP completa. Enviando um pacote com o <i>flag</i> Syn setado, como se fosse abrir uma

conexão real e esperando pela resposta, uma resposta Syn/ACK indica que a porta está no estado listening. Um *flag* RST indica que a porta não está escutando (non-listening). Se o flag Syn/ACK for recebido, o *flag* RST é imediatamente enviado para encerrar a conexão. A vantagem dessa tática de escaneamento é que poucos sites irão registrá-la em arquivos de log. Infelizmente, são necessários privilégios de *root* para a construção dos pacotes Syn customizados.

-sF -sX -sN Modos Stealth FIN, Xmas Tree ou Null scan

Algumas vezes nem mesmo a tática Syn scanning é furtiva o suficiente. Novos firewalls e filtros de pacotes observam por Syn's para portas restritas e programas como Synlogger e Courtney conseguem detectar esse tipo de escaneamento. Por outro lado, essas técnicas mais avançadas de escaneamento (*stealth* FIN, Xmas Tree, ou *Null scan*) podem ser capazes de passar através desses filtros sem muitos problemas.

-sP P Ping scanning

Algumas vezes somente se quer saber quais hosts da rede estão ativos. O NMAP pode fazer isso enviando um pacote de requisição ICMP para todo endereço IP

especificado da rede. Essa opção não modifica em nada o *scan* de portas.

-sU UDP scans

Esse modo é usado para determinar quais portas UDP (*User Datagram Protocol*, RFC 768) estão abertas no *host*. A técnica implica em enviar 0 *bytes* de dados de pacotes UDP para cada porta da máquina-alvo. Se receber a mensagem “ICMP *port unreachable*” (porta ICMP não alcançada), significa que esta porta está fechada. Senão, assume-se que a porta está aberta.

-sO Scan do Protocolo IP

Esse modo é usado para determinar quais protocolos IPs são usados no *host*. A tática consiste em enviar pacotes IP *raw* sem especificar nenhum cabeçalho para cada protocolo específico na máquina-alvo. Se não receber a mensagem do protocolo “ICMP *port unreachable*”, significa que o protocolo não está sendo usado. Por outro lado, assume-se que está aberto. Vale ressaltar que alguns *hosts* (AIX, HP-UX, *Digital UNIX*) e *firewalls* podem não enviar mensagens de protocolo *unreachable*. Assim, faz parecer que todos os protocolos estão abertos.

-sA A ACK scan

Esse modo é geralmente usado para mapear o conjunto de regras de um *firewall*. Em particular, ele pode ajudar

a determinar quando é um *firewall* completo ou somente um filtro de pacotes simples que bloqueia pacotes Syn de chegada.

-sW W Window scan

Esse modo é muito similar ao ACK scan, exceto que, às vezes, pode ser possível detectar portas abertas mesmo sendo filtradas, isso devido à anomalia do tamanho da janela TCP reportado por vários sistemas operacionais. Sistemas vulneráveis a isso incluem AIX, Amiga, BeOS, BSDI, Cray, Tru64 UNIX, DG/UX, OpenVMS, Digital UNIX, FreeBSD, HP-UX, OS/2, IRIX, MacOS, NetBSD, OpenBSD, OpenStep, QNX, Rhapsody, SunOS 4.X, Ultrix, VAX e VxWorks.

Fonte: Quadro adaptado do livro Segredos do Hacker Ético, 2014. p.80 e

81

A Figura 6 representa um NMAP feito em rede local para identificar quais serão os alvos dos testes, no caso as máquinas virtuais (VMs).

Figura 6 - Nmap em rede local

```
root@kali:/home/duelxk# nmap -sn -n 192.168.56.0/24 | grep 192 | cut -d ' ' -f 5
192.168.56.1
192.168.56.49
192.168.56.58
192.168.56.80
192.168.56.88
192.168.56.99
192.168.56.101
192.168.56.102
192.168.56.142
192.168.56.152
192.168.56.205
192.168.56.221
192.168.56.151
root@kali:/home/duelxk#
```

Fonte: Elaborada pelo autor, captura de tela

Marcados em vermelho são as VMs, de final 142, 205 e 221 (Figura 6). Agora com estes endereços dentro de um arquivo é possível fazer uma varredura de portas (Figura 7).

Figura 7 - Nmap em uma máquina virtual

```
root@kali:/home/duelxk# nmap -iL ips
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-13 15:07 -03
Nmap scan report for IE8Win7 (192.168.56.142)
Host is up (0.00045s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server
MAC Address: 08:00:27:3F:03:BC (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.205
Host is up (0.0020s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:4A:8D:D7 (Oracle VirtualBox virtual NIC)
```

Fonte: Elaborada pelo autor, captura de tela

A partir destes resultados é preciso identificar quais serviços estão sendo executados por trás destas portas, a porta 21 está rodando um servidor FTP, porém, para descobrirmos qual precisamos passar para a fase de enumeração.

3.3.1 Enumeração

A enumeração é o próximo passo após a varredura. Depois de ter os *hosts* ativos na rede e as portas abertas nos sistemas, é preciso descobrir quais serviços estarão executando, qual o sistema operacional do alvo, se é possível extrair nomes de usuários de alguns destes serviços e várias outras informações que serão relevantes durante a penetração, Assunção (2014).

Com os serviços identificados na Figura 7 deve ser feita uma captura de banner para saber qual o serviço está rodando por trás destes, para a máquina virtual de endereço final 205 (*metasploitable*) utilizando-se a porta 80 (uma das que estão abertas), Figura 8.

Figura 8 - Captura de banner de máquina virtual

```
root@kali:/home/duelxk# nc 192.168.56.205 80
HEAD
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.2.8 (Ubuntu) DAV/2 Server at metasploitable.localdomain Port 80</address>
</body></html>
root@kali:/home/duelxk#
```

Fonte: Elaborada pelo autor, captura de tela

Entretanto, há um possível problema já que na captura de banner o alvo pode ter mudado o serviço, e com isso, a informação pode ser falsa, ainda que tal possibilidade seja mínima, enfim para evitar essas situações pode-se utilizar o *fingerprint* (impressão digital).

“Através de recursos como identificar a pilha do TCP/IP através da análise do TTL (*Time to Live*) no recebimento de pacotes ICMP, alguns softwares conseguem diferenciar um sistema do outro. A implementação dessa pilha é diferente de um Linux para um Windows, por exemplo.” (Segredos do Hacker Ético, 2014, v. 5, p. 86).

O *fingerprint* irá identificar o banner, mesmo com possíveis alterações feitas, bem como o sistema operacional que está sendo rodado, Figura 9. Além disso, pode-se observar que ele é capaz de detectar o uso de uma máquina virtual “*Oracle VirtualBox virtual NIC*”.

Figura 9 - Captura de banner com *fingerprint* em máquina virtual

```

root@kali:/home/duelxk# nmap -O 192.168.56.221
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-13 19:42 -03
Nmap scan report for ie6winxp (192.168.56.221)
Host is up (0.00040s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2869/tcp   open  icslap
3389/tcp   closed ms-wbt-server
MAC Address: 08:00:27:9A:6C:07 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 2000|XP
OS CPE: cpe:/o:microsoft:windows_2000::sp3 cpe:/o:microsoft:windows_2000::sp4 cpe:/o:microsoft:windows_xp::sp1 cpe:/o:microsoft:windows_xp::sp2
OS details: Microsoft Windows 2000 SP3/SP4 or Windows XP SP1/SP2, Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.72 seconds
root@kali:/home/duelxk#

```

Fonte: Elaborada pelo autor, captura de tela

É necessário verificar, também os serviços que estão rodando por trás destas portas, utilizando o “nmap -sV IP-desejado ou arquivo com grupo de IPs”, Figura 10.

Figura 10 - Serviços rodando por trás das portas descobertas

```

root@kali:/home/duelxk# nmap -sV 192.168.56.221
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-13 19:44 -03
Nmap scan report for ie6winxp (192.168.56.221)
Host is up (0.00045s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE      VERSION
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds  Microsoft Windows XP microsoft-ds
2869/tcp   open  http          Microsoft HTTPAPI httpd 1.0 (SSDP/UPnP)
3389/tcp   closed ms-wbt-server
MAC Address: 08:00:27:9A:6C:07 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.85 seconds
root@kali:/home/duelxk#

```

Fonte: Elaborada pelo autor, captura de tela

Repare que na Figura 10 o nmap mostra as portas, o estado destas, os serviços que estão em execução e, principalmente, a versão utilizada neles. A versão será muito importante na realização da invasão caso optemos por algum destes. Para poder analisar todas as situações acima utiliza-se o comando -A.

Caso não sejam identificadas as máquinas mesmo estando ligadas utiliza-se o - Pn.

De acordo com Assunção (2014), algumas das formas para se evitar ou amenizar o problema são:

- Nunca deixar páginas não indexadas no servidor *Web*.
- Utilizar, sempre que possível, serviços de rede criptografados (como SSH e SFTP) para evitar a captura de banners.
- Se possível, configurar o servidor SMTP para não informar quando o usuário não existe e também não enviar resposta quando um e-mail destinado a uma conta inexistente chegar.
- Desabilitar recursos que permitam sessão nula, como o compartilhamento Netbios IPC\$.

3.3.2 Vulnerabilidades de um serviço

Para consultas de vulnerabilidades voltadas a um específico serviço basta fazer uma pesquisa manual, dando qual o serviço, sua versão, usuários e recursos no sistema, Assunção (2014). “Quando já sabemos a versão exata dos servidores que estão rodando, a pesquisa manual costuma ser a mais eficiente.” (Segredos do Hacker Ético, 2014, v. 5, p. 93).

