



Segurança Informática

Aula 9

Licenciatura em Engenharia Informática
Licenciatura em Informática Web

Sumário

Vulnerabilidades e ataques em redes locais e de larga escala. Análise do serviço de *Domain Name System* (DNS) como fonte importante de informação arquitetural de uma rede, foco e alavanca de ataques.

Computer Security

Lecture 9

Degree in Computer Science and Engineering
Degree in Web Informatics

Summary

Vulnerabilities and attacks in local and wide area networks. Analysis of the Domain Name System (DNS) service as an important source of network architecture information, and as a focus and starting point of attacks.

1 Contexto

Scope

A aula anterior focava-se no tema das vulnerabilidades em sistemas distribuídos, ou seja, nas máquinas terminais, sistemas operativos ou serviços instalados nessas máquinas, nomeadamente a nível da sua administração e implementação. Nesta aula converge-se para **as vulnerabilidades inerentes à interação entre esses sistemas sobre uma rede**, tanto **local** como de **larga escala**, e que se referem tanto a **fragilidade dos protocolos de comunicação** (nomeadamente protocolos de transporte e de rede), como às próprias **infraestruturas de suporte**, como é o caso do **Domain Name Server** (DNS).

Não se irão abordar vulnerabilidades a nível aplicacional, já que tal constitui, hoje em dia, uma tarefa imensa (estas serão alvo de estudo noutra cadeira). Ainda assim, haverá uma breve menção a problemas associados ao correio eletrónico. O foco estará sobretudo colocado sobre a suite de comunicações *Transmission Control Protocol/Internet Protocol* (TCP/IP), que é o **padrão de comunicação usado à escala mundial**.

2 Levantamento de Informação Arquitetural e Resolução de Endereços

Collecting Architectural Information and Address Resolution

2.1 Levantamento de Informação Arquitetural

Collecting Architectural Information

A **informação arquitetural** consiste na **descrição das máquinas** que fazem parte de uma rede e qual **a função que nela desempenham**. O **normal funcionamento de**

uma rede, bem como o **sucesso de potenciais ataques depende** normalmente desta informação. Como já foi mencionado antes, um ataque é tipicamente precedido por mecanismos de levantamento desta informação. Por isso, quando se desenham ou administram redes, é necessário **ter ambos os factos em conta**.

Uma das **principais fontes de informação** arquitetural são os **registos dos domínios DNS**. O DNS não fornece apenas serviços de resolução de nomes, como também **serviços informativos para gestão de rede** (por exemplo HINFO, que faculta informação acerca dos anfitriões da rede, como o processador e o sistema operativo que estão a correr)¹.

Repare que **o DNS é uma fonte succulenta de informação porque**, mesmo que não debite mais informação do que o seu serviço principal permite (i.e., resolução de nomes), basta pedir ao DNS de determinada rede privada que nos dê **os nomes dos endereços IP** que sabemos para **induzirmos a sua função na rede**. O procedimento de um atacante será semelhante a:

1. Faz um **varrimento aos endereços IP** (IP sweep) para saber todos os endereços que respondem a pedidos (obtém-se uma ideia da topologia da rede);
2. **Pergunta-se ao DNS quais os nomes do endereços** que obtivemos, de forma a deduzir-se o papel que as respetivas máquinas desempenham.

E.g., ao chegar a uma rede, um atacante começa por emitir *Internet Control Message Protocol* (ICMP) *echo requests* (vulgo PINGs) para todas as máquinas, obtendo resposta de várias delas, inclusive uma com endereço 192.168.1.253. Depois de emitir o comando `$ nslookup 192.168.1.253`, obtém o nome

¹ `$ nslookup -type=A www.di.ubi.pt` não mostra informação acerca do servidor do DI porque o administrador assim o quis.

estrela.di.ubi.pt, deduzindo que se deveria tratar de algum servidor.

2.2 Resolução de Endereços e de Nomes – DNS

Address and Name Resolution – DNS

O serviço DNS é **central ao bom funcionamento** da Internet. As **pessoas não decoram**, normalmente, **endereços IP**, mas **sim nomes** que lhes parecem lógicos, mas estruturados hierarquicamente: e.g., `www.di.ubi.pt/~inacio/`. Quando se começa uma comunicação na Internet, o cliente emite tipicamente um pedido DNS com o nome, que é respondido com o endereço IP da máquina a que corresponde esse nome. E.g., quando navega até um sítio na Internet, um utilizador começa por escrever um *Uniform Resource Locator* (URL) com palavras, não um endereço IP.

