doi: 10.5102/un.gti.v2i2.1458

Camuflagem de serviços na internet*

Internet service's camouflage

José Carlos Fernandes de Macedo[1]
Marco Antônio de Oliveira Araújo[2]

Resumo

Várias empresas utilizam a Internet para disponibilizar serviços de caráter crítico. Esses serviços, se comprometidos, podem permitir a realização de atividades maliciosas e desastrosas para a infraestrutura de Tecnologia da Informação. Este trabalho apresenta a técnica de camuflagem, que protege o perímetro da rede, pois evita a detecção de serviços por pessoas não autorizadas. A fim de se demonstrar a eficiência da camuflagem, um laboratório foi realizado para testar o aplicativo *Fwknop,* o qual implementa a técnica denominada *Single Packet Authorization* (SPA). Os resultados obtidos deixam claro que o SPA foi muito bem concebido e é bastante eficiente.

Palavras-chave: Camuflagem. *Firewall*. Internet. Segurança.

Abstract

Several companies use the Internet to deliver services of a critical nature. These services, if compromised, could allow for malicious and disastrous activities for the Information Technology infrastructure. This paper presents the camouflage technique that protects the network perimeter, as it avoids services detection by unauthorized persons. In order to demonstrate the camouflage effectiveness, a laboratory was designed to test the application Fwknop, which implements a technique called Single Packet Authorization - SPA. The results clearly show that the SPA was very well designed and is very efficient.

Keywords: Camouflage. Firewall. Internet. Security.

* Artigo recebido em 06/08/2012
Aprovado em 27/09/2012

[1] Formado em Matemática pelo Centro de Ensino Unificado de Brasília - CEUB. Pós-graduado em Análise de Sistemas pela GFI Consultoria e Treinamento e em Redes de Computadores pelo Centro Universitário de Brasília - Uniceub. Certificado como Microsoft Certified Systems Engineer pela Microsoft Corporation e como Computer Forensics Certified pela Axur Information Security. Trabalhou três anos no Departamento de Educação Física, Esportes e Recreação – DEFER na função de Chefe do Setor de Tecnologia da Informação, onde foi responsável pela montagem da infraestrutura de TI e pelo desenvolvimento dos sistemas utilizados pelo Órgão. Trabalha há dezessete anos na área de TI do Tribunal de Justiça do DF – TJDFT, ocupando há onze anos a função de chefe do setor de Redes e Segurança da Informação.

[2] Possui graduação em Ciência da Computação pela Universidade de Brasília (1997) e mestrado em Ciência da Computação pela Universidade de Brasília (2003). Foi supervisor, por 4 anos, da Seção de Segurança da Informação do Tribunal Superior do Trabalho. Hoje faz parte da equipe de Qualidade e Segurança do Sistema PJe da Justiça Trabalhista. É professor titular de redes e segurança do Centro Universitário de Brasília e da Universidade Católica de Brasília para a graduação e para a pós-graduação, atuando principalmente nos seguintes temas: detecção, segurança e redes.

1 Introdução

A camuflagem é uma técnica que pode ser utilizada para proteger os recursos de Tecnologia da Informação disponibilizados na Internet. Essa técnica de defesa consiste em ocultar os computadores servidores e seus serviços, permitindo que apenas seus usuários legítimos tenham o conhecimento de sua existência (DEGRAAF, 2007, p. 42-45).

Este artigo foi desenvolvido com a intenção de apresentar a técnica de camuflagem para auxiliar na defesa dos serviços críticos acessíveis pela Internet e permitir o fornecimento de um nível adequado de proteção aos recursos de Tecnologia da Informação (TI).

Apesar da ampla utilização de outras soluções eficientes, tais como *firewall*, *Intrusion Detection/ Prevention System* e *Virtual Private Network*, Degraaf (2007, p. 42-45) esclarece que, além de encobrir a existência do recurso de TI, a técnica de camuflagem oferece uma camada extra de proteção contra ataques, protege sistemas com vulnerabilidades que não possuem as respectivas correções aplicadas e adiciona autenticação a sistemas que possuem medidas de segurança inadequadas.

Neste documento, são indicadas algumas medidas a serem adotadas a fim de se evitar que os ativos de TI críticos e as redes corporativas fiquem expostos a análises de suas vulnerabilidades e a ataques que podem passar despercebidos pelas outras soluções de proteção do perímetro da rede.

A primeira seção deste artigo aborda as técnicas populares de ataque na Internet, enquanto a segunda seção apresenta algumas contramedidas geralmente utilizadas para evitar esses ataques. O funcionamento de algumas técnicas de camuflagem foi explicado na terceira seção, sendo que a técnica denominada *Single Packet Authorization* (SPA) foi abordada com detalhes. Para demonstrar a eficiência da camuflagem, a quarta seção descreve a realização de um laboratório de implementação de uma ferramenta de SPA (Fwknop). O aplicativo Fwknop foi posto em ação e alguns ataques foram executados ao mesmo tempo em que ocorriam os acessos legítimos, autorizados. Por último, são feitas as considerações finais e discutidos os pontos positivos e negativos da solução testada.

2 Ataque a computadores servidores

Nesta seção, serão abordadas algumas técnicas populares de ataques usadas para atingir os recursos de Tecnologia da Informação, tendo em vista que este trabalho procura apresentar uma solução que ajuda a evitá-los.

Para Mcclure, Scambray e Kurtz (2005, p. 4-40), o primeiro passo para a realização de um ataque é a verificação da postura da empresa em relação à segurança da informação. São levantadas informações sobre nome de domínio, blocos de rede, endereços de servidores, mecanismos de controle de acesso e suas listas de filtragem, sistemas de detecção de intrusão, nomes e grupos de usuários, tabelas de roteamento (ataque de varredura) etc. Essa fase é importante para a escolha do alvo.

Depois de o alvo ser selecionado, Northcutt et al. (2002, p. 536) explicam que o agressor passa a verificar os serviços e as respectivas versões instaladas no servidor que será atacado (ataque de enumeração). Com posse dessas informações, um agressor pode pesquisar as vulnerabilidades conhecidas nos *softwares* que disponibilizam os serviços, a fim de elencar os elementos necessários para a realização do ataque.