Um ótimo site para isso é o *SecurityFocus*, que fornece informações sobre o problema e o mais importante, a solução para a vulnerabilidade. Outro site que pode ser utilizado é o *Exploit Database*, que tem as ferramentas para exploração, mostrando como realizar o ataque. O problema destes sites é que seria necessário procurar uma a uma as vulnerabilidades e é aí que entram os *scanners* genéricos de falhas, que são programas capazes de analisar falhas em vários protocolos de rede analisando banco de dados, servidores FTP, POP3, SSH, entre outros. Consistem na ideia de um teste de penetração de caixa branca ou cinza no qual já temos certo conhecimento do alvo. Hoje um dos

scanners de vulnerabilidades mais conhecido é o Nessus, entretanto é uma ferramenta paga. Por isso uma opção mais viável é o Openvas.

Para o escaneamento web existem várias opções de softwares, com isso é possível realizar o “*crawler*” (tipo de escaneamento), que irá escanear cada subpágina e encontrar possíveis falhas. Usando o OWASP ZAP como exemplo pode-se identificar problemas como injeção de código, SQL *injection*, cmd *injection*, *cross site scripting* e vários outros.

Para sistemas com *wordpress* o wpscan, pois pode ser utilizado como a maioria dos sites hoje em dia roda em *wordpress*, é possível utilizar as funções do wpscan para descobrir páginas, versões, plugins, usuários e até senhas.

3.4 Privacidade (mascarando seu IP)

Para não deixar rastros, é necessário mascarar o endereço IP durante os testes, para isso pode-se utilizar servidores proxy ou VPN.

Uma prática muito conhecida é o uso do TOR, uma ferramenta P2P, ele monta diversos circuitos de endereçamentos que permita que se passe por vários endereços de IP antes de chegar no seu endereço final. É muito conhecido devido a *Deep Web* que são sites *Onion*.

Uma das formas mais simples é com um notebook utilizar um ponto de wi-fi público como de restaurantes ou shoppings, juntamente com as formas citadas abaixo.

O *ProxyChains* é um serviço que permite o encadeamento de outros servidores *proxy* em sequência para utilizar qualquer aplicação necessária, desde um navegador até alguma outra aplicação. Sinha (2017) defende que o nome sugere seu verdadeiro significado, para manter o anonimato, precisa-se de vários proxies. Por trás destes proxies, pode-se esconder a verdadeira identidade, entretanto o Kali Linux fornece uma forma de configuração através do arquivo “*prxychain.conf*” para que se possa configurar seu próprio tunelamento utilizando o TOR.

Figura 11 - Configuração do *ProxyChains*

```
# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted
#
dynamic_chain
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
```

Fonte: Livro Beginning Ethical Hacking with Python, (2017, v. 1, p. 179)

A utilização do comando “*proxychains*”, antes de um *nmap* por exemplo, mascarar o IP durante a varredura, ocasionando um tempo maior de execução pois ele irá passar por vários IPs antes de retornar o resultado da varredura.

VPN é uma forma de criar pontes de ligação entre dispositivos via internet, mantendo a comunicação mais segura e tornando a interceptação mais difícil. Pode-se utilizar uma VPN gratuita.

“Uma pluralidade de conexões é estabelecida com os respectivos nós pares na rede. Cada um dos pares é configurado para fornecer funções de aprimoramento de desempenho para a rede por meio das conexões. A abordagem permite que a seleção de uma das conexões, inclua um túnel seguro para o par associado à conexão solicitada.” (Dillon et al, 2003, v. 1, p. 1).

3.5 Exploração

A partir da fase de exploração necessita-se analisar quais dados foram colhidos anteriormente e considerar qual a forma mais viável de fazer a penetração. De acordo com Assunção (2014) para isso considera-se três tipos de ferramentas que podem abrir caminhos para outros ataques, o antivírus, o *firewall* e o IDS.

Um passo muito importante é impedir um programa malicioso de ser detectado pelo antivírus, onde uma detecção poderia comprometer todos os processos realizados anteriormente. Vale lembrar que, atualmente, a maioria

dos computadores possui um antivírus, que quantidade de vírus existentes é grande e que pelo volume cada vez maior de arquivos a serem analisados, o antivírus necessita de um processo rápido de identificação. Isso é feito através da análise da sequência de caracteres contidos no arquivo, caso haja algo contido em seu banco de dados, ele irá identificá-lo como malicioso.

Há também a necessidade de burlar o *firewall*, pois de nada adianta fazer um *backdoor* ou *shell* reverso e ser barrado pelos mecanismos de segurança. Normalmente o *firewall* é configurado para ter uma ótima proteção de fora para dentro, muitas vezes, deixando passar conexões a servidores *Web*, de correio e serviço remoto que possua autenticação segura.

IDS é um sniffer com regras simples e baseadas em texto puro (ASCII). “O objetivo de técnicas anti-IDS é modificar um pedido de tal maneira que os sistemas de detecção fiquem confusos, mas o servidor *Web* ainda conseguira entender o que estamos pedindo.” (Assunção, 2014, p. 142)

Um software muito utilizado nos testes de penetração é o NetCat, que permite que haja conexão como cliente ou servidor e independente da necessidade, servirá como uma ponte. Por meio dele é possível criar uma *bind shell*, caso haja um *backdoor* (porta dos fundos) um sistema pode se conectar ao outro com um *port scan*. Como a possibilidade de isso acontecer é pequena, entra-se no conceito de conexão reversa e caso não haja IP público o atacante pode estabelecê-lo. Caso a vítima tenha o NetCat instalado o atacante conseguirá copiar qualquer tipo de arquivo da máquina dela para a dele.

Outra opção é utilizar o NCAT que basicamente é um Netcat melhorado, permite o uso do SSL para burlar IPs e sistemas de detecção em geral.

“Se não conseguirmos nos conectar a alguém dentro dessa barreira, podemos fazer o contrário: o próprio programa se conectará ao nosso sistema de volta, burlando, assim, a maioria dos filtros de segurança e permitindo que acessemos endereços privados”. (Segredos do Hacker Ético, 2014, v. 5, p. 132).

O reverse *shell* ou *shell* reverso força a conexão do alvo ao atacante. Este se conectará a uma porta local aberta esperando a entrada do alvo, o que o torna

mais possível passar por um *firewall* de modo que ele se conecte de forma independente.

3.5.1 Metasploit

O *metasploit* foi criado pela RAPID7 para divulgar informações relacionadas a *exploits* (vulnerabilidades) com a intenção de facilitar os testes de penetração e o desenvolvimento de detecção de invasores. Ele irá encontrar o endereço na memória durante o *overflow* e executar o *payload*, assim, lhe dando acesso ao sistema operacional.

“O *metasploit* Framework é um projeto *open source* desenvolvido para diversos sistemas, como Linux, Windows e *FreeBSD*. É um ambiente completo para escrever, testar e utilizar códigos de exploits. Ele nos fornece uma plataforma sólida para desenvolvimento de *shellcode* e pesquisa de vulnerabilidades. O *Framework* é desenvolvido em *Perl*, *C*, *Assembler* e *Python*. Essa ferramenta possui uma poderosa interface em modo texto e também uma interface *Web*”. (Segredos do Hacker Ético, 2014, v. 5, p. 207).

Antigamente era necessário ter vasto conhecimento em algumas linguagens específicas como *Assembly* por exemplo para gerar estes acessos. O *metasploit* possibilitou uma mudança no *penetration test* justamente por estes fatores, seus scripts são baseados em *python*, o que torna muito mais fácil desenvolver uma extensão.

De acordo com Kennedy et al (2011) o *Metasploit* é um *software* de código aberto gratuito, com muitos colaboradores na comunidade de segurança. Disponibilizado em duas versões, a *Community* e a *Pro*, para quem irá trabalhar com isso há muitas vantagens em adquirir a *Pro* devido à variedade de recursos.

Kennedy et al (2011) defende que *exploit* é uma sequência de comandos dados por um *software* para tirar proveito de uma falha ou vulnerabilidade. Seu objetivo é causar um comportamento imprevisto na execução de um *software* ou *hardware*. Para fins maléficos este pode dar a um *cracker* o controle do SO, permitindo a conexão indevida e provavelmente prejudicial.

O Msfconsole é uma das ferramentas mais flexíveis, rica em recursos e bem suportada dentro do *Framework* e também fornece uma interface multifuncional útil para quase todas as opções e configurações disponíveis no *Framework*. Pode ser usado para tudo, incluindo iniciar um *exploit*, carregar módulos auxiliares, executar enumeração, criar *listeners* (ouvintes) e até mesmo executar uma exploração em massa na rede inteira. (Kenedy et all, 2011). Depois de feita a conexão tem-se total acesso para baixar, enviar, deletar arquivos, *webcams*, *keyloggers* e muitas outras opções de acordo com a necessidade.

Ao contrário dos *hackers*, os *crackers* são pessoas que quebram a segurança para fins maliciosos ou até mesmo criminosos. Embora o termo *cracker* não seja muito conhecido e diferente do que a mídia prega há uma grande diferença entre os dois, pois a prática de “crackear” se caracteriza crime, devido ao seu conceito malicioso, “Eles são os únicos responsáveis por roubar identidades online e explorar fraquezas no sistema. Os *crakers* fazem uso malicioso do conhecimento dos *hackers*”. (Hackers not Crackers, 2015, v. 10, p. 3).

Conforme Kennedy et all (2011), *payload* é o *software* que permite controlar um sistema depois de explorado, este é fornecido pelo *exploit* e é um código malicioso que vai ser rodado para ter acesso ao sistema operacional após corrompê-lo. O mais popular deles é o *Meterpreter*.

