Segurança Aplicada a Redes Corporativas

Prof. Me. Wallace Rodrigues de Santana

www.neutronica.com.br

© 2018 neutronica.com.br – Parte III – Versão 1.0

Atribuição-NãoComercial-CompartilhaIgual 3.0 Brasil (CC BY-NC-SA 3.0)

Você tem a liberdade de:

Compartilhar — copiar, distribuir e transmitir a obra.

Remixar — criar obras derivadas.

Sob as seguintes condições:

Atribuição — Você deve creditar a obra da forma especificada pelo autor ou licenciante (mas não de maneira que sugira que estes concedem qualquer aval a você ou ao seu uso da obra).

Uso não comercial — Você não pode usar esta obra para fins comerciais.

Compartilhamento pela mesma licença — Se você alterar, transformar ou criar em cima desta obra, você poderá distribuir a obra resultante apenas sob a mesma licença, ou sob uma licença similar à presente.

Ficando claro que:

Renúncia — Qualquer das condições acima pode ser **renunciada** se você obtiver permissão do titular dos direitos autorais.

Domínio Público — Onde a obra ou qualquer de seus elementos estiver em **domínio público** sob o direito aplicável, esta condição não é, de maneira alguma, afetada pela licença.

Outros Direitos — Os seguintes direitos não são, de maneira alguma, afetados pela licença:

- Limitações e exceções aos direitos autorais ou quaisquer **usos livres** aplicáveis;
- Os **direitos morais** do autor;
- Direitos que outras pessoas podem ter sobre a obra ou sobre a utilização da obra, tais como **direitos de imagem** ou privacidade.

Aviso — Para qualquer reutilização ou distribuição, você deve deixar claro a terceiros os termos da licença a que se encontra submetida esta obra. A melhor maneira de fazer isso é com um link para esta página.

Módulo 13

Técnicas de Invasão, Análise e Gestão de Vulnerabilidades

Introdução

A vulnerabilidades de um ativo ou sistema podem surgir a partir de três causas principais:

- Erros de implementação;
- Falhas de projeto;
- Erros de configuração ou de infraestrutura do sistema.

Uma forma de verificar estas vulnerabilidades é executar contra o ativo um teste de intrusão, também conhecido como teste de penetração (do inglês Penetration Test ou pentest).

O pentest é um método que avalia a segurança de um sistema computacional ou de uma rede ao simular um ataque de forma maliciosa.

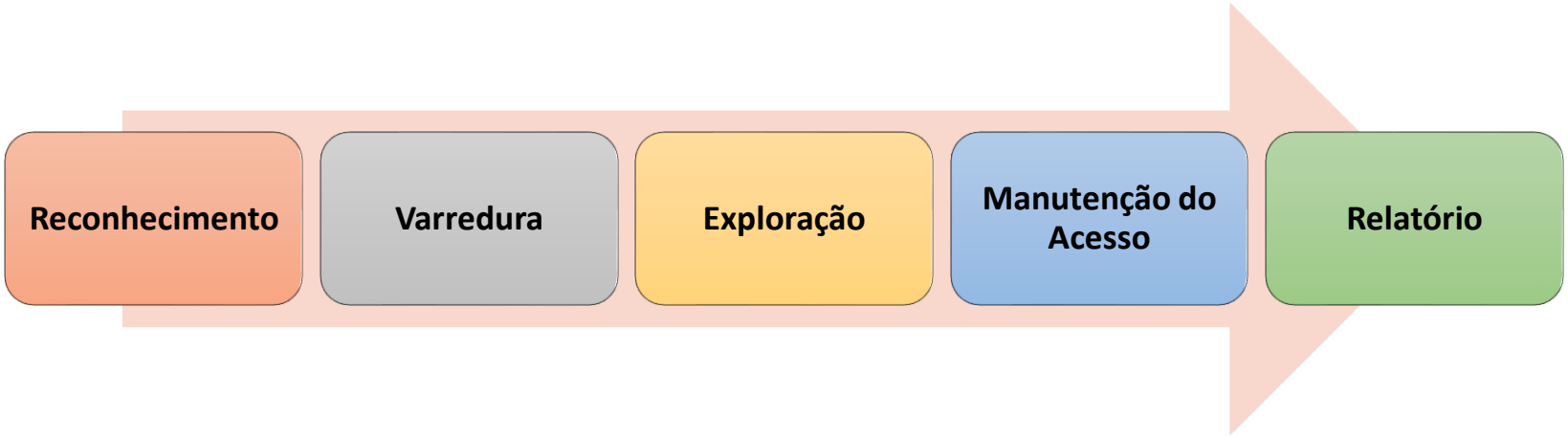
Teste de intrusão

De acordo com P. Engebretson em seu livro “The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy”, um teste de intrusão legítimo (também conhecido como *pentest*, *ethical hacking* ou *White hat hacking*) é definido como uma tentativa autorizada de localizar e invadir sistemas computacionais, com o propósito de tornar esses sistemas mais seguros.

Tais testes se diferenciam de outras formas de avaliação pois não consistem apenas do levantamento de ameaças em potencial, mas também da efetiva construção das ferramentas necessárias e da execução propriamente dita do ataque invasivo, gerando uma PoC (Proof of Concept ou Prova de Conceito) cujo objetivo é demonstrar a possibilidade de consolidação do ataque.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

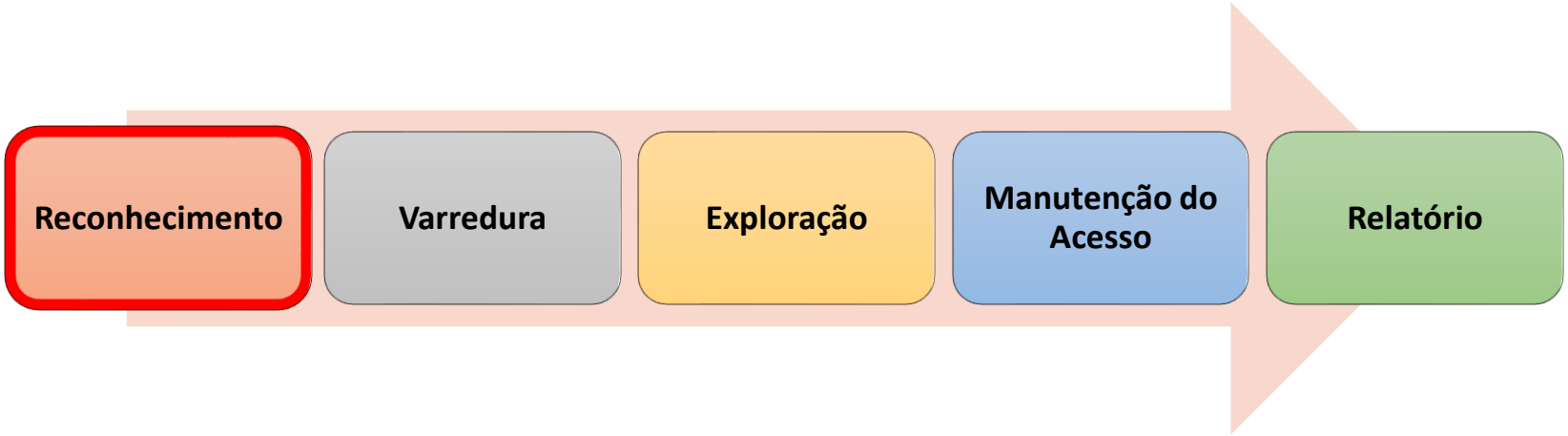
Teste de intrusão



O processo de Pentest ou Teste de Intrusão é divido em cinco fases: Reconhecimento, Varredura, Exploração, Manutenção do Acesso e Relatório.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

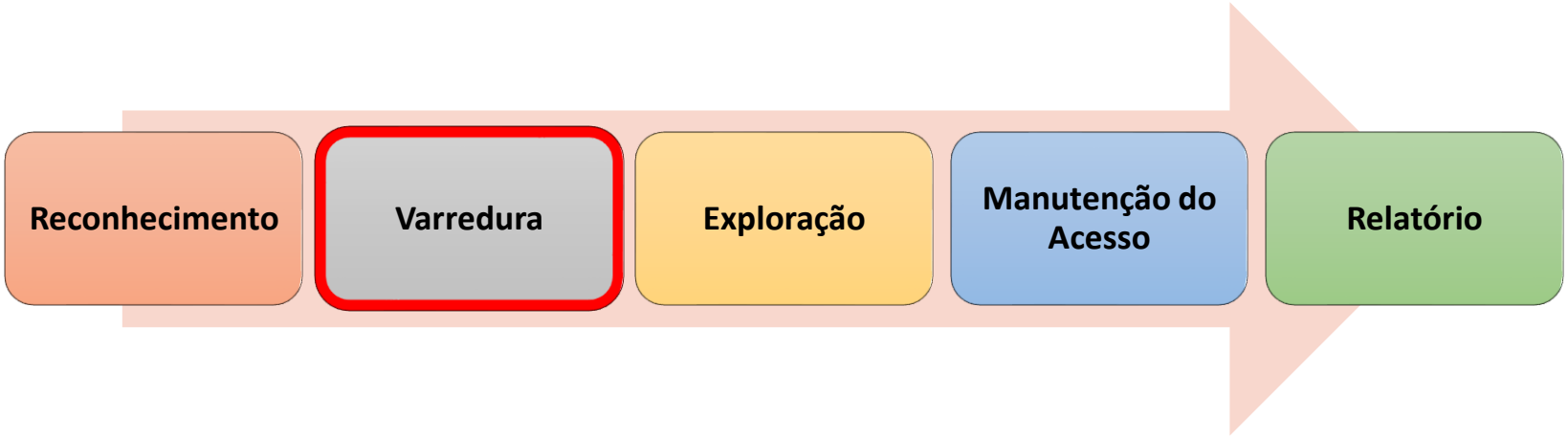
Teste de intrusão



Na fase de Reconhecimento deve ser levantada a maior quantidade possível de informações a respeito do objeto ou sistema a ser atacado, além de definir as estratégias que serão empregadas e o planejamento de todas as ações que serão executadas.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

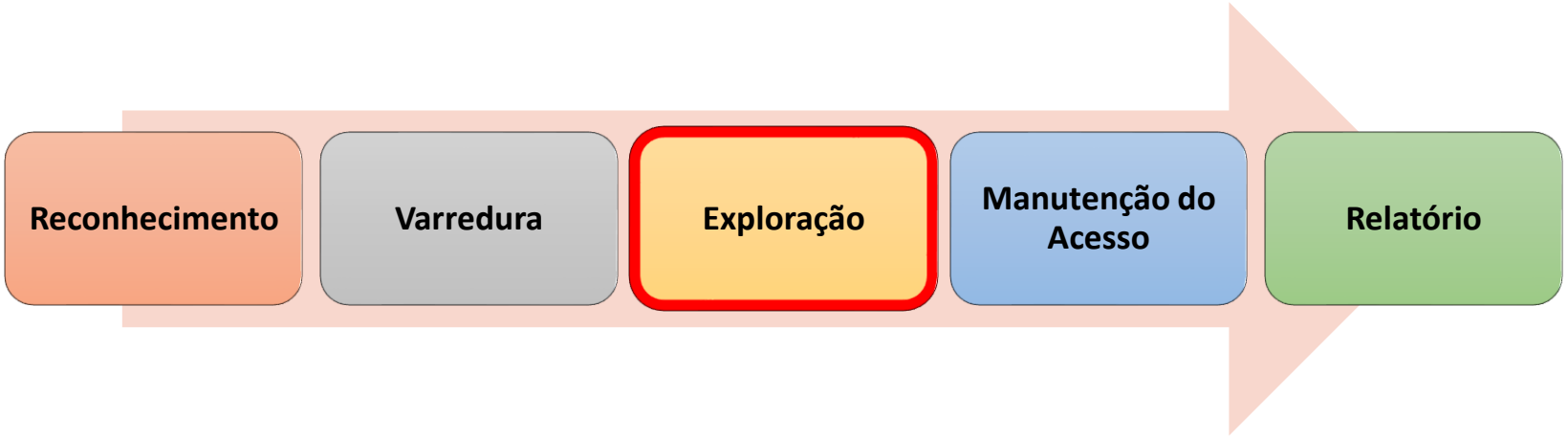
Teste de intrusão



Na fase de Varredura são identificadas as vulnerabilidades específicas dos serviços ou elementos de *software* que serão a base das ações invasivas executadas na fase seguinte.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

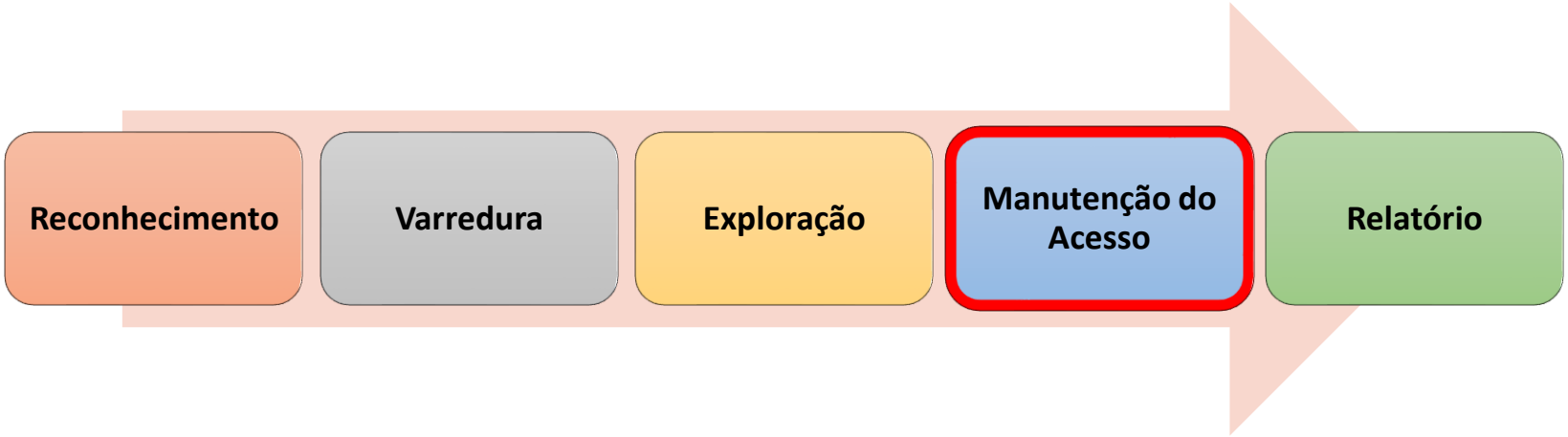
Teste de intrusão



Na fase de Exploração todas as informações colhidas nas etapas anteriores são utilizadas para a execução do ataque propriamente dito, tendo como principal meta utilizar as falhas ou vulnerabilidades identificadas anteriormente para alterar o comportamento natural de um *software*, de forma a obter acesso e controle do sistema que foi alvo da invasão.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

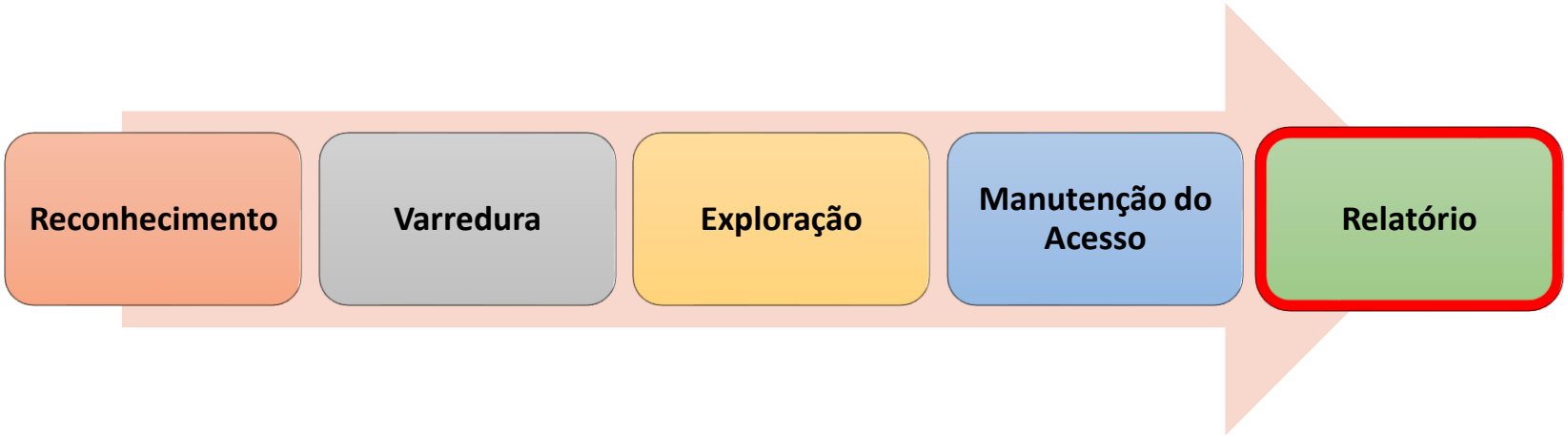
Teste de intrusão



Na fase de Manutenção do Acesso o atacante vai garantir a continuidade do acesso ao sistema invadido (normalmente através do uso de ferramentas como *backdoors* e *rootkits*), evitando que todo o processo de invasão tenha que ser executado novamente em um acesso posterior ao sistema invadido. Essa etapa pode ser considerada opcional, mas é necessária para construir uma PoC que espelhe fielmente os ataques reais.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Teste de intrusão



Na fase de Relatório é elaborado um documento onde são detalhadas as informações obtidas durante a execução do processo de ataque. Esse relatório deve ser escrito de forma clara, contendo: a) um resumo executivo, com foco no corpo gerencial da instituição que solicitou o teste; b) o detalhamento técnico de todo o processo invasivo; e, se possível; c) uma descrição das ações necessárias para mitigar a falha e eliminar as vulnerabilidades que o sistema apresenta.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Tipos de Pentest

Caixa Preta	Caixa Cinza	Caixa Branca
• Sem conhecimento sobre o alvo	• Conhecimento parcial sobre o alvo	• Conhecimento completo sobre o alvo

São três os tipos de Pentest: Caixa Preta (Black Box), Caixa Cinza (Grey Box) e Caixa Branca (White Box).

Fonte: Auditoria de Segurança da Informação em Sistemas e Aplicações, Carlos Magno B. R. da Cruz

Tipos de Pentest

Caixa Preta	Caixa Cinza	Caixa Branca
• Sem conhecimento sobre o alvo	• Conhecimento parcial sobre o alvo	• Conhecimento completo sobre o alvo

Caixa preta (*Black Box*) - neste método o testador não possui conhecimento da infraestrutura a ser testada. Os testadores devem primeiro determinar a localização e a extensão dos sistemas, antes de iniciar sua análise. O teste de penetração de caixa preta executa ataques como alguém que não está familiarizado com o sistema.

Fonte: Auditoria de Segurança da Informação em Sistemas e Aplicações, Carlos Magno B. R. da Cruz

Tipos de Pentest

Caixa Preta	Caixa Cinza	Caixa Branca
• Sem conhecimento sobre o alvo	• Conhecimento parcial sobre o alvo	• Conhecimento completo sobre o alvo

Caixa cinza (*Gray Box*) - é uma variação entre os testes caixa preta e caixa branca. No tipo caixa cinza, os testadores terão conhecimentos parciais sobre a infraestrutura ou sistema que serão testados.

Fonte: Auditoria de Segurança da Informação em Sistemas e Aplicações, Carlos Magno B. R. da Cruz

Tipos de Pentest

Caixa Preta	Caixa Cinza	Caixa Branca
• Sem conhecimento sobre o alvo	• Conhecimento parcial sobre o alvo	• Conhecimento completo sobre o alvo

Caixa branca (*White Box*) - neste método os testadores possuem conhecimento da infraestrutura ou aplicação a ser testada, muitas vezes incluindo diagramas de rede, código fonte e informações de endereçamento IP. Este teste demonstra o que pode acontecer durante um “trabalho interno“ ou após um “vazamento” de informações sensíveis, onde o atacante tem acesso ao código fonte, *layouts* de rede e possivelmente até senhas.

Fonte: Auditoria de Segurança da Informação em Sistemas e Aplicações, Carlos Magno B. R. da Cruz

Fase de Varredura – ferramentas

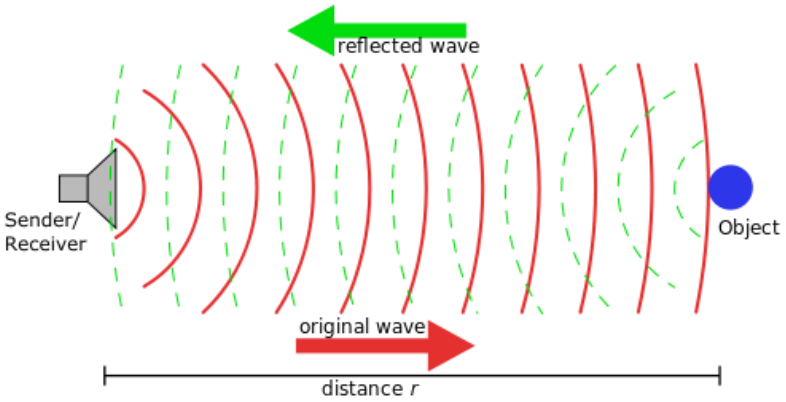
Nesta fase pode-se utilizar ferramentas de auditoria para identificar os ativos alvo.

Duas ferramentas das mais simples podem ser utilizadas para este fim. São elas o PING e o NMAP.

PING

Ping é um utilitário de software usado para testar a acessibilidade de um host em uma rede de computadores usando a pilha de protocolos TCP/IP. Ele mede o tempo de ida e volta das mensagens enviadas do host de origem para um host de destino usando pacotes de solicitação de eco ICMP (Internet Control Message Protocol). O programa relata erros, perda de pacotes e um resumo estatístico dos resultados, geralmente incluindo o mínimo, o máximo, os tempos médios de ida e volta e o desvio padrão da média.

O nome vem da terminologia do sonar ativo que envia um pulso de som e ouve o eco para detectar objetos embaixo d'água.



PING – determinando o Sistema Operacional

Device/OS	Version	Protocol	TTL
AIX	3.2, 4.1	ICMP	255
BSD	BSD/OS 3.1 and 4.0	ICMP	255
Compaq	Tru64 v5.0	ICMP	64
Cisco		ICMP	254
DEC Pathworks	V5	TCP and UDP	30
Foundry		ICMP	64
FreeBSD	5	ICMP	64
HP-UX	9.0x	TCP and UDP	30
HP-UX	10.01	TCP and UDP	64
Linux	2.4 kernel	ICMP	255
Linux	Red Hat 9	ICMP and TCP	64
MacOS/MacTCP	2.0.x	TCP and UDP	60
MacOS/MacTCP	X (10.5.6)	ICMP/TCP/UDP	64
NetBSD		ICMP	255
Netgear FVG318		ICMP and UDP	64
OpenBSD	2.6 & 2.7	ICMP	255
OS/2	TCP/IP 3.0		64
Solaris	2.5.1, 2.6, 2.7, 2.8	ICMP	255
Windows	98	ICMP	32
Windows	Server 2008	ICMP/TCP/UDP	128

NMAP

O Nmap (“Network Mapper”) é uma ferramenta de código aberto amplamente utilizado para exploração de redes e auditoria de segurança.

O Nmap usa pacotes IP brutos para determinar quais hosts estão disponíveis na rede, quais serviços (nome do aplicativo e versão) esses hosts estão oferecendo e quais sistemas operacionais (e versões do SO) estão executando, entre dezenas de outras características.

Embora o Nmap seja comumente usado para auditorias de segurança, ele também é utilizado para tarefas rotineiras tais como: inventário de rede, gerenciamento de agendas de atualização de serviços e monitoramento do tempo de atividade do host ou serviço.

Fonte: www.nmap.org

NMAP – Exemplo

```
[root@server ~]# nmap host1.acme.com
Starting Nmap 6.47 ( http://nmap.org ) at 2018-10-29 15:13 -03
Nmap scan report for host1.acme.com (192.168.0.10)
Host is up (0.0000080s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
111/tcp  open  rpcbind
3306/tcp open  mysql
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

NMAP – Exemplo