No sentido **direto**, este serviço chama-se **resolução de nome DNS**. No sentido **contrário** dá-se o nome de **resolução de nomes inversa** (*reverse name lookup*).

Por ser central ao bom funcionamento da Internet e redes em geral, o **DNS** é também **um polo de atração para ataques ou atividades maliciosas**, podendo estruturarem-se as razões para esta atração da seguinte forma:

1. Explorando o seu normal funcionamento, um **sujeito mal intencionado pode obter informação arquitetural** de uma rede;
2. Se o **serviço for comprometido, pode permitir personificação** de entidades na Internet.
3. Se o **serviço for atacado** de modo a negar o serviço que fornece, **outras aplicações Internet falharão**.
4. Os **servidores DNS existem em todo o lado e podem ser usados na amplificação de ataques** distribuídos e refletidos.

2.3 Resolução Errada de Nomes DNS

DNS Spoofing

Uma das **formas mais simples de usar o DNS contra utilizadores** da Internet (ou de servidores em redes privadas) **consiste em registar e divulgar nomes DNS enganadores**, de modo a redirecionar os clientes para sites maliciosos. E.g., divulgar o site `www.whitehouse.gov`, quando o site real é `www.whitehouse.org`. A forma mais elaborada de otimizar o efeito supramencionado consiste em levar a cabo **um ataque** designado por **DNS spoofing**. Este tipo de ataques é possível porque, **quando o DNS foi desenhado, não havia preocupações com os ataques** a que hoje está sujeito. O protocolo contemplava aspetos de eficácia, eficiência e escalabilidade. A técnica mais comum para efetuar **DNS spoofing** recorre ao **envenenamento da cache DNS** (*DNS cache poisoning*), e explora partes do funcionamento do DNS, nomeadamente:

1. Para além da memória persistente onde se guardam os registos locais, **todos os servidores DNS têm uma memória (a cache) onde guardam**, durante algum tempo, os **registos dos últimos pares que traduziram**, nomeadamente aqueles para os quais não possuem registo e que tiveram de perguntar a outro DNS.
2. Quando recebem um pedido de resolução de nome ou endereço, um servidor **DNS procura primeiro nos seus registos e, caso não possua o que lhe diz respeito, faz uma query com um identificador** (um número) a outro servidor (mais alto na cadeia hierárquica), à espera de resposta satisfatória. O **problema** é que os pedidos e as respostas são feitas sobre *User Datagram Protocol* (UDP) e **autenticadas de uma forma fraca**.

Considere que um atacante consegue escutar a rede onde está a atuar ou parte dos dispositivos onde passam os pedidos DNS. Para tentar envenenar a *cache* DNS, procede da seguinte forma:

1. Numa primeira fase, envia vários pedidos DNS para endereços que sabe **que não existem** (e.g., `www.algures.net`, `www.nenhures.net`, etc.) e observa o servidor DNS local enviar dezenas de pedidos para outro servidor com o tal identificador. Esta fase também serve para limpar a *cache* do DNS.
2. Numa fase intermédia ou de forma paralela à fase anterior, o atacante **tenta perceber qual é o algoritmo usado para criar o identificador** dos pedidos DNS.
3. Numa terceira fase, **envia um pedido para o servidor DNS local** (e.g., `www.millenniumbcp.pt`) e, logo após, **envia várias respostas com vários identificadores** (um deles deve servir) para o DNS **com uma resolução errada** do URL. Para maximizar o seu proveito, pode publicitar um endereço IP de uma máquina controlada por ele. A partir desse momento, **quem tentar aceder ao URL, vai obter e ser redirecionado para um IP comprometido** (e provavelmente controlado pelo atacante, que vai tentar fazer *fishing* da informação).

2.4 Resolução Errada de Nomes DNS – Soluções

DNS Spoofing – Solutions

O problema apontado antes parece ter os seus **pontos mais fracos no identificador e na fraca autenticação** (se assim pode ser considerada) dos pedidos e resposta DNS. Assim, algumas propostas apontadas para a resolução deste problema passam por:

1. Utilizar **identificadores aleatórios** entre pedidos, que parece ser uma ideia interessante, mas que se baseia num princípio que pode ser falível.