“Um *shell* reverso é um *payload* que cria uma conexão da máquina alvo de volta ao invasor como um prompt de comando do Windows, enquanto um *shell* de ligação é um *payload* que liga um prompt de comando a uma porta de escuta na máquina alvo, que o invasor pode conectar” (Metasploit The Penetration Tester's Guide, 2011, v. 1, p. 8)

A exploração “*client-side*” tem foco no cliente e não no servidor, onde pode-se focar na criação e utilização de *backdoor*, cavalo de Tróia, determinadas falhas *web*, arquivos pdf, Java e vários outros tipos. Para que a vítima abra o arquivo como de *backdoor* por exemplo, uma das formas mais fáceis é utilizar engenharia social de modo que consiga convencer o alvo de que ele precisa abrir o arquivo.

No caso de aparelhos *Android* fica mais fácil ainda pois boa parte das pessoas não utilizam antivírus nestes dispositivos. Por isso é tão importante tomar muito cuidado ao baixar aplicativos em sites de terceiros.

Vários navegadores de internet têm diversas vulnerabilidades descobertas através dos anos, caso encontre alguma destas pode-se embutir um *payload* neste computador. É importante que haja uma migração do processo ao executar o arquivo onde está o *payload*, assim mantendo a conexão mesmo depois do fechamento deste.

O *Meterpreter* é uma *payload* do tipo *staged* muito usada que usa injeção de DLL na memória estendendo-se para rede em tempo de execução. É uma ferramenta para ser usada depois de obter acesso ao alvo, e que permite fazer várias coisas como *dump* de *hashes*, migrar processos, parar antivírus, entre outras funções. (Kennedy et al, 2011).

3.5.2 Vulnerabilidades Web

As vulnerabilidades *web* são erros de um *firewall* de aplicação web e no servidor, os mais explorados hoje em dia são do tipo *buffer overflow*, que em sua maioria podem dar privilégios de administrador. Os tipos de vulnerabilidade *web* mais críticos são: *Cross Site Scripting* (XSS) e Falhas de injeção.

XSS ocorrem quando a aplicação obtém informações fornecidas que voltam sem validação ou codificação, permitindo a execução de *scripts*.

“É uma técnica que visa roubar *cookies* de usuários através de seus navegadores. Geralmente o invasor injeta comandos HTML e Java *Script* em alguma função, conseguindo obter sessões de usuários mesmo sem ter autorização para isso”. (Segredos do Hacker Ético, 2014, v. 5, p. 97).

Falhas de injeção, com ênfase em SQL *Injection*, são comuns em aplicações *web* e enganar o interpretador, fazendo-o executar comandos maliciosos. A injeção de SQL pode ocorrer em dois métodos o *GET* e *POST*, este é considerado a pior vulnerabilidade e mais frequente em aplicações web.

“Esse tipo de falha não é do servidor de banco de dados, e sim, de um programa feito para interagir com esse banco. Seja ASP, PHP, JSP ou qualquer outro tipo de programação para a *Web*, se o programa não interpretar corretamente certos caracteres como (/) e aspas simples (‘), eles podem ser usados para injetar comandos naquele sistema, burlando sistemas de login e senha e, muitas vezes, fornecendo acesso completo ao banco de dados”. (Segredos do Hacker Ético, 2014, v. 5, p. 95).

Através de softwares como o *BurpSuite* é possível escanear e injetar scripts ou alterar informações relevantes para realizar estes procedimentos.

A ideia de uma *webshell* é construir um arquivo no qual se consiga jogá-lo dentro do servidor *web*, assim executando comandos dentro um servidor vulnerável. Geralmente o programa cria *backdoors* para reinstalar o *webshell* periodicamente, caso este seja descoberto ou removido

3.5.3 Senhas

Assunção (2014) diz que nunca se deve ter uma conta com senha fácil ou serviço que dependa de autenticação sem senha, onde o ideal é nunca utilizar senhas do tipo:

- Data de nascimento
- Nome de familiar ou amigo
- Local de trabalho
- Nome de personagens e filmes
- Outros nomes conhecidos

Mesmo acrescentando números nas citadas acima, ainda há a possibilidade de invasão, por isso deve-se considerar os seguintes padrões para montar uma senha:

- Letra maiúsculas e minúsculas
- Números

- Caracteres estendidos como /!@#\$%`&*()[{};
- Tamanhos de no mínimo 10 caracteres

Logo uma senha considerada segura poderia ser “EhAV9\$29x}@”. As Figuras 12, 13, 14 e 15 a seguir listam algumas senhas padrões de diferentes fabricantes.

Figura 12 - Hardwares e senhas padrões

Fabricante	Modelo	Versão do SO	Login	Senha
APC	qualquer	Firmware Pri	apcuser	apc
APC	MasterSwitches	-	apc	apc
Apple	Airport	1.1	none	public
Apple	Network Assistant	3.X	None	xyzy
Arrowpoint	qualquer?	-	admin	system
Ascend	todos TAOS models	todos	admin	Ascend
AT&T	Starlan SmarHUB	9.9	N/A	manager
AWARD	qualquer BIOS	-	AWARD SW	-
Axent	NetProwler manager	WinNT	administrator	admin
Axis	200 V1.32	-	admin	-
Axis	2100 Network Camera	Linux ETRAX	root	pass
Axis	NPS 530	5.02	root	pass
Axis	StorPoint CD100	4.28	root	pass
Bay Networks	ASN / ARN Routers	qualquer	manager	manager
Bay/Nortel Networks	Accelar 1xxx switches	qualquer	rwa	rwa
Bay/Nortel Networks	Remote Annex 2000	qualquer	admin	IP address
BEA	Weblogic	5.1	system	weblogic
Bintec	todos Routers	qualquer	admin	bintec
Borland	Interbase	qualquer	polictodosy	correct
Borland/Inprise	Interbase	qualquer	SYSDBA	masterkey
Brocade	Silkworm	-	admin	password
Buffalo/MELCO	AirStation WLA-L11	-	root cannot be changed	no password by default
Cabletron routers and switches	*	*	vazio	vazio
Cayman	3220-H DSL Router	GatorSurf 5.	qualquer	-
Cisco	3600	12	bumhole	sniffer
Cisco	-	12	turd	burgular
Cisco	qualquer	aqualquer IOS	no default login	no default password
Cisco	qualquer Router and Switch	10 thru 12	cisco	cisco
Cisco	ConfigMaker Software	qualquer	n/a	emaker
Cisco	IDS netranger	-	root	attack
Cisco	MGX	*	superuser	superuser
Cisco	N/A	N/A	privadmin	privadmin
Cisco	Net Ranger 2.2.1	Sol 5.6	root	attack
Cisco	Network Registrar	3	admin	changeme
Cisco	VPN 3000 Concentrator	-	admin	admin
Cisco	xxx	12	rob's	knob
Cobalt	RaQ * Qube*	qualquer	admin	admin
Comersus Shopping Cart	3.2	Win 95/98/NT	admin	dmr99
Compaq	Insight manager	-	administrator	administrator
Compaq	Insight manager	-	operator	operator
Compaq	Management Agents	todos	administrator	none
Coyote-Point	Equaliser 4	Free BSD	eqadmin - Serial port only	equalizer
Coyote-Point	Equaliser 4	Free BSD	root - Serial port only	-
Coyote-Point	Equaliser 4	Free BSD	look - Web Browser only	look
Coyote-Point	Equaliser 4	Free BSD	Touch - Browser only	touch
Cyclades	MP/RT	-	super	surt
Dell	Powerapp Web 100 Linux	RedHat 6.2	root	powerapp
Dell	PowerVault 35F	-	root	calvin
Dell	PowerVault 50F	WindRiver	root	calvin
Digiboard	Portserver 8 & 16	qualquer	root	dbps
DLINK	DI 106	winnt	administrator	@*nigU^D ha.;
DLINK	DI-206 ISDN router	1.*	admin	admin