```
[root@server ~]# nmap host2.acme.com
Starting Nmap 6.47 ( http://nmap.org ) at 2018-10-29 15:15 -03
Nmap scan report for host2.acme.com (192.168.0.20)
Host is up (0.0014s latency).
Not shown: 989 filtered ports
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
8000/tcp  open  http-alt
8081/tcp  open  blackice-icecap
27000/tcp open  flexlm0
49153/tcp open  unknown
49154/tcp open  unknown
Nmap done: 1 IP address (1 host up) scanned in 4.74 seconds
```

Fase de Exploração – métodos

Dentre as muitas alternativas na busca por vulnerabilidades nos software, a abordagem de caixa preta denominada *Fuzzing* é uma das mais utilizadas. Trata-se de um método de testes de software onde a aplicação é bombardeada com casos de testes gerados por outro programa.

A ideia desse método é verificar o comportamento do software nos casos em que o sistema recebe entradas mal formadas ou fora do padrão esperado. Isto possibilita a descoberta de vulnerabilidades à partir de combinações de entradas não testadas originalmente pelo desenvolvedor.

O método *Fuzzing* permite descobrir vulnerabilidades no sistema através de um monitoramento das exceções lançadas pelo programa. É um processo geralmente automatizado, que envolve a manipulação dos campos de entrada da aplicação. A técnica é dividida em duas categorias: *Fuzzing* baseado em mutação e *Fuzzing* baseado em geração.

No *Fuzzing* baseado em mutação, os dados são coletados da aplicação e depois modificados. Essa mutação pode ser efetuada de forma aleatória ou através de uma heurística. No *Fuzzing* baseado em geração, um modelo para os dados de entrada especifica como os casos de testes devem ser criados.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Método Fuzzing

A forma de condução da metodologia *Fuzzing* depende de vários fatores, como a habilidade do testador e a complexidade da aplicação. Em geral a aplicação dos testes ocorre em seis etapas, conforme descrito a seguir:

a) Identificação do alvo – o testador procura por dados sobre o sistema a ser explorado, para encontrar informações relevantes que ajudem na condução do teste.

b) Identificação das entradas – nesta etapa, o testador procura identificar todas as interfaces e campos de entrada para o sistema alvo. Entre outras possibilidades, podem ser utilizados como vias de entrada de dados: soquetes (*sockets*) de rede, variáveis de ambiente, arquivos, argumentos da linha de comandos e interfaces remotas.

c) Geração de dados – a terceira fase é a responsável pela geração das entradas que serão utilizados durante o *Fuzzing*. A decisão de utilizar *Fuzzing* baseado em mutação ou *Fuzzing* baseado em geração depende do alvo e das informações coletadas.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Método Fuzzing

d) Execução dos dados – nesta etapa, as entradas geradas no passo anterior são transferidas para a aplicação. Esse processo pode envolver o ato de enviar pacotes de dados ao alvo, abrir arquivos, enviar argumentos pela linha de comandos, etc.

e) Monitoramento das exceções – esse passo é responsável por colher as informações das exceções geradas no passo anterior. O tipo de falha ocasionada e o estado da aplicação após a ocorrência do erro determinará a estratégia de desenvolvimento de *exploit* a ser adotada.

f) Determinação dos *exploits* – na última fase do *Fuzzing*, as informações colhidas no passo anterior são estudadas para determinar se vulnerabilidades identificadas podem ser exploradas por algum tipo de *exploit*.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Explorando falhas – shellcode

Shellcode é um conjunto de instruções, tipicamente escritas em *assembly*, que são injetadas e executadas por um programa através de um *exploit*. Para funcionar adequadamente, um *shellcode* deve ser desenvolvido seguindo as seguintes premissas:

a) Deve ser pequeno, pois geralmente o espaço que o atacante possui para injetar o código no *buffer* é limitado;

b) Não deve possuir bytes nulos (0x00), pois esse valor é um delimitador de *string* na linguagem C. Se a entrada maliciosa for recebida na forma de um *string*, o byte nulo causará a interrupção da leitura do *shellcode*;

c) Deve ser escrito em código hexadecimal, para representar diretamente os bytes a serem gravados na memória;

d) Deve ser específico para um ambiente computacional (hardware e sistema operacional). Os *shellcodes* desenvolvidos para os sistemas operacionais Windows, por exemplo, não funcionam em sistemas Linux, pois, entre outras coisas, as chamadas de sistema são diferentes.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Explorando falhas – shellcode

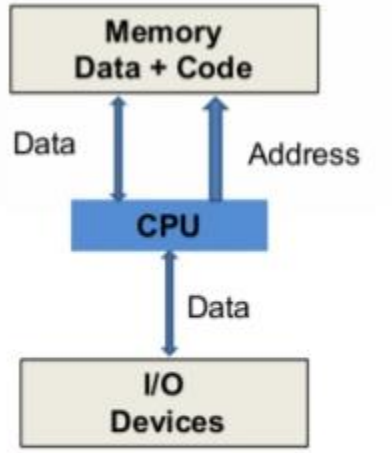
A meta de um *shellcode* é fazer com que um programa vulnerável funcione como uma porta de acesso ao sistema operacional hospedeiro. A maneira mais fácil de interagir com o sistema operacional é através de suas chamadas de sistema (*syscalls*).

Por isso, normalmente os *shellcodes* preparam os parâmetros para a execução de chamadas de sistema que possibilitem o acesso ao sistema alvo.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Explorando falhas – memória

Quando um programa é executado seus componentes são mapeados para um espaço de memória designado para este fim pelo sistema operacional. Esse espaço reservado é utilizado para o armazenamento tanto das instruções que serão executadas como dos dados (variáveis) que serão processados.

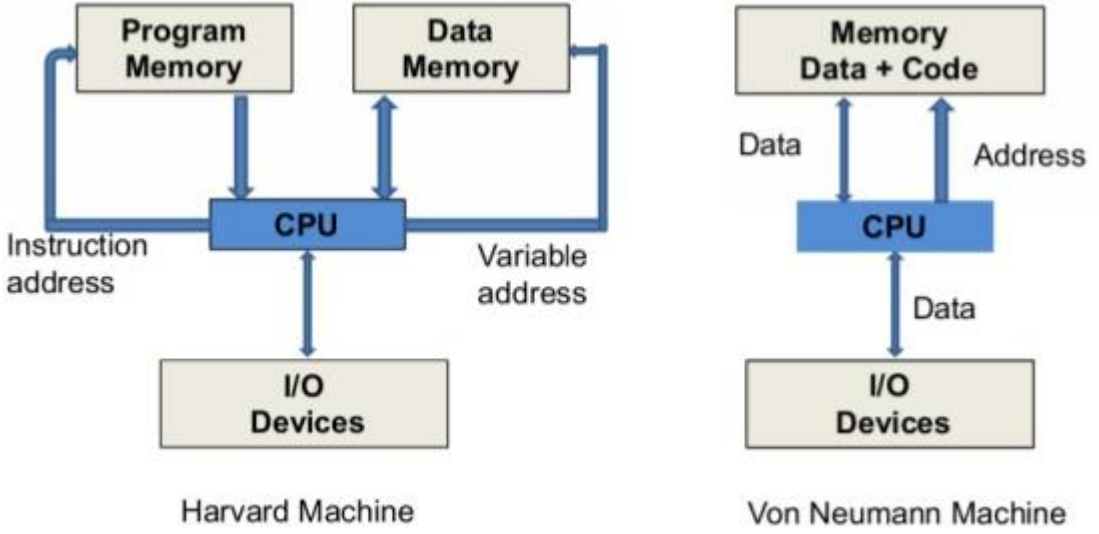


Von Neumann Machine

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

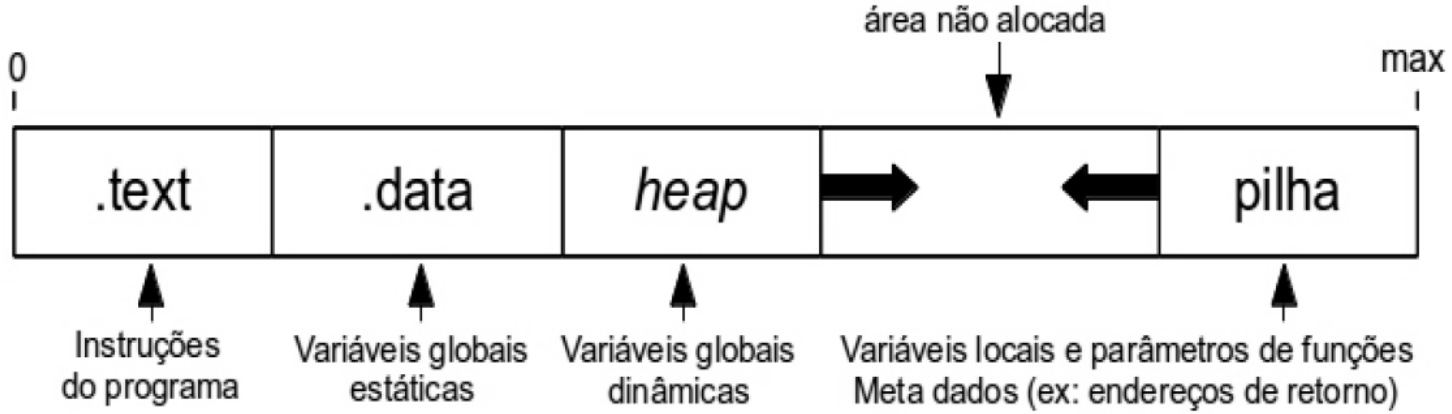
Arquitetura Harvard vs Von Neumann

A título de curiosidade, máquinas que seguem a especificação da arquitetura Harvard, como por exemplo o Arduino e outros dispositivos baseados em microcontroladores, possuem espaços de memória separados para o código do programa e para os dados, ao passo que máquinas que seguem a arquitetura Von Neumann, como por exemplo o Raspberry Pi e todos aqueles baseados em microprocessadores, compartilham o mesmo espaço de memória para código e dados.



Explorando falhas – memória

De forma mais detalhada, em máquinas Von Neumann esse espaço é subdividido em quatro segmentos. O segmento .data é reservado para o armazenamento das variáveis estáticas e o segmento .text corresponde às instruções do programa. Ambos possuem um tamanho fixo. Por fim, existem as estruturas de pilha (*stack*) e o *heap*. A primeira é responsável pelo armazenamento de variáveis locais (pertencentes ao escopo de uma única função) e pelo registro de metadados usados pelo processador para controlar chamadas de funções. O *heap* é responsável pelo armazenamento de variáveis alocadas dinamicamente.



Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Explorando falhas – registradores

A manipulação de registradores é um elemento fundamental no processo de criação de *exploits*. Para esse fim, merecem destaque os seguintes registradores:

ESP (*Extended Stack Pointer*): um registrador acessível pelo programa, que armazena o endereço de memória correspondente ao topo da pilha. Sua manipulação é muito importante na aplicação de técnicas de estouro de pilha (*stack overflow*);

EIP (*Extended Instruction Pointer*): um registrador de controle, acessível somente pelo processador, responsável por armazenar o endereço da próxima instrução que será executada. Induzir a alteração do valor desse registrador torna possível a subversão do fluxo de execução do programa, o que é fundamental para a criação de um *exploit* efetivo.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Explorando falhas
Transbordamento de memória

Uma das técnicas mais usadas pelos atacantes para explorar vulnerabilidades no software são o *buffer overflow* (estouro ou transbordamento de memória).

Os *buffers* são porções de memória, com tamanho definido, usados para armazenar algum dado. Um transbordamento de memória consiste em estourar a capacidade do *buffer*, inserindo mais informação do que ele é capaz de armazenar. Isso faz com que os dados sejam inseridos em espaços de memória além do limite previsto (tamanho do *buffer).*

Basicamente, existem dois tipos de transbordamentos de *buffer*:

- *Stack overflow* (estouro da pilha);
- *Heap overflow* (estouro do *heap*).

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

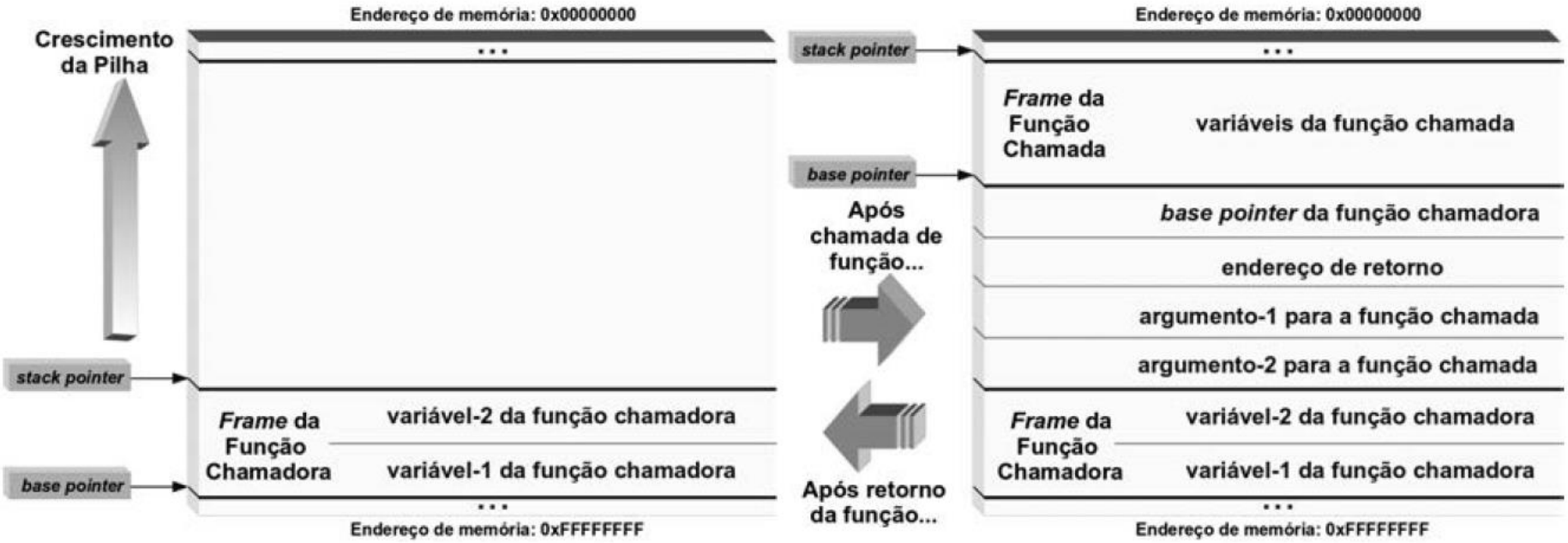
Transbordamento de memória
Estouro da pilha

Na maioria das linguagens de programação de alto nível, o código é quebrado em frações menores para que os desenvolvedores possam chamar a mesma função em várias partes do programa.

Durante essas chamadas entre funções, parâmetros e metadados precisam ser salvos em algum lugar, para que o fluxo de execução possa fluir entre as funções. Esse lugar é o segmento de pilha.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Transbordamento de memória
Estouro da pilha



Estado da pilha durante chamadas e retornos de funções

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Transbordamento de memória
Estouro da pilha

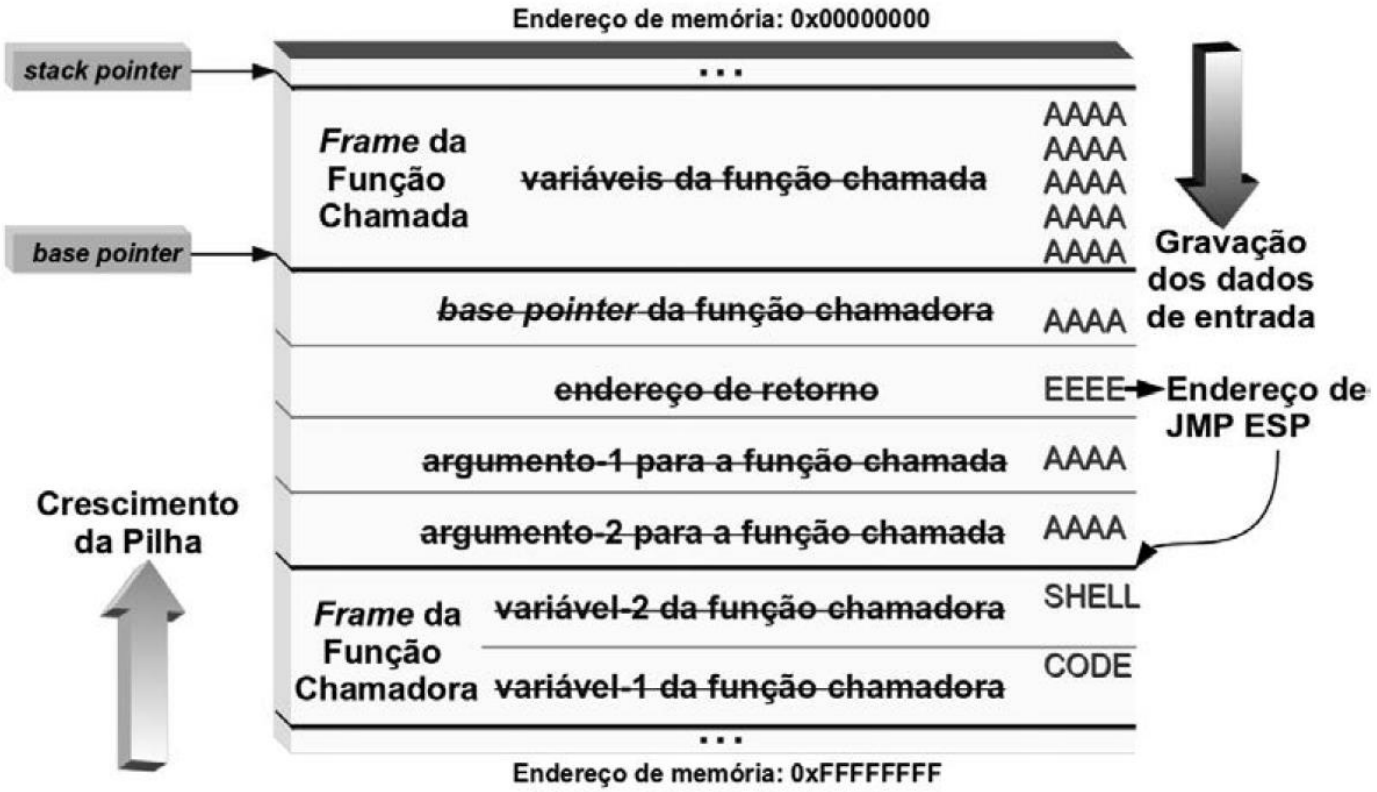
Imagine que a função chamada pelo programa principal não possua checagem do tamanho de entrada dos dados. Caso o usuário informe uma *string* de entrada maior do que o *buffer* alocado dentro da função, ocorrerá um estouro.

As áreas adjacentes a esse *buffer* na pilha serão sobrescritas e se o tamanho da *string* de entrada for suficientemente grande, ela poderá inclusive sobrescrever o endereço de retorno anotado na pilha por ocasião da chamada da função.

Assim, um atacante é capaz de subverter o fluxo de execução do programa, desviando-o para um endereço de sua escolha.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Transbordamento de memória
Estouro da pilha



Estado da pilha após transbordamento com sobrescrita do endereço de retorno

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Transbordamento de memória
Estouro do heap

Heap é uma região de memória reservada pelo sistema operacional para o processo armazenar dados referentes a variáveis alocadas dinamicamente, ou seja, durante a execução do programa.

Essa alocação dinâmica ocorre por dois motivos: primeiro, porque não é possível prever, durante a compilação, a quantidade de blocos de memória necessários para armazenar dados cujo tamanho é variável; segundo porque com a alocação dinâmica de memória as áreas livres podem ser liberadas e reaproveitadas quando não estiverem mais sendo utilizadas.

Um estouro de *heap* ocorre quando um programa possibilita a cópia de dados para uma área de memória dinâmica (*heap*) sem checar se todos os dados de origem irão caber no destino.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Transbordamento de memória
Estouro do heap

As técnicas de exploração das falhas de estouros de *heap* podem ser divididas em dois grupos:

O primeiro é constituído por ataques onde o estouro de uma variável localizada no *heap* possibilita a alteração do conteúdo dos blocos de memória adjacentes, enquanto o segundo grupo compreende os ataques que alteram as informações de gerenciamento do *heap*.

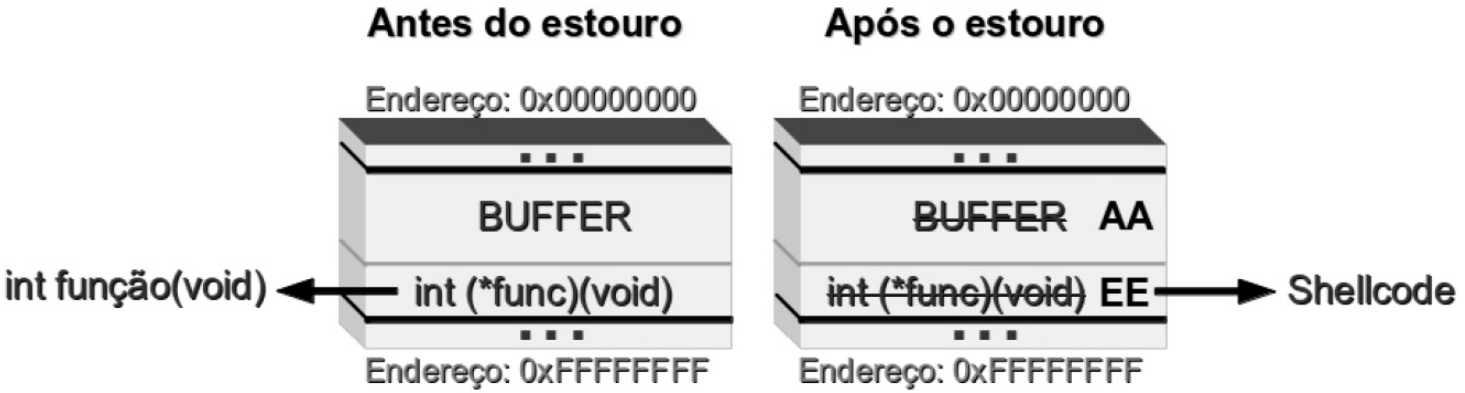
Aqui, o atacante utiliza um estouro no *heap* para sobrescrever dados relativos a um arquivo ou a uma senha. Para que isso ocorra, o ponteiro que o atacante deseja sobrescrever deve estar localizado após o *buffer* que será sobrescrito, pois o *heap* é escrito no sentido crescente da memória.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Transbordamento de memória
Estouro do heap

No segundo grupo de ataques de estouro de *heap* ocorre a sobrescrita de um ponteiro para função.

Nessa situação, o atacante sobrescreve um ponteiro fazendo-o apontar para outro endereço de memória, onde se localiza um *shellcode*. Após essa sobreposição de endereços, se a aplicação chamar a função através do ponteiro sobrescrito, o fluxo de execução será desviado para o *shellcode*, ao invés de saltar para o código originalmente apontado.



Sobrescrita de ponteiro para função no *heap*.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Mitigando as falhas
Proteção de memória

Assim como as estratégias para explorar vulnerabilidades de *buffer overflow* evoluíram ao longo do tempo, surgiram e foram aprimorados mecanismos de proteção, que têm a incumbência de bloquear ou – pelo menos – dificultar a consolidação desses ataques.

Entre as classes de mitigações propostas contra ataques de exploração de softwares, os mecanismos de proteção da memória, implementados pelos sistemas operacionais na tentativa de impor barreiras contra a execução de códigos maliciosos, podem ser considerados os dispositivos que mais evoluíram ao longo do tempo.

Além disso, a maior parte das técnicas modernas de desenvolvimento de *exploits* surgiram da necessidade de adaptar os ataques para funcionamento em ambientes com proteções de memória. Apesar dessa evolução nas estratégias de exploração, quando empregadas em conjunto, as proteções de memória descritas a seguir impõem um elevado nível de dificuldade à criação de *exploits* eficazes.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Proteção de memória
Pilha não-executável (nx-stack)

Como os ataques de estouro de pilha foram os primeiros a se popularizar, surgiu inicialmente a necessidade de se proteger a pilha contra a execução de códigos, principalmente porque a estratégia mais óbvia de exploração desse tipo de vulnerabilidade é a inserção do código malicioso na própria pilha, junto com o código que gera o estouro do *buffer*.

Em decorrência da constatação dessa possibilidade, surgiram em 1996 dispositivos para impedir a execução de instruções oriundas da área de pilha (*non-executable stack* ou *nx-stack*). Atualmente esse tipo de proteção está presente por padrão na maioria dos sistemas operacionais, incluindo Linux, OpenBSD, Mac OS X, Solaris e Windows, entre outros.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Proteção de memória
Prevenção de Execução de Dados (DEP)

Uma extensão natural do mecanismo de proibição da execução de instruções armazenadas na pilha (*nx-stack*), concebida para bloquear esse tipo de tentativa em outras áreas da memória – como o *heap* e o BSS* –, recebeu as designações "NX" (*No eXecute, na AMD*) e "XD" (eXecute Disable, na Intel).

Essa estratégia de proteção, batizada no Windows como DEP (*Data Execution Prevention*), baseia-se na utilização de um recurso incorporado aos processadores, em 2004, para marcar as páginas de memória com um bit de execução.

Assim, o sistema operacional pode garantir que áreas de memória destinadas a armazenar dados – como a pilha e o *heap* – não sejam executáveis.

*O BSS (Block Started by Symbol) é uma área de memória usada por muitos compiladores para armazenar a parte de um arquivo objeto ou executável que contém variáveis alocadas estaticamente que não são explicitamente inicializadas com um valor definido.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Proteção de memória
Prevenção de Execução de Dados (DEP)

O “NX/XD” torna praticamente impossível a injeção de código externo em programas vulneráveis. Entretanto, técnicas de reutilização de código, por exemplo, podem obter êxito contra um sistema protegido apenas pelo bit de execução.

Além disso, nas arquiteturas de 32 bits de alguns sistemas operacionais, como em algumas versões do Windows, a proteção oferecida pelo DEP pode ser desativada durante a execução do programa.

Essa é uma brecha comumente explorada por atacantes. Também são comuns ataques que usam funções do próprio sistema operacional para copiar o *shellcode* em uma área executável da memória ou para atribuir permissões de execução à região onde o *shellcode* se localiza.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Proteção de memória
Prevenção de Execução de Dados (DEP)

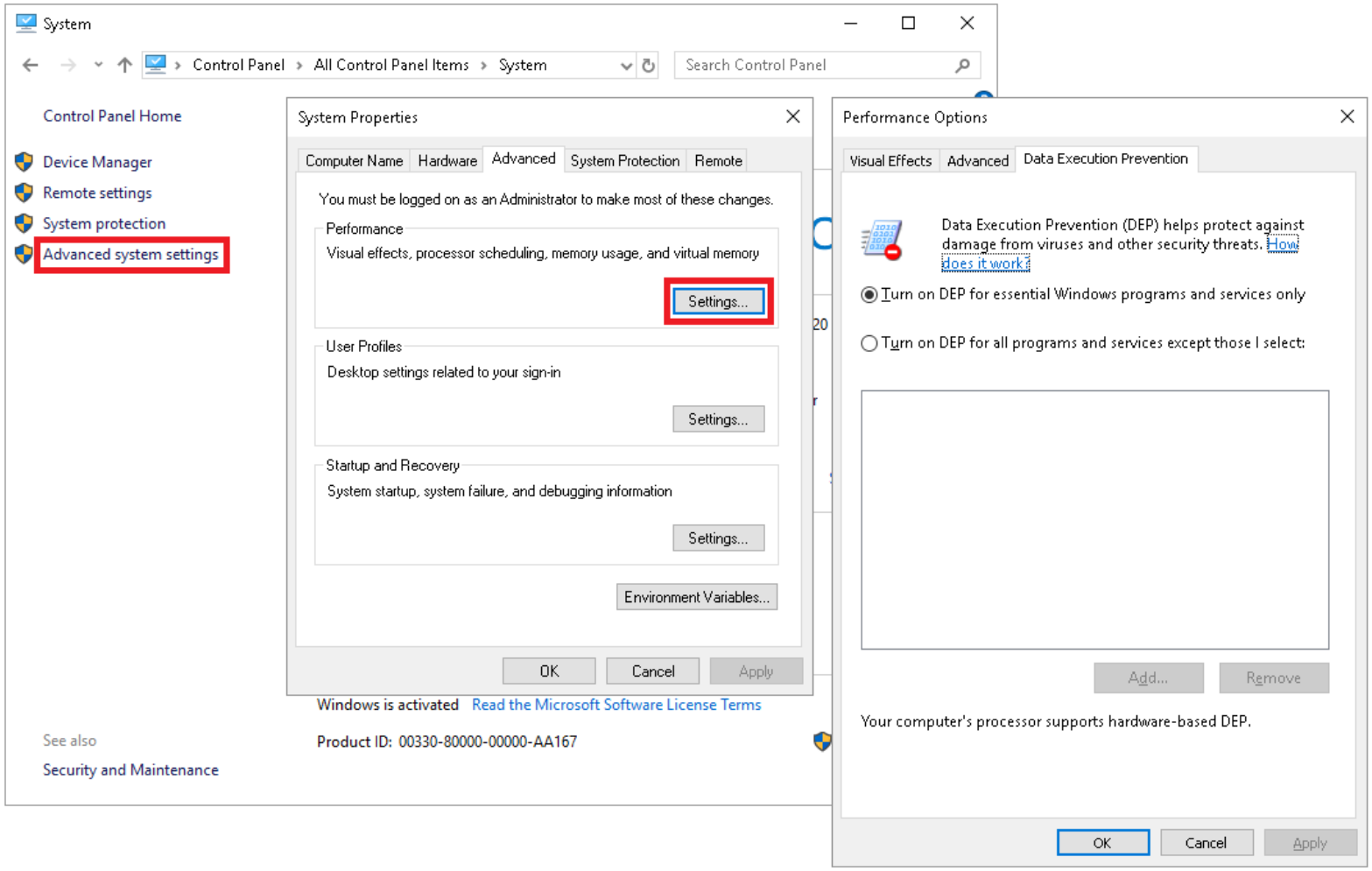
O Windows implementa o DEP desde o lançamento do *Service Pack* 2 para o XP.

O funcionamento do DEP requer que as aplicações sejam ligadas com uma opção especial: /NXCOMPAT. Essa opção encontra-se disponível no compilador Visual Studio desde a versão 2005.

A tecnologia “NX/XD” também está presente no Linux, desde o *kernel* 2.6.8.

Fonte: Análise de vulnerabilidades, Mateus F. T. Ferreira et al.

Proteção de memória
Prevenção de Execução de Dados (DEP)



Para saber mais...

… leia o artigo Análise de vulnerabilidades em Sistemas Computacionais Modernos: Conceitos, Exploits e Proteções, Mateus F. T. Ferreira et al, in Minicursos do XII Simpósio SBSEG, 2012

... leia a dissertação Auditoria de Segurança da Informação em Sistemas e Aplicações, de Carlos Magno B. R. da Cruz

Módulo 14

Criptografia, Certificados Digitais e Public Key Infrastructure

Introdução

Desde a antiguidade sempre houve a necessidade de se transmitir mensagens de forma que somente o destinatário pudesse acessá-la e compreendê-la.

Generais precisam dar ordens a seus comandados sem que essas caiam nas mãos do inimigo, e líderes políticos precisam trocar informações com seus aliados e estas devem estar a salvo de adversários.

Enfim, a arte de disfarçar, tornar secreta, codificar uma mensagem e transmiti-la de forma que somente o destinatário possa compreendê-la, evitando que qualquer outro possa roubar esta informação, tem sido vital em várias áreas.

Fonte: Fundação CECIERJ

Introdução

Existem duas formas básicas de proteger informações de forma que pessoas não autorizadas não tenham acesso ao seu conteúdo. São elas a esteganografia e a criptologia.

Esteganografia: deriva das palavras gregas *steganos*, que significa “coberto, oculto ou protegido”, e *gráphein* que significa “escrita”. Consiste na ocultação de mensagens dentro de outras mensagens. Um exemplo bastante comum é a utilização de imagens para a ocultação de textos.

Criptografia: deriva das palavras gregas *kryptós*, que significa “escondido ou secreto”, e *gráphein*, que significa “escrita”. Consiste de princípios e técnicas para transformar a informação de sua forma original para outra ilegível, de forma a proteger seu conteúdo do acesso de pessoas não autorizadas.

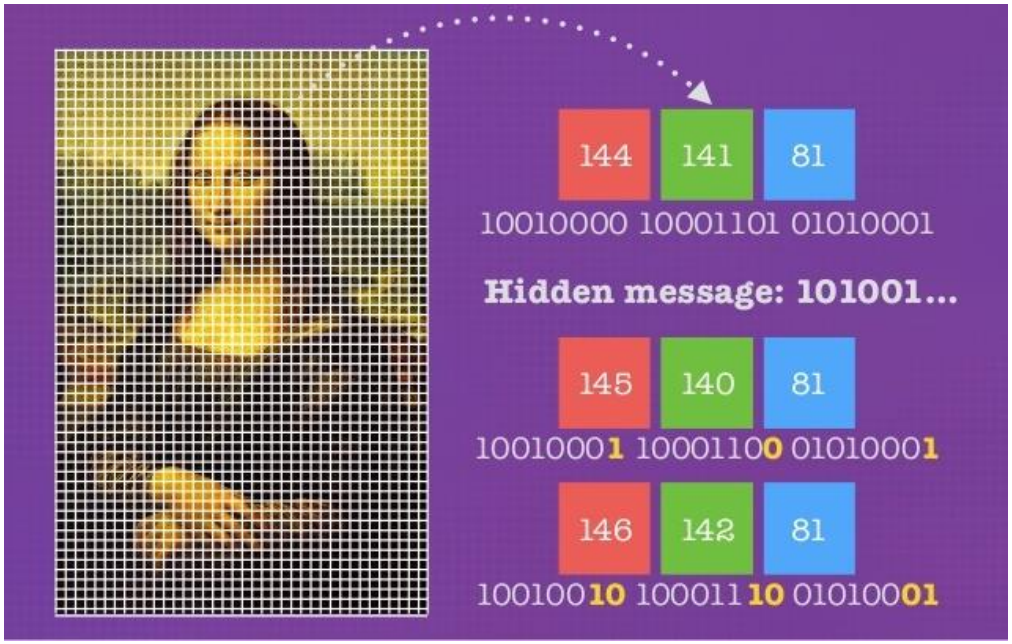
A diferença entre as duas é que a esteganografia oculta a existência da mensagem, enquanto a criptografia oculta o conteúdo da mesma. Pode-se utilizar as duas técnicas simultaneamente, cifrando uma mensagem e depois ocultando-a dentro de outra.

Fonte: Escola Superior de Redes RNP

Esteganografia

Muitas técnicas modernas possibilitam esconder informações dentro de imagens.

A forma mais utilizada emprega a técnica denominada LSB (Least Significant Bit, ou Bit Menos Significativo), que consiste em utilizar o bit menos significativo de uma determinada informação para armazenar um bit de uma nova informação.



Fonte: Wikipedia

Esteganografia

O Caso Abadia

Quando o traficante colombiano Juan Carlos Ramírez Abadía foi preso em São Paulo em agosto de 2007, os delegados da Polícia Federal ficaram intrigados com a quantidade de imagens da Hello Kitty que ele guardava nos computadores. Eram quase 200 imagens, quase todas enviadas por e-mail.

A perícia da Polícia Federal descobriu que haviam mensagens de voz e de texto escondidas nas imagens.

Fonte: Folha de São Paulo

Esteganografia – demonstração

Uma forma menos sutil de esconder mensagens em uma imagem é anexar um arquivo texto a uma imagem JPEG.

Pode-se usar o seguinte comando no console do Windows para combinar de forma binária os arquivos HelloKitty.jpg e Mensagem.txt no arquivo Desenho.jpg:

```
copy /b HelloKitty.jpg + Mensagem.txt Desenho.jpg
```

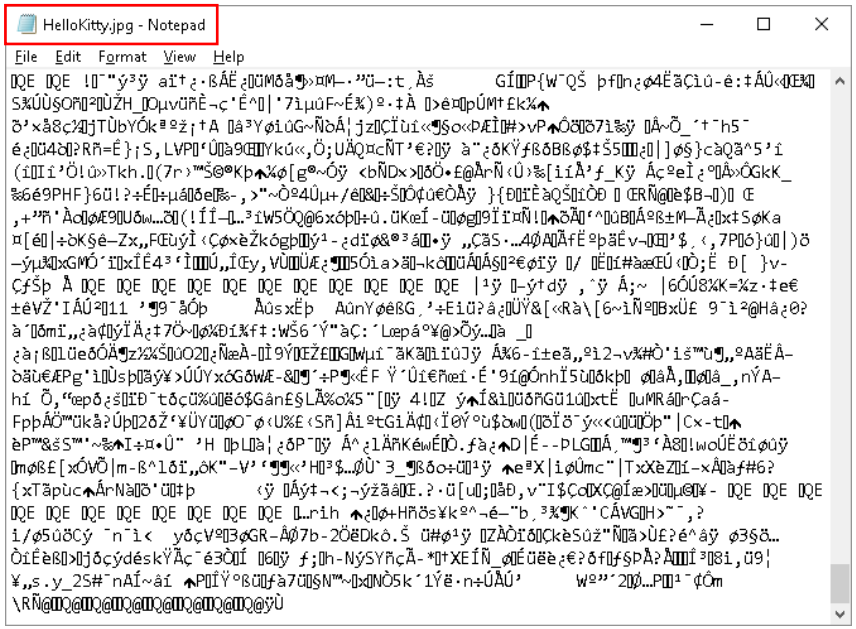
HelloKitty.jpg | Mensagem.txt | Desenho.jpg

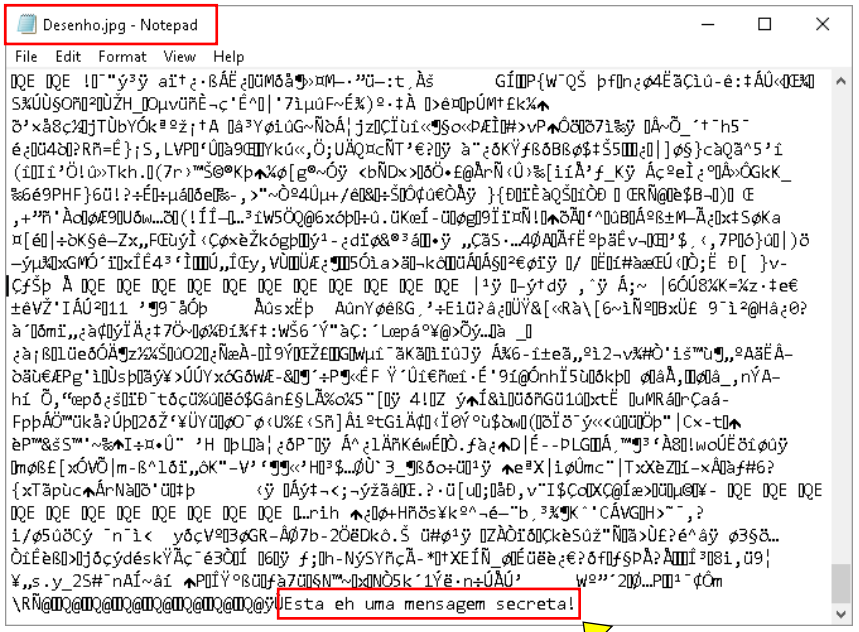
O arquivo Mensagem.txt possui o seguinte texto: Esta eh uma mensagem secreta!

Esteganografia – demonstração

Depois de realizada a cópia binária, basta abrir os dois arquivos no Bloco de Notas. A diferença entre eles poderá ser visualizada no final do arquivo.

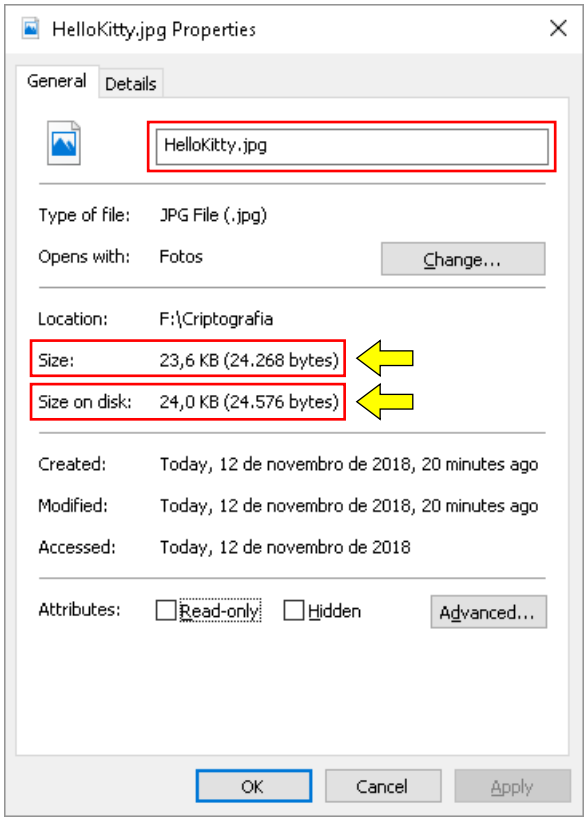
Se o usuário abrir estes dois arquivos em um visualizador de imagens, a diferença não poderá ser notada.

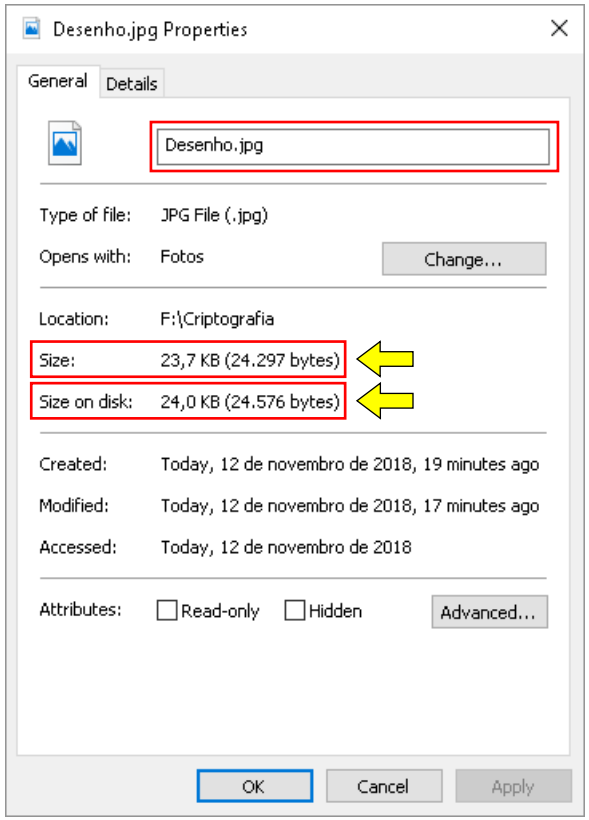




Esteganografia – demonstração

É possível verificar que o arquivo Desenho.jpg ficou ligeiramente maior que o arquivo original HelloKitty.jpg, uma vez que ele contém a mensagem escondida.





Criptografia

A criptografia, considerada como a ciência e a arte de escrever mensagens em forma cifrada ou em código, consiste em transformar a informação de sua forma original e legível para outra ilegível, protegendo assim de pessoas não autorizadas o acesso ao seu conteúdo.

É hoje um dos principais mecanismos de segurança que pode-se usar para proteger dos riscos associados ao uso da Internet.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Criptografia

Por meio do uso da criptografia pode-se:

- proteger os dados sigilosos armazenados no computador;
- criar uma área (partição) específica no computador, na qual todas as informações que forem lá gravadas serão automaticamente criptografadas;
- proteger os *backups* contra acesso indevido, principalmente aqueles enviados para áreas de armazenamento externo de mídias;
- proteger as comunicações realizadas pela Internet, como os *e-mails* enviados/recebidos e as transações bancárias e comerciais realizadas.

As técnicas básicas de criptografia por embaralhamento de mensagem são a transposição e a substituição.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Conceitos

REMETENTE: Pessoa ou serviço que envia a informação;

DESTINATÁRIO: Pessoa ou serviço que recebe a informação;

CANAL DE COMUNICAÇÃO: Meio utilizado para a troca de informações;

TEXTO CLARO: Informação legível (original) que será protegida, ou seja, que será cifrada;

TEXTO CIFRADO: Texto ilegível, gerado pela codificação de um texto claro;

CIFRAR: Ato de transformar um texto claro em um texto codificado;

DECIFRAR: Ato de transformar um texto codificado em um texto claro;

MÉTODO CRIPTOGRÁFICO: Conjunto de programas responsável por cifrar (codificar) e decifrar (decodificar) informações;

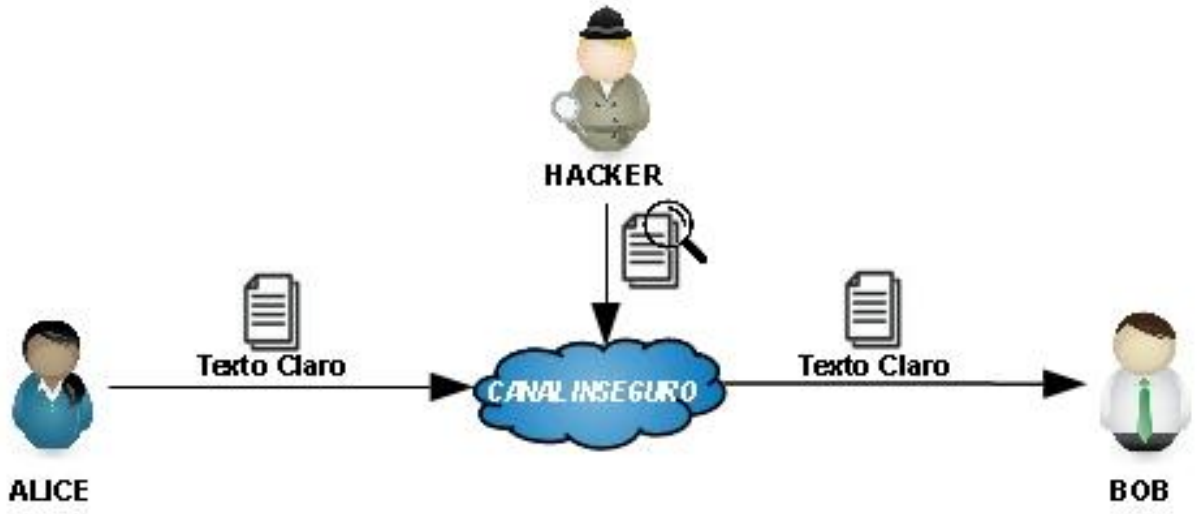
CHAVE: Similar a uma senha, é utilizada como elemento secreto pelos métodos criptográficos. Seu tamanho é geralmente medido em quantidade de bits;

ESPAÇO DE CHAVES: Quantidade de combinações possíveis de chave.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

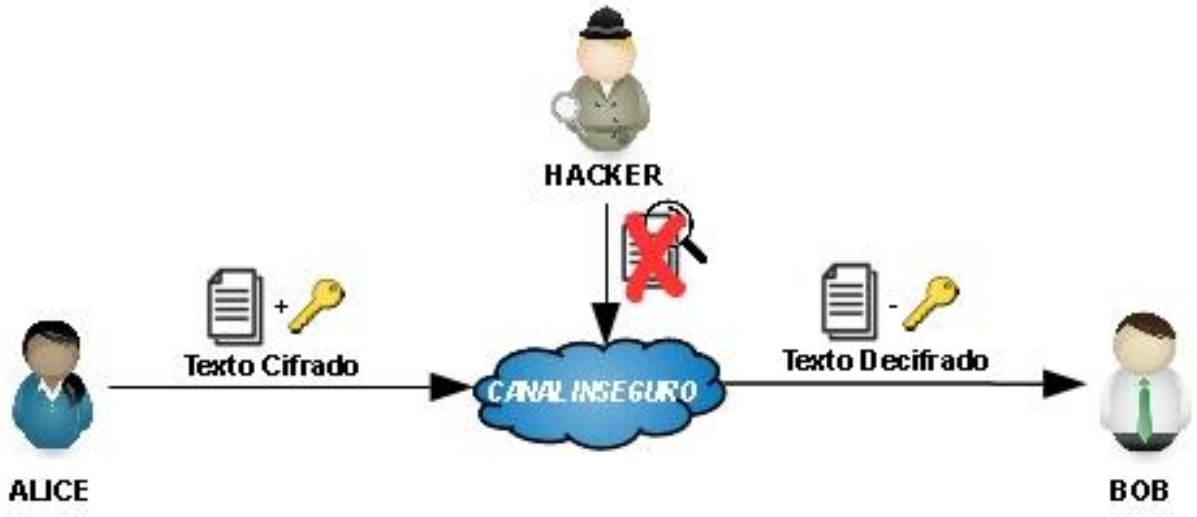
Troca de mensagens

Quando Alice deseja enviar uma mensagem para Bob por meio de um canal inseguro, um terceiro (Hacker) pode interceptar a mensagem e verificar o seu conteúdo.



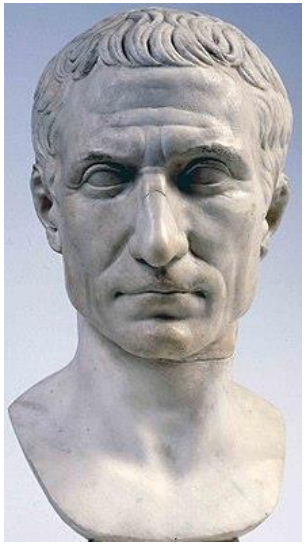
Troca de mensagens

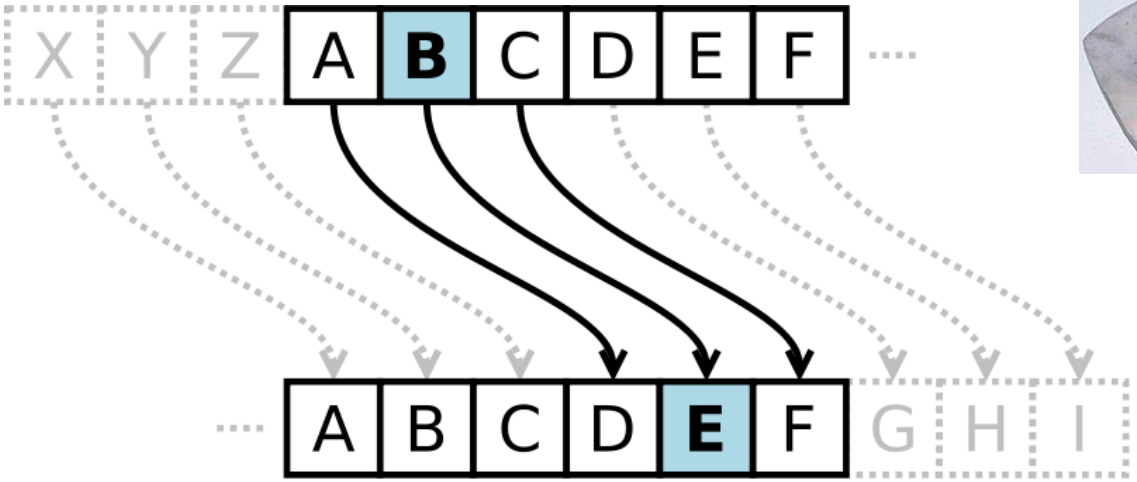
Uma forma de proteger a comunicação seria adotar um canal seguro onde somente Alice e Bob tivessem acesso, ou então codificar (cifrar) a mensagem de tal forma que um terceiro (Hacker) não conseguiria verificar o seu conteúdo, ainda que estivesse de posse da mensagem.



Cifra de César

O Imperador Romano Júlio César, que viveu de 100 AC à 44 AC, usava na sua correspondência militar uma chave de substituição muito simples, na qual cada letra da mensagem original era substituída pela letra que a seguia em três posições no alfabeto. A letra A era substituída pela D, a B pela E, e assim sucessivamente.



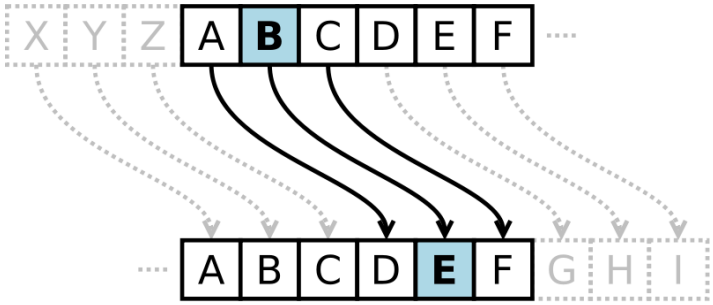


Fonte: Fundação CECIERJ

Cifra de César – algoritmo

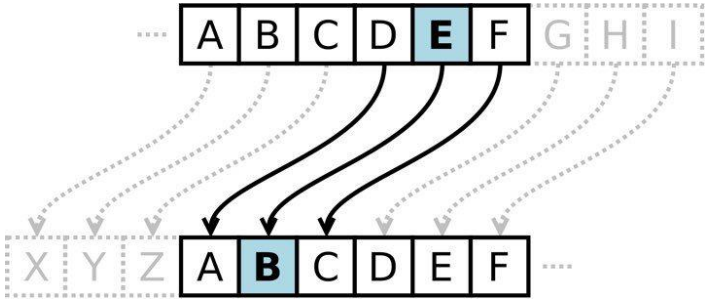
Para cifrar a mensagem, aplica-se a função de encriptação $E_k(x)$, onde ***x*** é a mensagem original e ***k*** a chave. Para decriptar a mensagem, usa-se a função $D_k(x)$.

ENCRIPTAR (CIFRAR)



$$E_k(x) = (x + k)\ mod\ 26$$

DECRIPTAR (DECIFRAR)



$$D_k(x) = (x - k)\ mod\ 26$$

⚠ Como a chave usada para encriptar e decriptar a mensagem é a mesma, a chamamos de **Chave Simétrica**.

⚠ Como o alfabeto usado possui 26 letras, significa que temos ***k-1*** (25) combinações possíveis, o que chamamos de **Espaço de Chaves**.

⚠ O operador ***mod 26*** (resto de divisão inteira) garante que a função seja limitada ao espaço de chaves do alfabeto.

Cifra de César – exemplo

**Leônidas I, Rei e General da cidade-estado de Esparta (491 AC à 480 AC).*

Cifra de César – exemplo

A RESISTENCIA SE DARA NO DESFILADEIRO DAS TERMOPILAS

operação[+] → chave[3]

D UHVLVWHQFLD VH GDUD QR GHVILODGHLUR GDV WHUPRSLODV

operação[-], chave[3]

A RESISTENCIA SE DARA NO DESFILADEIRO DAS TERMOPILAS

Cifra de César – força bruta

TEXTO CIFRADO

D UHVLVWHQFLD VH GDUD QR
GHVILODGHLUR GDV WHUPRSLODV

FORÇA BRUTA

CHAVE

X OBPFPQBKZFX PB AXOX KL ABPCFIXABFOL AXP QBOJLMFIXP	-6
Y PCQGQRCLAGY QC BYPY LM BCQDGJYBCGPM BYQ RCPKMNGJYQ	-5
Z QDRHRSDMBHZ RD CZQZ MN CDREHKZCDHQN CZR SDQLNOHKZR	-4
A RESISTENCIA SE DARA NO DESFILADEIRO DAS TERMOPILAS	-3
B SFTJTUFODJB TF EBSB OP EFTGJMBEFJSP EBT UFSNPQJMBT	-2
C TGUKUVGPEKC UG FCTC PQ FGUHKNCFGKTQ FCU VGTOQRKNCU	-1

Cifra de César – conclusão

O que garante a segurança do método criptográfico não é o seu algoritmo, mas sim o seu espaço de chaves!

Cifra de César – exercício

**Elfiates, filho de Euridemo de Malis e nascido em Traquis, na Tessália.*

Métodos criptográficos

De acordo com o tipo de chave usada, os métodos criptográficos podem ser subdivididos em duas grandes categorias:

- criptografia de chave simétrica;
- criptografia de chaves assimétricas.

Chave
Compartilhada

Chave Simétrica

A mesma chave compartilhada é usada para cifrar e decifrar a mensagem

Chave
Pública

Chave
Privada

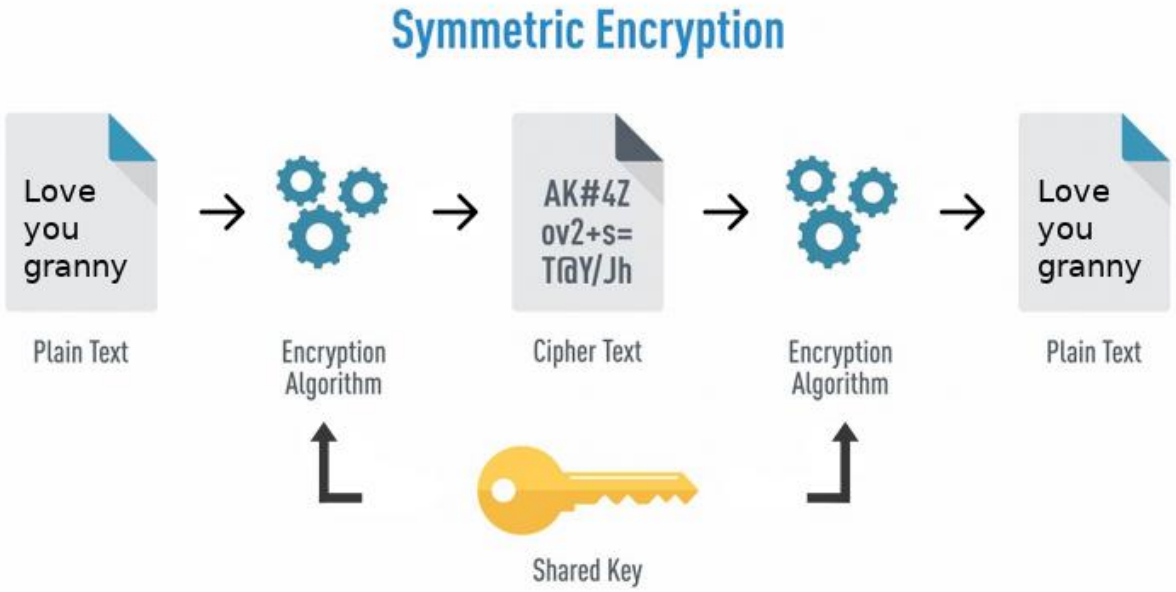
Chaves Assimétricas

São usadas duas chaves, uma para cifrar a mensagem (chave pública) e outra para decifrar (chave privada)

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Criptografia de chave simétrica

Criptografia de chave simétrica ou criptografia de chave secreta ou única, utiliza uma mesma chave tanto para codificar como para decodificar informações, sendo usada principalmente para garantir a confidencialidade dos dados.



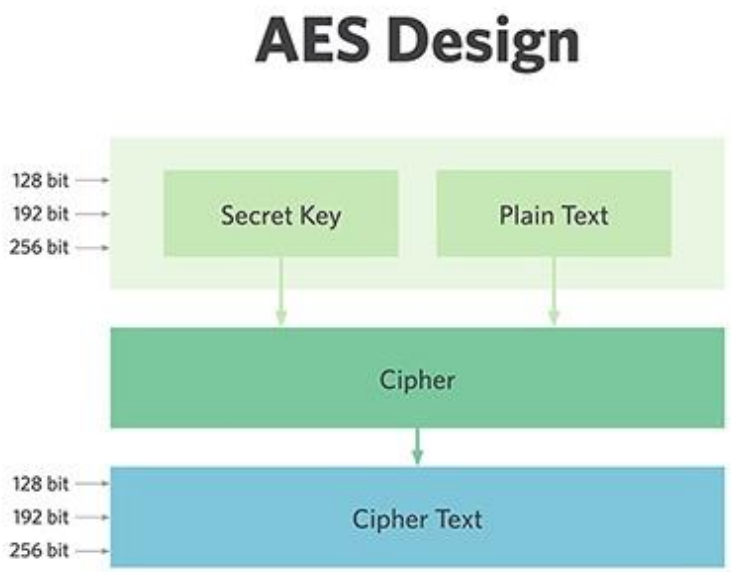
Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Criptografia de chave simétrica

Casos nos quais a informação é codificada e decodificada por uma mesma pessoa não há necessidade de compartilhamento da chave secreta. Entretanto, quando estas operações envolvem pessoas ou equipamentos diferentes, é necessário que a chave secreta seja previamente combinada por meio de um canal de comunicação seguro (para não comprometer a confidencialidade da chave).

Exemplos de algoritmos criptográficos que usam chave simétrica são:

- AES;
- Blowfish;
- RC4;
- 3DES;
- IDEA.



Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Criptografia de chave simétrica

A criptografia de chave simétrica, quando comparada com a de chaves assimétricas, é a mais indicada para garantir a confidencialidade de grandes volumes de dados, pois seu processamento é mais rápido.

Todavia, quando usada para o compartilhamento de informações, se torna complexa e pouco escalável, em virtude da:

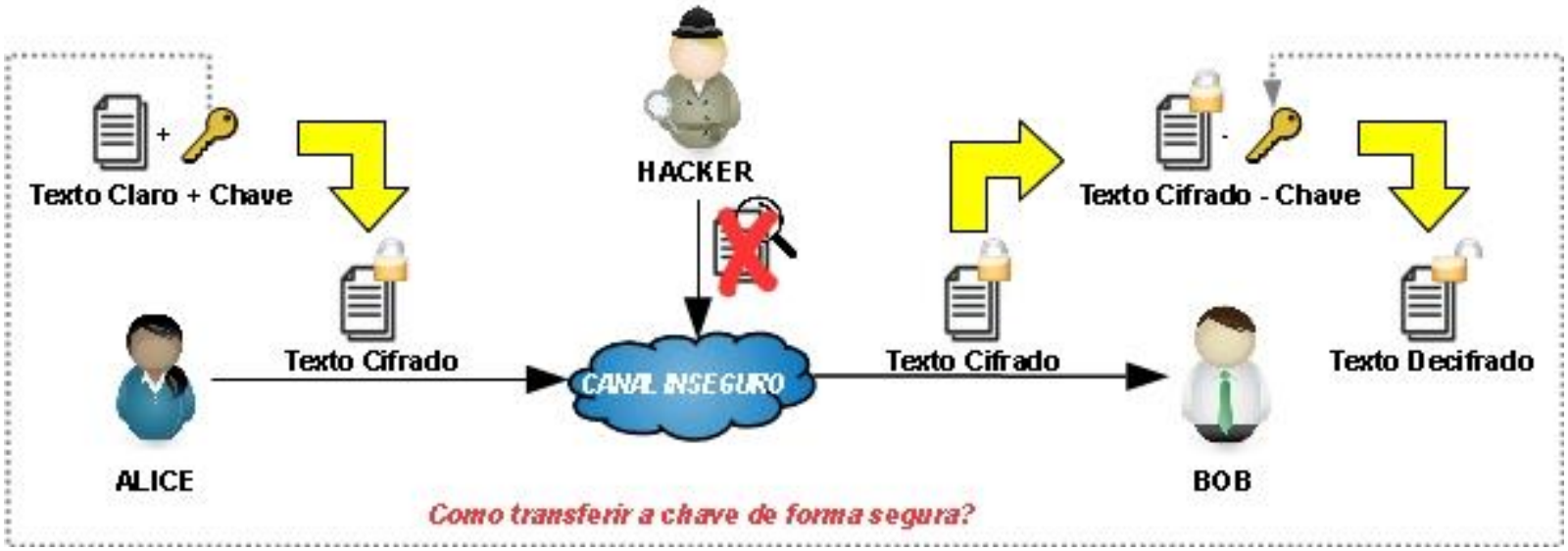
- necessidade de um canal de comunicação seguro para promover o compartilhamento da chave secreta entre as partes (o que na Internet pode ser bastante complicado);
- dificuldade de gerenciamento de grandes quantidades de chaves, uma vez que seriam necessárias várias chaves secretas distintas para cada pessoa com quem quisesse comunicar-se.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Criptografia de chave simétrica

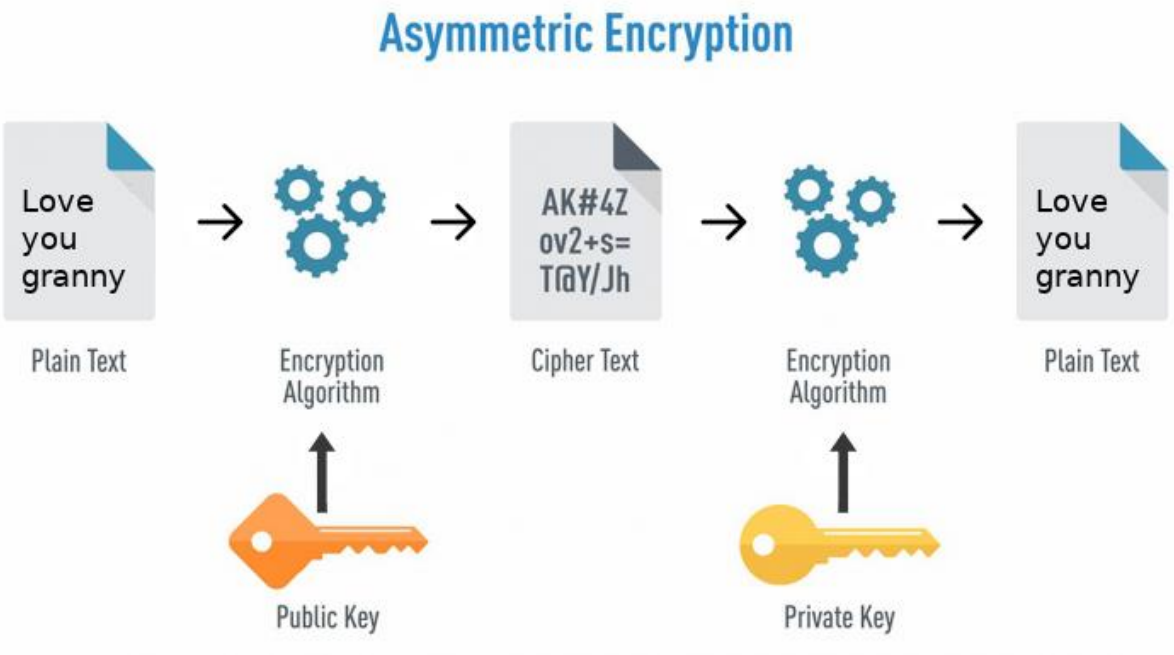
Para Alice enviar uma mensagem para Bob por meio de um canal inseguro, sem que um terceiro (Hacker) intercepte a mensagem e verifique o seu conteúdo, Alice deve cifrar a mensagem e compartilhar uma chave simétrica com Bob.

A questão que fica é: como transferir a chave simétrica de forma segura?



Criptografia de chaves assimétricas

Criptografia de chaves assimétricas, também conhecida como criptografia de chave pública, utiliza duas chaves distintas: uma pública, que pode ser livremente divulgada, e uma privada, que deve ser mantida em segredo por seu dono.



Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Criptografia de chaves assimétricas

Quando uma informação é codificada com uma das chaves, somente a outra chave do par pode decodificá-la. Qual chave usar para codificar depende da proteção que se deseja, se confidencialidade ou autenticação, integridade e não-repúdio.

A chave privada pode ser armazenada de diferentes maneiras, como um arquivo no computador, um *smartcard* ou um *token*, e a sua guarda é de responsabilidade de seu proprietário.

Já a chave pública pode ficar armazenada em um repositório de acesso público.

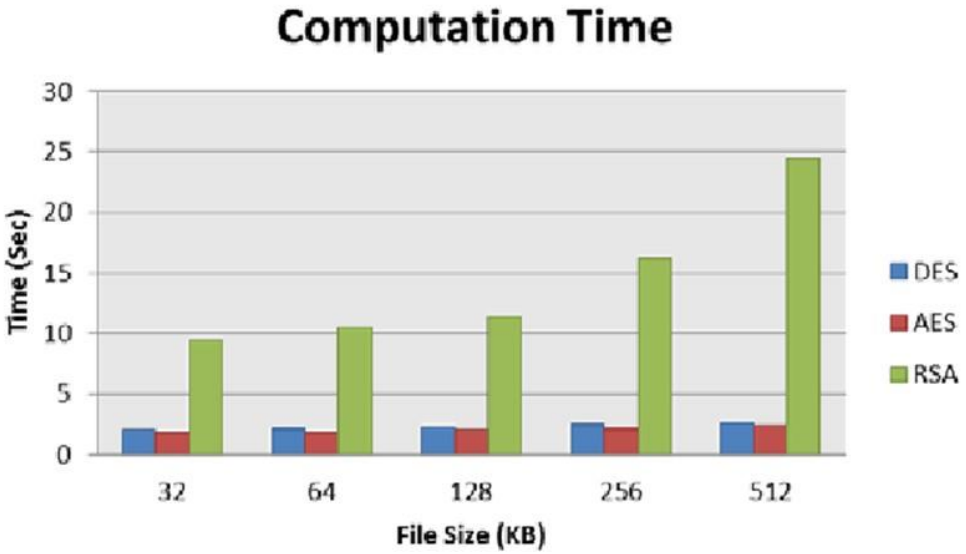
Exemplos de algoritmos criptográficos que usam chaves assimétricas são:

- RSA;
- DSA;
- ECC;
- Diffie-Hellman.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Criptografia de chaves assimétricas

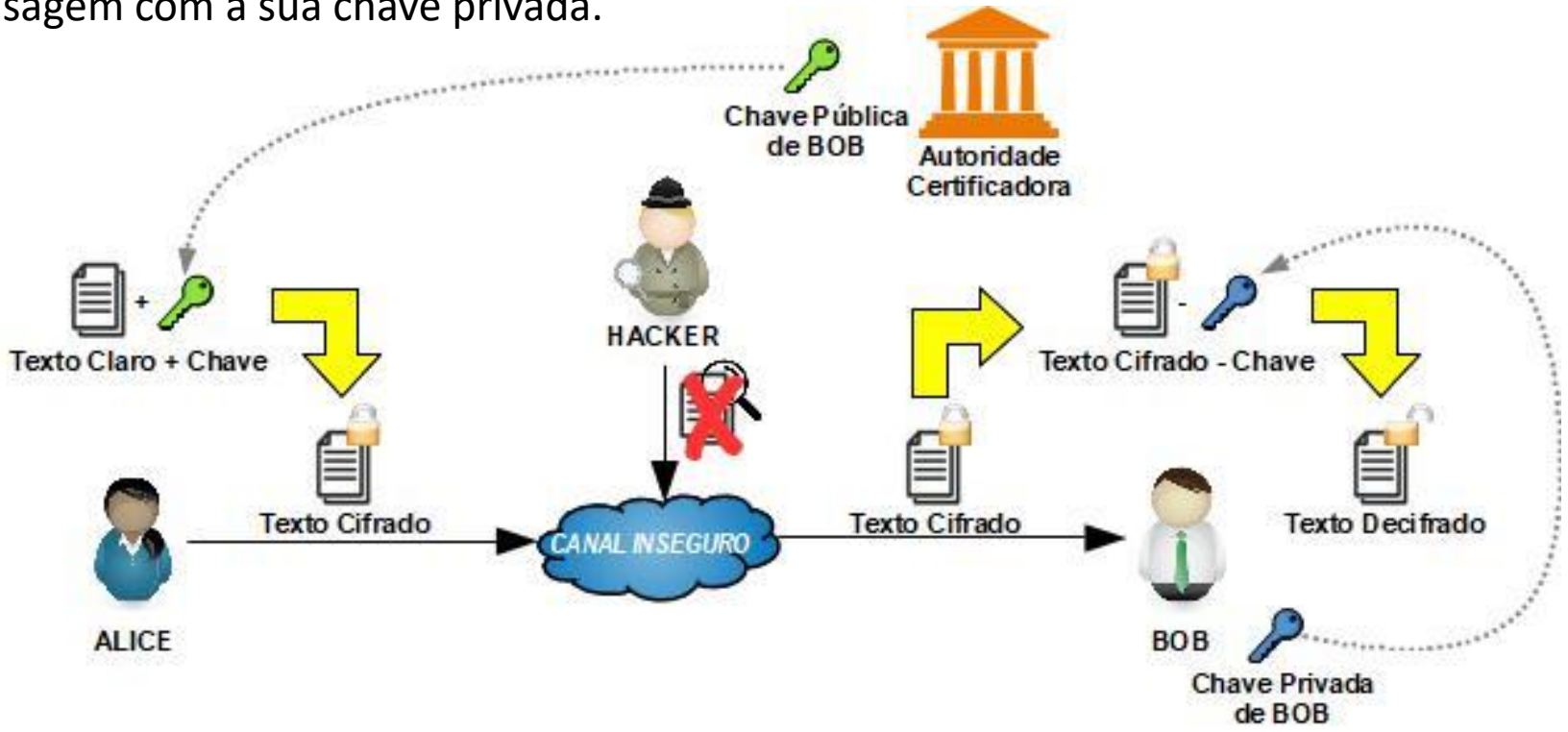
A criptografia de chaves assimétricas, apesar de possuir um processamento mais lento que a de chave simétrica, resolve os problemas de canal seguro e de gerenciamento de chaves, uma vez que facilita o seu gerenciamento (pois não requer que se mantenha uma chave secreta com cada um que desejar se comunicar) e dispensa a necessidade de um canal de comunicação seguro para o compartilhamento de chaves.



Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Criptografia de chaves assimétricas

Para Alice enviar uma mensagem para Bob por meio de um canal inseguro sem que um terceiro (Hacker) intercepte a mensagem e verifique o seu conteúdo, Alice deve cifrar a mensagem com a chave pública de Bob e este deve decifrar a mensagem com a sua chave privada.

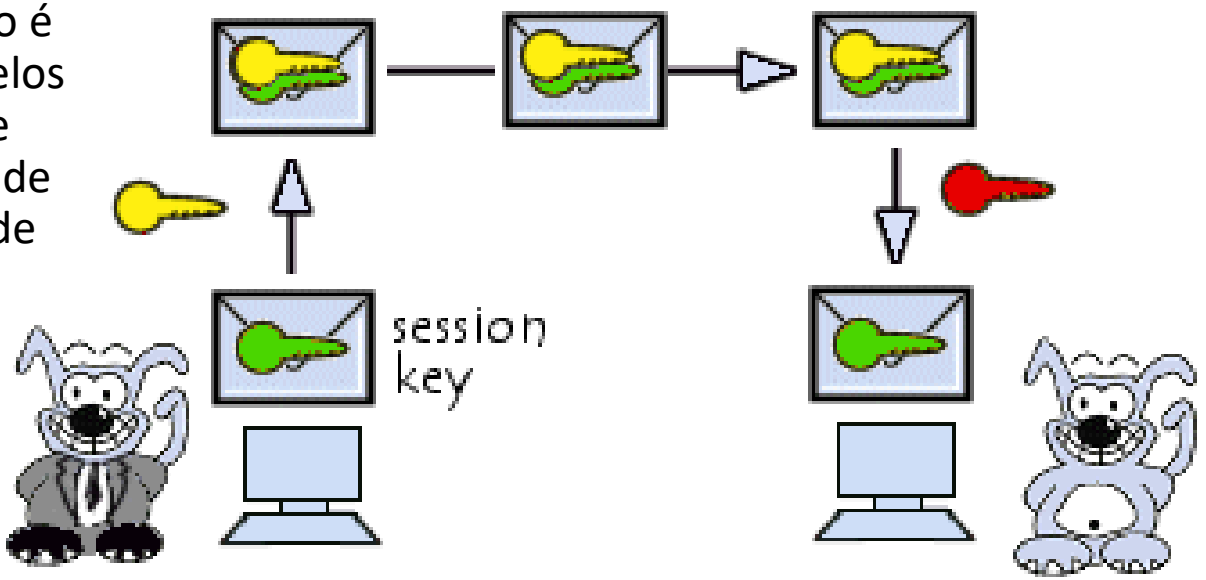


Criptografia – aplicações

Para aproveitar as vantagens dos métodos de chave simétrica e assimétricas, o ideal é o uso combinado de ambos, onde o primeiro é usado para codificar a informação e o segundo é utilizado para o compartilhamento da chave secreta (neste caso, também chamada de chave de sessão).

Este uso combinado é o que é utilizado pelos navegadores Web e programas leitores de e-mails. Exemplos de uso deste método combinado são:

- SSL;
- PGP;
- S/MIME.

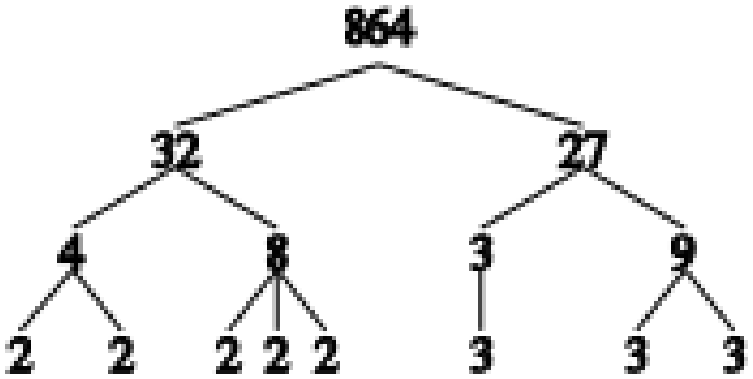


Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

RSA

RSA (Rivest-Shamir-Adleman) é um sistema de criptografia de chave pública amplamente utilizado para transmissão segura de dados. Seu acrônimo vem dos sobrenomes dos criptógrafos Ron Rivest, Adi Shamir e Leonard Adleman, que descreveram o algoritmo em 1978.

Neste sistema de criptografia de chaves assimétricas, a assimetria é baseada na dificuldade prática da fatoração do produto de dois números primos grandes, conhecido como “problema de fatoração”.

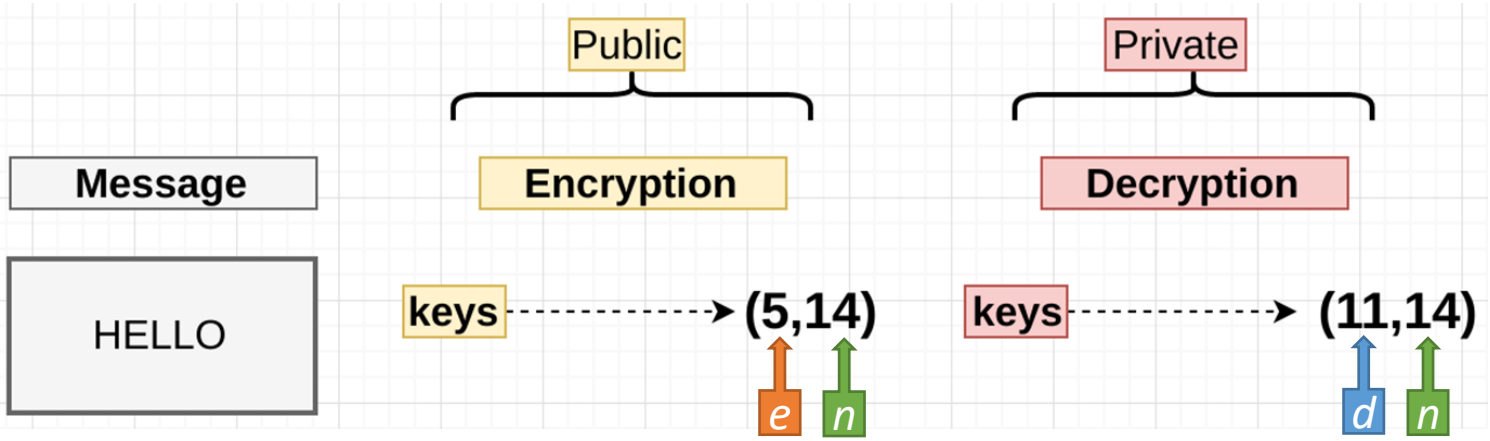


Fonte: www.wikipedia.org

RSA

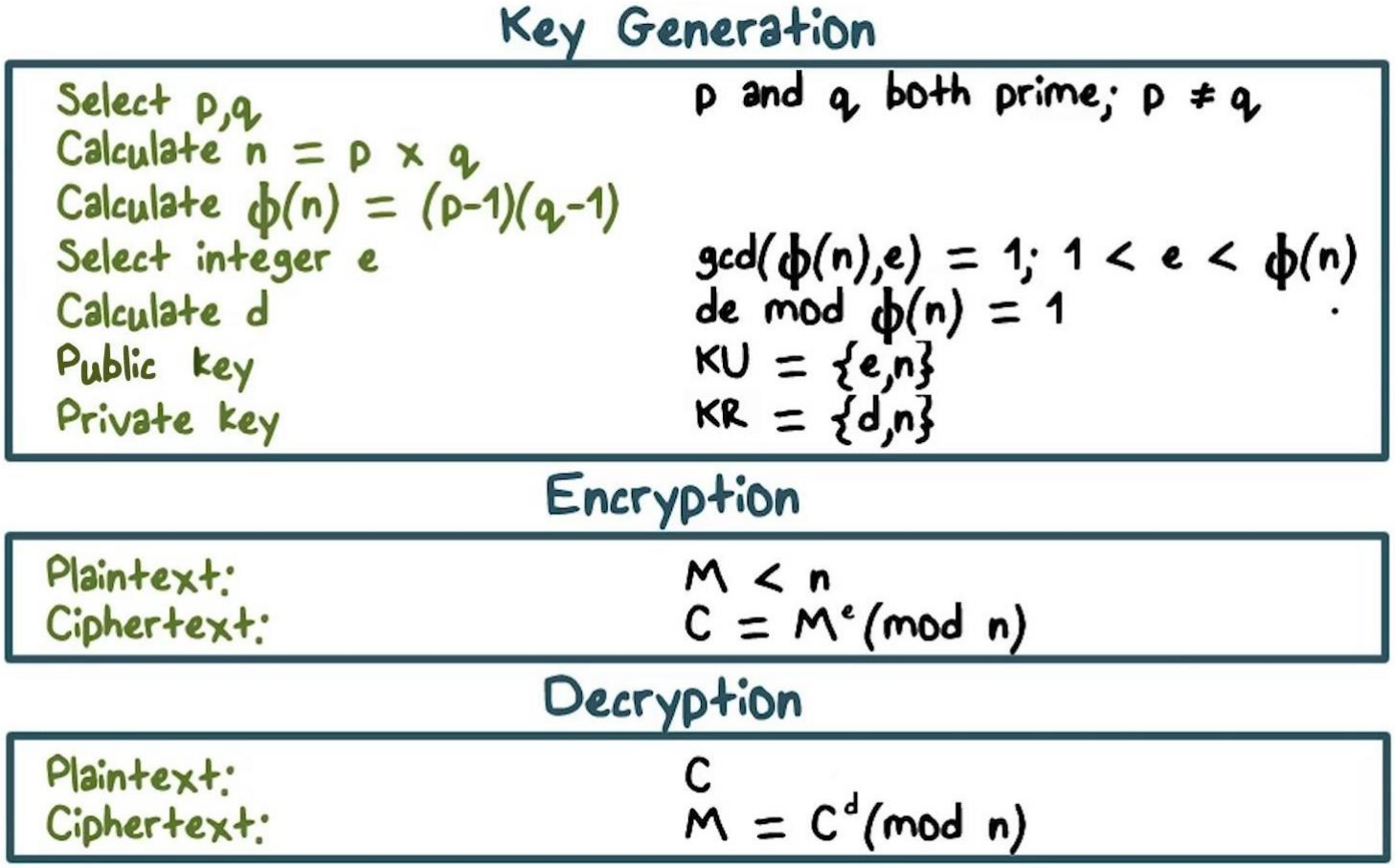
Um usuário do RSA cria e divulga uma chave pública composta pelo produto ***n*** de dois números primos grandes e por um valor auxiliar ***e***. Os números primos devem ser mantidos em segredo. Qualquer um pode usar a chave pública para cifrar a mensagem, mas para decifrá-la é necessária a chave privada composta por ***n*** e o número calculado ***d***, de conhecimento apenas do dono da chave.

Com os métodos atualmente conhecidos e se a chave pública for muito grande, só é possível decifrar a mensagem conhecendo-se os números primos usados originalmente. Quebrar a encriptação RSA é conhecido como problema RSA.



Fonte: www.wikipedia.org

RSA – algoritmo

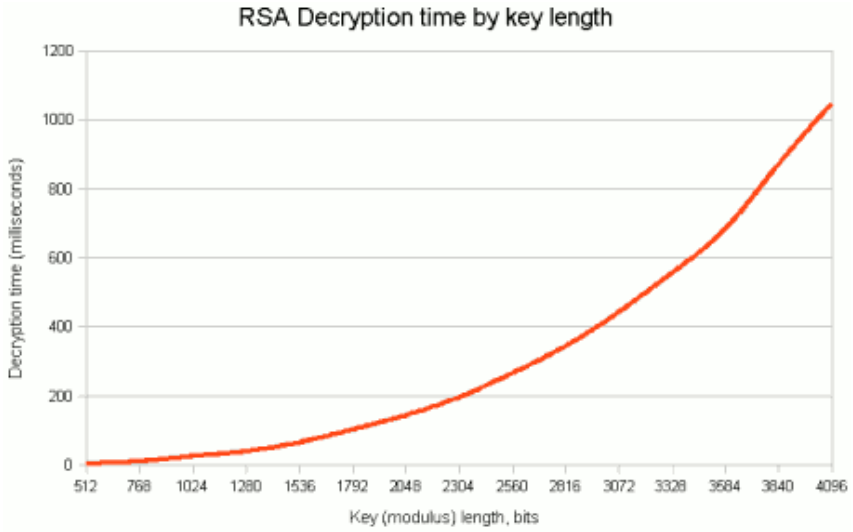


Fonte: www.digicert.com

RSA

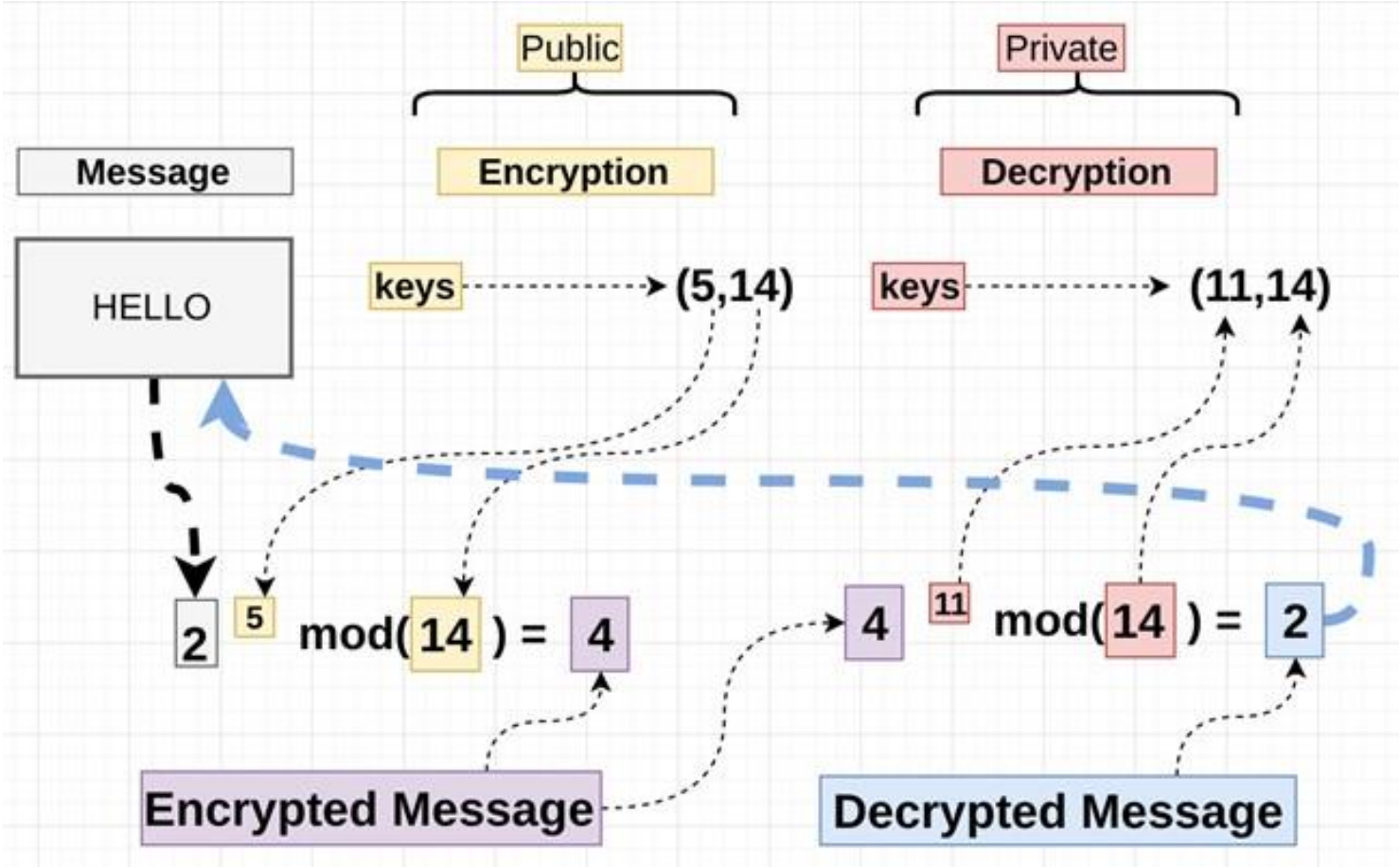
O RSA é um algoritmo relativamente lento e, por isso, é menos usado para cifrar diretamente os dados a serem transmitidos.

O RSA geralmente é usado em conjunto com algum algoritmo de chave simétrica, permitindo que a chave compartilhada possa ser transmitida de forma segura, e esta por sua vez, possa executar operações de cifrar-decifrar em massa a uma velocidade muito maior.



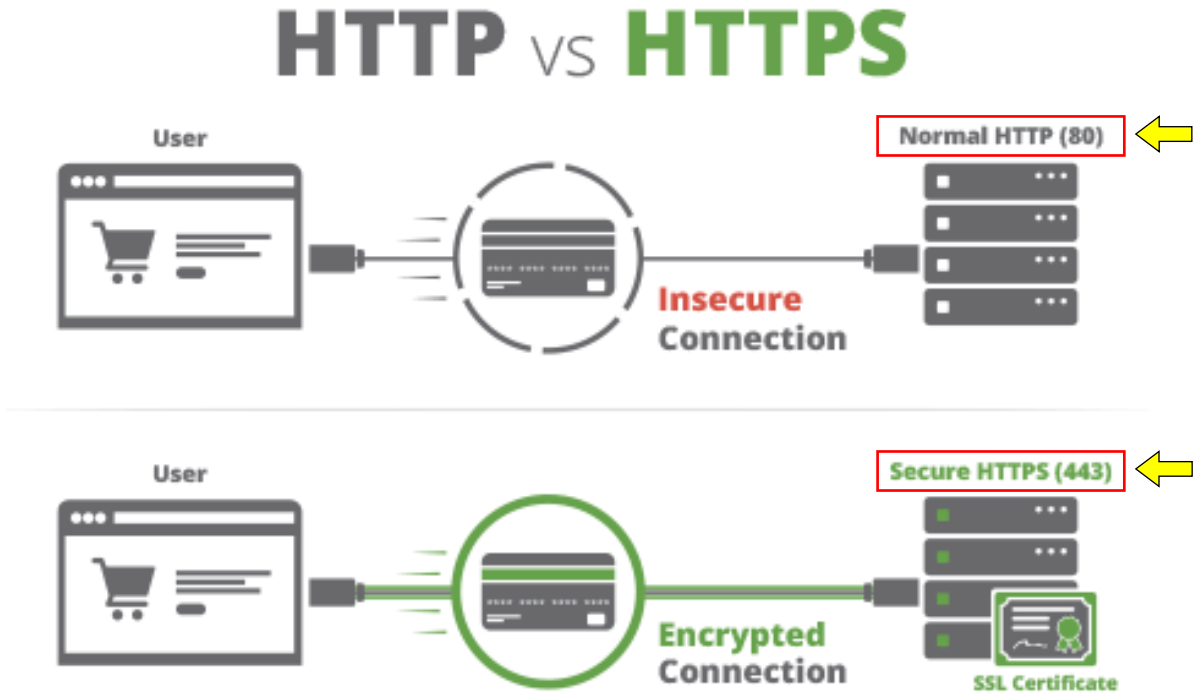
Fonte: www.wikipedia.org

RSA – exemplo


Public
Private
Message
Encryption
Decryption
HELLO
keys
(5,14)
keys
(11,14)
2
5
mod(14) = 4
4
11
mod(14) = 2
Encrypted Message
Decrypted Message

Secure Sockets Layer – protocolo

SSL (Secure Sockets Layer) é uma tecnologia de segurança padrão para estabelecer uma conexão criptografada entre um servidor e um cliente, que pode ser um servidor Web (site) e um navegador, ou um servidor de e-mail e um cliente de e-mail.



Fonte: www.digicert.com

Secure Sockets Layer – protocolo

O SSL permite que informações confidenciais, como números de cartão de crédito e credenciais de login sejam transmitidas com segurança, ao invés de serem enviados em texto simples.

O SSL é um protocolo de segurança que descreve como os algoritmos devem ser usados. Nesse caso, o protocolo SSL determina as variáveis da criptografia para o enlace e os dados que estão sendo transmitidos.

Todos os navegadores têm a capacidade de interagir com servidores da Web protegidos usando o protocolo SSL. No entanto, o navegador e o servidor precisam do que é chamado de Certificado SSL para poder estabelecer uma conexão segura.

Fonte: www.digicert.com

Secure Sockets Layer – certificado

O protocolo SSL requer autenticação entre o servidor e o cliente para proteger uma conexão, e para tal é necessário um certificado SSL.

O certificado SSL é emitido por um terceiro confiável, normalmente uma autoridade de certificação (AC).

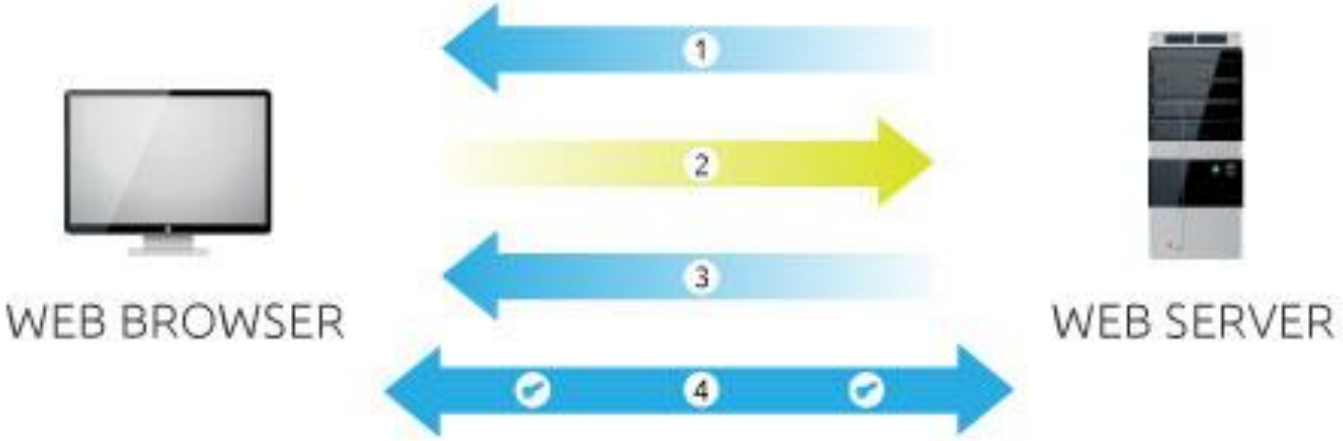
O certificado SSL vincula o domínio, o servidor ou o nome do servidor à identidade e localização de uma organização.

O certificado é instalado no servidor web da organização e fornece uma conexão segura entre o navegador do usuário e o servidor quando solicitado.

ATENÇÃO: protocolo SSL é diferente de certificado SSL!

Fonte: www.digicert.com

Secure Sockets Layer – aplicação

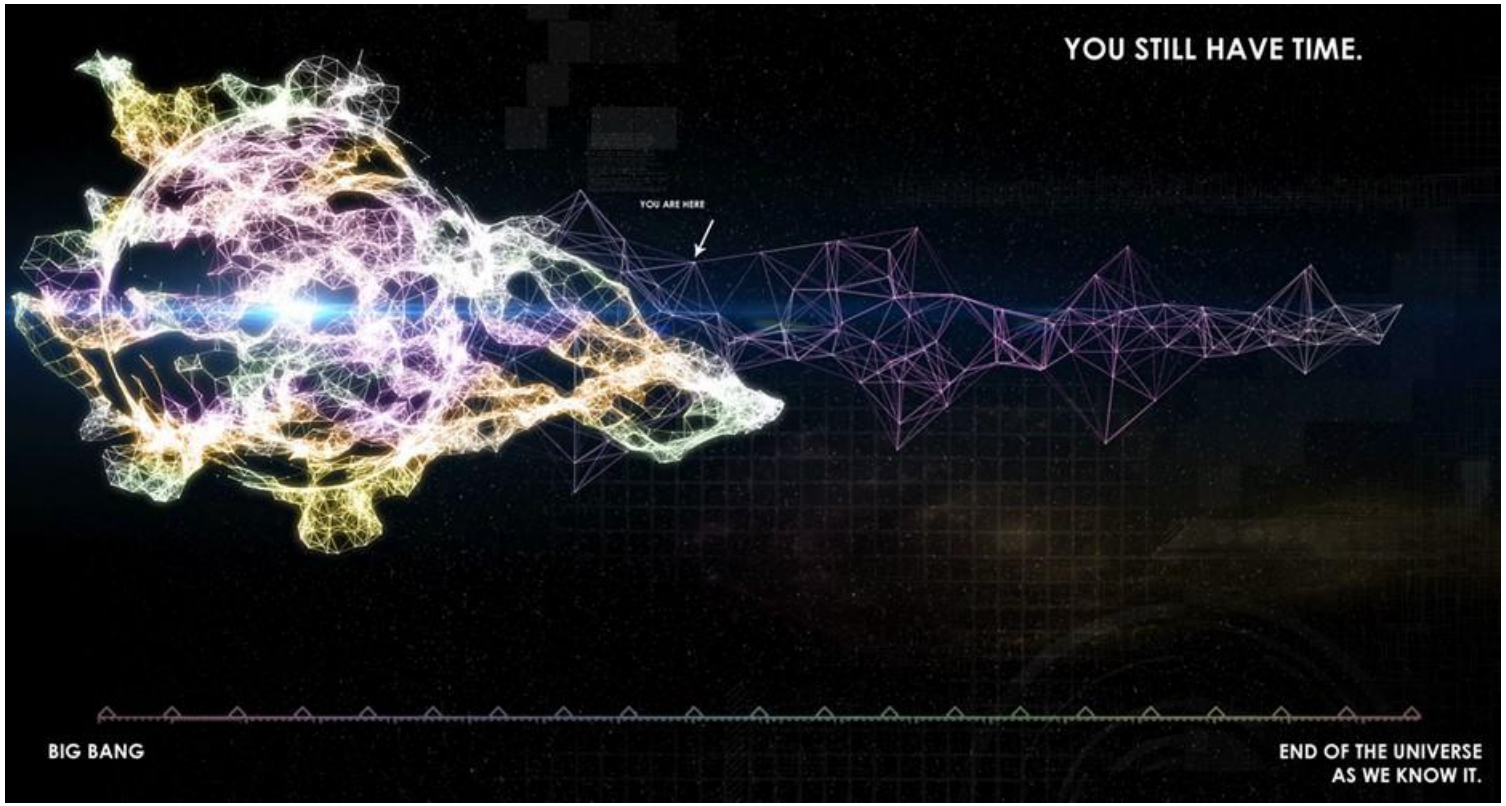


1. O Web Server envia uma cópia de sua chave pública assimétrica para o Web Browser;
2. O Web Browser cria uma chave de sessão simétrica e a criptografa com a chave pública assimétrica do Web Server e em seguida a envia para o servidor;
3. O Web Server decifra a chave de sessão criptografada usando sua chave privada assimétrica para obter a chave de sessão simétrica;