2. Utilização de **chaves públicas e certificados para autenticar clientes, servidores e conteúdos nas comunicações DNS**, que acarreta **custos computacionais elevados**, nomeadamente armazenamento de chaves públicas, certificados e cadeias de certificados. Esta solução necessita que os servidores dediquem tempo de processamento a verificação de cadeias de certificação e assinaturas digitais, por cada pedido que recebem ou façam.

2.5 Resolução de Endereços e de Nomes – MAC

Address and Name Resolution – MAC

As comunicações entre cada dois nós numa rede *Ethernet* fazem-se através de um endereço *Medium Access Control* (MAC). Para comunicar, o computador constrói uma trama *Ethernet*, onde insere o pacote IP. No cabeçalho da trama vão os endereços MAC fonte e destino, correspondentes à máquina fonte e destino dentro da rede local. Repare: (i) **no cabeçalho do pacote IP vai o endereço IP destino da máquina terminal**; (ii) **no cabeçalho da trama *Ethernet*, vai o endereço MAC destino da próxima máquina na comunicação**. Caso o computador não tenha o MAC destino relativo ao IP destino do pacote encapsulado ou do próximo nó na comunicação, **envia um pedido *Address Resolution Protocol* (ARP) para o endereço *broadcast* (difusão)**. A máquina que tem o endereço IP em questão, **envia-lhe o seu MAC**.

2.6 Obtenção Errada de Endereços MAC

MAC Spoofing

Cada sistema operativo guarda uma *cache ARP*², que tem o mapeamento entre o endereço IP e o respetivo endereço MAC. Esta tabela vai sendo preenchida das mais variadas formas, nomeadamente **através de respostas a pedidos ARP**.

Para envenenar a *cache*, basta que o atacante faça chegar uma resposta a um pedido ARP com **informação falsa ao recetor**. Por exemplo, considere que o seu computador emitia um pedido ARP para o endereço MAC relativo ao **IP do servidor DNS e que o atacante lhe enviava uma resposta forjada** (*ARP response*) dizendo que o endereço MAC relativo àquele IP era **o dele**. A partir daí, e enquanto o sistema operativo tivesse aquela entrada errada na *cache ARP*, iria sempre enviar pedidos DNS para o computador do atacante.

Note que se costuma usar a palavra **envenenamento porque o efeito do ataque persiste enquanto a memória (*cache*) não for limpa**. É muito difícil impedir ações de envenenamento de *caches ARP*. Pode-se tentar **minimizar o seu impacto através da monitorização da rede e da deteção de situações anómalas** (e.g., várias respostas ARP para o mesmo pedido e vindas de endereços MAC diferentes ao mesmo tempo).

²O comando `$ ip neigh` mostra a tabela ARP em sistemas operativos Linux.

A monitorização de toda a rede de área local só é possível em pontos centrais (e.g., *switch*), que têm de possuir uma porta de *mirroring* para o efeito ou suportar funcionalidades adicionais sendo, portanto, mais caras.

3 Confidencialidade

Confidentiality

Uma das motivações para o grande desenvolvimento de técnicas da criptografia moderna deve-se à notável adoção e crescimento das redes de computadores, que constituem um *habitat* natural para estas técnicas. É natural o requisito/pilar da confidencialidade tomar especial relevância neste contexto.

3.1 Confidencialidade de Interações em Rede

Confidentiality in Network Interactions

A **confidencialidade de interações em redes** pode ser analisada de **dois** ângulos diferentes:

- Confidencialidade **do tráfego**, que se prende com a preocupação **em esconder os interlocutores finais de uma comunicação**. De notar que as versões iniciais dos protocolos de comunicação não contemplavam mecanismos para assegurar este tipo de confidencialidade, e tal facto ainda hoje se mantém parcialmente inalterado.
- Confidencialidade **de conteúdos**, que se prende com a preocupação **em esconder dados úteis trocados entre os interlocutores**. Existe hoje uma grande preocupação com a confidencialidade dos conteúdos que circulam na Internet. Essa preocupação reflete-se no esforço de normalizar soluções de segurança e numa **proliferação de protocolos e aplicações de segurança**.

A proteção dos interlocutores é normalmente conseguida através de **mecanismos de encapsulamento e desencapsulamento de pacotes em túneis cifrados** (redes privadas virtuais). O **projeto Tor**³ (e aplicações associadas) é um exemplo de **um esforço feito na direção da proteção de interlocutores no acesso a servidores ou até em redes *Peer-to-Peer* (P2P) para *file sharing***.