Todavia, as falhas não estão presentes somente nos programas e nos aplicativos. Existem falhas nas pilhas de protocolos e nas estruturas de rede que compõem a Internet. Essas falhas, de acordo com Cheswick, Bellovin e Rubin (2005, p. 117-118), muitas vezes são exploradas na forma de tentativas de inanição de recursos, para que o serviço ou o servidor fique "fora do ar" (ataque de *Deny of Service* – DDoS).

Comer (2007, p. 143-144) comenta que há outras formas de se aproveitar da rede e realizar ataques. Ele menciona a captura de pacotes que trafegam sem criptografia, que pode levar ao roubo das credenciais de usuários de um sistema. As credenciais são cobiçadas uma vez que, com elas, podem-se controlar os recursos de uma rede.

O roubo de credenciais também pode ser efetuado com o ataque de força-bruta, que consiste em várias tentativas de autenticação com senhas diferentes (ERICKSON, 2008, p. 422). Ives (apud LEMOS 2008, p. 54, tradução nossa) declara que:

> Dado tempo suficiente, qualquer senha pode ser quebrada e muitas delas o são com relativa facilidade, porque, até certo ponto, os seres hu-

> manos são preguiçosos e quase sempre optam por senhas não aleatórias e fáceis de lembrar – e, portanto, fáceis de adivinhar.

Existem muitos riscos de comprometimento dos serviços de TI, e a Internet é uma rede perigosa – no ano de 2011 foram reportados ao CERT.br mais de trezentos e noventa e nove mil incidentes de segurança na Internet.[3] Qualquer organização que disponibilize recursos nessa rede deve prover à sua infraestrutura de TI soluções de segurança que atendam às políticas para os negócios e forneçam um bom nível de proteção.

3 Técnicas de defesa

Esta seção apresenta as técnicas de defesa que geralmente são utilizadas para proteger os recursos de TI dos ataques que foram descritos na seção anterior.

Northcutt et al. (2002, p. 5) define *firewall* como sendo o equipamento instalado entre as redes externa e interna de uma organização que tem como função filtrar o tráfego e liberar apenas o que for permitido na política de segurança.

Na opinião de Cheswick, Bellovin e Rubin (2005, p. 177), o *firewall* [...] "é qualquer dispositivo, *software,* arranjo ou equipamento que limita o acesso à rede". Complementam, dizendo que a filtragem pode ocorrer em diversos níveis e até operar na camada onde as aplicações atuam. Essa filtragem consegue evitar ataques de varredura, enumeração, *DOS* e força bruta nos servidores e serviços da organização cujos acessos são negados. Porém, há os serviços disponíveis para acesso pela Internet e, para esses, o *firewall* não consegue oferecer proteção, sendo necessário o uso de ferramentas capazes de detectar tentativas de intrusão.

Um sistema de detecção de intrusão *Intrusion Detection System* (IDS) é um sistema que monitora os pacotes que trafegam pela rede à procura de algum comportamento que indique uma violação de segurança (COMER, 2007, p. 556). Northcutt et al. (2002, p.157) fazem um alerta sobre a necessidade da utilização desse sistema como ferramenta de segurança: "[...] sem detecção de intrusão, você pode não perceber muitos ataques que ocorrem".

Fuchsberger (2005, p. 136-137) ensina que o sistema de detecção de intrusão pode gerar alertas e até tomar medidas para interromper ataques de varredura, enumeração, DoS e força bruta. Contudo, há a possibilidade de que alguns ataques ainda não estejam catalogados ou gerem um comportamento aparentemente normal, fazendo com que eles não sejam percebidos.

O conteúdo do tráfego também necessita de cuidados e a solução mais adequada para proteger a comunicação entre um cliente e um servidor, evitando a análise de pacotes capturados na rede, é a criptografia dos pacotes.

Tanenbaum (2003, p. 784) comenta que, para a Tecnologia da Informação, a criptografia é a cifragem dos dados e utiliza chaves para codificar a mensagem, sendo que apenas o detentor da chave consegue enxergar o dado original.

A criptografia aplicada ao tráfego, quando conta com técnicas de ocultação, dá origem a Redes Virtuais Privativas (*Virtual Private Network-VPN).* A VPN, informam Northcutt et al. (2002, p. 181), "[...] é uma conexão que é estabelecida por uma infraestrutura 'pública' ou compartilhada existente, usando tecnologia de criptografia ou autenticação para proteger o seu *payload*".

A utilização das técnicas e soluções de segurança aludidas neste capítulo é de grande importância para uma rede presente na Internet. No entanto, como tais ferramentas possuem limitações, medidas adicionais, tal como a camuflagem, são recomendáveis para melhorar a segurança oferecida aos serviços críticos (ex.: *shell* remoto de computadores servidores, utilizado para administração de serviços).

4 Camuflagem

Esta seção apresenta a técnica de defesa denominada camuflagem, que consiste em ocultar a presença de serviços na rede e tem como objetivo complementar a segurança oferecida pelas técnicas de defesa informadas na seção anterior.

A camuflagem pode ser utilizada na Internet, para que alguns serviços críticos possam ser acessados de qualquer lugar sem que pessoas não autorizadas tenham o conhecimento da existência do recurso.

[3] Disponível em: <http://www.cert.br/stats/incidentes/>. Acesso em: 27 maio. 2012.

Quando há necessidade de um recurso estar disponível na *Internet* e não se sabe o endereço de onde partirá o pedido de acesso, as regras do *firewall* não podem filtrar a origem da conexão e, por isso, acabam permitindo a realização de varreduras no computador servidor por qualquer pessoa (CHESWICK; BELLOVIN; RUBIN, 2005, p. 131).

A estratégia aqui proposta é a criação automática de regras de filtragem mediante algum tipo de autenticação. Trata-se de regras temporárias configuradas para um acesso específico (origem, destino e serviço a ser acessado). Em virtude de essas regras serem apagadas logo após a conexão ser estabelecida e informarem o endereço de origem da conexão, a presença do serviço dificilmente é notada.