4. Web Server e Web Browser agora cifram e decifram todos os dados transmitidos com a chave de sessão simétrica. Isso permite um canal seguro porque somente o Web Server e o Web Browser conhecem a chave de sessão simétrica e esta é usada apenas para essa sessão. Se o Web Browser tivesse que se conectar ao mesmo Web Server no dia seguinte, uma nova chave de sessão seria criada.

Fonte: www.digicert.com

Secure Sockets Layer – segurança

Quanto tempo seria necessário para quebrar um certificado SSL de 2048 bits?*



Fonte: www.digicert.com

**Usando um desktop com processador AMD Opteron de 2.2 GHz com 2GB RAM.*

Transport Layer Security

SSL e TLS são protocolos criptográficos que fornecem autenticação e criptografia de dados entre servidores, máquinas e aplicativos que operam em uma rede.

O TLS é o sucessor do SSL, e ao longo dos anos novas versões destes protocolos foram lançadas para solucionar vulnerabilidades e oferecer suporte a conjuntos de algoritmos de codificação mais fortes e seguros.

O SSL foi originalmente desenvolvido pela Netscape e apareceu pela primeira vez em 1995 com o SSL 2.0, também conhecida como SSLv2. Em 1996 foi lançada a versão 3.0 (SSLv3), depois que várias vulnerabilidades foram encontradas na versão anterior.

O TLS foi introduzido em 1999 como uma nova versão do SSL e foi baseado no SSL 3.0. Atualmente o TLS está na versão 1.2.

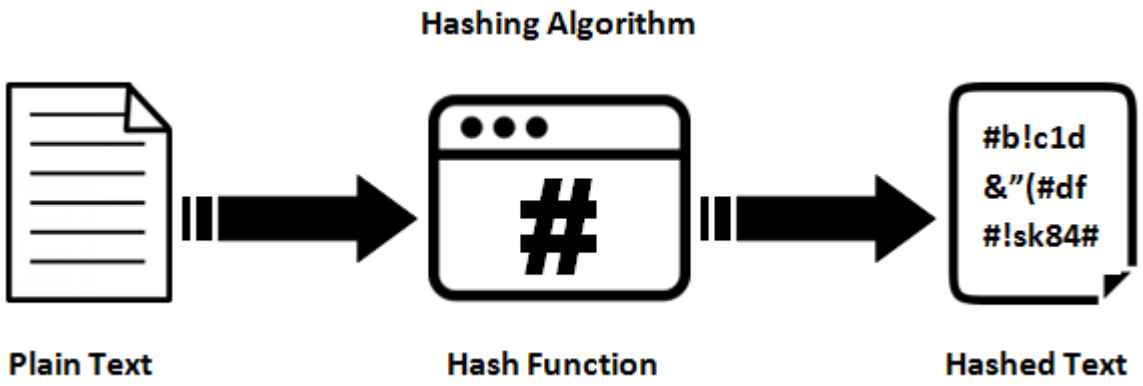
Nota: o SSL 1.0 nunca foi lançado ao público.

Fonte: www.globalsign.com

Função de resumo (*Hash*)

Uma função de resumo é um método criptográfico que, quando aplicado sobre uma informação, independente do tamanho que ela tenha, gera um resultado único e de tamanho fixo, chamado *hash*. Pode-se utilizar *hash* para:

- verificar a integridade de um arquivo obtido da Internet (alguns *sites*, além do arquivo em si, também disponibilizam o *hash* correspondente, para que se possa verificar se o arquivo foi corretamente transmitido e gravado);
- gerar assinaturas digitais.



Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Função de resumo (*Hash*)

Exemplos de métodos de *hash* são:

- SHA-1;
- SHA-256;
- MD5.

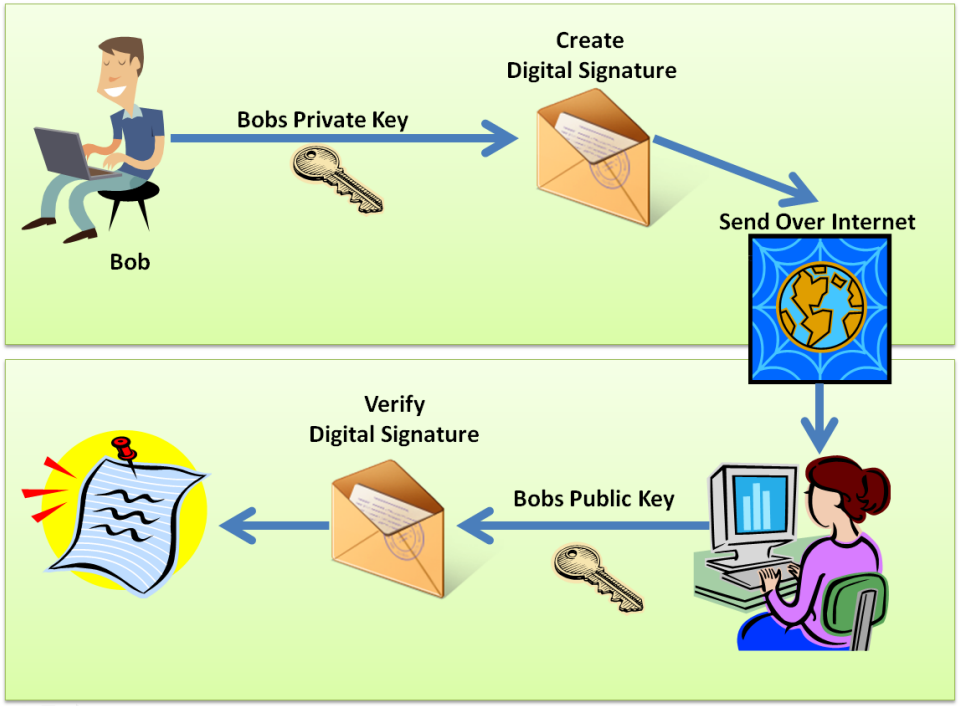
O *hash* é gerado de tal forma que não é possível realizar o processamento inverso para se obter a informação original e qualquer alteração na informação original produzirá um *hash* distinto.

Apesar de ser teoricamente possível que informações diferentes gerem *hashes* iguais, a probabilidade disto ocorrer é bastante baixa.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Assinatura digital

A assinatura digital permite comprovar a autenticidade e a integridade de uma informação, ou seja, que ela foi realmente gerada por quem diz ter feito isto e que ela não foi alterada.



Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Assinatura digital

A assinatura digital baseia-se no fato de que apenas o dono conhece a chave privada e que, se ela foi usada para codificar uma informação, então apenas seu dono poderia ter feito isto.

A verificação da assinatura é feita com o uso da chave pública, pois se o texto foi codificado com a chave privada, somente a chave pública correspondente pode decodificá-lo.

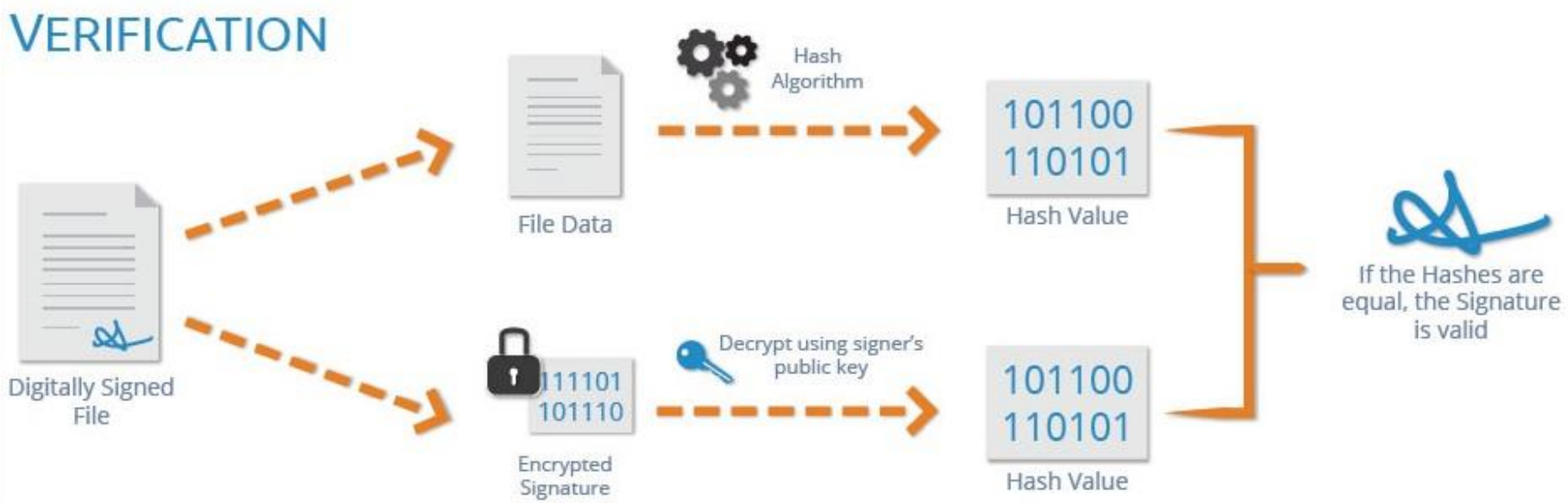
Para contornar a baixa eficiência característica da criptografia de chaves assimétricas, a codificação é feita sobre o *hash* e não sobre o conteúdo em si, pois é mais rápido codificar o *hash* (que possui tamanho fixo e reduzido) do que a informação toda.

A assinatura digital garante apenas que a informação foi de fato gerada pelo proprietário da chave privada.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Assinatura digital – assinando um documento

Assinatura digital – verificando um documento



Certificado digital

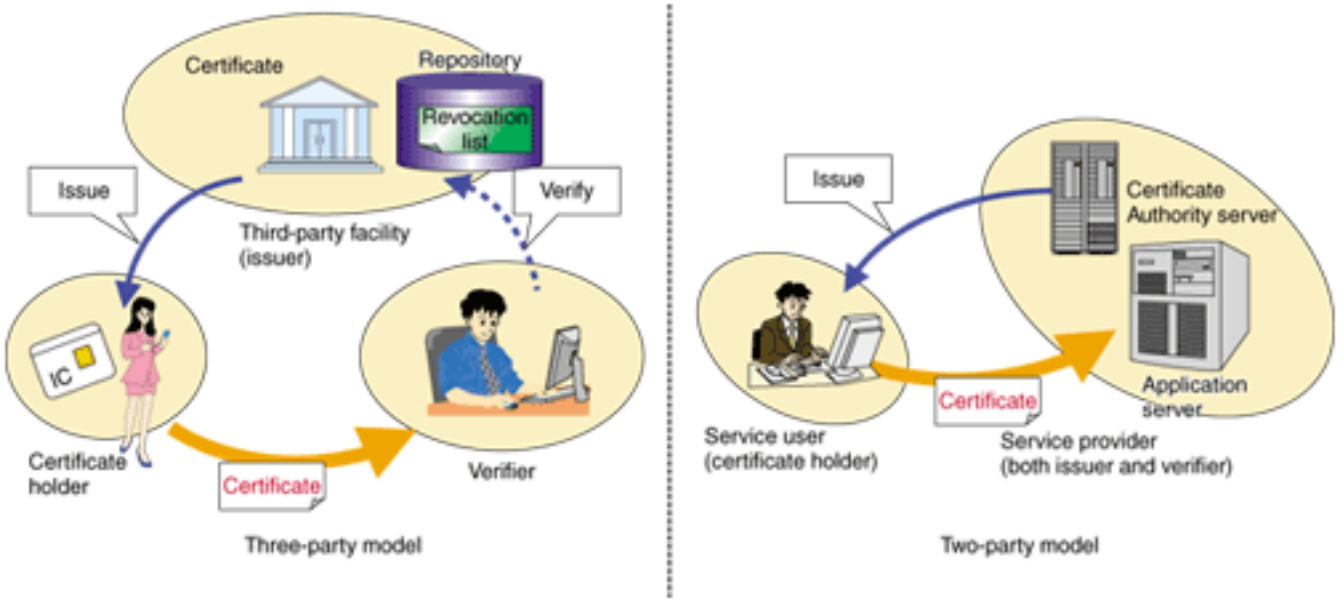
Na assinatura digital a chave púbica pode ser livremente divulgada, mas não há como comprovar a quem ela pertence de fato, e pode ocorrer em uma troca de mensagens da comunicação estar se dando com um impostor.

Um impostor pode criar uma chave pública falsa para um amigo seu e enviá-la para você ou disponibilizá-la em um repositório. Ao usá-la para codificar uma informação para o seu amigo, você estará, na verdade, codificando-a para o impostor, que possui a chave privada correspondente e conseguirá decodificar a mensagem. Uma das formas de impedir que isto ocorra é usando certificados digitais.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Certificado digital

O certificado digital é um registro eletrônico composto por um conjunto de dados que distingue uma entidade e associa a ela uma chave pública. Ele pode ser emitido para pessoas, empresas, equipamentos ou serviços na rede (por exemplo, um *site Web*) e pode ser homologado para diferentes usos, como confidencialidade e assinatura digital.



Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

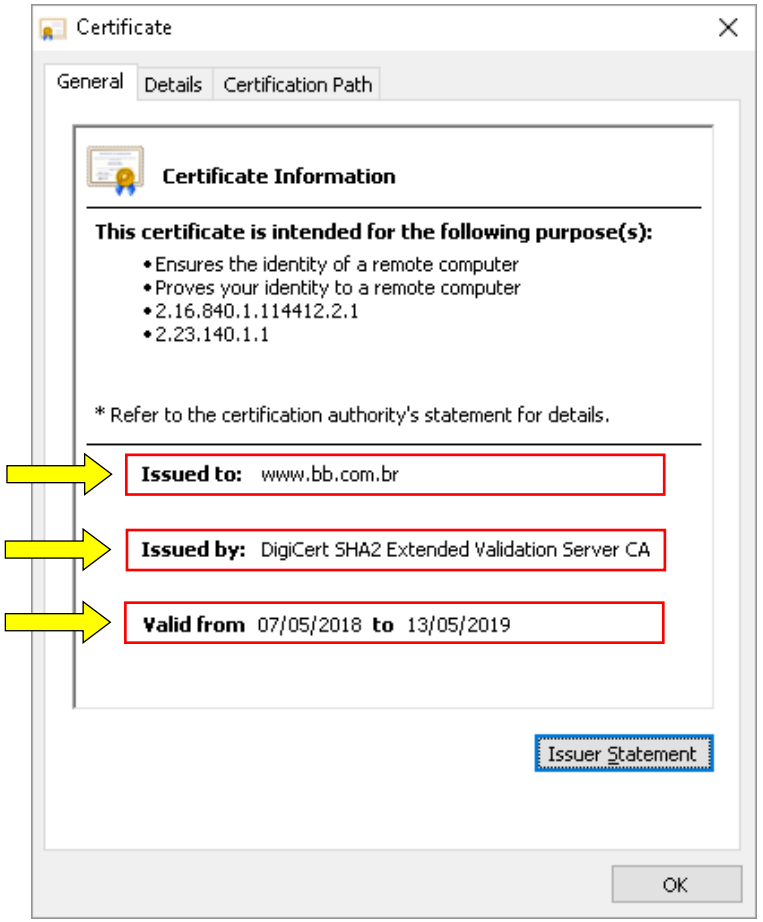
Certificado digital

Um certificado digital pode ser comparado a um documento de identidade, como o passaporte, por exemplo, no qual constam o dados pessoais do requerente e a identificação de quem o emitiu.

No caso do passaporte, a entidade responsável pela emissão e pela veracidade dos dados é a Polícia Federal.

No caso do certificado digital esta entidade é uma Autoridade Certificadora (AC).

O certificado deve conter os dados do emissor (Issued by), do requerente (Issued to) e a data de validade.

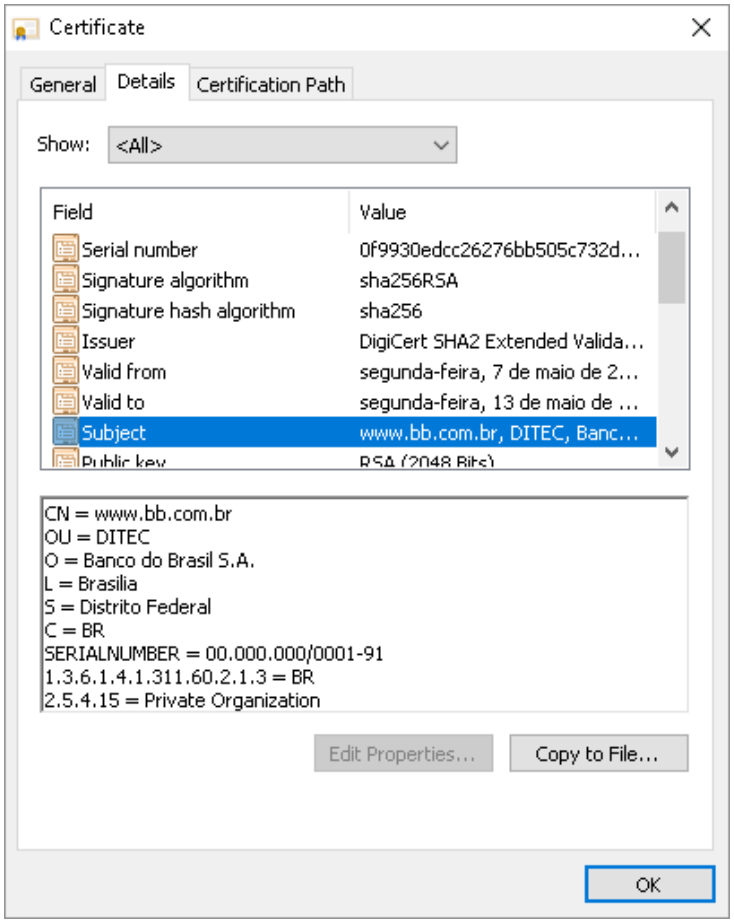


Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Certificado digital

De forma geral, os dados básicos que compõem um certificado digital são:

- versão e número de série do certificado;
- dados que identificam a AC que emitiu o certificado;
- dados que identificam o requerente do certificado (para quem ele foi emitido);
- chave pública do dono do certificado;
- validade do certificado (quando foi emitido e até quando é válido);
- assinatura digital da AC emissora e dados para verificação da assinatura.

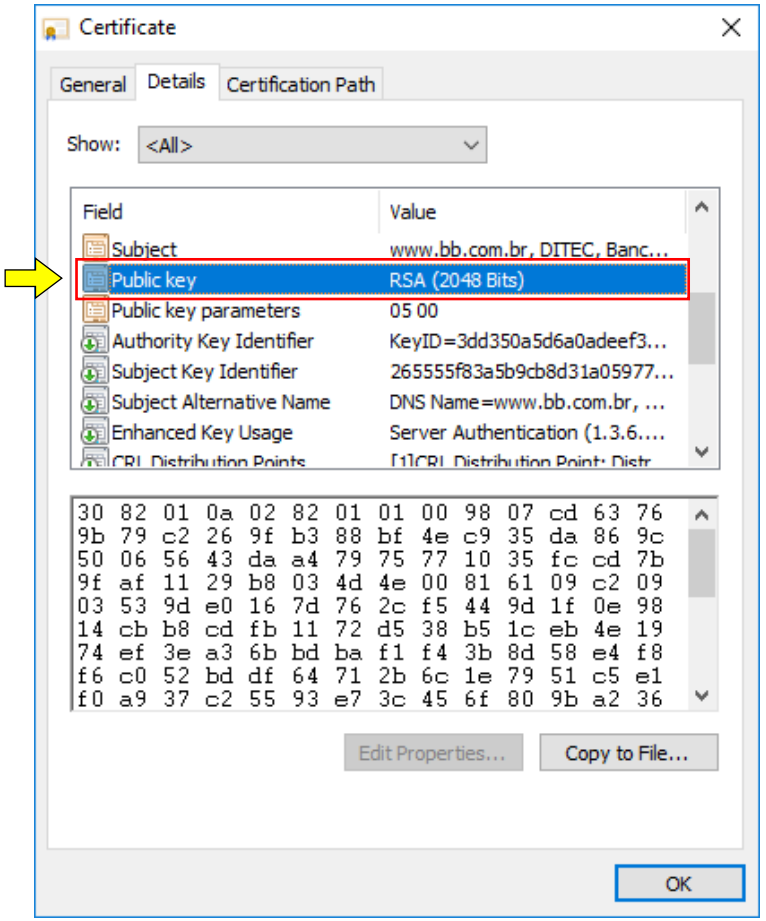


Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Certificado digital

Para verificar o algoritmo usado para gerar a chave pública e o tamanho da mesma em bits, deve-se olhar o campo Public Key.

No exemplo ao lado a chave foi gerada usando-se o algoritmo RSA e a mesma possui 2048 bits de tamanho.

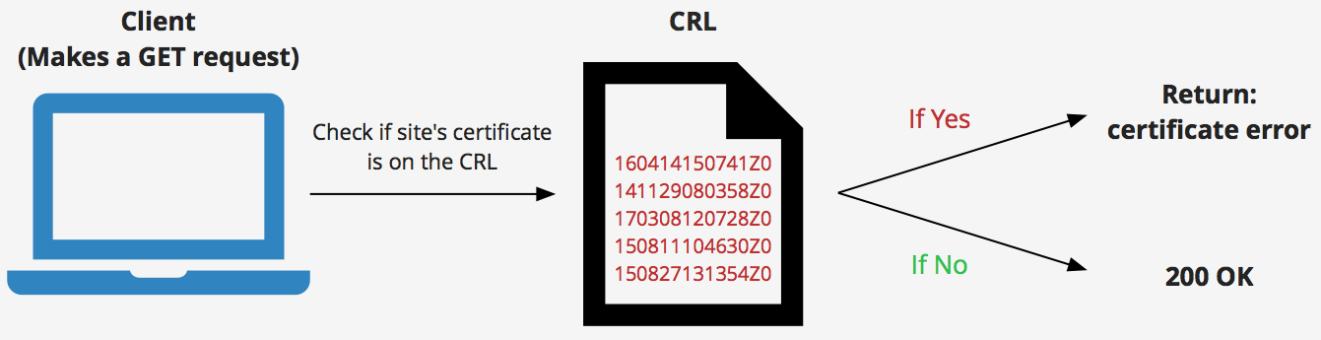


Certificado digital

Uma Autoridade Certificadora (AC ou CA, Certification Authority), além de emitir certificados, também é responsável por publicar informações sobre certificados que não são mais confiáveis.

Sempre que a AC descobre ou é informada que um certificado não é mais confiável, ela o inclui em uma "lista negra", chamada de "Lista de Certificados Revogados" (LCR ou CRL, Certificate Revocation List) para que os usuários possam tomar conhecimento.

A LCR é um arquivo eletrônico publicado periodicamente pela AC, contendo o número de série dos certificados que não são mais válidos e a data de revogação.



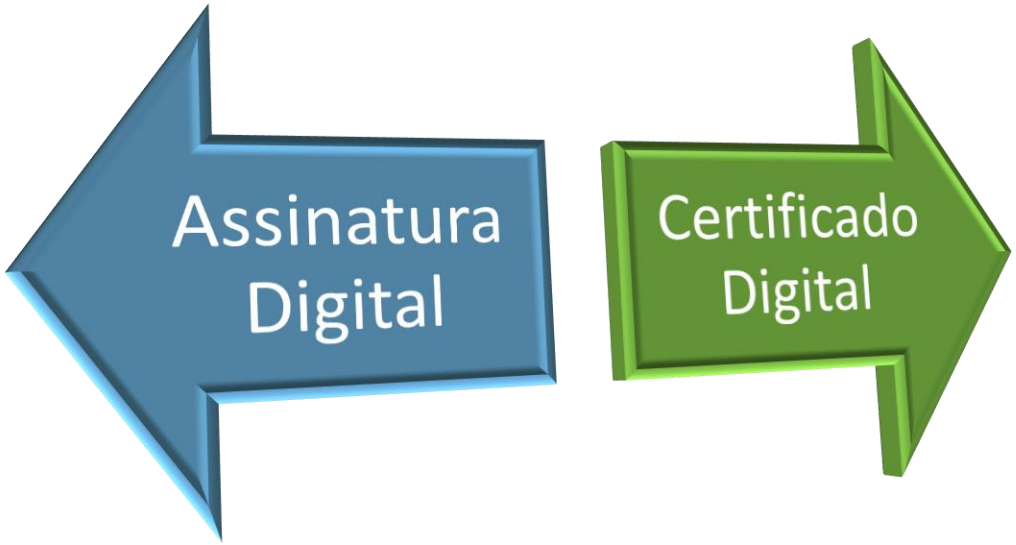
Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Certificado digital

Exemplo de site com certificado inválido

Assinatura digital vs Certificado digital

A assinatura digital garante apenas que a informação foi de fato gerada pelo proprietário da chave privada.



O certificado digital serve para verificar quem é o proprietário da chave privada usada, mas é necessário confiar em um terceiro (AC) que ateste tal propriedade.