Existem inúmeras **ferramentas de inspeção de tráfego e de conteúdo**. Algumas dessas ferramentas são **gratuitas** (e.g., *Wireshark*) e **open source** (e.g. *tcpdump*). Caso não se tenha acesso físico a determinado troço da rede para colocar a escuta, podem-se tentar técnicas de redirecionamento como as estudadas antes (e em baixo) para colmatar o problema. Outrora, bastava observar o tráfego que fluía na rede para obter segredos como palavras- ou frases-chave (hoje em dia ainda é possível

³<https://www.torproject.org/>

fazer isso nalguns casos)⁴.

O **risco** associado à possibilidade de inspeção de tráfego **varia de acordo com o que é transmitido**. Contudo, a **possibilidade de alguém obter palavras- ou frases-chave** de acesso a serviços privilegiados é uma das que normalmente **causa mais preocupação**.

3.2 Captura de Palavras-passe

Password Capturing

A **autenticação usando uma palavra-passe memorizada** é uma das formas **mais antigas de identificação de entidades**, mesmo antes de existirem computadores. Aliás, **antes até existia o hábito de haver uma contra-senha que autenticava o autenticador**. A autenticação por palavras-passe **baseia-se na partilha de um segredo entre duas pessoas**, e na **demonstração**, por parte de **uma delas**, que **sabe esse segredo** (*something you know factor*). Muitas aplicações distribuídas que utilizam a Internet adotaram o **mesmo mecanismo de autenticação**, mas a **segurança do canal** utilizado para trocar a palavra ou frase-chave **não está garantida à partida**. Nas aplicações computacionais de hoje em dia, há não só que **garantir que o caminho que transporta a palavra-passe entre o autenticado e o autenticador é seguro**, como também que as palavras- ou frase-passe são **guardadas e manuseadas de forma segura em ambos os lados da comunicação**. E.g., se estivermos a escrever a senha num terminal para autenticação, estamos perante um cenário inerentemente inseguro.

O **risco** de captura das palavras-chave **pode ser reduzido através de soluções como:**

1. **Cifrar** as comunicações antes de transmitir a palavra- ou frase-chave.
2. Usar **palavras-chave únicas** já que, se a palavra-chave for única, quem a escuta não a poderá usar no futuro. Para isto utiliza-se um método para derivar palavras-chave de uma palavra-chave inicial.
3. Usar **protocolos de troca de desafios e respostas**, em que o autenticador normalmente envia um desafio, e o autenticado responde com algo que só quem tem a palavra-passe correta conseguiria derivar. Um exemplo deste protocolo é o conhecido *Challenge-Handshake Authentication Protocol* (CHAP), em que
 - (a) Bob: gera $c \xleftarrow{r} C$;
Bob gera um número aleatório (um desafio);
 - (b) Bob→Alice: c
Bob envia c à Alice;

⁴Em sistemas operativos da família Linux com o `tcpdump` instalado, costuma ser suficiente fazer `$ sudo tcpdump -i wlan0 -s 1500 -A -v -t` para observar o tráfego que chega ou sai da máquina via *interface* wlan0.

(c) Alice: calcula $r = \text{SHA1}(c || \text{palavra-passe} || \text{minuto})$;

(d) Alice→Bob: r
Alice envia a resposta r ao Bob;

(e) Bob: verifica e dá acesso se
 $r = \text{SHA1}(c || \text{palavra-passe} || \text{minuto})$

Desta forma, a **palavra-chave nunca vai em claro**, e só quem a tem é que consegue responder ao desafio. Mais, se uma resposta for escutada, **não pode ser usada em autenticações futuras**, porque o **desafio será potencialmente diferente**.

4. Usar sistemas complexos de distribuição de credenciais cifradas com a palavra ou frase chave. Neste esquema, a **palavra-chave é pré-distribuída durante a instalação do sistema**. A partir daí, a palavra-chave **só é usada para cifrar credenciais que são usadas para autenticar o cliente noutros servidores**.