Fonte: Livro Segredos do Hacker Ético, 2014, p. 226

Figura 13 - Hardwares e senhas padrões

Fabricante	Modelo	Versão do SO	Login	Senha
Dupont Digital Water Proofer	Sun Sparc	qualquer	root	par0t
Elron	Firewtodos	2.5c	hostname/ ip address	sysadmin
Ericsson	ACC	-	netman	netman
Ericsson formerly ACC	qualquer router	todos	netman	netman
Extended Systems	ExtendNet 4000 / Firewtodos	todos versions	admin	admin
Extended Systems	Print Servers	-	admin	extendnet
General Instruments	SB2100D Cable Modem	-	test	test
gonet	-	-	fast	abd234
Hewlett Packard	HP Jetdirect todos Models	qualquer	none	none
Hewlett Packard	MPE-XL	-	HELLO	manager.SYS
Hewlett Packard	MPE-XL	-	HELLO	MGR.SYS
Hewlett Packard	MPE-XL	-	HELLO	FIELD SUPPORT
Hewlett Packard	MPE-XL	-	MGR	CAROLIAN
Hewlett Packard	MPE-XL	-	MGR	CCC
Hewlett Packard	MPE-XL	-	OPERATOR	COGNOS
Hewlett Packard	MPE-XL	-	manager	HPOFFICE
IBM	2210	RIP	def	trade
IBM	-	OS/400	QSECOFR	QSECOFR
IBM	AS/400	-	qsysopr	qsysopr
IBM	AS/400	-	qpgrmr	qpgrmr
IBM	AS/400	OS/400	QUSER	QUSER
IBM	AS400	qualquer	QSECOFR	QSECOFR
IBM	AS400	-	QSRVBAS	QSRVBAS
IBM	AS400	-	QSRV	QSRV
IBM	DB2	WinNT	db2admin	db2admin
IBM	LAN Server / OS/2	2.1, 3.0, 4.	username	password
IBM	Lotus Domino Go WebServer net.commerce edition	qualquer	webadmin	webibm
IBM	NetCommerce PRO	3.2	ncadmin	ncadmin
IBM	RS/6000	AIX	root	ibm
Imperia Software	Imperia Content Managment System	Unix/NT	superuser	superuser
Ipswitch	Whats up Gold 6.0	Windows 9x a	admin	admin
Janta sales	254	Compaq	janta sales	janta211
Juniper	todos	Junos 4.4	root	none
Lantronix	LPS1-T Print Server	j11-16	qualquer	system
Lantronix	LSB4	qualquer	qualquer	system
LGIC	Goldstream	2.5.1	LR-ISDN	LR-ISDN
Linksys	BEFSR41	-	vazio	admin
Linksys	BEFSR71 OR 4	Standalone R	vazio	admin
Livingston	Livingston officrouter	-	!root	vazio
Livingston	Livingston portmaster2/3	-	!root	vazio
Lucent	AP-1000	-	public	public
Lucent	Cajun Family	-	root	root
Lucent	Packetstar PSAX	-	readwrite	lucenttech1
Lucent	Portmaster 2	-	!root	none
Lucent	Portmaster 3	unknown	!root	!ishtar
MacSense	X-Router Pro	-	admin	admin
Microcom	hdms	unknownen	system	hdms
Microsoft	NT	-	-	start
Microsoft	NT	4	free user	user
Microsoft	NT	4	free user	user
Microsoft	SQL Server	-	sa	-
Microsoft	Windows NT	todos	administrator	-
Microsoft	Windows NT	todos	Guest	-
Microsoft	Windows NT	todos	Mail	-
Microsoft	Windows NT	4	pkoolt	pkooltPS
Motorola	Motorola-Cablerouter	-	cablecom	router

Fonte: Livro Segredos do Hacker Ético, 2014, p. 227

Figura 14 - Hardwares e senhas padrões

Fabricante	Modelo	Versão do SO	Login	Senha
Multi-Tech	RASExpress Server	5.30a	guest	none
Nanoteq	NetSeq firew todos	*	admin	NetSeq
NetApp	NetCache	qualquer	admin	NetCache
Netgear	RT311	qualquer	admin	1234
Netgear	RT311/RT314	-	admin	1234
Netgear	RT314	qualquer	admin	1234
Netopia	R7100	4.6.2	admin	admin
Netscreen	-	-	net screen	net screen
Netscreen	NS-5, NS10, NS-100	2	net screen	net screen
Nokia - Telecom NZ	M10	-	Telecom	Telecom
Nortel	Contivity Extranet Switches	2.x	admin	setup
Nortel	Meridian 1 PBX	OS Release 2	0	0
Nortel	Norstar Modular ICS	qualquer	**admin **23646	admin 23646
Nortel	Norstar Modular ICS	qualquer	**CONFIG 266344	CONFIG 266344
Nortel	Shasta	qualquer	admin	admin
Novell	NetWare	qualquer	guest	-
Novell	NetWare	qualquer	PRINT	-
Novell	NetWare	qualquer	LASER	-
Novell	NetWare	qualquer	HPLASER	-
Novell	NetWare	qualquer	PRINTER	-
Novell	NetWare	qualquer	LASERWRITER	-
Novell	NetWare	qualquer	POST	-
Novell	NetWare	qualquer	MAIL	-
Novell	NetWare	qualquer	GATEWAY	-
Novell	NetWare	qualquer	GATE	-
Novell	NetWare	qualquer	ROUTER	-
Novell	NetWare	qualquer	BACKUP	-
Novell	NetWare	Arserve	CHEY ARCHSVR	WONDERLAND
Novell	NetWare	qualquer	WINDOWS PASS THRU	-
ODS	1094 IS Chassis	4.x	ods	ods
Optivision	Nac 3000 & 4000	qualquer	root	mpegvideo
Oracle	7 or later	-	system	manager
Oracle	7 or later	-	sys	change on inst todos
Oracle	7 or later	qualquer	Scott	Tiger
Oracle	8i	8.1.6	sys	change on inst todos
Oracle	8i	todos	internal	oracle
Oracle	Internet Directory Service	qualquer	cn=orcladmin	welcome
Oracle Co.	Database engines	todos	sys	change on inst todos
Osicom Datacom	Osicom Datacom	-	sysadm	sysadm
Pandastel	EMUX	todos	admin	admin
RapidStream	RS4000-RS8000	Linux	rsadmin	rsadmin
Research Machines	Classroom Assistant	Windows 95	manager	changeme
Rodopi	Rodopi billing software 'AbacBill' sql database	-	rodopi	rodopi
Securicor3NET	Cezanne	qualquer	manager	friend
Securicor3NET	Monet	qualquer	manager	friend
SGI	todos	todos	root	n/a
SGI	Embedded Support Partner	IRIX 6.5.6	administrator	Partner
SGI	Embedded Support Partner	IRIX 6.5.6	administrator	Partner
SGI	IRIX	todos	lp	lp
SGI	IRIX	todos	OutOfBox, demos, guest, 4DGifts	none by default
SGI	IRIX	todos	EZsetup	-
Shiva	AccessPort	qualquer	hello	hello
Shiva	qualquer	-	Guest	vazio
Soho	nbg800	todos	admin	1234
SonicW	Firew todos Device	-	admin	password
Sun	-	SunOS 4.1.4	root	-
Surecom	ep3501/3506	todos	admin	surecom

Fonte: Livro Segredos do Hacker Ético, 2014, p. 228

Figura 15 - *Hardwares e senhas padrões*

Fabricante	Modelo	Versão do SO	Login	Senha
Tekelec	Eagle STP	-	eagle	eagle
Telebit	netblazer 3.*	-	setup/smp	setup/nopasswd
Terayon	-	6.29	admin	nms
Terayon	TeraLink 1000 Controller	-	admin	password
Terayon	TeraLink 1000 Controller	-	user	password
Terayon	TeraLink Getaway	-	admin	password
Terayon	TeraLink Getaway	-	user	password
Tiara	Tiara	-	tiara	tiaranet
Tubas	-	SCO	haasadm	lucy99
TopLayer	AppSwitch 2500	qualquer	siteadmin	toplayer
Toshiba	TR-650	V2.01.00	admin	tr650
TrendMicro	ISVW VirusW todos	qualquer	admin	admin
Trintech	eAcquirer App/ Data Servers	-	t3admin	Trintech
Ulla ka partha	Gand mara	Gandoo	Bhosda	Lund
USR	TOTALswitch	qualquer	none	amber
Vina Technologies	ConnectReach	3.6.2	none	none
Webmin	Webmin	qualquer Unix/Lin	admin	-
Webramp	410i etc.	-	wradmin	trancell
Wireless Inc.	WaveNet 2458	n/a	root	rootpass
Xylan	Omnistack 1032CF	3.2.8	admin	password
Xylan	Omnistack 4024	3.4.9	admin	password
Xylan	Omniswitch	3.1.8	admin	switch
Xyplex	mx-16xx	-	setpriv	system
Zyxel	641 ADSL	-	-	1234
Zyxel	ISDN Router Prestige 100H	-	-	1234
Zyxel	ISDN-Router Prestige 1000	-	-	1234
Zyxel	prestige 128 modem-router	qualquer	-	1234
Zyxel	prestige 300 series	zynos 2.*	-	1234

Fonte: Livro Segredos do Hacker Ético, 2014, p. 229

Vale ressaltar dois pontos importantes, Hash e Ataques de Força Bruta (*Bruteforce*). Podemos dividir isto em duas categorias, ou seja, quebra de senhas online onde há a necessidade de conectar-se no serviço e usar um *software* para realizar a força bruta, e o *offline* que é quando há um hash por exemplo e não se sabe qual o texto por trás dele. Para isso precisa-se criar uma *wordlist* (lista de palavras) ou baixar alguma via internet. (Assunção, 2014).

O tipo de ataque denominado como força bruta trata-se de uma técnica de sucessivas tentativas de acertar a combinação de senha e usuário para assim conseguir acesso no que se deseja.

O processo de força bruta remoto não é dos mais eficientes, a menos que a pessoa utilize uma senha muito simples, no entanto, não deve ser descartado. Já o força bruta local é mais eficiente devido a quantia de senhas testadas por segundo, “na local você pode conseguir até 3,4 milhões por segundo, dependendo da capacidade da máquina e do algoritmo de encriptação.” (Segredos do Hacker Ético, 2014, v. 5, p. 235).

Hash é uma defesa secundária onde há necessidade de uma autenticação da informação para validar a senha, isto se dá a comparação de *strings*.

“Para tentar descobrir qual é a senha criptografada, os programas de *bruteforce* usam um método interessante: eles codificam a informação a ser testada com o mesmo algoritmo e testam os dois. Se coincidirem, a senha foi descoberta”. (Segredos do Hacker Ético, 2014, v. 5, p. 235).

3.5.4 Engenharia Social

Quando não temos falha nenhuma no sistema precisamos partir para vulnerabilidade humana. Hadnagy e Fincher (2015) defendem que uma estatística afirma que mais de 60 por cento de todos os ataques tiveram o “fator humano” como ponto crucial ou a parte principal do ataque. A engenharia social nada mais é do que a arte de enganar, o engenheiro social utiliza técnicas para explorar vulnerabilidades humanas, como por exemplo o *pishing* e *fake mail*.