Public Key Infrastructure

Com a evolução da criptografia de chaves públicas e o conceito da existência de uma chave pública que pode ser distribuída livremente, surgiram novas necessidades e questões a serem tratadas.

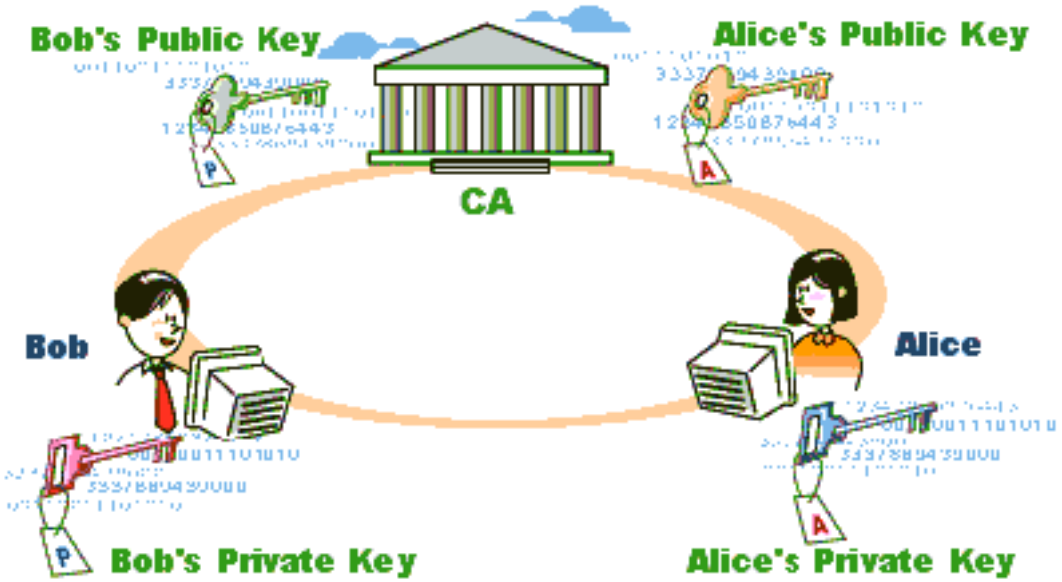
Entre elas merece destaque a questão de como associar uma chave pública ao seu autêntico proprietário. Esta associação pode ser feita através dos certificados digitais.

No entanto, também se faz necessária uma entidade confiável para atestar a ligação entre o proprietário e/ou responsável pela chave e a sua respectiva chave pública.

Fonte: Escola Superior de Redes RNP

Public Key Infrastructure

De acordo com a especificação X.509, foi criado o conceito de uma entidade chamada de Autoridade Certificadora (AC), que é responsável pela identificação do usuário e por atestar que ele possui a chave privada correspondente à chave pública.



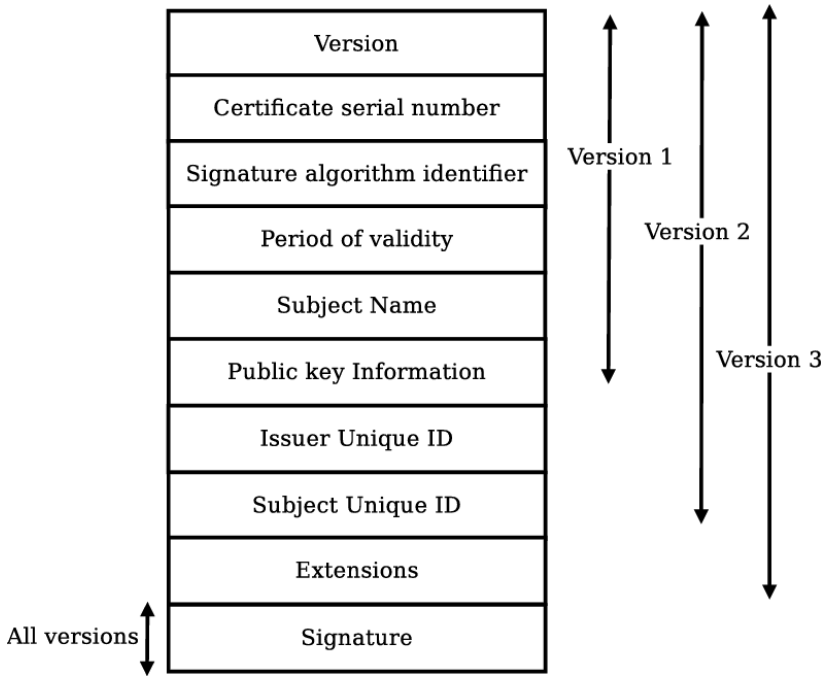
Fonte: Escola Superior de Redes RNP

Public Key Infrastructure

O processo de identificar o usuário e atestar que ele possui a chave privada correspondente à chave pública é realizado através da assinatura de um documento pela AC, que contém dados de identificação do usuário, sua chave pública e outros atributos necessários.

Este documento é chamado de Certificado Digital X.509 e representa o mais básico elemento de uma PKI (ou ICP, Infraestrutura de Chaves Públicas).

O padrão X.509 foi definido pela ITU-T (International Telecommunication Union – Telecommunication Standardization Sector) e especifica, entre outras coisas, o formato dos certificados digitais.



Formato do certificado digital X.509

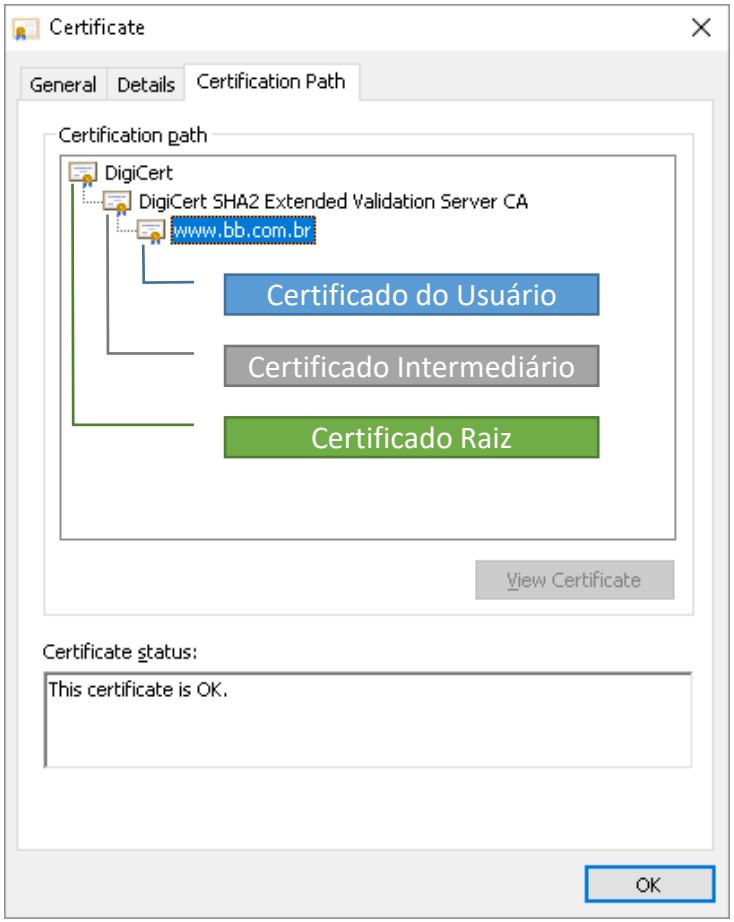
Fonte: Escola Superior de Redes RNP

Public Key Infrastructure

O certificado digital de uma AC é emitido, geralmente, por outra AC, estabelecendo uma hierarquia conhecida como “cadeia de certificados” ou “caminho de certificação”.

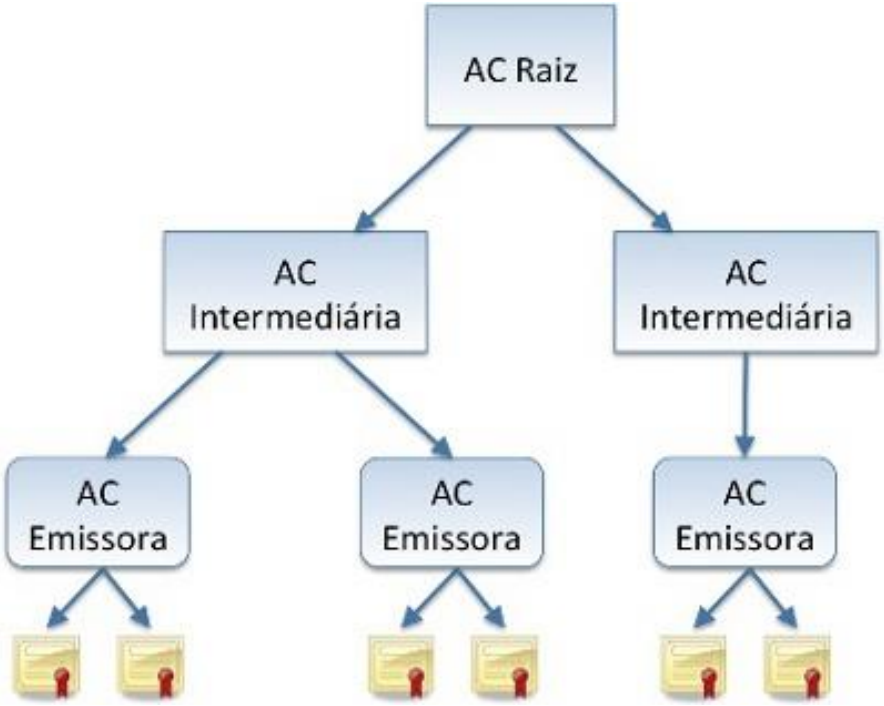
A AC raiz, primeira autoridade da cadeia, é a âncora de confiança para toda a hierarquia e, por não existir outra AC acima dela, possui um certificado autoassinado.

Um certificado autoassinado é aquele no qual o dono e o emissor são a mesma entidade.



Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Public Key Infrastructure



Hierarquia de Entidades Certificadoras

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Public Key Infrastructure

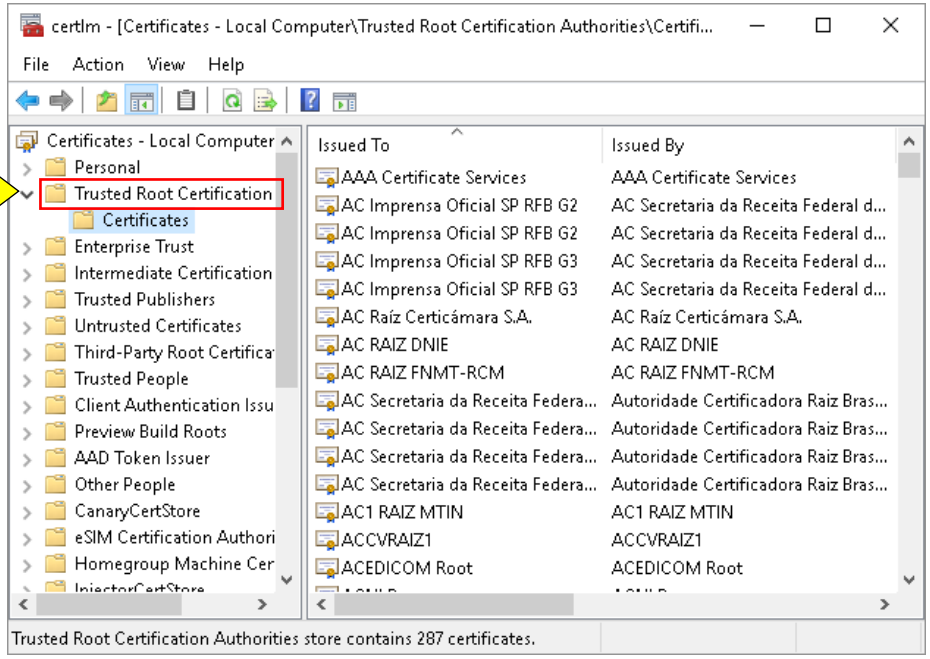
Os certificados das ACs raízes publicamente reconhecidas já vêm inclusos, por padrão, em grande parte dos sistemas operacionais e navegadores e são atualizados juntamente com os próprios sistemas.

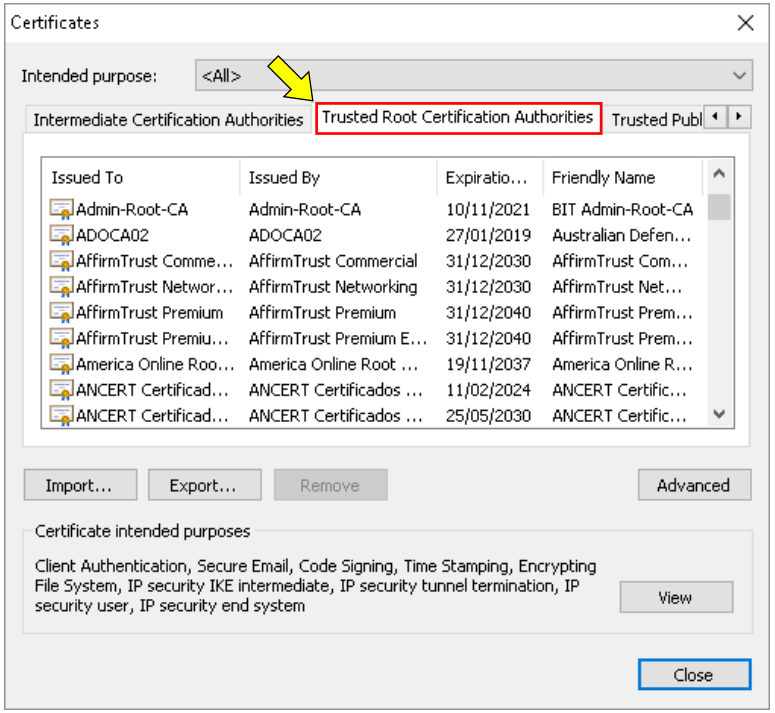
Alguns exemplos de atualizações realizadas na base de certificados dos navegadores são:

- inclusão de novas ACs;
- renovação de certificados vencidos;
- exclusão de ACs não mais confiáveis.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Public Key Infrastructure





ACs reconhecidos pelo sistema operacional Windows 10*

ACs reconhecidos pelo navegador Google Chrome*

*As ACs podem ser gerenciadas (incluídas, alteradas ou removidas) pelo administrador do sistema.

Public Key Infrastructure
Autoridade Certificadora

A Autoridade Certificadora (AC) é composta por hardware, software e pessoas que a operam. É o elemento de uma ICP responsável pela emissão de certificados, emissão de LCRs, gerenciamento e publicação das informações sobre certificados revogados, além de ser capaz de delegar determinadas funções a outras entidades.

Fonte: Escola Superior de Redes RNP

Public Key Infrastructure
Autoridade Certificadora

Ao emitir um certificado, uma AC assegura que a entidade requisitante detém a chave privada correspondente à chave pública contida no certificado. Os certificados emitidos podem ser para outras ACs (conhecidas como ACs intermediárias), para entidades finais ou para ambos.

Quando emite LCRs, uma AC gera uma lista assinada contendo informações sobre os certificados revogados, como a data e o motivo da revogação. De maneira semelhante ao certificado, quando uma AC assina sua LCR, ela atesta seu conhecimento e a autenticidade do conteúdo da lista.

Fonte: Escola Superior de Redes RNP

Public Key Infrastructure
Autoridade Certificadora

Uma infraestrutura de chaves públicas pode ser constituída por uma única AC, porém em muitos casos faz-se necessário que determinadas tarefas sejam delegadas a outras entidades a fim de minimizar a carga de tarefas sobre a AC.

Por exemplo, uma AC pode delegar a outra AC, denominada AC intermediária, a emissão de certificados em seu nome, ou então delegar a emissão da LCR a outra AC.

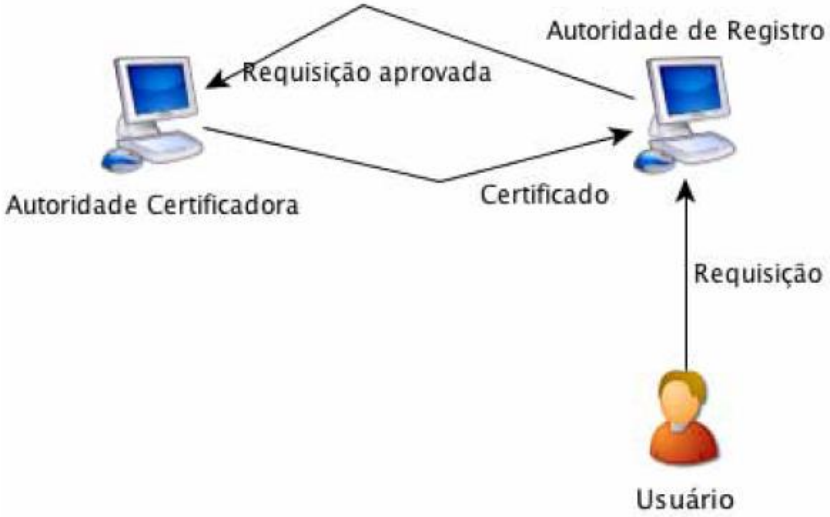
Outra delegação de tarefa bastante comum em uma AC é a de delegar o processo de identificação dos usuários para uma entidade chamada Autoridade de Registro (AR).

Fonte: Escola Superior de Redes RNP

Public Key Infrastructure
Autoridade de Registro

A Autoridade de Registro (AR) é uma entidade composta por software, hardware e operadores para os quais a AC delega a tarefa de verificar o conteúdo de requisições de certificados. Uma AC pode delegar a tarefa de verificação de informações para várias ARs, que podem desempenhar seu papel para várias ACs.

A existência desta entidade em uma ICP faz-se necessária de acordo com a abrangência que uma AC pode ter, seja ela por sua distribuição geográfica, ou por um elevado número de usuários.



Fonte: Escola Superior de Redes RNP

Public Key Infrastructure
Repositório

O Repositório de Certificados Digitais também atua por delegação da AC, e é normalmente composto por software e hardware com o objetivo de publicar os certificados digitais e listas de certificados revogados atuais emitidos por uma ou mais ACs.

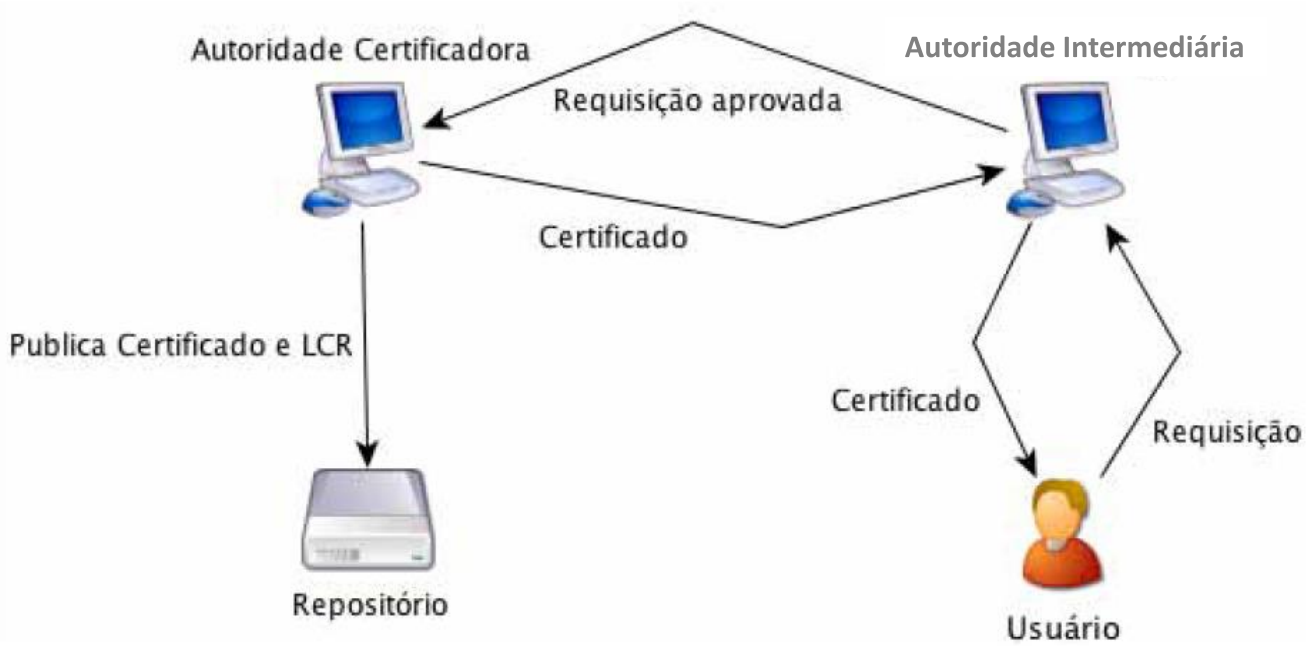
Os dados disponibilizados e armazenados pelo Repositório de Certificados Digitais são assinados pela AC representada por ele, garantindo sua integridade e sua autenticidade e tornando-o imune a ataques de substituição e fabricação.

O Repositório de Certificados Digitais é uma parte da ICP que precisa estar sempre disponível, e por isso necessita de medidas de segurança que garantam a sua disponibilidade.

Fonte: Escola Superior de Redes RNP

Public Key Infrastructure
ACs Intermediárias

Uma AC pode delegar a responsabilidade de emissão de certificados para uma ou mais ACs Intermediárias.



Fonte: Escola Superior de Redes RNP

Public Key Infrastructure
ACs Intermediárias

Os motivos pelos quais uma AC pode fazer isto são:

- Redução da carga de trabalho sobre uma AC, fazendo com que a AC Raiz tenha que emitir um número menor de certificados, dividindo esta tarefa com outras Acs;
- Facilitar o crescimento de toda a estrutura da AC;
- Aumentar a abrangência (se necessário), já que com mais ACs é possível distribuir melhor a localização e o escopo das emissões de certificados digitais;
- Melhorar a capacidade de tolerância a erros, uma vez que se uma AC Intermediária tiver problemas, apenas o que está abaixo desta AC será comprometido. Se houvesse apenas a AC Raiz, a estrutura toda seria comprometida.

Se a AC Raiz autorizar, uma AC Intermediária pode delegar a tarefa de emissão para outras ACs abaixo dela. Além disso, pode limitar o número de ACs abaixo dela através do uso de uma extensão específica para este fim.

Fonte: Escola Superior de Redes RNP