3.3 Senhas Fortes e Ataques de Dicionário

Strong Passwords and Dictionary Attacks

O uso de mecanismos para cifragem ou ocultação da palavra-chave **não vale de nada se a palavra-chave for fraca em termos de facilidade de adivinhação por parte alheia**. Por arrasto, as palavras-chave **costumam ser palavras ou combinações de palavras fáceis de lembrar para um humano**, o que **facilita** a tentativa de um atacante adivinhar essa palavra-chave. De modo a reduzir o universo de palavra-chave possíveis, um atacante **pode tentar palavras ou combinações de palavras de um dicionário de palavras**, em vez de tentar todas as combinações de caracteres possíveis para o tamanho da senha. Este ataque é conhecido como **Ataques usando Dicionários**. Infelizmente, o recente **comprometimento de grandes bases de dados de palavras-passe** ⁵ tem ajudado a construir **dicionários com milhões de combinações** usadas por **humanos**. Por outro lado, o **desenvolvimento tecnológico**, nomeadamente de **processadores com funções otimizadas**, ditam **um débito de teste cada vez maior de palavras-passe contra resumos ou cifras das mesmas**.

Note que, na grande maioria das situações, quando mais o atacante sabe da vida pessoal de uma vítima, mais hipóteses tem de adivinhar as suas senhas (e.g., imagine que sabe que a vítima gosta de banda desenhada da DC Comics...).

⁵Ver, e.g., <http://www.zdnet.com/blog/btl/6-46-million-linkedin-passwords-leaked-online/79290>.

4 Autenticidade

Authenticity

A **autenticidade de interações** em redes constitui outro dos grandes problemas associados a este diversificado *habitat*, e pode também ser analisada de **dois** ângulos diferentes:

1. Autenticidade **de interlocutores**, que se prende com a prova de que **cada um dos intervenientes da comunicação é realmente quem diz que é**, e não um impostor.
2. Autenticidade **de conteúdos**, que se prende com a **prova ou a certeza de que os conteúdos da comunicação chegam ao destino tal como foram enviados** (i.e., não foram modificados em trânsito). Prende-se ainda com a possibilidade de detectar que não foram injetadas mensagens ou repetidas de forma fraudulenta durante a comunicação.

Como apontado antes, as versões iniciais dos protocolos de comunicação em redes de computadores não dispunham de mecanismos para garantia de autenticidade. Por exemplo, **o Protocolo IP não garante, por defeito, a autoria dos endereços nos pacotes** (e.g., pode-se fazer IP *spoofing*).

A autenticidade dos interlocutores e conteúdos pode ser assegurada usando **mecanismos da criptografia moderna** já estudados, nomeadamente **Message Authentication Codes (MACs)** e **Assinaturas Digitais**. Para motivar a necessidade destes mecanismos, descrevem-se alguns ataques que derivam de problemas de autenticidade.

4.1 Pedidos Fraudulentos sobre UDP

Fake Requests Using UDP

Uma das **técnicas mais elementares** de efetuar controlo de acesso em redes IP é **usando o valor do endereço IP da origem de um pedido**. Este controlo consiste em **negar o serviço aos IPs que não estão definidos na tabela de clientes autorizados do serviço**. O problema que está normalmente associado a esta técnica é que **o controlo de acesso só é normalmente efetuado para o primeiro ou primeiros pacotes do pedido**, i.e., durante o serviço propriamente dito, não é mais verificado o IP. Uma das formas de subverter este mecanismo consiste, então, em **enviar um pedido com o endereço IP falsificado** (*spoofed*) e em arranjar maneira de observar a resposta para saber como obter o serviço.

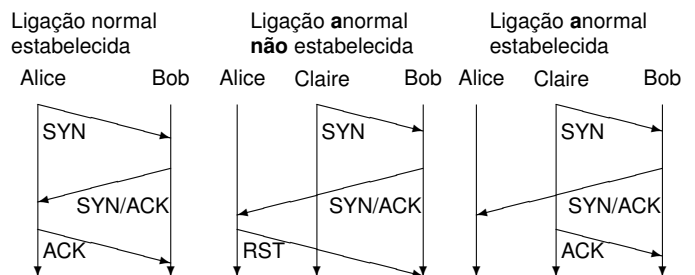
Considere o protocolo *Network File System* (NFS), usado para acesso a ficheiros distribuídos e que corre sobre UDP. Quando **recebe um pedido de uma máquina autorizada**, o protocolo **responde com um *file handler* específico para aquele IP**. O cliente pode então usar esse *file handler* para **aceder à diretoria ou ficheiro daí para a frente**. Para um atacante, bastava enviar um **pedido bem formatado, mas com IP fonte falsificado**, e

observar (de alguma forma) a resposta contendo o *file handler*. Depois disso, era só usar esse *file handler* para aceder ao ficheiro.