Alguns *firewalls* proprietários do mercado permitem a filtragem de acesso mediante autenticação (ex.: *Check Point Transparent Session Authentication*[4] e Cisco ASA *Cut-Through Proxy Feature*[5]). O problema é que, para realizar essa tarefa, soquetes[6] de serviços são disponibilizados para os usuários da rede. Caso tais serviços sejam comprometidos, o invasor poderá configurar as regras de filtragem no *firewall* de forma a disponibilizar para si todos os recursos da rede protegida pelo equipamento.

4.1 Port knocking

Uma forma mais segura de criação automática de regras de filtragem em *firewalls* é o *Port Knocking*. Esse método não disponibiliza soquetes e, consequentemente, não possui portas que expõem o serviço a ataques ao receberem requisições de conexões.

Jeanquier (2006, p. 4-5) conta que a ideia do método surgiu no ano de 2001, em uma lista de discussões do *site German Linux User Group* (www.guug.de), e que se concretizou em 2003, quando Martin Krzywinski[7] lançou um artigo que dava a ele o nome *Port Knocking*.

Krzywinski (2003, p. 12-17) explica a técnica, informando que, quando um computador deseja acessar um serviço crítico, uma sequência de pacotes com características específicas é enviada para o *firewall*. Esses pacotes são interceptados e analisados e, caso a sequência esteja correta, a regra de filtragem permitindo o acesso ao recuso desejado é criada.

É recomendável que a sequência de pacotes enviada ao *firewall* seja criptografada, evitando, assim, o monitoramento e a análise do conteúdo dos pacotes.

Para que a segurança oferecida pela técnica de camuflagem seja eficaz, a regra criada automaticamente deve ser apagada pelo próprio *firewall* após um tempo determinado ou por uma sequência de pacotes enviada pelo cliente.

Temos a seguir um exemplo do que acontece em algumas implementações de *Port Knocking* (KRZYWINSKI, 2003, p. 12-17):

Fase 1 - O cliente não consegue se conectar a uma aplicação que escuta em determinada porta;

Fase 2 - O cliente envia pedidos criptografados de conexões, utilizando uma sequência pré-definida de portas;

Fase 3 - O *firewall* intercepta os pedidos de conexão, decifra-os e os interpreta;

Fase 4 - O *firewall* executa a tarefa solicitada na sequência de portas enviada (a regra é criada no *firewall*);

Fase 5 - O cliente consegue finalmente se conectar ao recurso desejado.

Fase 6 – A regra de acesso criada pela solução de *Port Knocking* é apagada automaticamente após algum tempo.

Embora o *Port Knocking* seja uma técnica de camuflagem promissora, Rash (2006, p. 64-65) chama a atenção para alguns problemas, tais como: demora em estabelecer uma conexão devido à sequência extensa de pacotes; dificuldade de autenticação em virtude dos pacotes chegarem fora de ordem ao *firewall*; vulnerabilidade a ataques de repetição, já que a sequência para a criação de regras é sempre a mesma.

Algumas variações[8] do *Port Knocking* surgiram na tentativa de corrigir as falhas. Veremos na próxima subseção uma dessas variações.

[4] Disponível em: <http://www.dm-int.com/Authentication.htm>. Acesso em: 05 maio 2011.

[5] Disponível em: <http: //www.ci sco.com/en/ US/products / hw/vpndevc/ps2030/ products_configuration_ example-09186a00807349e7.shtml>. Acesso em: 05 maio 2011.

[6] Segundo Comer (2007, p. 398), soquete é "[...] a interface entre um programa aplicativo e os protocolos de comunicação em um sistema operacional".

[7] Disponível em: <www.portknocking.org>. Acesso em: 05 maio 2011.

[8] Disponível em: <www.portknocking.org/view/implementations>. Acesso em: 10 maio 2011.

4.1.1 Single Packet Authorization – SPA

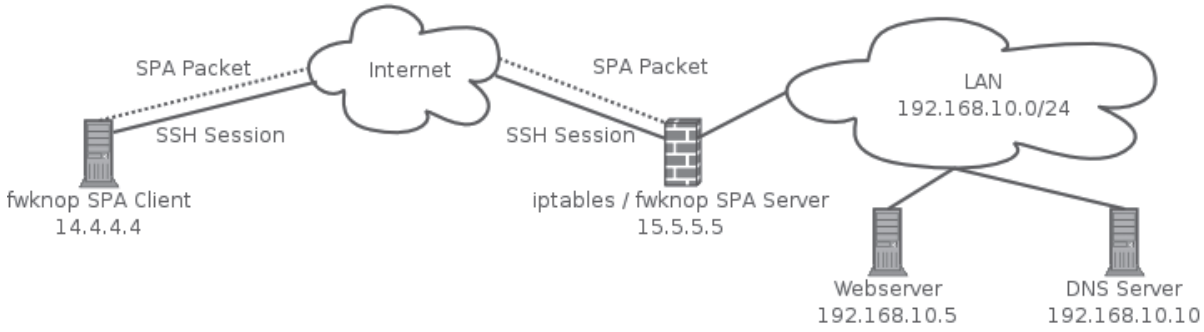
O Single Packet Authorization - SPA é uma variação da técnica de Port Knocking.

Isabel (2005) explica que, para evitar alguns dos problemas da técnica original de Port Knocking, o SPA se baseou no sistema Cryptographic One-Time Knocks - COK, o qual trabalha com apenas um único pacote criptografado que possui as credenciais e os demais elementos necessários para a criação automática das regras de filtragem no firewall.

Jeanquier (2006, p. 48) esclarece que os dados necessários à criação das regras são informados como campos da mensagem, sendo eles: dado randômico (caracteres gerados aleatoriamente), usuário do computador cliente, hora do computador cliente, versão do aplicativo que implementa o SPA, modo, acesso desejado ou linha de comando e o digest (código gerado pelo algoritmo de hash).

Rash (2006, p. 66–68) explica que o SPA trabalha em conjunto com o firewall, criando automaticamente regras de filtragem de acesso. Para que as regras sejam criadas, o cliente da solução deve enviar ao firewall uma solicitação assinada digitalmente (vide Figura 1).

Figura 1. Fluxo dos pacotes com a utilização do SPA



Fonte: Disponível em: <www.cipherdyne.org/fwknop/>. Acesso em: 07 maio 2012.