Para saber mais...

… leia a Cartilha de Segurança para Internet do Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

... leia a apostila Introdução a Infraestrutura de Chaves Públicas e Aplicações, da Escola Superior de Redes RNP

Módulo 15

Proteção de Redes de Computadores

Introdução

Quando se conecta uma rede à Internet, assegurar que ela estará livre de intrusos é uma questão crítica.

As organizações de um modo geral sentem a necessidade de proteger as comunicações entre seus sistemas espalhados pelas mais diversas localidades.

A forma mais efetiva de proteger a ligação com a Internet é colocar um sistema de firewall entre a rede local e a Internet, de modo a garantir que toda a comunicação entre a rede corporativa e a Internet esteja de acordo com a política de segurança da empresa.

Para a interligação de sistemas dispersos geograficamente que usam uma rede pública para se comunicar, o uso de VPN (Virtual Private Network) fornece a proteção necessária.

Fonte: Getting Started with Check Point FireWall-1, Check Point Software Technologies Ltd.

Firewall

O firewall é um dispositivo de segurança que tem por objetivo aplicar uma política de segurança a um determinado ponto da rede.

O firewall oferece uma das mais importantes opções para segurança da informação porque ele é o primeiro item a ser testado por quem, de fora, tenta acessar os computadores de uma rede corporativa. Como o firewall fica na borda do perímetro de segurança da rede, sua função é a de servir como primeiro ponto de verificação dos pacotes que tentam entrar na rede.

Firewall

Para fazer com que efetivamente seja oferecida uma segurança realista, o firewall deve rastrear e controlar o fluxo de toda comunicação que passa através dele.

Para decidir sobre quais ações se deve tomar sobre serviços baseados em TCP/IP (como por exemplo, o que deve passar, ser rejeitado, encriptado ou feito log de tentativas de comunicação), um firewall deve obter, armazenar, restaurar e manipular informações derivadas de todas as camadas de comunicação e de outras aplicações.

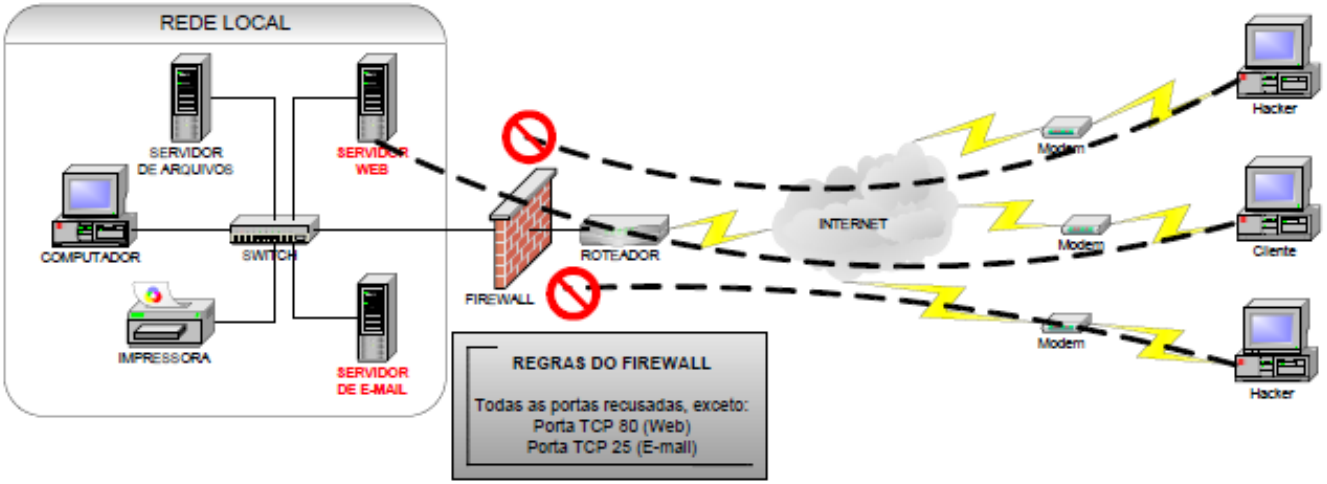
Isto não é suficiente para examinar pacotes isoladamente. O estado da comunicação – derivado de transmissões anteriores e de outras aplicações – é um fator essencial para estabelecer o controle sobre tentativas de comunicação.

Ambos o estado de comunicação (derivado de transmissões anteriores) e o estado da aplicação (derivado de outras aplicações) talvez devam ser considerados quando se quer tomar decisões de controle.

Fonte: Getting Started with Check Point FireWall-1, Check Point Software Technologies Ltd.

Firewall – funções
Filtro de pacotes

O firewall do tipo “filtro de pacotes” analisa individualmente os pacotes à medida que estes são transmitidos, verificando apenas o cabeçalho das camadas de rede e de transporte.



Fonte: Firewalls, de Matthew Strebe e Charles Perkins

Firewall – funções
Filtro de pacotes

Os protocolos TCP, UDP, UDP-Lite, SCTP (Stream Control Transmission Protocol) e DCCP (Datagram Congestion Control Protocol) usam *namespaces* de 16 bits para seus registros de número de porta. Os registros de porta para todos esses protocolos de transporte são subdivididos em três intervalos de números:

- Portas do Sistema, também conhecidas como Portas Bem Conhecidas, de 0 a 1023 (atribuídas pela IANA);
- Portas do Usuário, também conhecidas como Portas Registradas, de 1024-49151 (atribuídas pela IANA);
- Portas Dinâmicas, também conhecidas como Portas Privadas ou Efêmeras, de 49152-65535 (nunca atribuídas).

Fonte: RFC 6335 - Service Name and Port Number Procedures

Firewall – funções
Network Address Translation

O mecanismo *Network Address Translation* foi criado para permitir que redes locais pudessem acessar a Internet sem a necessidade de ter um IP público para cada máquina, pois a quantidade de endereços IPv4 disponíveis estava esgotando-se rapidamente.

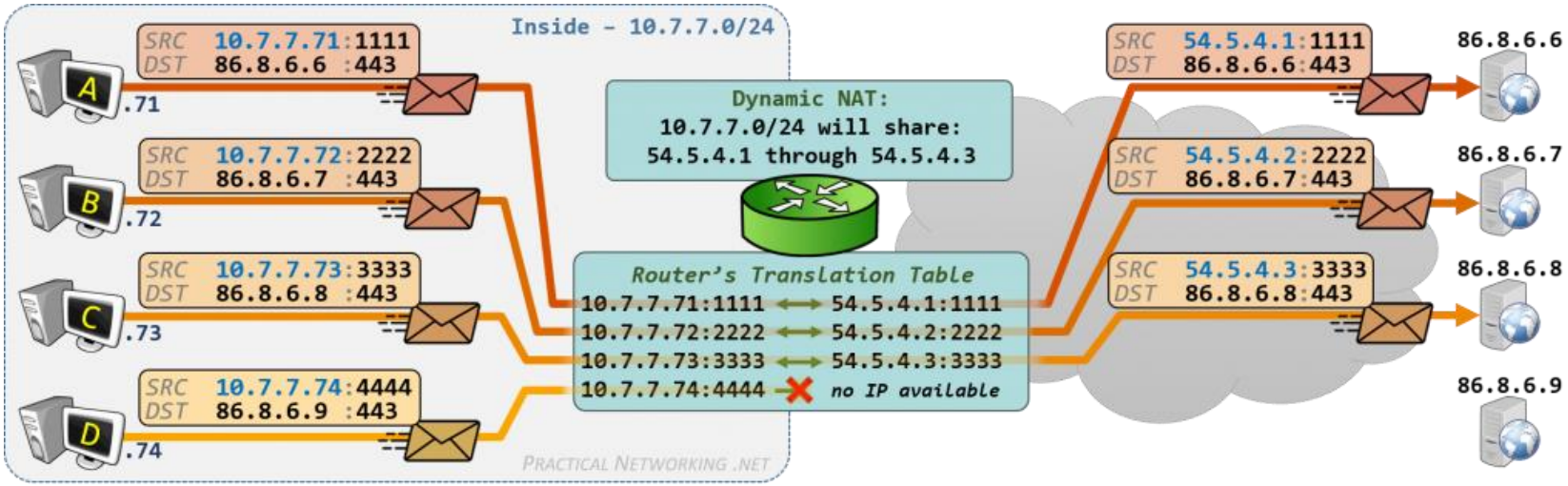
O NAT permite que uma rede local possa navegar na Internet usando uma quantidade de IP´s públicos menor que a quantidade de máquinas.

O que o NAT faz é ocultar os endereços IP internos privados, convertendo-os em um ou mais endereços IP públicos do firewall ou do roteador de borda.

Os dados desta conversão ficam armazenados em uma tabela própria no firewall.

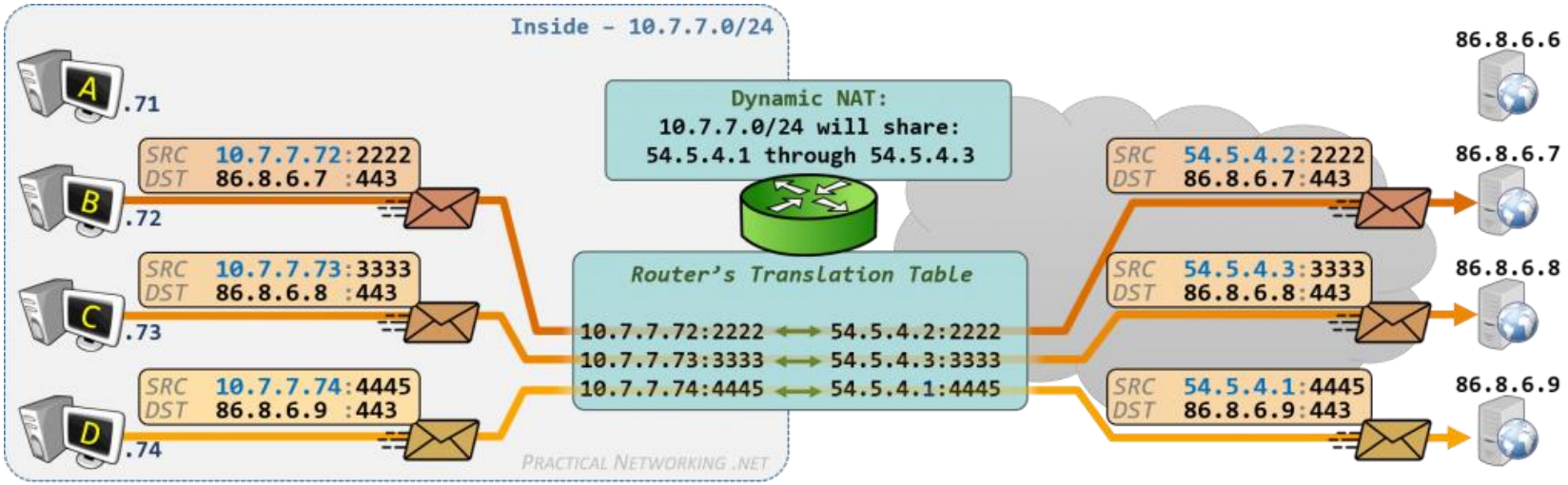
Fonte: Firewalls, de Matthew Strebe e Charles Perkins

Firewall – funções
Network Address Translation



Fonte: Practical Networking .net

Firewall – funções
Network Address Translation



Fonte: Practical Networking .net

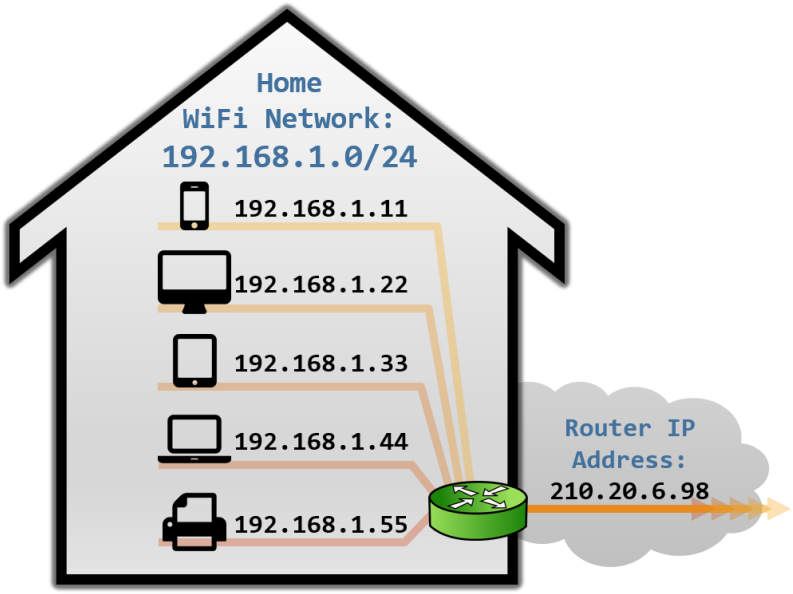
Firewall – funções
Port Address Translation

Da mesma forma que o NAT, o mecanismo *Port Address Translation* também foi criado para permitir que redes locais pudessem acessar a Internet sem a necessidade de ter um IP público para cada máquina. Assim, o PAT permite que os computadores de uma rede local possam navegar na Internet usando um único de IP público.

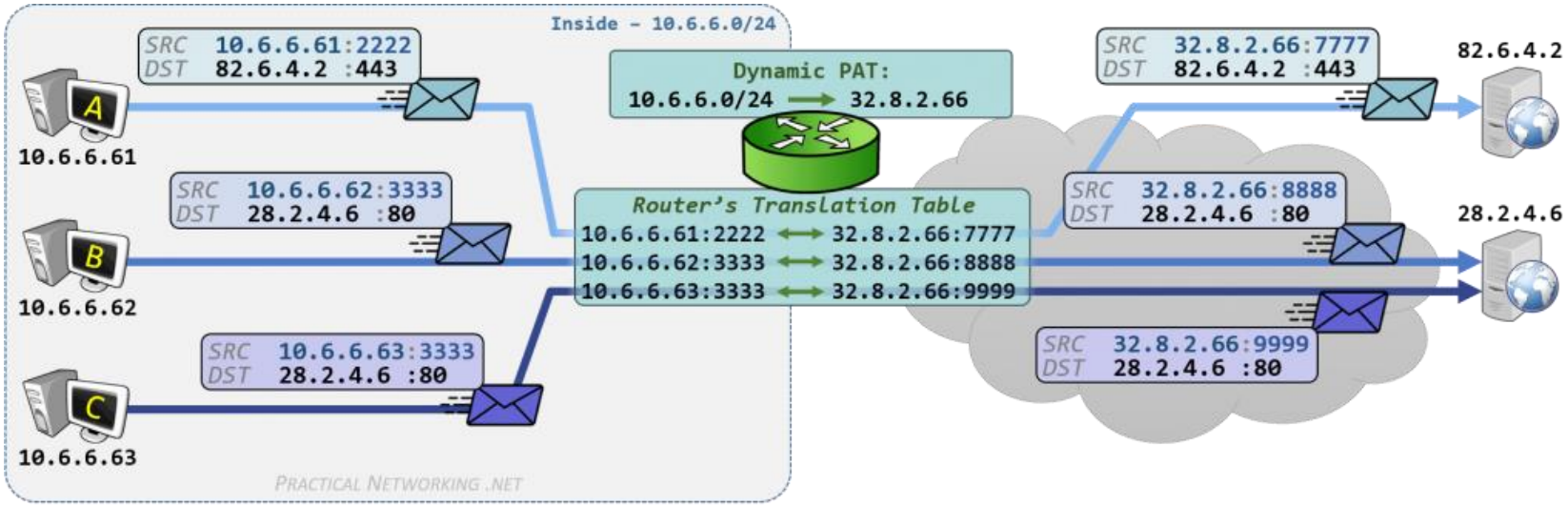
O que o PAT faz é ocultar os endereços IP internos privados, convertendo-os em um endereço IP público do *firewall* ou do roteador de borda.

O firewall então retransmite os dados dos hosts internos a partir de seu próprio endereço, usando o número da porta TCP para saber quais conexões do lado público devem ir para que hosts do lado privado interno. Para a Internet, todo o tráfego da rede parece vir de um único computador "extremamente ocupado".

Os dados desta conversão ficam armazenados em uma tabela própria no firewall.



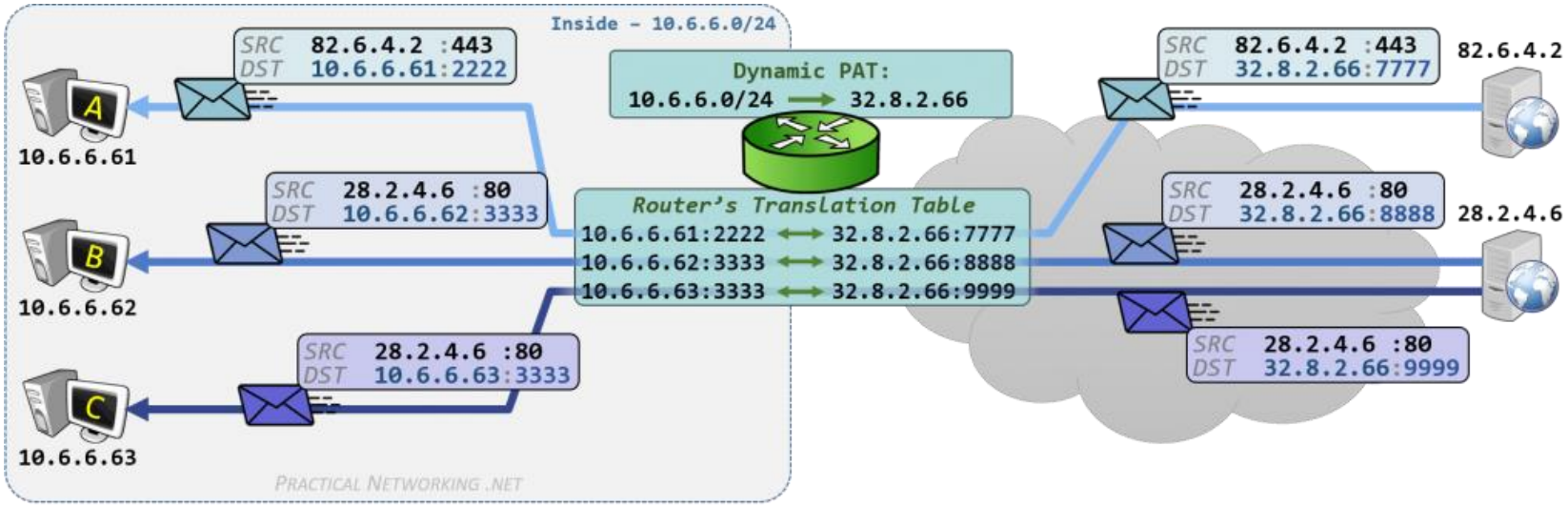
Firewall – funções
Port Address Translation



Tráfego de saída

Fonte: Practical Networking .net

Firewall – funções
Port Address Translation



Tráfego de resposta

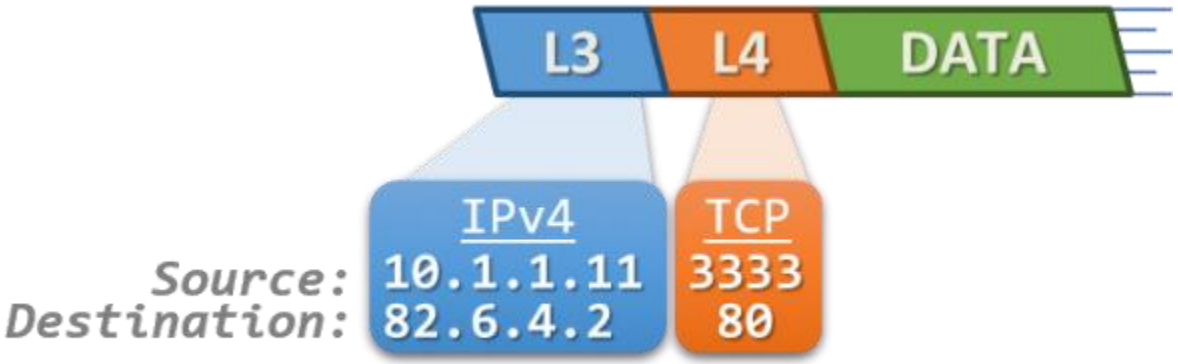
Fonte: Practical Networking .net

Firewall – funções
NAT vs PAT

O NAT refere-se estritamente à conversão de um endereço IP privado em um endereço IP público, ou vice-versa, atuando apenas na camada de rede (L3).

Já o PAT refere-se à conversão de um endereço IP privado mais a porta em um endereço IP público mais a porta, ou vice-versa, atuando na camada de rede (L3) e de transporte (L4).

Na prática o termo NAT é usado indistintamente para os dois casos.



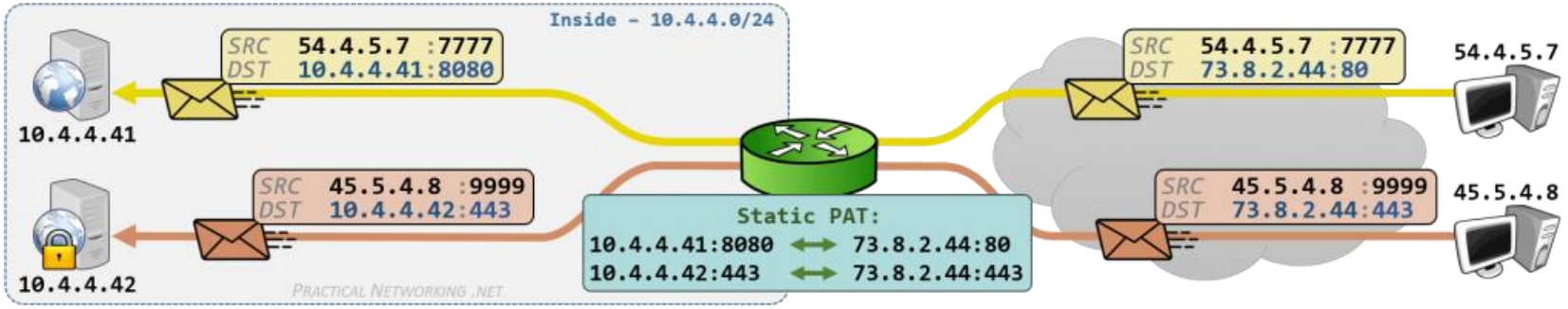
Fonte: Practical Networking .net

Firewall – funções
NAT e PAT estático ou dinâmico

Tanto o NAT quanto o PAT podem ser estáticos ou dinâmicos.

Eles serão estáticos quando a conversão dos endereços e/ou portas internas forem convertidos para um endereço e/ou porta externa fixos (SNAT e SPAT).

Eles serão dinâmicos quando a conversão dos endereços e/ou portas internas forem convertidos para um endereço e/ou porta externa alocados dinamicamente pelo firewall (DNAT e DPAT).



⚠ O NAT/PAT estático (SNAT e SPAT) é comumente usado quando serviços internos à rede precisem ser acessados externamente.

Fonte: Practical Networking .net

Firewall – funções
Faixa de IP´s privados

A IANA (Internet Assigned Numbers Authority) reservou três blocos do espaço de endereço IP para redes privadas:

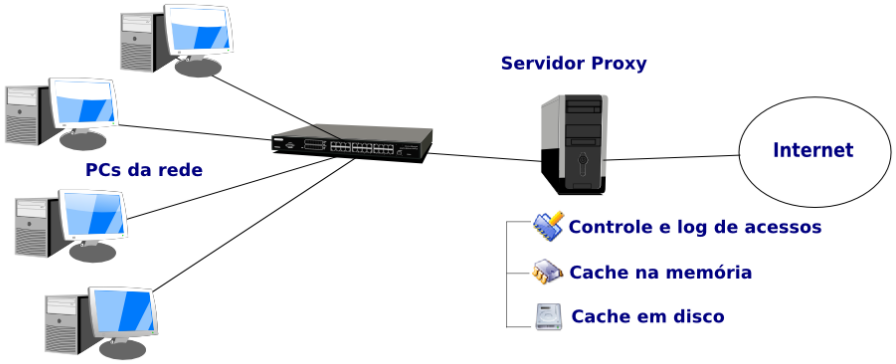
- 10.0.0.0 - 10.255.255.255 (prefixo 10/8);
- 172.16.0.0 - 172.31.255.255 (prefixo 172.16/12);
- 192.168.0.0 - 192.168.255.255 (prefixo 192.168/16).

Fonte: RFC 1918 - Address Allocation for Private Internets

Firewall – funções
Proxy