4.2 Iniciação Fraudulenta de Ligações TCP

Fake Initiation of TCP Connections

A **inicialização fraudulenta de ligações TCP é mais difícil que para UDP**. Se enviarmos um pacote com endereço fonte forjado, teremos de observar a resposta e possivelmente **continuar a forjar pacotes**, já que a **ligação não é estabelecida se não houver um *acknowledgement* da parte da entidade que iniciou a ligação**. Só é possível manter a farsa **se o IP que estamos a falsificar não estiver presente**, caso contrário, basta um RST para terminar a ligação ilegítima. Caso não consiga



escutar a resposta emitida pelo Bob para obter o valor de sequência enviado no SYN/ACK, a Claire pode sempre tentar responder com vários ACKs falsificados, na esperança que um deles tenha o valor de sequência correto.

4.3 Redirecionamento de Tráfego IP – ICMP Redirect

Redirecting IP Traffic ICMP Redirect

O **redirecionamento de tráfego é uma técnica que pode complementar as antes enunciadas**. Para além dos envenenamentos referidos anteriormente, há outras técnicas que podem ser utilizadas para o mesmo efeito. Por exemplo, os pacotes do tipo *Internet Control Message Protocol* (ICMP) **redirect** permitem que **um gateway mude o endereço de gateway que um host está a utilizar**. Se o atacante forjar um destes datagramas e o enviar para um *host* com o endereço fonte do seu *gateway* actual, pode conseguir que esse *host* altere o *gateway* para, e.g., uma máquina controlada por si. O resultado disto já é o esperado.

4.4 Redirecionamento de Tráfego IP – via Source Route

Redirecting IP Traffic – via Source Route

O protocolo IP especifica **uma opção chamada Source Route**, que permite **definir a rota que um pacote IP deve seguir em termos de routers** (e.g., 192.168.0.254 → 133.22.22.1 → 133.233.1.25 → 93.0.93.150). Se o sistema operativo de uma máquina *host* responder a um pacote IP **invertendo a rota definida no pacote**, então

um atacante pode enviar um pacote com **endereço fonte falsificado** (para parecer que vem de alguém de confiança), e com uma **rota que inclua uma máquina que consegue escutar**. Por este motivo, **esta opção costuma ser desativada** a nível do sistema operativo.

4.5 Problemas de Autoria em Correio Electrónico *Authorship Problems in E-mails*

O protocolo que suporta o envio de correio electrónico, designado por *Simple Mail Transfer Protocol (SMTP)*, tem um **mecanismo fraco de verificação da autenticidade e origem das mensagens**. Normalmente, os servidores **verificam apenas o endereço IP de origem da mensagem** para se assegurarem que a mensagem vem do domínio que estão a servir. Um nome de utilizador / palavra-passe pode ainda servir para autenticar o cliente perante o servidor, mas **o conteúdo dos campos do e-mail são, frequentemente, de exclusiva autoria do emissor**. Assim, campos como *From* e *Reply-to* podem ser **preenchidos ao critério do autor da mensagem, seja esse autor humano ou virtual (vírus)**.

O problema antes mencionado é **exaustivamente explorado por** dois tipos de ameaças, nomeadamente **spammers e vírus**:

1. Os **spammers** colocam endereços de *e-mail* falsos no campo *From* e *Reply-to*, de modo a que **não sejam identificados como fonte de spam**.
2. Os **vírus** fazem uso das listas de endereços de um cliente de *e-mail* para se propagarem, enviarem *spam* e **evitam que um aviso de infeção volte para a máquina infetada**. Repare que, se o vírus colocar um endereço falso nos dois campos mencionados em cima, qualquer aviso de infeção vão para uma máquina que quase certamente não está infetada (porque não é a origem real do *e-mail*).

5 Ataques à Prestação de Serviços

Denial of Service Attacks

Existem **inúmeras motivações** para ataques à prestação de serviços (mais conhecidos por *Denial of Service (DoS) Attacks*), mas **o resultado desejado é sempre, ou quase sempre, impedir que uma empresa ou serviço consigam operar normalmente e prestar um serviço** aos seus clientes. Os ataques DoS seguem normalmente uma de **três estratégias**:

1. **Exploração de vulnerabilidades** conhecidas que induzam **o sistema vítima numa falha** (e.g., *Ping-of-Death*);
2. **Inundação dos servidores** com pedidos bem formados, mas falsos;
3. **Inundação da rede de acesso** de um servidor com tráfego inútil.