A fim de se evitar que o serviço disponibilizado pela solução de SPA fique exposto a ataques, nenhuma porta de serviço é levantada. Em vez disso, o serviço rastreia os pacotes que trafegam por uma determinada interface, procurando informações específicas. Esse cuidado não impede que, no caso de uma varredura de rede, as particularidades das requisições enviadas ao *firewall* sejam notadas, mas, como os pacotes são criptografados, um agressor não consegue entender o conteúdo da mensagem.

Uma medida adotada pela solução a fim de se evitarem falhas de *buffer overflow*, é a limitação do tamanho do pacote com o qual ela trabalha. Dessa forma, uma sobrecarga causada por um pacote muito grande pode ser evitada.

Rash (2006, p. 66-68) conta que a ferramenta trabalha com chaves criptográficas assimétricas, que são utilizadas para assinar e criptografar a mensagem referente à criação de regras no firewall. Quando as credenciais são reconhecidas, uma regra é criada no firewall, contendo o IP de origem, o IP de destino e a porta a ser acessada. A regra, após o tempo determinado nas configurações da solução, é apagada e novas conexões são bloqueadas. As conexões já estabelecidas são mantidas, desde que existam regras criadas no firewall para esse fim.

Existem elementos randômicos dentro de cada pacote criptografado, fazendo com que as mensagens sejam únicas, e são gerados *digests* desses pacotes, que são armazenados pelo servidor. O objetivo dessas medidas é bloquear pacotes já recebidos, evitando os ataques de repetição.

Jeanquier (2006, p. 50-51) declara que o SPA possui venerabilidades a alguns ataques, estando entre eles:

> *a) man-in-the-middle* – o pacote enviado para o *firewall* pode ser interceptado e retido. Como o pacote não chega a seu destino, ele pode ser utilizado por quem o interceptou, uma vez que o *digest* dele não constará na relação de pacotes recebidos. Entretanto os pacotes possuem um selo de tempo para evitar esse tipo de ataque. Para haver sucesso, o agressor precisaria saber a hora exata do computador cliente que enviou o pacote para o *firewall* – caso o computador cliente faça requisições de NTP para sincronizar seu relógio, essas requisições podem ser

capturadas e utilizadas para determinar a hora do sistema operacional.

b) piggyback – um agressor que conhece a rotina de acesso a um serviço pode se aproveitar de um momento de criação da regra de filtragem e, se passando pelo equipamento que solicitou a conexão (IP *Spoofing*), conseguir acessar o serviço. Contudo, o agressor precisaria adivinhar a porta a ser liberada (as portas liberadas são aleatórias e mudam a cada requisição).

c) replay attacks – se houver *firewalls* realizando balanceamento de carga, a lista de *digests* de um será diferente da lista do outro. Isso pode permitir que pacotes capturados possam ser reutilizados, tendo em vista que a relação de pacotes da solução de SPA em cada *firewall* estará incompleta.

Mesmo havendo a possibilidade de que a técnica de SPA seja burlada, algumas medidas simples, como por exemplo, a não utilização do protocolo NTP e a replicação da tabela de *digests* entre *firewalls* redundantes, poderiam evitar os ataques mencionados.

O SPA possui um bom nível de segurança e, baseando-se nisso, Rash (2006, p. 68, tradução nossa) conclui:

> O *Single Packet Authorization* tem diversas características que o tornam mais poderoso e flexível para proteger os serviços de rede do que o *port knocking*. Sua capacidade de transmissão de dados, juntamente com a sua estratégia para a prevenção de ataques de repetição, fazem dele um candidato ideal para expandir a configuração dos filtros de pacotes para descartar, por padrão, todas as conexões com alguns serviços essenciais.

Há alguns aplicativos que usam a técnica de SPA, tais como o Fwknop[9], o Aldaba[10] e o Knockknock.[11] Durante o laboratório, o aplicativo utilizado será o Fwknop (***FireWall KNock OPerator***).

4.1.2 Implementando a camuflagem para o acesso a serviço de SSH

A camuflagem do serviço de SSH presente em um servidor hospedado em *Data Center* se baseia em *firewall* com uma solução de *Single Packet Authorization*-SPA instalada. O SPA cria automaticamente regras no *firewall*, permitindo o acesso ao serviço SSH apenas no momento em que o administrador estiver estabelecendo uma conexão. A regra criada é apagada, também automaticamente, após um tempo determinado. Como consequência, caso haja uma varredura de serviços no servidor, o SSH não será detectado, pois todas as regras permitindo o início de conexões a esse serviço terão sido apagadas (essa situação está representada na Figura 1). Deve haver uma regra liberando os acessos das conexões informadas como estabelecidas na tabela de estados das conexões. Tal regra é imprescindível para que o administrador consiga manter a comunicação com o serviço SSH após a regra criada pelo SPA ter sido apagada.

Figura 2 - Proposta de Implementação

Fonte: Do autor

A autenticação no serviço de SSH deve ocorrer em poucos segundos, para que a regra criada no *firewall* possa ser apagada o mais rapidamente possível. Uma forma de agilizar a autenticação no serviço de SSH é a utilização de certificados digitais, já que os certificados digitais possuem as credencias do usuário e, sendo assim, dispensam a necessidade de digitação de senhas.

O endereço IP da rede interna do serviço de SSH não é divulgado, sendo utilizada a conversão do endereço interno para o endereço da rede pública mediante a técnica de *Network Address Translation* – NAT.

[9] Disponível em: < http://cipherdyne.org/fwknop>. Acesso em: 17 abr. 2011.

[10] Disponível em: <http://www.aldabaknocking.com/download >. Acesso em: 17 abr. 2011.

[11] Disponível em: < http://www.thoughtcrime.org/software/knockknock>. Acesso em: 17 abr. 2011.

No tocante ao computador utilizado pelo administrador, é necessária a instalação e configuração do cliente do serviço de SPA.

Os certificados digitais que são utilizados pelo serviço SSH e pela solução de SPA são gerados tanto para a estação de trabalho do administrador quanto para o *firewall*. Um programa de gerência de certificados digitais deve ser instalado em ambos os computadores.