De acordo com Hadnagy (2011) a *Webster's Dictionary* define social como “de ou pertencente à vida, bem-estar e relações de seres humanos em uma comunidade”, também define engenharia como “a arte ou ciência de fazer aplicação prática do conhecimento das ciências puras, como física ou química, como na construção de motores, pontes, edifícios, minas, navios e fábricas químicas ou dispositivos habilidosos ou engenhosos”. Combinando essas duas definições, é fácil compreender que a engenharia social é a arte ou ciência de manobrar habilmente os seres humanos para agirem em algum aspecto de suas vidas. Essa definição amplia os horizontes dos engenheiros sociais em todos os lugares, pois a engenharia social é usada na vida cotidiana, na forma como as crianças fazem seus pais cederem as suas demandas, na interação professores com seus alunos, médicos, advogados ou psicólogos obtêm informações de seus pacientes ou clientes. Enfim, pode-se dizer que uma verdadeira definição de engenharia social é o ato de manipular uma pessoa para realizar uma ação que pode ou não ser interessante para o “alvo”.

Este é um dos fatores mais importantes em uma organização pois trata de uma situação simples, onde qualquer pessoa pode ligar para a empresa se passando por um funcionário e pedir, a quem atendeu a ligação, que digite um comando ou abra um arquivo enviado sem que o alvo tenha noção alguma do que está fazendo.

Hadnagy e Fincher (2015) dizem que *phishing* é uma mensagem enganosa tentando utilizar artifícios de engenharia social como manipular a curiosidade, confiança e simpatia da pessoa, capaz de anexar um *malware* (software malicioso) dentro de um *e-mail* ou também um *link* para um *site* ilegítimo. Os invasores dedicam tempo para pesquisar os alvos e criar mensagens pessoais e relevantes. O objetivo dos *phishers* tende a ser bem típicos, dinheiro ou informações (que geralmente se converte em dinheiro).

Supondo que em um serviço realizado não seja encontrada nenhuma vulnerabilidade, é preciso que um funcionário da empresa execute um *meterpreter* reverso e a partir disso, é necessário usar métodos para convencê-lo a executar o arquivo ou programa. Para isso pode-se esconder um executável dentro de um *link* ou algo do tipo como pdf, rar e outros arquivos.

O *fake mail* (e-mail falso) nada mais é do que enviar um *e-mail* se passando por outra pessoa, este pode ser enviado pelo domínio que quiser. Assunção (2014) defende que como a maioria dos usuários é leiga e nunca irá conferir o endereço do remetente para ver se é real, essa se torna uma maneira muito eficiente. Existem vários sites com essa função, que permite gerar o *e-mail* desejado.

3.5.5 Rede Local

Quando há acesso a uma rede local independente se é cabeada ou *wireless*, pode-se redirecionar o tráfego, realizar o envenenamento das máquinas e várias funções para obter informações sobre os usuários desta rede.

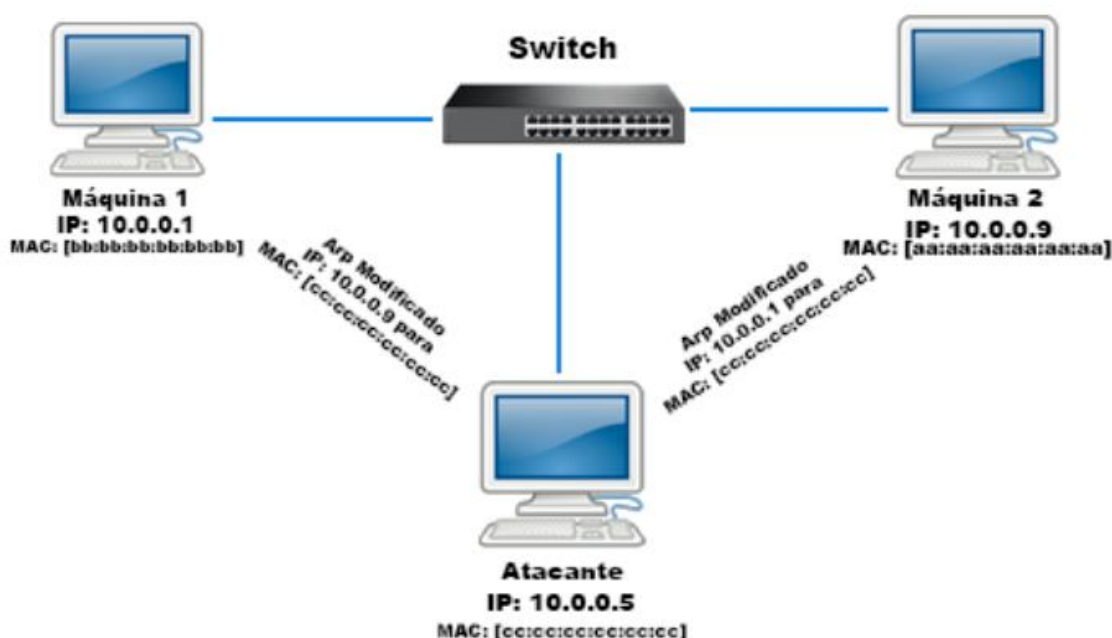
De acordo com Whalen (2001) um computador conectado a uma LAN (*Local Area Network* ou Rede Local) possui dois endereços. Um é o da placa de

rede, denominado endereço MAC, e que em teoria é um endereço globalmente único e imutável, armazenado na própria placa de rede, pois os endereços MAC são necessários para que o protocolo *Ethernet* possa enviar dados para frente e para trás, independentemente de quaisquer protocolos de aplicação usados sobre ele. A Ethernet constrói “frames” de dados, consistindo em blocos de 1500 bytes, cada quadro possui um cabeçalho *Ethernet*, contendo o endereço MAC da fonte e do computador de destino. O Segundo é o endereço IP, que é um protocolo usado por aplicativos, independentemente de qualquer tecnologia de rede que opere sob ele. Cada computador em uma rede deve ter um endereço IP exclusivo para se comunicar.

IP e *Ethernet* devem funcionar juntos, o IP se comunica construindo “pacotes” que são semelhantes a quadros, mas têm uma estrutura diferente, esses pacotes não podem ser entregues sem a camada de enlace.

O ARP *spoofing* (falsificação de ARP) é uma das técnicas para redirecionar o tráfego e capturar o que os usuários estão fazendo na rede.

Figura 16 - ARP Poisoning



Fonte: Elaborada pelo autor

O ARP é responsável por traduzir os endereços IPS para os físicos (MAC). Na imagem, a máquina 1 não sabe o endereço físico da máquina 2 e irá mandar

um *broadcast* na rede onde a máquina 2 é que irá detectar e responder com o próprio endereço MAC. Agora entrando com o ARP *poisoning* tem a máquina do atacante que também irá receber o pedido do ARP e irá se passar pela máquina 2, o mesmo acontecerá com ela, dando a entender que o atacante é a máquina 1.

3.5.6 Wireless

Assunção (2014) relata que devido ao custo decrescente e as facilidades de acesso, as redes *wireless* estão se tornando cada vez mais acessíveis e fáceis de serem configuradas, pelo fato de muitas pessoas não terem uma noção básica de segurança. “Atualmente o Wi-Fi 802.11 é o tipo de rede LAN sem fio mais utilizada. Existem vários programas interessantes para farejar ou quebrar senhas de redes wireless como: AirSnort, Kismet, WireShark, etc.” (Segredos do Hacker Ético, 2014, v. 5, p. 246).

Este processo requer um dispositivo para capturar redes *wireless*, apesar de notebooks possuírem placas internas o ideal é utilizar um *hardware* externo que possua antena, o que aumentará o alcance, facilitando a conexão. Este dispositivo deverá estar no modo de monitoração para poder enxergar as redes disponíveis.

3.5.6.1 Quebra de chaves

Os protocolos de segurança evitam que terceiros se conectem a sua rede sem fio, já que diferente das redes cabeadas há uma grande insegurança nas redes sem fio. Considerando que grande parte dos roteadores oferecem diferentes tipos de opções de segurança para a rede, em caso de má escolha o resultado será uma rede lenta e menos segura.

Bulbul, Batmaz e Ozel (2008) dizem que quando o WEP está ativo cada pacote 802.11 é criptografado separadamente, com um fluxo de criptografia RC4 gerado por uma chave RC4 de 64 bits. Essa chave é composta por um vetor de inicialização (IV) de 24 bits e uma chave WEP de 40 bits. Como sua tecnologia

é um pouco ultrapassada, cada arquivo WEP tem este vetor, porém depois de uma grande quantidade de arquivos ele começa a repetir o padrão. Então se for possível capturar um grande número de “*data*” (dados) pode-se simplesmente deduzir a chave. Existem várias ferramentas que fazem este processo.

Sari e Karay (2015) citam que o protocolo WPA foi introduzido em 2003 pela *Wi-Fi alliance* (Aliança *Wi-Fi*) para tentar eliminar ou superar os problemas do WEP, resolvendo as falhas de criptografia. A WPA (*Wi-Fi Protected Access*) traz a encriptação 256 bits e uma segurança maior para as redes, além de um sistema de análise para verificar invasões, que é uma melhoria significativa do WEP.

De acordo com Alblwi e Shujaee (2017) o WPA2 (*Wi-Fi Protected Access II*) também conhecido como IEEE 802.11i, que é o resultado da falha dos protocolos anteriores. A camada 3 (rede) era vulnerável a ataques devido aos protocolos anteriores fazerem a rede sem fio depender da VPN para fornecer segurança à camada. O WPA2 inicializa os fundamentos da criptografia e fornece a privacidade e integridade dos dados de segurança sem fio (*wireless*). Devido aos principais problemas de segurança do RC4 (utilizado no WEP e WPA) foi necessário criar um algoritmo de criptografia mais poderoso, em outubro de 2000, o NIST (*National Institute of Standards and Technology* ou Instituto Nacional de Padrões e Tecnologia) designou o AES como o substituto do RC4. O WPA2 é o padrão atual mais utilizado atualmente (2020) e a forma que o sistema lida com algoritmos e senhas diminui de forma significativa de um ataque de força bruta.