5.1 Ataques DoS e DoS Distribuídos

DoS and Distributed DoS Attacks

Se os ataques DoS são **levados a cabo usando várias máquinas**, então designam-se por *Distributed Denial of Service (DDoS) attacks*. Estes ataques **surgiram em larga escala em 1999** e são usados, sobretudo, para efetuar **ataques baseados em inundações**. A ideia de um DDoS é a de obter um **efeito amplificado do ataque DoS original**.

A **amplificação** de ataques pode ser conseguida de duas formas básicas:

1. Usando as chamadas **Botnets**, **redes** constituídas por **computadores em estado zombie, controlados por worms**, que infetam máquinas remotas sem conhecimento do utilizador legítimo;
2. Usando mecanismos de **amplificação de tráfego**, nomeadamente através da exploração de **funcionalidades inerentes aos protocolos** (e.g., **endereços de difusão**) ou da exploração de **mecanismos de pergunta resposta**, em que **a resposta é maior do que a pergunta** (e.g. Serviços de resolução de nomes DNS).

5.2 Redes de Bots

Botnets

As *Botnets* constituem **um dos maiores problemas de segurança informática da atualidade**. Para dar uma **ideia da extensão destas redes**, pode referir-se uma notícia de **Março de 2010**, em que foi desmantelada a **rede Mariposa, constituída por mais de 12 milhões** de endereços IP distintos. O grande problema associado aos ataques despoletados nestas redes é que **os pedidos falsos vêm de endereços legítimos, e podem desgastar a comunicação ao nível da camada da aplicação**. Contrariamente a ataques com endereço falsificado, o *worm* pode **interagir com o servidor até ao ponto crítico da vulnerabilidade**, e depois explorá-la.

Os ataques por *botnets* seguem uma primeira diretiva simples. Vários computadores **têm de ser infetados com um programa malicioso** (um *worm*), que pode procurar disseminar-se para outros computadores automaticamente (e.g., via lista de contactos do *e-mail*). O programa **pode anunciar o estado da infeção à fonte ou criador**, usando os mais diversos canais, nomeadamente *e-mail*, *Internet Relay Chat (IRC)*, etc. Normalmente, o *worm* **configura uma porta na firewall pessoal e instala um daemon que fica à escuta** sempre que o sistema operativo está a correr, e por onde **recebe comandos de ataque ou pedidos para transmitir informação** acerca do utilizador legítimo da máquina.

5.3 Amplificação de Ataques – Smurf e Fraggle *Attack Amplification – Smurf and Fraggle*

De modo a perceber-se melhor como funcionam os ataques por amplificação, descrevem-se os ataques conhecidos por *Smurf* e *Fraggle*. Os ataques *Smurf* e *Fraggle* exploram a possibilidade de amplificar o tráfego usando **funcionalidades simples** dos protocolos de comunicação:

- O ataque *Smurf* consiste no **envio de pacotes ICMP Echo Request (Ping) para o endereço de difusão (broadcast)** de uma rede, **colocando o endereço da vítima como sendo quem originou essa**

1. O atacante envia uma mensagem mal formada para o endereço de *broadcast* da rede, contendo como endereço fonte o endereço IP da vítima.
2. O router e o switch fazem o trabalho de difundir a mensagem para todos os equipamentos ligados à rede.

IP packet (ICMP echo request)		
10.0.0.1	10.0.0.255	PING

3. Todos respondem para a vítima, inundando-a.

Repare-se que o ataque tem **efeito triplo** em termos de DoS:

1. A **largura de banda de saída da rede amplificadora é reduzida**, por causa das mensagens geradas;
2. A **largura de banda da rede local da vítima também é reduzida**;
3. A **capacidade de resposta da vítima fica reduzida**, já que tem de processar as mensagens que recebe.

A **solução parcial** para estes ataques passa por **resolver o problema do IP spoofing** (quando tal é possível) e por **definir e concretizar as políticas de segurança em routers e switches**, de modo a que **não deixem fazer difusão para dentro da rede que servem**. Note-se que, dentro de uma rede local, o uso de endereços *broadcast* é necessário ao seu bom funcionamento, e não pode ser desativado. Já o impedimento que um pacote IP vindo do exterior de uma rede para o endereço local de uma determinada rede já é possível sem, provavelmente, qualquer impacto a nível de funcionamento.