Um último detalhe importante é a criação de *scripts* que automatizem todos os procedimentos de acesso aos serviços. Com a utilização dos *scripts*, os acessos serão rápidos e a vida do usuário (administrador do servidor) será facilitada – com o uso de *scripts*, não há a necessidade da digitação de comandos complexos.

4.2 Proposta de um laboratório de implementação de spa para ssh

A utilização de máquinas virtuais torna a montagem do ambiente dos testes bastante prática. Dependendo da capacidade de processamento do computador *host* das máquinas virtuais, várias podem ser criadas em uma única máquina física.

O modelo empregado no laboratório proposto simula uma empresa que possui um servidor *Web* instalado em *data center*. O servidor *Web* é administrado remotamente por meio do serviço de SSH.

Para avaliar o bloqueio da tentativa de acesso não autorizado ao servidor, enquanto ocorrem conexões legítimas, são necessários computadores que cumpram quatro papéis diferentes, sendo eles: o *firewall*, onde estará instalada a solução de SPA; o servidor, que receberá os pacotes de conexão do agressor e do cliente; o agressor, que tentará realizar os acessos não autorizados; e o cliente/administrador do serviço, que fará as conexões autorizadas.

Como tanto o agressor quanto o cliente/administrador do serviço farão acesso ao servidor, esses dois papéis podem ser atribuídos a um mesmo equipamento, tornando possível a utilização de apenas três máquinas virtuais.

Uma rede virtual entre o *firewall* e o servidor e outra entre o *firewall* e a máquina cliente/agressora são criadas no ambiente de virtualização.

São instalados nas máquinas virtuais os seguintes aplicativos e ferramentas: servidor da solução de SPA (*firewall*), cliente da solução de SPA (cliente/agressor), serviço de SSH (servidor), cliente de SSH (cliente/agressor), programa de geração de certificados digitais (*firewall* e cliente/agressor), servidor *Web* (servidor), navegador *Web* (cliente/agressor), *firewall* (*firewall*), programa de varredura de portas (cliente/agressor), programa de levantamento de vulnerabilidades (cliente/agressor) e programas que permitam o monitoramento da rede e das regras criadas pelo *firewall*.

A política padrão de filtragem a ser seguida no *firewall* é a de bloqueio geral para todos os pacotes que vêm da rede externa. O acesso ao serviço *Web* é liberado para todos os computadores da rede externa e as regras de acesso ao serviço de SSH são criadas dinamicamente pela solução de SPA.

Além das regras de filtragem, o *firewall* possui regra de *Network Address Translation*-NAT para o serviço *Web*. A regra de NAT para o serviço de SSH é criada dinamicamente pela solução de SPA.

A solução de SPA deve ser configurada no *firewall*.

Devem ser criados: os certificados digitais no *firewall* e no computador cliente/agressor, o *script* de acesso ao serviço de SSH no computador cliente/agressor e uma página *Web* de teste no computador servidor.

Os procedimentos de teste devem compreender: acesso ao serviço *Web*, a fim de se verificar a sua disponibilidade; varredura de portas no computador servidor, para que se possam relacionar as portas disponíveis para conexões; acesso do computador cliente/agressor à solução de SPA, que fará com que as regras de filtragem sejam criadas dinamicamente; monitoramento do tráfego de rede, para verificar a existência de criptografia no tráfego; conexão com serviço de SSH, objetivando constatar a disponibilidade do serviço, cujas regras permitindo o acesso ao recurso são controladas pelo aplicativo que implementa o SPA; verificação da criação dinâmica de regras de filtragem, que permitirá a constatação do funcionamento da solução; levantamento de vulnerabilidades no servidor, a fim de que possa ser analisado o nível de segurança proporcionado pela utilização da técnica de camuflagem; leitura dos arquivos dos *logs* do sistema operacional e do SPA, para que se possam ver as ações tomadas pela solução.

As ações propostas neste subcapítulo foram postas em prática no laboratório ao qual se refere o capítulo 5.

5 Laboratório de implementação: metodologia e análise

5.1 Ambiente

Foram criadas três máquinas virtuais, que assumiram os seguintes papeis:

- máquina Firewall = papel do Firewall;
- máquina Servidor = papel do servidor *Web*;
- máquina AgressorAdmin = papéis de agressor e de administrador.

A máquina hospedeira das máquinas virtuais possuía a seguinte configuração: 4 GBytes de memória RAM, 300 GBytes de disco rígido, processador Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz e placa de vídeo com 512 MBytes de memória (memória da própria placa de vídeo).

O sistema operacional da máquina hospedeira era o Windows XP com o Service Pack 3.

O aplicativo de virtualização utilizado era o Sun VirtualBox®, versão 3.2.10.

Foram criadas três máquinas virtuais, conforme foi explicado anteriormente, e dois seguimentos de rede virtuais chamados de intnet e intnet2. Esses seguimentos de rede não se comunicavam e, portanto, atuavam como redes distintas.

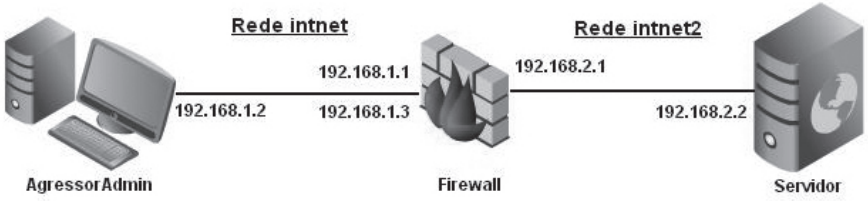
Cada máquina virtual possuía 512 MBytes de memória RAM, 10 Gbytes de disco rígido, 64 MBytes de memória de vídeo, interfaces de rede *FastEthernet* (100Mbps) e emulavam um único processador com um núcleo.

O sistema operacional instalado nas três máquinas virtuais era o Ubuntu GNU/Linux, versão 10.04.1 LTS, cuja versão do *kernel* era 2.6.32-25-generic, o Linux foi escolhido para ser o sistema operacional utilizado no laboratório, pois era muito utilizado mundialmente, possuía grande quantidade de *softwares* gratuitos disponíveis e era seguro.