Kohlios e Hayajneh (2018) mencionam que o WPA3 foi lançado ao público pela *Wi-Fi Alliance*, em 25 de junho de 2018, utiliza a técnica de autenticação SAE (*Simultaneous Authentication of Equals* ou Técnica de Autenticação Simultânea) baseada em senha para autenticar o cliente no AP. SAE foi um protocolo introduzido pela primeira vez para uso em redes *mesh* WLAN (IEEE 802.11s) por Dan Harkins, em 2008, tendo mais tarde se provado vulnerável a ataques passivos e ativos. Após uma revisão do padrão RFC 7764, em 2015, o protocolo melhorado mostrou oferecer a proteção prometida. Essa resistência é alcançada usando o “*dragonfly handshake*” para alavancar a criptografia de

curva logarítmica e elíptica discreta, o resultado do *handshake* gera um PMK, que é então usado no aperto de mão de quatro vias, padrão usado no esquema WPA2. No protocolo *dragonfly*, um elemento de senha PE é usado em vez da senha para as chaves de computação, o PE é determinado no momento da sessão, usando um conjunto acordado de parâmetros de curva elíptica p , que é um grande número primo utilizado para determinar o campo primo para a curva elíptica, e q , que é outro grande número primo na ordem de um grupo G , acordada entre o cliente e o AP usando computação logarítmica discreta e uma técnica de “*hunting-and-pecking*” com a senha como valor inicial. O WPA3 é uma melhoria do WPA2, utilizando uma encriptação de 384 bits, este adiciona uma melhoria na proteção do *handshake*. Entretanto são poucos os dispositivos que se conectam a este tipo no momento (2020), conforme vierem atualizações isso será implementado.

3.5.6.2 *Evil Twin*

De acordo com Kohlios e Hayajneh (2018) uma prática muito comum é a utilização do *Evil Twin* (Gêmeos Malvados), fazendo o cliente pensar que está se conectando a um AP genuíno quando na verdade ele está se conectando a um AP não autorizado. O invasor representa um AP específico na esperança que um usuário se conecte a ele e assim que isto ocorrer, o invasor será um MITM (*Man-In-The-Middle*), capaz de descriptografar, ver e manipular o tráfego que o usuário está recebendo e enviando de seu dispositivo. O invasor encaminhará o acesso à internet para usuário de forma neutra, porém, estará atuando como um *proxy*, que visualiza todos os dados primeiro.

O *Evil Twin* é um *software* que cria um ponto de acesso falso com o mesmo nome de uma rede, para isso é necessário o SSID, endereço MAC, esquema de segurança e senha para enganar o dispositivo, o invasor configurará o AP não autorizado como tal e, em seguida, o invasor terá que desautenticar o usuário do AP real enviando sinalizadores de desautenticação, de forma que faça com que os clientes conectados a uma rede verdadeira se reconectem na falsa (atacante).

4. Honeypot

Joshi e Sardana (2011) dizem que *honeypot* (pote de mel) é uma tecnologia que trabalha para reunir o conhecimento prioritário sobre ataques, atraindo os *hackers* para atacá-los, atuando como uma isca que induz os usuários suspeitos a tentarem cometer um ato malicioso previsto, e então, os movimentos do invasor suspeito são monitorados e analisados. O termo *honeypot* foi cunhado pela primeira vez durante a Guerra Fria como uma técnica de espionagem, o ano de 1990 marca o início da utilização do conceito no campo de segurança da informação com a publicação de “*The cuckoos-Egg*” e “*Na Evening with Berferd in which a Cracker is Lured, Endured and Studied*”. Um grande desafio para uma organização é saber quem são seus inimigos, bem como quando e como podem atacar, pois o que fazem pode comprometer um sistema e, o mais importante, o porquê atacam. Em outras palavras, é uma ferramenta com a função proposital de simular falhas de segurança para colher dados de um invasor, não tem o intuito de resolver problemas específicos e sim entender onde estão as vulnerabilidades e ameaças existentes ao seu sistema ou negócio.

Um *honeypot* compreende:

- Sistema de produção de *Honeypot*: não é um verdadeiro sistema de produção, mas uma presa para invasores. Isso fornece os melhores arquivos e recursos do sistema falso para o invasor lidar. Respostas automáticas às ações do intruso são configuradas para mostrar o *honeypot* como um verdadeiro sistema de produção
- *Firewall*: fornecem logs sobre como um intruso está tentando entrar em um *Honeypot*. O *Firewall* é configurado para registrar todos os pacotes que vão para o sistema *honeypot*, pois não deve haver nenhuma razão legítima para o tráfego indo ou vindo deste
- Unidade de monitoramento: é uma unidade de avaliação de ameaças que monitora as atividades da rede e do sistema em busca de ações maliciosas ou violações de política e produzem relatórios para uma estação de gerenciamento. Revisar a ordem,

sequência, carimbos de data e hora, tipo de pacotes utilizados por um intruso para obter acesso ao *honeypot*, pressionamentos de teclas, acessos ao sistema, arquivos alterados, etc., ajuda também a identificar as ferramentas, metodologia usada e suas intenções. Um IDS pode fazer o trabalho de uma unidade de monitoramento.

- Unidade de alerta: o *Honeypot* deve ser capaz de gerar alertas por *e-mail* para enviar notificações para o administrador sobre o tráfego indo ou vindo, para deixá-lo revisar a atividade enquanto a intrusão estiver acontecendo.
- Unidade de registro: essa unidade fornece armazenamento eficiente para todos os registros do *firewall*, sistema e do tráfego que vai entre o *firewall* e o *honeypot*.

Os *honeypots* podem ser diferenciados por tipos e níveis, sendo os tipos:

- *Honeypot* de produção: utilizados para proteger organizações, seu objetivo é ajudar a mitigar o risco reduzindo significativamente a chance de intrusão ao descobrir vulnerabilidades e alertar os administradores sobre ataques, assim, agregando valor às medidas de segurança de uma organização, que ainda precisa depender de suas políticas de segurança, procedimentos e práticas recomendadas, como desabilitar serviços não utilizados, gerenciamento de *patches*, implementação de mecanismos de segurança como *firewall*, sistemas de detecção de intrusão, antivírus e mecanismos de autenticação segura.
- *Honeypot* de pesquisa: são usados para pesquisar ameaças que as organizações enfrentam e como se proteger melhor de tais ameaças (são mais focados na pesquisa das ações do intruso). Tradicionalmente, organizações como universidades, governo, militares ou organizações de pesquisa de segurança optam por essa opção, o que não é tão comum em organizações comerciais.

E os níveis:

- **Baixa Interatividade:** são caracterizados por sua interação mínima com os invasores e irão emular serviços falsos, como ftp ou http por exemplo. Não há sistemas operacionais reais ou serviços rodando neles, são apenas emulações acima da camada do sistema operacional (o que salva o sistema de controle do invasor), são mais simples de implementar e manter, porém, registram uma quantidade limitada de informações. O dano máximo que pode ser causado pelo atacante é derrubar a emulação do *honeypot*. Estes são muito úteis na identificação de endereços IP de invasores. Alguns exemplos comerciais são Honeyd e Specter.
- **Média interatividade:** este combina as vantagens de baixa e alta interatividade (sendo mais avançado que o de baixa e menos que o de alta), não possuem um ambiente de sistema operacional real, nem implementam todos os detalhes do protocolo de aplicativo, eles tem uma camada de virtualização e fornecem apenas as informações aguardadas pelo invasor. Essas respostas são utilizadas para que atacantes enviem seu *payload*, sendo que, ao recebimento deste, extrai o código do *shell* para investigação e análise detalhada. São bastante complexos e demandam muito tempo em sua implantação, além de exigirem grande esforço e profundo conhecimento de protocolos, serviços e segurança para criá-los.
- **Alta interatividade:** fornecem um sistema operacional real para o ataque, o que torna o sistema mais exposto a riscos. Devido a esse fornecimento completo, a possibilidade de acumular informações e atratividade aumentam muito, por isso são especialmente utilizados para fins de pesquisa. Estes são muito difíceis de implantar, pois várias ferramentas são utilizadas para sua execução. Pode ser muito útil na descoberta de novos *exploits*, *worms*, vírus e vulnerabilidades.

5. Precauções e Contramedidas

Com o entendimento das vulnerabilidades e pontos a demandarem maior atenção, faz-se necessário encontrar formas de evitá-los ou até mesmo corrigi-los. Vale lembrar que o hacker ético não é o profissional que resolve, mas é quem indica quais medidas tomar.

O Quadro 3 mostra algumas situações que explicam por que não estamos seguros

Quadro 3 - Por que não estamos seguros

Configurações malfeitas	O primeiro motivo da falta de segurança é a existência de recursos mal configurados. São usuários que utilizam uma senha de acesso muito fácil ou que contenham permissões excessivas, pastas e arquivos que podem ser gravados/sobrescritos desnecessariamente, máquinas da rede que podem ser acessadas por qualquer usuário, roteadores e <i>switchs</i> com contas de usuário-padrão, portas de serviços sem a proteção adequada, etc.
Softwares com falhas	Todo <i>software</i> que o usuário utiliza em sua máquina – sem exceção – possui falhas. A calculadora, o tocador de mp3, o leitor de arquivos PDF, o <i>Office</i> , o navegador de Internet, absolutamente, todos os programas. Isto cria uma situação complicada, pois nós, literalmente, trabalhamos utilizando um grande “queijo suíço”, que chamamos de computador.