5.4 Amplificação de Ataques usando Servidores

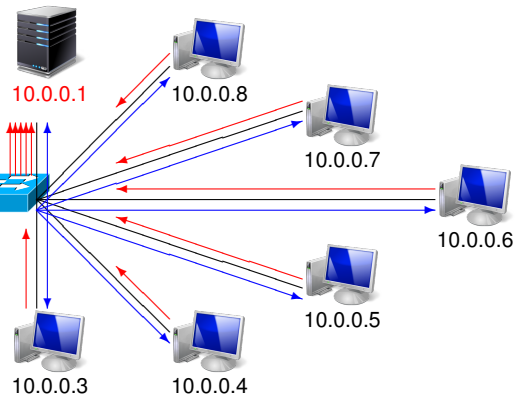
Attack Amplification using Servers

Como já foi referido antes, uma das formas de concretizar DDoSs sem recorrer a *worms* é abusar de servidores, que funcionam como amplificadores, e que apresentam as seguintes características:

1. O **atacante pode usar** os seus serviços indiscriminadamente;

mensagem.

- O ataque *Fraggle* é parecido, mas usa datagramas UDP. O atacante envia um pacote para o endereço difusão da rede, que é redirecionado pelo *router* ou *switch* para todos os computadores. Caso os equipamentos terminais tenham o serviço a correr, respondem para a vítima; caso contrário, enviam um ICMP *Port Unreachable* também para a vítima.



2. Usem o **protocolo UDP para troca de perguntas e respostas** (caso contrário teria de estabelecer uma ligação, o que não é bom para um atacante);
3. Gerem **respostas maiores do que as perguntas**.

Não causará surpresa que **um dos exemplos** de servidores que podem ser usados para este efeito é o **servidor DNS**. Um atacante pode **questionar um servidor DNS com pedidos** de resolução iterativos **até encontrar uma resposta cujo tamanho suplante**, em muito, **o tamanho da pergunta**. De seguida, pode enviar **pedidos repetidos** para esse servidor (inundação de pequenos pedidos), pedindo a resolução do endereço amplificador e **falsificando o endereço origem** do pedido. O Servidor DNS **responde com mensagens muito maiores para o endereço da vítima**. Este ataque tem ainda o efeito colateral (se não for o primário) de **saturar a largura de banda do próprio servidor**.

Caso o **IP spoofing não fosse possível** (nalguns casos pode ser severamente limitado), **nenhum atacante poderia efetuar o ataque a uma vítima remota**, e caso quisesse inundar o servidor DNS (ou a sua rede) teria de se expor. Contudo, existem outras medidas que podem ajudar a diminuir o risco, nomeadamente:

- **Desativar serviços** desnecessários;
- **Limitar o número de pedidos atendidos**, e.g., negar a resolução de mais do que 10 pedidos por segundo vindos de endereços não confiáveis;
- Não permitir transferência de zona DNS.

5.5 Ataques ao Serviço DNS

Attacks to the DNS Service

Devido ao facto do serviço DNS ter uma importância vital no funcionamento da Internet, os ataques aos respectivos servidores são frequentes, mas **não tão devastadores como inicialmente se podia pensar**. A grande resistência deste serviço resulta da forma como está **estruturado** (hierarquicamente) e da **quantidade enorme de servidores que o disponibilizam**, desde a rede local até mundial. Por um lado, **atacar um servidor local** pode **afetar** as comunicações de uma **rede local**, dependendo da quantidade de servidores DNS que se possui, mas o seu **impacto é limitado**. Por outro lado, **atacar um servidor global** pode dar **alguns problemas**, mas os **outros servidores DNS têm uma cache** que suporta a maior parte dos pedidos de resolução em caso de falha dos servidores acima.

A título de exemplo pode-se dizer que, em **Outubro de 2002**, ocorreu um ataque DDoS aos servidores-raiz do serviço DNS através de **uma inundação de datagramas ICMP durante 1 hora**. Durante este ataque, **9 das 13 máquinas falharam, mas o impacto** na maioria dos utentes da Internet foi **quase inexistente**. Foi maior o impacto psicológico e na imprensa que na atividade computacional que depende deste serviço.

Nota: o conteúdo exposto na aula e aqui contido não é (nem deve ser considerado) suficiente para total entendimento do conteúdo programático desta unidade curricular e deve ser complementado com algum empenho e investigação pessoal.