As redes, o endereçamento e as rotas utilizados no laboratório (representados na Figura 2) atendiam ao seguinte *layout:*

- interface do computador AgressorAdmin na rede virtual intnet, configurada com o endereço IP 192.168.1.2;
- interface do computador Firewall na rede virtual intnet, configurada com os endereços IPs 192.168.1.1 e 192.168.1.3;
- interface do computador Firewall na rede virtual intnet2, configurada com o endereço IP 192.168.2.1;
- interface do computador Servidor na rede virtual intnet2, configurada com o endereço IP 192.168.2.2; e
- o computador AgressorAdmin se comunicava com diretamente com o computador Firewall e, via roteamento, com o computador Servidor.

Figura 3 - Rede entre as máquinas virtuais



Fonte: Do autor

Os aplicativos utilizados no laboratório foram:

- computador AgressorAdmin: fwknop v1.9.12 (file revision: 1533), cliente ssh OpenSSH_5.3p1 Debian-3ubuntu4, gpg (GnuPG) 1.4.10, Nmap 5.00, OpenVAS 2.0.5 e Firefox versão 3.6.12;
- computador Firewall: fwknopd v1.9.12 (file revision: 1533), servidor ssh OpenSSH_5.3p1 Debian-3ubuntu4, gpg (GnuPG) 1.4.10, iptables v1.4.4, iptstate Versão 2.2.1 e Wireshark 1.2.7 ; e

- computador Servidor: servidor ssh OpenSSH_5.3p1 Debian-3ubuntu4 e servidor *Web* apache 2.2.14.

A política de segurança que norteou a criação de regras no *firewall* é a que se segue:

- todas as conexões partindo do *firewall* são permitidas;
- os pacotes referentes a conexões já estabelecidas são permitidos;
- as conexões do *firewall* para o próprio *firewall* são permitidas (alguns serviços internos do sistema operacional necessitam dessa liberação);
- conexões originadas externamente para o servidor *Web* são permitidas, sendo que há a necessidade de conversão NAT do endereço público (rede externa) do servidor para o endereço de rede interna – o endereçamento interno não é divulgado; e
- todas as demais conexões são bloqueadas.

Os usuários do serviço SSH se autenticam utilizando certificados digitais.[12]

A solução de SPA utilizada no laboratório é o *fwknop*. Ela utiliza criptografia entre o cliente de SPA (AgressorAdmin) e o servidor de SPA *(Firewall)*. Para esse fim, são criadas chaves assimétricas para os computadores AgressorAdmin e Firewall.[13]

5.2 Roteiro da realização dos testes

As três máquinas virtuais foram postas em funcionamento pela console do VirtualBox.

Os serviços SSH e *Web* foram carregados no computador Servidor.

O acesso ao serviço *Web* do computador Servidor foi realizado com o navegador Firefox do computador AgressorAdmin e a página *Web* foi apresentada.

O aplicativo *nmap* foi executado no computador AgressorAdmin a procura de portas de serviço disponíveis no computador Servidor.

O cliente do serviço de *fwknopd* foi executado no computador AgressorAdmin, a fim de se verificar a mensagem com as informações sobre o comando envidado para o *firewall*. Nesse momento, não houve a intenção de se estabelecer nenhuma conexão.

A ferramenta *Wireshark f*oi iniciada no computador Firewall e os pacotes enviados pelo cliente do serviço de *fwknop,* vindos do computador AgressorAdmin, passaram a ser monitorados.

O *script* do cliente do serviço *fwknopd* foi executado no computador AgressorAdmin. A sessão de SSH foi estabelecida automaticamente com o computador Servidor.

O aplicativo *nmap* foi imediatamente executado no computador AgressorAdmin, para verificar a existência de alguma outra porta de serviço no computador Servidor além da porta 80 (*Web*), pois a sessão SSH ainda estava estabelecida.

O *script* do cliente do serviço *fwknopd* foi executado novamente no computador AgressorAdmin.

As regras do IPTables foram verificadas no computador Firewall.

Dado algum tempo, fez-se nova verificação das regras de filtragem no computador Firewall.

O aplicativo *iptstate* foi carregado no computador Firewall e as conexões foram verificadas.

Com a sessão de SSH ainda estabelecida, o aplicativo gráfico OpenVAS foi utilizado no computador AgressorAdmin à procura de falhas de segurança no computador Servidor.

Os procedimentos de teste foram repetidos várias vezes.

Vários dados foram gerados e registrados nos arquivos de *log* do sistema operacional Ubuntu/Linux e da própria ferramenta de SPA. Os dados do *log* do sistema operacional foram consultados, pois mostram as ações tomadas no momento em que os pacotes foram recebidos pelo servidor *fwknopd*. O segundo arquivo (*log* da solução *fwknop*) foi examinado em virtude de possuir os *digests* de cada pacote do comando de criação das regras de filtragem no *firewall*.

[12] Disponível em: < http://www.vivaolinux.com.br/artigo/Conexao-com-chaves-assimétricas-sem-uso-de-senha-em-servidor-sshd>. Acesso em: 15 maio 2011.

[13] Disponível em: < http://cipherdyne.org/fwknop/docs/gpghowto.html>. Acesso em: 15 maio 2011.

5.3 Resultados

a) Comando NMAP antes do acesso ao serviço SSH

- Resultado: O serviço Web foi detectado na porta 80. Embora o serviço de SSH estivesse presente, não se pôde detectar nada, pois não existiam regras no firewall liberando acesso a esse serviço.

b) Execução do cliente do serviço fwknopd

- Resultado: Entre os campos que compunham o pacote, existia informação gerada aleatoriamente (Random data) que, com a ajuda do selo de tempo (Timestamp), garantia que um pacote jamais fosse igual a outro. A exclusividade de cada

A porta para o acesso ao serviço SSH era escolhida aleatoriamente e fazia parte de uma regra de *Network Address Translation*-NAT criada no firewall.

c) Payload dos pacotes capturado no Wireshark

- Resultado: O comando enviado para a criação de regras de filtragem estava criptografado com a chave pública da máquina Firewall.

d) Resultado da execução do script que roda o cliente do serviço fwknopd

- Resultado: Após a digitação do PIN do certificado digital, o acesso ao serviço SSH foi estabelecido automaticamente na porta aleatória.