	Qualquer pessoa com um conhecimento um pouco mais avançado pode debugar qualquer um dos programas utilizados, descobrir falhas e explorá-las. Muitas vezes isso nem é necessário, pois muitos sites na internet divulgam falhas para <i>softwares</i> conhecidos e nos fornecem ferramentas para explorar tais falhas.
Redes desprotegidas	As redes locais das empresas e instituições são vulneráveis a um enorme número de ataques.
Falta de criptografia	A utilização de serviços que não realizam nenhum tipo de codificação do tráfego (http, ftp e dns), que permitem que um atacante fareje o tráfego da rede e capture senhas e outros dados importantes. O ideal seria a utilização de versões “seguras” como HTTPS (HTTP com SSL), SFTP e DNSSEC.
Redirecionamento de tráfego	Quando a utilização do HUB era comum, o farejamento do tráfego da rede podia ser feito com maior facilidade, o que já não acontece com o switch. Mas ao utilizar técnicas como ARP <i>POISONING</i> , DHCP <i>SPOOFING</i> , ICMP <i>REDIRECT</i> ou <i>PORT STEALING</i> , pode-se fazer com que as máquinas da rede (ou em

	alguns casos até o próprio <i>switch</i>) nos enviem todo o tráfego local.
Spoofing	Tendo acesso ao tráfego local e se este tráfego não for criptografado, um atacante pode facilmente realizar ataques de <i>spoofing</i> ou falsificação. Dois exemplos comuns são o IP <i>Spoofing</i> , que permite falsificar o endereço de origem, e o DNS <i>Spoofing</i> , que permite enviar respostas DNS falsas.
Proteções ineficazes	As ferramentas de proteção que utilizamos como antivírus, filtros de pacotes e <i>proxys</i> sofrem de muitos problemas de segurança. A maior parte delas pode ser facilmente burlada por alguém com um pouco mais de conhecimento. Um exemplo simples é o filtro de pacotes: mesmo negando o acesso a todas as portas no tráfego de entrada da rede, um atacante ainda poderia acessar uma máquina realizando uma conexão reversa com tunelamento por http ou ICMP.
Falta de atualizações	Muitas empresas fornecem patches de correção para falhas, o que teoricamente, deveria tornar os <i>softwares</i> mais seguros, porém existem alguns problemas:

	<ul style="list-style-type: none"> • Quando a atualização do software não é automática, como no Windows Update, muitos usuários têm resistência a baixar e instalar patches por contra própria. • Muitas vezes, os próprios patches de segurança abrem novas falhas, o que pode comprometer o sistema. • Softwares muito antigos, como o Windows 98 por exemplo, já foram abandonados há tempos pelos seus desenvolvedores e não possuem mais patches de correção, portanto, utilizá-los é um risco enorme.
Fator humano	<p>Além de todos os problemas técnicos que podem causar a falta de segurança, ainda tem o pior deles, e justamente o não técnico. Através de técnicas de Engenharia Social, a manipulação do fator humano causa enormes desastres como: fazer um usuário rodar um cavalo de troia sem saber, conseguir informações privilegiadas sobre a empresa, obter especificações de um novo produto, etc.</p>

Fonte: Quadro adaptado do livro Segredos do Hacker Ético, 2014. p. 29,

O Quadro 4 mostra algumas formas de evitar problemas ou ataques citados ao decorrer do documento e outros que podem vir a ocorrer, embora não sejam as únicas, podendo variar a cada caso

Quadro 4 - Vulnerabilidades e soluções

Problemas	Soluções
Varredura de endereços IP	- Bloquear ICMP de entrada
Varredura de portas	- IPS (Sistema de prevenção de intrusos) - Uso de serviços com <i>Port knocking</i> - Bloqueio no <i>firewall</i>
FootPrinting e Fingerprint (Enumeração)	- IPS e/ou <i>Honeypot</i> - Atualizar o arquivo <i>robots.txt</i> - Verificar os dados expostos no Whois - <i>Hardening</i> das permissões de arquivos, diretórios e usuários
Falhas de Software	- Atualizações - Serviços rodando como usuários sem privilégios
Força-Bruta	- Bloqueio de conta após 3 erros - Bloqueio de IP no <i>firewall</i> - Política rígida de senhas (mínimo de 10 ou 12 caracteres)
Redirecionamento de Tráfego (ARP Poisoning)	- Switch com <i>ARP Inspection</i> - ARP estático para o <i>gateway</i> nas máquinas - ArpON instalado nas máquinas
SNIFFER (Farejamento)	- Evitar o redirecionamento do tráfego - Criptografar os dados

MITM Remoto	<ul style="list-style-type: none"> - Verificar certificado do site - Verificar se o <i>proxy</i> está ativo e remover se necessário
MITM Local	<ul style="list-style-type: none"> - Verificar certificado - Impedir redirecionamento do tráfego
Vulnerabilidades de aplicações em ambiente Web	<ul style="list-style-type: none"> - Refazer os filtros de entrada e saída de dados (<i>SQL Injection</i> e <i>XSS</i>) - Implementar verificação SOP (<i>Same Origin Policy</i>) - Implementar <i>tokens</i> para evitar <i>CSRF</i> - Implementar um <i>Web Application Firewall</i>
Recusa de Serviço (<i>Denial of Service</i>)	<ul style="list-style-type: none"> - Configurar o <i>firewall</i> para impedir <i>SYN flood</i> - Detectar e mitigar ataques <i>Smurf</i> - Utilizar serviços como <i>CloudFlare</i> para mitigar ataques <i>DDoS</i>
Exploração de Falhas	<ul style="list-style-type: none"> - Atualizar os <i>softwares</i> da máquina - Utilizar programas menos conhecidos - Melhorar regras do <i>IPS</i> para detectar os <i>exploits</i> e <i>payloads</i>
Keylogger, Vírus, Cavalos de Tróia, Worms, Spywares e Rootkits	<ul style="list-style-type: none"> - Antivírus corporativo com <i>Internet Security</i> - <i>Firewall</i> local - Anti-spywares - <i>HIDS</i> - <i>Chkrootkit</i> ou similar
Ataques Wireless	<ul style="list-style-type: none"> - Utilizar Chave <i>WPA2-PSK</i> complexa, se possível <i>wpa2-ENTERPRISE</i> juntamente com um servidor <i>Radius</i> - Utilizar um certificado nos clientes - Ocultar a rede sem fio - Realizar um controle de acesso por <i>MAC</i> - Separar rede pública da administrativa por <i>VLANs</i>

- Utilizar um WIPS (*Wireless Intrusion Prevention System*)

- Evitar o uso de *wi-fi* em redes desconhecidas

Fonte: Elaborada pelo autor com informações do livro Segredos do Hacker Ético

Uma das primeiras formas é fazer o *Hardening* (Endurecimento) do sistema. Para reforçar a segurança, contra possíveis ameaças, como problemas de acesso de arquivos e pastas (usuários com acesso indevido), configurações de senha padrões, arquivos de instalação esquecidos no servidor *web*, usuários inativos, serviços desnecessários e muitas outras, que geralmente as empresas não dão a devida importância. Estes geralmente não são vulnerabilidades em si, mas podem levar a gerar maiores problemas.

“*Hardening* é o processo que visa proteger um sistema contra ameaças desconhecidas. Os administradores de sistema se protegem contra tudo o que eles acreditam que pode ser uma ameaça à segurança tanto de softwares quando de empresas” (Hardening Windows, 2006, v. 2. P. 2)

Caso não haja redirecionamento de tráfego já pode-se eliminar o *Spoofing* e MITM local (*Man in the Middle* Local). O MITM Remoto por ser mais complexo, geralmente é barrado pelo antivírus (com *Internet Security*), em alguns casos até o próprio navegador pode barrá-lo.

Para resolver o redirecionamento é preciso deixar a entrada estática ao invés de dinâmica, então, associar estaticamente o endereço IP com o MAC. (Figura 17).

Figura 17 - Entradas do tipo dinâmico e estático

```

Microsoft Windows [versão 10.0.19041.572]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

C:\Users\>arp -a

Interface: 192.168.56.109 --- 0xe

```

Endereço IP	Endereço físico	Tipo
192.168.56.1	a8-5e-45-e9-b9-a0	dinâmico
192.168.56.50	d0-d0-03-c9-ba-8d	dinâmico
192.168.56.102	0c-9d-92-6d-6b-54	dinâmico
192.168.56.114	ec-fa-5c-78-46-e1	dinâmico
192.168.56.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático
255.255.255.255	ff-ff-ff-ff-ff-ff	estático

Fonte: Elaborada pelo autor, captura de tela

Assim o IP não pode ser substituído através de uma requisição ARP. Também há a possibilidade de utilizar o ArpON, software que permite a verificação da tabela ARP da máquina (deverá instalar em todos os computadores da rede).

A nível de switch e roteador (camada 2 e 3) pode-se habilitar algumas proteções, no caso do switch *Cisco Catalyst* 6500+ (utilizados em empresas) e alguns outros dispositivos temos o *Dynamic Arp Inspection* (DAI), permite que o monitoramento, pelo *switch*, das requisições enviadas e recebidas pelas portas. E também o *DHCP Snooping*, se houver um servidor DHCP falso na rede ele verifica e derruba algumas requisições falsas para não redirecionar o tráfego.

Após o descobrimento destas vulnerabilidades o ideal é pesquisar se já há um método mais eficaz de combatê-las (considerando específica versão da vulnerabilidade), em sua maioria atualizar versões, *patch* e outros passos. Primeiramente é necessário escanear a rede, identificar os sistemas vulneráveis, indicar a vítima que faça o *download* das correções, gerar relatórios das atualizações e verificar se há mais alguma atualização. Feito isso repetir o escaneamento para verificar se foi corrigido. Vale ressaltar que existem muitas ferramentas de gerenciamento de patches hoje em dia, as quais facilitam o processo.