e) Execução do nmap com a sessão SSH estabelecida

- Resultado: Apenas o serviço Web continuou a ser detectado. Apesar de a conexão SSH estar ativa, a porta de serviço não podia ser notada.

f) Verificação das regras do firewall, logo após a conexão com o serviço SSH

- Resultado: Ao receber o comando do cliente do aplicativo Fwknop, o firewall criou regras de filtragem, permitindo o acesso ao recurso solicitado (serviço SSH), e criou regras de NAT, convertendo a porta gerada aleatoriamente para a porta do serviço SSH (porta 22).

g) Verificação das regras do firewall, algum tempo depois e com a sessão de SSH inda estabelecida

- Resultado: Após o tempo determinado nas configurações do serviço fwknopd, as regras de filtragem e de NAT foram apagadas e novas conexões para o serviço de SSH passaram a ser rejeitadas.

h) Verificação das conexões ativas com o aplicativo iptstate

- Resultado: Mesmo após as regras de acesso ao serviço SSH terem sido apagadas, a conexão do computador AgressorAdmin permaneceu ativa. Isso foi possível devido à existência de uma regra permitindo que as conexões já estabelecidas continuassem ativas (iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT).

i) Verificação de falhas em aplicativos do computador Servidor

- Resultado: A única porta que pôde ser analisada foi a porta 80 do serviço Web, mas, mesmo assim, não foram encontradas vulnerabilidades nesse serviço, uma vez que o seu software estava devidamente atualizado.

j) Log do arquivo /var/log/messages no computador Firewall

- Resultado: O relatório de log informou a detecção do pacote de comandos e as ações tomadas para atender à solicitação de acesso. As regras criadas bem como as regras que foram remo-

vidas eram apresentadas com todos os seus detalhes.

k) Log do arquivo / var/log /fwknop/digest. cache no computador Firewall

- Resultado: Os resultados colhidos nesse relatório de log mostraram que os digests (resultados dos algoritmos de hash) eram diferentes para cada comando que foi enviado para o firewall. Isso deixou claro que as informações enviadas em cada pacote do cliente do aplicativo Fwknop eram diferentes.

6 Considerações finais

Este trabalho apresentou uma solução que tenta resolver algumas deficiências específicas da segurança de perímetro das redes.

Geralmente, o *firewall* é a ferramenta que possui o encargo de proteger o perímetro das redes corporativas (CHESWICK; BELLOVIN; RUBIN, 2005, p. 30). Northcutt et al. (2002, p. 5) também indicam os sistemas de detecção de intrusão como auxiliares do *firewall* em sua missão.

Ambas as ferramentas citadas possuem deficiências, podendo ser "enganadas" em alguns casos. A técnica de camuflagem conhecida como *Single Packet Authorization* - SPA evita a exploração dessas falhas.

A comprovação da eficiência do SPA se deu com a avaliação do aplicativo *Fwknop*. Foram precisos alguns ajustes no sistema operacional, no *firewall* e no serviço SSh, a fim de que a solução apresentasse a eficiência desejada. Após os ajustes necessários, os objetivos foram atingidos; porquanto, foi possível realizar a administração remota do servidor *Web* e quaisquer tentativas de detecção do serviço SSh foram infrutíferas, ou seja, o serviço referido permaneceu disponível para o acesso autorizado e, ao mesmo tempo, estava invisível (camuflado) para os demais usuários da rede.

Apesar do sucesso na ocultação do serviço, a camuflagem proporcionada pelo SPA não é completa. O tráfego entre o cliente e o servidor poderia ser monitorado e o padrão de comunicação do protocolo permitiria a dedução do serviço que é acessado.

Os pontos positivos da camuflagem com SPA podem ser assim identificados:

As regras de filtragem criadas no firewall e apagadas automaticamente após um curto período de tempo proporcionam uma camuflagem eficiente;

A criptografia das mensagens enviadas para o firewall impede o acesso às informações pertinentes à criação dinâmica das regras de filtragem;

A geração aleatória de algumas informações que compõem os comandos enviados para o firewall faz com que cada pacote seja exclusivo. Dessa forma, evita-se a realização de um ataque de repetição; e

A não disponibilização de soquetes e a ausência de reposta para os pacotes recebidos pela solução de SPA ajudam a autopreservar o serviço que implementa a solução.

E como pontos negativos da camuflagem com SPA, podem ser citados:

Não há proteção dos serviços contra a análise do tráfego e os ataques do tipo man-in-the-middle, sendo aconselhável o uso de algum tipo de criptografia na comunicação entre o cliente e o serviço acessado; e

O computador cliente que utiliza a solução não é tratado e pode ser comprometido, o que permitiria a um hacker acessar o serviço camuflado.

Referências

CHESWICK, Willian R.; BELLOVIN, Steven M.; RUBIN, Aviel D. *Firewalls e segurança na Internet***:** repelindo o hacker ardiloso. 2. ed. Porto Alegre: Bookman, 2005.

COMER, Douglas E. *Interligação em rede com TCP/IP:* princípios, protocolos e arquitetura**.** 3. ed. Rio de Janeiro: Campus, 1998. v. 1.

COMER, Douglas E. *Redes de computadores e internet.* 4. ed. Porto Alegre: Bookman, 2007.

DEGRAAF, Reinderd G. N. *Enhancing firewalls:* conveying user and application identification to network firewalls. 2007. Dissertação (mestrado). The University of Calgary, 2007. Disponível em: <http://www.ciphertext.info/papers/thesis-degraaf.pdf>. Acesso em: 27 maio 2012.

ERICKSON, Jon. *Hacking:* the art of exploitation. 2. ed. San Francisco: No Starch Press, 2008.

FUCHSBERGER, Andreas. Intrusion detection systems and intrusion prevention systems. *Information Security Technical Report*, Londres, n. 10, p. 134 – 139. 2005. Disponível em:<http://faculty.kfupm.edu.sa/ics/salah/misc/courses/cse551/slides/IDS%20and%20IPS.pdf>. Acesso em: 27 maio 2012.

ISABEL, Dawn. Port Knocking: beyond the basics. *InfoSec Reading Room,* v. 9, mar. 2005. Disponível em: </www.sans.org/reading_room/whitepapers/sysadmin/port-knocking-basics_1634 >. Acesso em: 25 maio 2012.

JEANQUIER, Sebastien. *Analysis of port knocking and single packet authorization.* 2006. Dissertação (Mestrado)-University of London. Disponível em: <http://www.securitygeneration.com/wp-content/uploads/2010/05/An-Analysis-of-Port-Knocking-and-Single-Packet-Authorization-Sebastien-Jeanquier.pdf>. Acesso em: 26 maio 2012.

KRZYWINSKI, M. Port Knocking**:** Network authentication across closed ports. *SysAdmin Magazine 12*, p. 12–17, dez. 2003. Disponível em: <http://www.portknocking.org/>. Acesso em: 03 dez. 2010.

LEMOS, Robert. Admins warned of brute-force SSH attacks. *SecurityFocus,* maio 2008. Disponível em: <http://www.securityfocus.com/news/11518>. Acesso em: 21 nov. 2010.

MCCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. *Hacking exposed:* network security secrets & solutions. 5. ed. Emeryville: McGraw-Hill, 2005.

NORTHCUTT, Stephen et al. *Desvendando segurança em redes.* Rio de Janeiro: Campus, 2002.

RASH, Michael. Single packet authorization with fwknop. *Usenix login:* Magazine, v. 31, n. 1, fev. 2006. p. 63-69. Disponível em: <http://www.usenix.org/publications/login/2006-02/pdfs/rash.pdf>. Acesso em: 02 dez. 2010.

TANENBAUM, Andrew S. *Redes de computadores.* 4. ed. Rio de Janeiro: Elsevier, 2003.

Anexo A

Alguns comandos utilizados e arquivos verificados durante a realização do laboratório

Execução do aplicativo nmap à procura de portas de serviço no computador Servidor: nmap 192.168.1.3

Execução do cliente fwknop, para verificar a mensagem com o comando envidado para o firewall: fwknop -A tcp/22 --gpg-recip EE2E4F82 --gpg-sign A7B82C42 --NAT-rand-port --Forward-access 192.168.2.2 -a 192.168.1.2 -D 192.168.1.1

Verificação das regras do firewall IPTables: iptables –nvL e fwknopd --fw-list

Execução do aplicativo Iptstate: iptstate

Arquivo de log do Ubuntu/Linux: /var/log/messages

Arquivo de log do fwknop: /var/log/fwknop/digest.cache

Anexo B

Configuração da ferramenta da solução de spa

Para que camuflagem possa ser posta em prática, devem ser realizadas as configurações a seguir:

a) Arquivo */etc/fwknop/access.conf* do computador Firewall

- Autorização do serviço SSH

OPEN_PORTS: tcp/22;

- ID do certificado digital do cliente

GPG_REMOTE_ID: A7B82C42;

- ID do certificado digital do servidor.

GPG_DECRYPT_ID: EE2E4F82;

- Diretório onde se encontram os arquivos do gnupg.

GPG_HOME_DIR: /root/.gnupg;

- PIN necessário para a utilização da chave privada do servidor

GPG_DECRYPT_PW: posgraduacao;

- Tempo, em segundos, de duração das regras de filtragem no firewall

FW_ACCESS_TIMEOUT: 30;

- Autorização para a criação de regras na chain FORWARD

ENABLE_FORWARD_ACCESS: Y;

b) Arquivo *fwknop.conf* do computador Firewall

- Nome do servidor fwknopd

HOSTNAME Firewall;

- Tipo de firewall com o qual o serviço fwknopd irá trabalhar

FIREWALL_TYPE iptables;

- Modo de captura das mensagens do cliente fwknop

AUTH_MODE PCAP;

- Interface que será monitorada pelo aplicativo PCAP

PCAP_INTF eth1;

- Permissão para que o aplicativo PCAP trabalhe em modo promíscuo

ENABLE_PCAP_PROMISC Y;

- Faixa de portas e protocolo analisados pelo aplicativo PCAP

PCAP_FILTER udp dst portrange 10000-65535;

- Instrução para a utilização da chain FORWARD

ENABLE_IPT_FORWARDING Y;

- Tamanho máximo, em bytes, dos pacotes que serão analisados

MAX_SNIFF_BYTES 1500;

- Localização dos arquivos do gnupg

GPG_DEFAULT_HOME_DIR /root/.gnupg;

- Parâmetros que serão usados na criação das regras no firewall IPTables

IPT_INPUT_ACCESS ACCEPT, src, filter, INPUT, 1, FWKNOP_INPUT, 1;

IPT_FORWARD_ACCESS ACCEPT, dst, filter, FORWARD, 1, FWKNOP_FORWARD, 1;

IPT_DNAT_ACCESS DNAT, src, nat, PREROUTING, 1, FWKNOP_PREROUTING, 1;

c) Script *ConectaFwnop*, utilizado pelo computador AgressorAdmin, para criação e envio de comando ao *firewall*

```
# Mensagem solicitando a digitação de senha.

echo " "

echo "Digite a senha do certificado digital:"

# Comando de criação de regra de filtragem, com o retorno do resultado sendo armazenado pela variável denominada porta.

porta=`fwknop -A tcp/22 --gpg-recip EE2E4F82 --gpg-sign A7B82C42 --NAT-rand-port--Forward-access 192.168.2.2 -a 192.168.1.2 -D 192.168.1.1 | grep Request | cut -d" " -f14`

# No caso de a senha ter sido digitada corretamente, a variável denominada porta armazenará o número da porta do servico SSH. Sendo assim a estrutura de decisão irá realizar o acesso ao serviço mencionado.

# Caso a senha esteja incorreta, uma mensagem de erro será apresentada ao usuário.

if test `echo $((porta+1))` -gt 1

then

 echo " "

 echo " "

 echo "Abrindo conexao com o servidor de ssh..."
```

```
 echo " "
 echo " "
 ssh -p $porta -2 root@192.168.1.3
else
 echo " "
 echo "**********************************
**********************************
**********"
 echo "* A senha do certificado digital
foi digitada incorretamente. O pro-
grama sera finalizado. *"
 echo "**********************************
**********************************
**********"
 echo " "
fi
```