É importante também fazer o *hardening* no *firewall* e no DNS e para isso pode-se utilizar o *PfSense*, que é um *software* gratuito e ajuda em todo o processo. Muitas empresas ainda hoje não o utilizam o DNS ou não utilizam de forma integrada com o *firewall*. É recomendável instalar nas máquinas o HIPS, que é um sistema de prevenção de intrusos local.

No caso de vulnerabilidades *web* não é tão simples quanto aos outros. Uma das principais recomendações é dar um treinamento aos programadores da empresa focado em desenvolvimento seguro. Existem certificações para o desenvolvimento seguro, como a ECSP (*EC-Council Certified Secure Programmer*) e a EXIN da *Secure Programming Foundation*, além de várias outras.

Agora voltado para o lado da rede, existe um tipo de *firewall* mais específico que é o *Web Application Firewall*, utilizado para inspecionar os pedidos HTTP e requisições e procurar por padrões que podem ser ataques, diferentemente do *Sniffer* ele irá monitorar as requisições tornando mais lento, mesmo assim é uma prática interessante de se utilizar. Para as redes *wi-fi* pode-se optar pela utilização do WPA2 (de preferência ao *Enterprise* ao invés do *pre-shared key*). Pode-se utilizar também certificados, implementar o WIPS, HIDS, VLANs, filtros MAC, ocultar redes, muito importante o uso de criptografias nos protocolos e se possível uma segunda camada de autenticação, proteger os hosts contra ARP *Poisoning* e localizar e remover *Evil Twin* utilizando a trilateração do sinal. O WIPS, assim como o IPS, tem a ideia de ser um sistema de proteção contra intrusos, porém ele é mais específico para *wireless*. Ele ajuda muito a detectar um *Access Point* falso na rede e impede que o cliente fique conectado em um Evil Twin. Vale ressaltar que este só funcionará dentro da empresa.

6 Conclusão

Conforme discussões anteriores o projeto cumpriu com sua proposta, onde o intuito é instruir empresas e profissionais relacionados à área de tecnologia da informação e desenvolvimento. Foram abordados os princípios de um hacker ético e onde ele se aplica neste meio, dadas as situações que são os pontos críticos e quais devem ser devidamente entendidas para que haja uma análise de cada caso.

Além do desenvolvimento de caminhos para realizar testes de ataque e explorações de vulnerabilidades, foram citados meios como softwares e mudança de padronização visando a prevenção de ataques futuros, inclusive os mesmos meios podem ser funcionais para outras vulnerabilidades não citadas no documento devido a inúmera quantia existente.

Como mencionado, as vulnerabilidades estão por todo canto, desde o meio humano até o mais profundo que um sistema pode alcançar, por isso podemos concluir que não há apenas um método eficaz para todos os casos, pois muitos tem a mesma solução e alguns, no entanto, podem não ter sequer uma forma de reparação. Com base nisto há a necessidade de os testes serem realizados por profissionais qualificados e, principalmente, as empresas instruírem formas de desenvolvimento seguro desde a base da carreira de um profissional. Para próximos passos é recomendável a realização dos testes com buscas mais aprofundadas, a cada vulnerabilidade encontrada, com intuito de prevenir o roubo de informações, que pode acarretar desde pequenos problemas até danos irreparáveis.

Finalmente, o projeto demonstrou que pessoas mal intencionadas, desde que tenham um conhecimento mínimo em ferramentas de invasão, podem causar grandes danos a pessoas e principalmente a empresas. Contudo, havendo bom preparo e prevenção, por parte da vítima, essa não virá a sofrer drasticamente com nenhum ou quase nenhum dos ataques citados ao decorrer do documento. Também foi ressaltada a importância do profissional hacker ético bem como os benefícios que o mesmo agrega ao mundo tecnológico.

Referências Bibliográficas

ALBLWI, Samia. SHUJAE, Khalil. **A Survey on Wireless Security Protocol WPA2.** Disponível em: <

<https://csce.ucmss.com/cr/books/2017/LFS/CSREA2017/SAM3445.pdf> >.

Acesso em: 10 nov. 2020.

ASSUNÇÃO, Marcos. **Segredos do Hacker Ético – 5ª edição.** ed. Visual Books – 2014, p. 29 – 246.

Brasil, Lei Nº 12.737, de 30 de novembro de 2012. **Tipificação Criminal de Delitos Informáticos.** Disponível em: <

http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12737.htm >.

Acesso em: 04 nov. 2020.

BRISCOE, Neil. **Understanding The OSI 7-Layer Model.** Disponível em: <
http://sdcc.vn/template/266_osi7layer_t04124.pdf >. Acesso em: 5 nov. 2020.

BULBUL, Halil. BATMAZ, Ihsan. OZEL, Mesut. **Wireless Network Security: Comparison of WEP (Wired Equivalent Privacy) Mechanism, WPA (Wi-Fi Protected Access) and RSN (Robust Security Network) Security Protocols.**

Disponível em: < <https://dl.irstu.com/wp-content/uploads/Download/Education/Book/Network/Network%20Security/WEP-WPA-Article/Wireless%20Network%20Security.pdf> >. Acesso em: 10 nov. 2020.

DILLON, Douglas et al. **Method and System for Utilizing Virtual Private Network (VPN) Connections in a Performance Enhanced Network.** Disponível em: <

<https://patentimages.storage.googleapis.com/2a/3b/89/cd8ba425e7943b/US20030219022A1.pdf> >. Acesso em: 10 nov. 2020.

ENGEBRETSON, Patrick. **The Basics of Hacking And Penetration Testing – Second Edition.** ed. Elsevier – 2013, p. 1.

GOURLEY, David. TOTTY, Brian. **HTTP: The Definitive Guide.** ed. O'REILLY – 2002, p. 308 – 322.

HADNAGY, Christopher. FINCHER, Michele. **Phishing Dark Waters: The Offensive and Defensive Sides of Malicious E-mails**. ed. Wiley Publishing – 2015, p. 1 – 3.

HADNAGY, Christopher. **Social Engineering: The Art of Human Hacking**. ed. Wiley Publishing – 2011, p. 11 – 13.

HASSELL, Jonathan. **Hardening Windows – Second Edition**. ed. Apress – 2006. p. 2

JOSHI, R. SARDANA, Anjali. **Honeypots: A New Paradigm to Information Security**. ed. CRC Press – 2011. p. 1 – 17.

KENNEDY, David. O’GORMAN, Jim. AHARONI, Mati. **Metasploit The Penetration Tester’s Guide**. ed. No Starch Press – 2011. p. 8 – 10.

KHAN, Mohd. KHAN, Farmeena. **A Comparative Study of White Box, Black Box and Gray Box Testing Techniques**. p.12. 2012. Disponível em: <
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.1887&rep=rep1&type=pdf#page=22> >. Acesso em: 04 nov. 2020.

KOHLIOS, Christopher. HAYAJNEH, Thaier. **A Comprehensive Attack Flow Model and Security Analysis for Wi-Fi and WPA3**. Disponível em: <
<https://www.mdpi.com/2079-9292/7/11/284> >. Acesso em: 10 nov. 2020

LONG, Johnny. **Google Hacking for Penetration Tester – Volume 2**. ed. Syngress – 2008, p. 1 – 50.

MAHALA, Ilan et all. **Protocol Conversion “Bearer Independent Protocol (BIP)” – TCP/IP for Communication Between SIM and Terminal**. Disponível em: <
<https://patentimages.storage.googleapis.com/e8/4c/89/2a38f89d1b2862/US8447836.pdf> >. Acesso em: 6 nov. 2020.

MATA, Saulo. GUARDIEIRO, Paulo. **Uma Introdução ao Protocolo ENUM**. Disponível em: <
https://www.peteletricaufu.com/static/ceel/doc/artigos/artigos2011/IX_CEEL_049.pdf >. Acesso em: 8 nov. 2020.

McCLURE, Stuart. SCAMBRAY, Joel. KURTZ, George. **Hacking Exposed: Network Security, Secrets and Solutions, Third Edition**. ed. McGraw-Hill Osborne – 2001, p. 4 – 5.

SARI, Arif. KARAY, Mehmet. **Comparative Analysis of Wireless Security Protocols: WEP vs WPA**. Disponível em: < https://www.scirp.org/pdf/IJCNS_2015121715205514.pdf >. Acesso em: 10 nov. 2020

SCHÖTTLE, Markus. **Hackers not Crackers**. ATZ elektronik. v. 10, p. 3. abr. 2015

TINGLEY, Chase. WALSH, Robert. **Extension of Address Resolution Protocol (ARP) for Internet Protocol (IP) Virtual Networks**. Disponível em: < <https://patentimages.storage.googleapis.com/ea/73/dc/a4db263462ac96/US7260648B2.pdf> >. Acesso em: 8 nov 2020.

VAL, David. KLEMENTS, Anders. **Method and Apparatus for Communication Media Commands and Media Data Using the HTTP Protocol**. Disponível em: < <https://patentimages.storage.googleapis.com/bf/51/a2/787d180a796dde/US6128653.pdf> >. Acesso em: 6 nov. 2020.

WHALEN, Sean. **An Introduction to ARP Spoofing**. Disponível em: < http://67.225.133.110/~gbpprorg/2600/arp_spoofing_intro.pdf >. Acesso em: 10 nov 2020.

WITMAN, Paul. **The Art and Science of Non-Disclosure Agreements**. Disponível em: < <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=3029&context=cais> >. Acesso em: 5 nov. 2020.