Os proxies foram desenvolvidos originalmente para colocar em cache as páginas Web que eram acessadas frequentemente.

Nos primeiros dias da Internet, os enlaces de longa distância remotos eram lentos, a Web era relativamente pequena e as páginas Web eram estáticas.

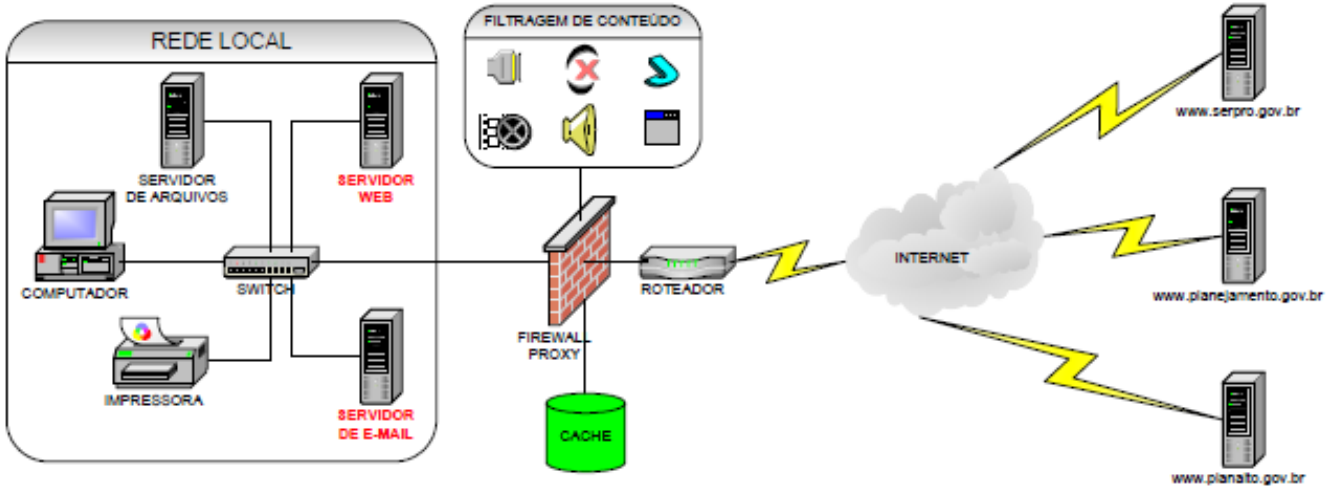


Quando a Web cresceu explosivamente, os proxies começaram a ficar menos eficazes em sua finalidade original – pois a Web agora é muito ampla e muitas páginas Web são dinâmicas, perdendo a validade assim que são transmitidas, fazendo com que os usuários de uma única organização tenham de passar por “um milhão” de páginas Web antes de uma delas ser visitada três vezes.

Fonte: Firewalls, de Matthew Strebe e Charles Perkins

Firewall – funções
Proxy

Esses fatores representam, verdadeiramente, um problema de cache difícil de resolver. Mas, a nova Web também apresenta novos elementos e os servidores proxy mostraram um efeito colateral surpreendente e notável: eles podem ocultar todos os usuários reais de uma rede por trás de uma única máquina, eles podem filtrar URLs e podem impedir a passagem de conteúdo suspeito ou ilegal. A finalidade principal da maioria dos servidores proxy atuais é operar como firewall em vez de cache de páginas Web.



Fonte: Firewalls, de Matthew Strebe e Charles Perkins

Firewall – tipos
Sem inspeção de estados – *stateless*

Os firewalls são roteadores nas fronteiras que aumentam a segurança determinando se um pacote deve ou não ser encaminhado com base na informação contida no cabeçalho de cada pacote individual.

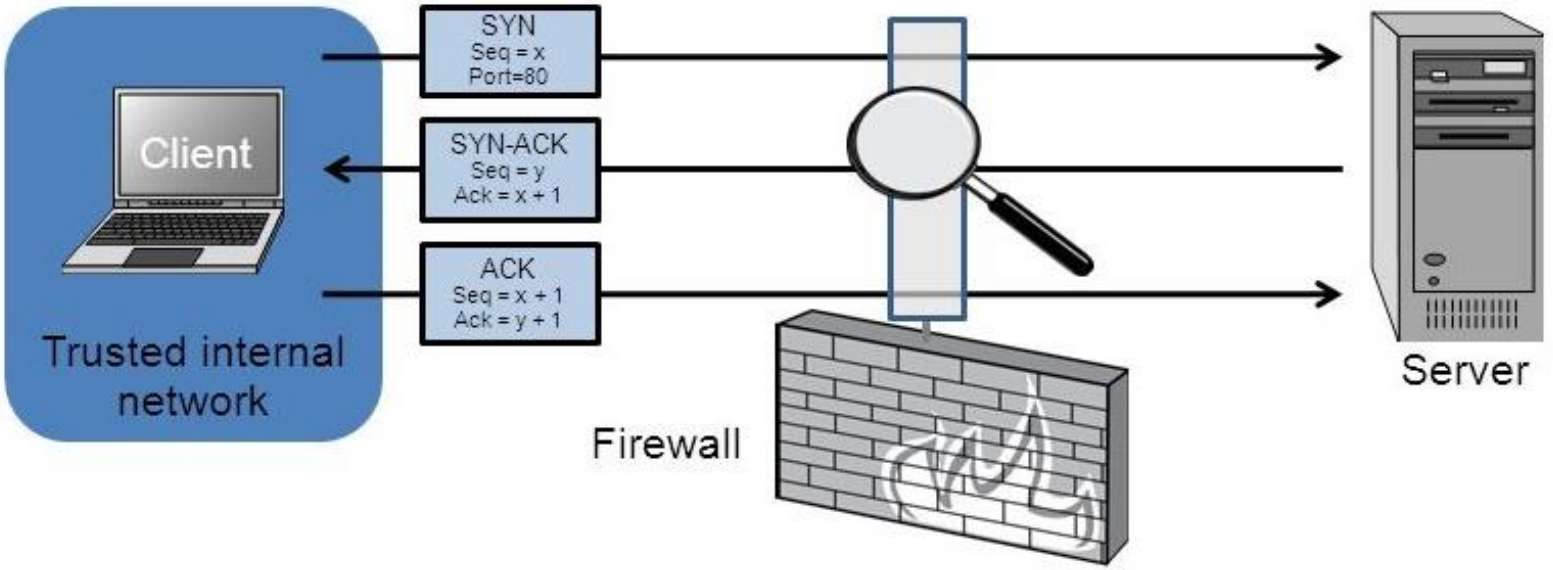
Teoricamente, os filtros podem ser configurados para operar com base em qualquer parte do cabeçalho do protocolo, mas a maioria só pode ser configurada para filtrar os campos de dados mais úteis, como:

- tipos de protocolo;
- endereços IP;
- portas TCP e UDP;
- número de fragmentos;
- informações sobre roteamento de origem.

Este tipo de firewall é conhecido como *stateless*, ou sem inspeção de estados, pois verifica cada pacote individualmente.

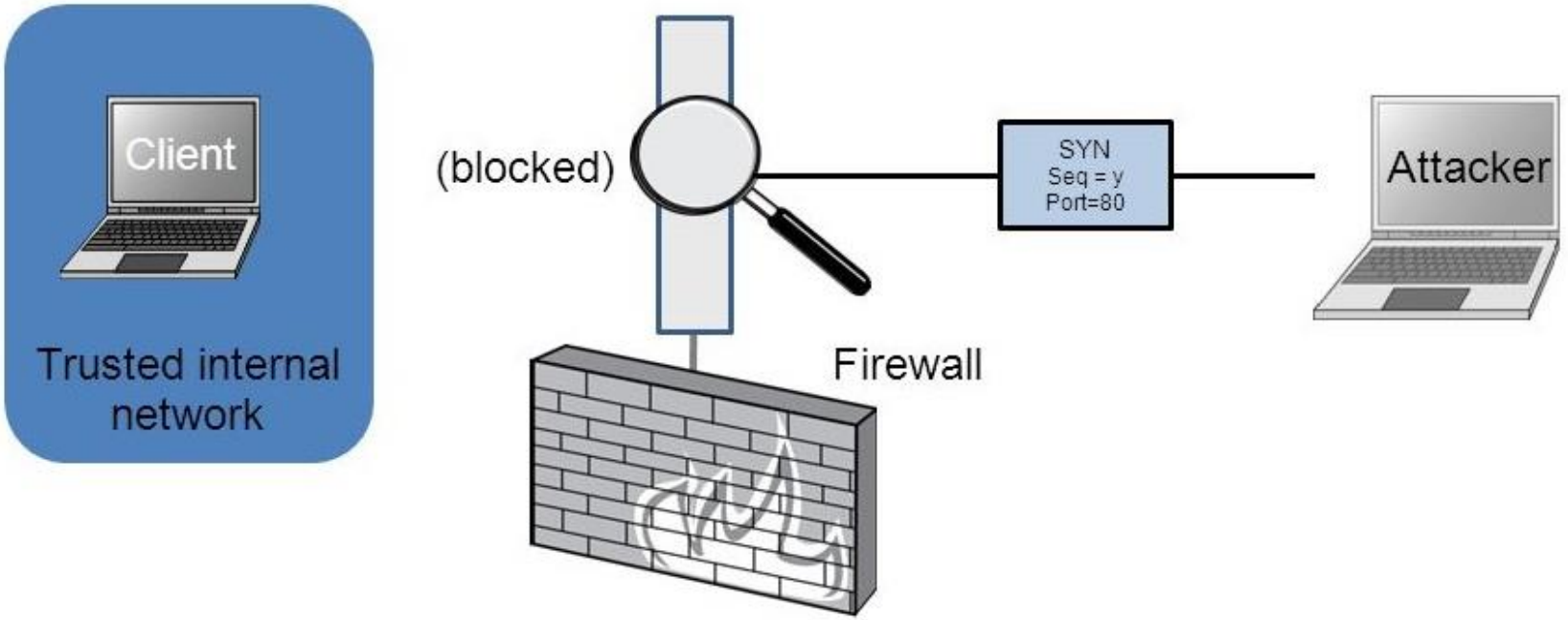
Fonte: Firewalls, de Matthew Strebe e Charles Perkins

Firewall – tipos
Sem inspeção de estados – *stateless* – exemplo



Allow outbound SYN packets, destination port=80
Allow inbound SYN-ACK packets, source port=80

Firewall – tipos
Sem inspeção de estados – *stateless* – exemplo



Allow outbound SYN packets, destination port=80
Drop inbound SYN packets,
Allow inbound SYN-ACK packets, source port=80

8

Firewall – tipos
Com inspeção de estados – *stateful*

Os firewalls sem inspeção de estados têm várias falhas, todas nascendo do fato de que um único pacote em uma comunicação não contém informações suficientes para determinar se ele deve ou não ser recusado, porque ele faz parte de uma comunicação maior.

Os firewalls que realizam a inspeção de estados resolvem esse problema retendo na memória os estados de todas as comunicações passando pelo firewall e usando esse estado guardado para determinar se os pacotes individuais devem ou não ser abandonados.

A inspeção com estados filtra fluxos de comunicação inteiros, não apenas os pacotes, e conseguem se lembrar do estado das conexões da rede e das camadas da sessão, gravando informações sobre o estabelecimento da sessão que passa através do gateway do filtro.

Os filtros usam então essa informação para discriminar pacotes de retorno válidos das tentativas de conexão inválidas ou de invasão.

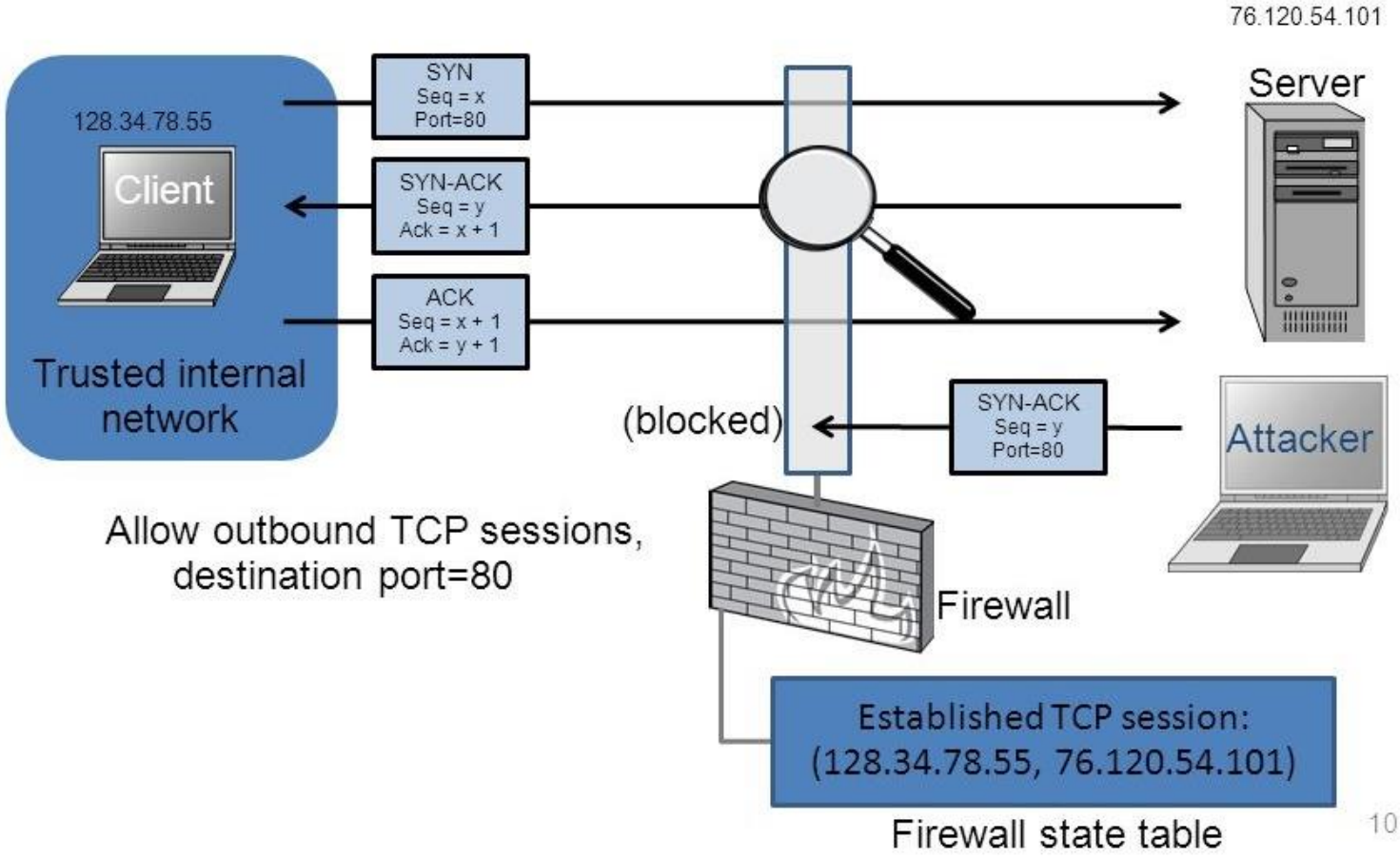
Fonte: Firewalls, de Matthew Strebe e Charles Perkins

Firewall – tipos
Com inspeção de estados – *stateful*

Os firewalls com inspeção de estados ou *stateful* armazena o estado das conexões e as filtra com base nesse estado. Existe três estados para uma conexão:

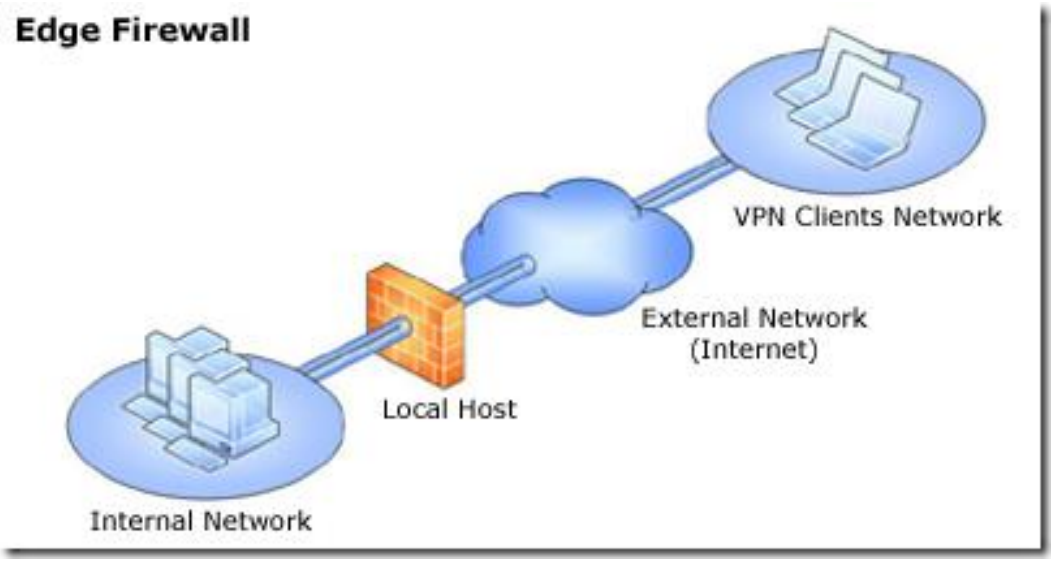
- NEW, para novas conexões;
- ESTABLISHED, para conexões já estabelecidas;
- RELATED, para conexões relacionadas a outras existentes.

Firewall – tipos
Com inspeção de estados – *stateful* – exemplo



Firewall – topologia
Firewall de borda

Nessa topologia, o firewall está localizado na borda da rede, onde atua como o firewall de borda da organização e está conectado a duas redes: a rede interna e a rede externa (geralmente a Internet).

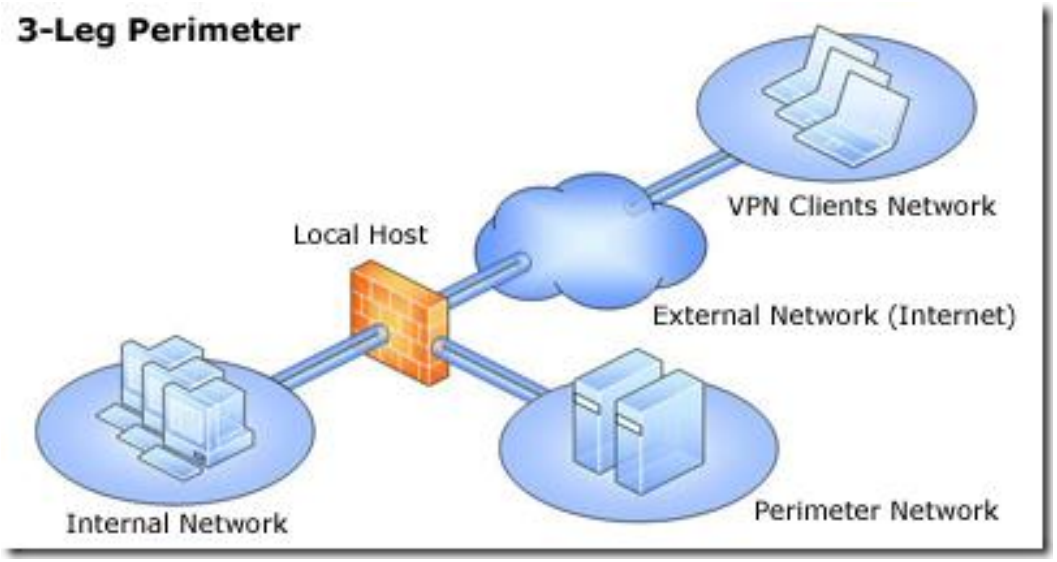


Fonte: Microsoft Forefront Threat Management Gateway

Firewall – topologia
Firewall com DMZ

Esta topologia implementa uma rede de perímetro (DMZ). O firewall está conectado a pelo menos três redes físicas: a rede interna, uma ou mais redes de perímetro e a rede externa.

DMZ significa *demilitarized zone*, ou zona desmilitarizada.

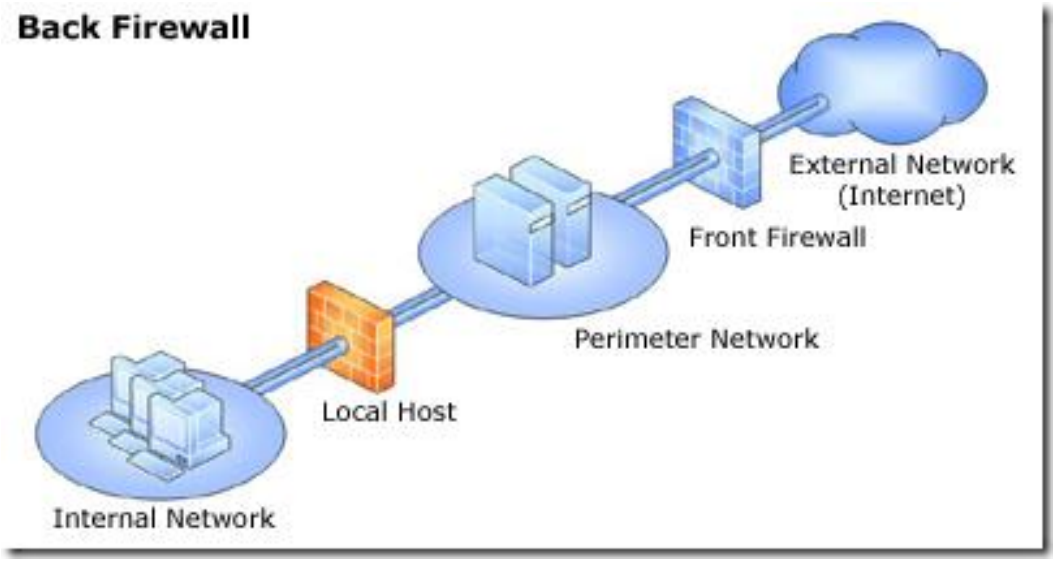


Fonte: Microsoft Forefront Threat Management Gateway

Firewall – topologia
Firewall de *backend*

Nessa topologia, o firewall está localizado no *backend* da rede. Essa topologia deve ser combinada com outro firewall localizado no *frontend*. É usada para aumentar a segurança como um todo, ao impor mais de uma barreira de checagem e isolar a rede de perímetro.

O firewall, neste caso, está conectado à rede interna e ao elemento de rede na frente dele.

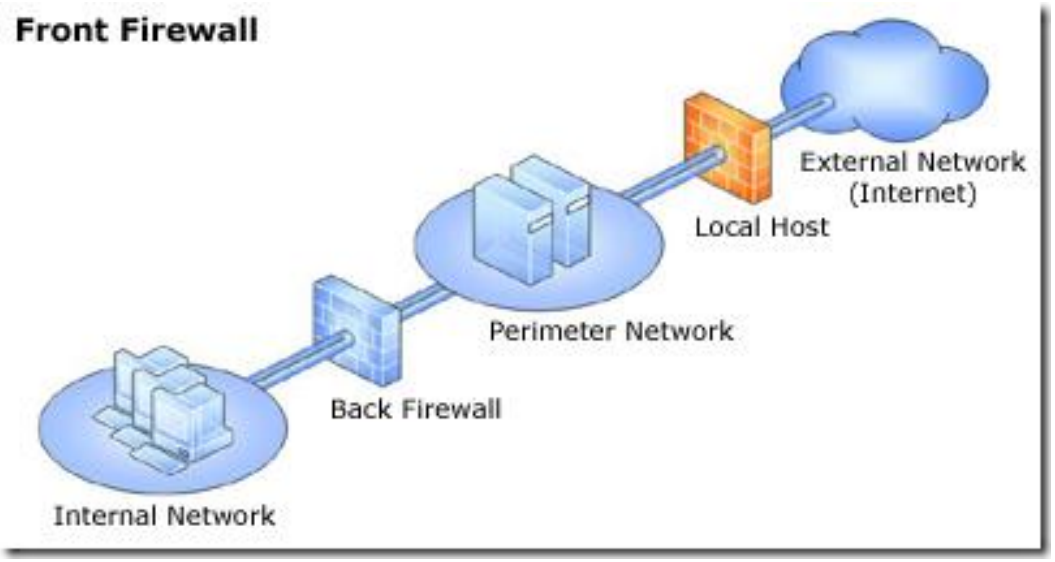


Fonte: Microsoft Forefront Threat Management Gateway

Firewall – topologia
Firewall de *frontend*

Nessa topologia, o firewall está localizado no *frontend* da rede. Essa topologia deve ser combinada com outro firewall localizado no *backend*. É usada para aumentar a segurança como um todo, ao impor mais de uma barreira de checagem e isolar a rede de perímetro.

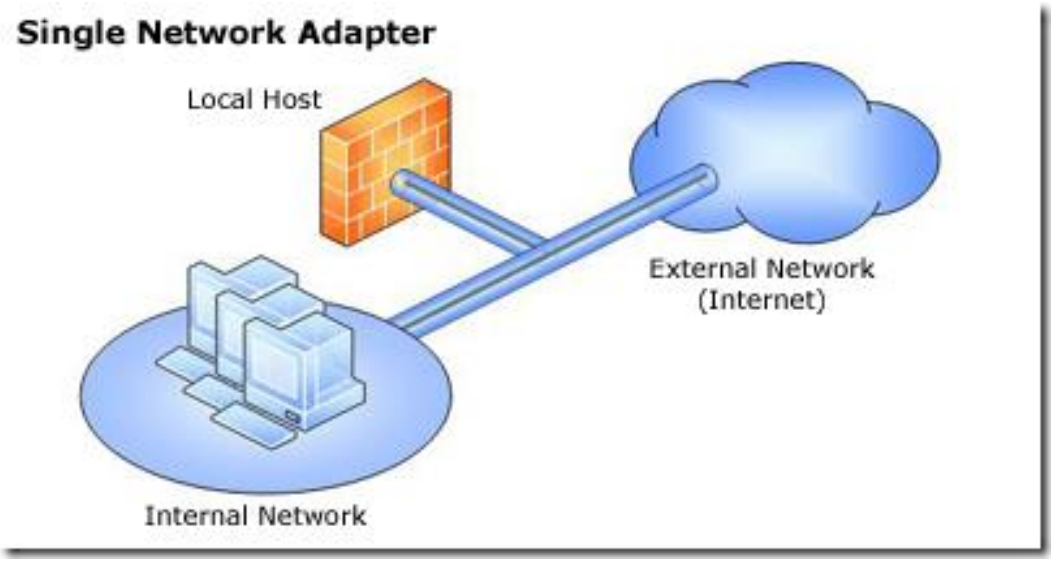
O firewall, neste caso, está conectado à rede externa e ao elemento de rede atrás dele.



Fonte: Microsoft Forefront Threat Management Gateway

Firewall – topologia
Firewall de rede interna

Nessa topologia o firewall está conectado a apenas uma rede, seja a rede interna ou uma rede de perímetro. Normalmente, essa configuração é usada quando o firewall estiver localizado na rede corporativa interna ou em uma rede de perímetro e outro firewall estiver localizado na borda, protegendo os recursos corporativos da Internet.



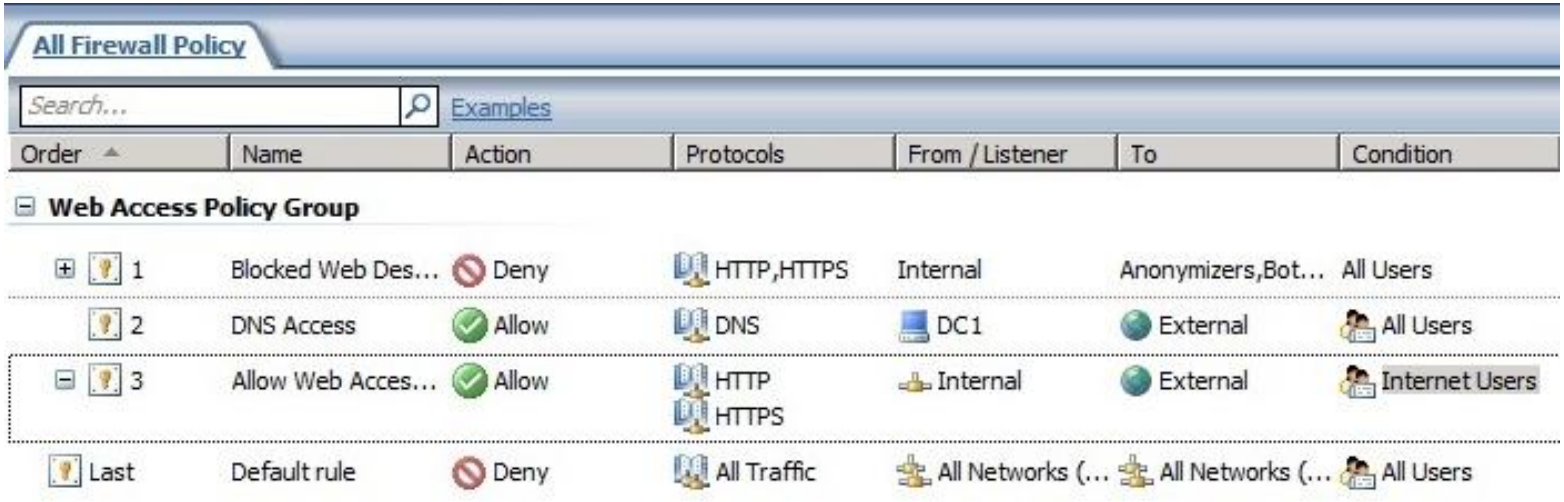
Fonte: Microsoft Forefront Threat Management Gateway

Firewall – regras

As regras em um firewall possuem geralmente uma ação (allow/deny), origem e destino dos pacotes, protocolos atendidos pela regra e os usuários afetados, bem como um nome e uma ordem de precedência.

A ordem de precedência é importante pois quando um pacote encontra uma regra que o atenda, as demais regras serão ignoradas.

Na figura a regra 1 deve vir antes da regra 3. Se fosse ao contrário ela seria inócua.



Fonte: Microsoft Forefront Threat Management Gateway

IPsec

O IPsec (Internet Protocol Security) é um conjunto de protocolos que fornece segurança às comunicações da Internet na camada IP.

O uso atual mais comum do IPsec é fornecer uma Rede Privada Virtual (VPN), seja entre dois locais (gateway a gateway) ou entre um usuário remoto e uma rede corporativa (host a gateway). Ele também pode fornecer segurança de ponta a ponta, ou de host para host.

O IPsec também é usado por outros protocolos da Internet, como por exemplo, o Mobile IP versão 6 (MIPv6), para proteger parte ou todo o tráfego deles.

O IKE (Internet Key Exchange) é o principal protocolo de negociação e de gerenciamento que é mais comumente usado para fornecer os recursos de atualização e negociação dinâmica de chaves secretas usadas no IPsec.

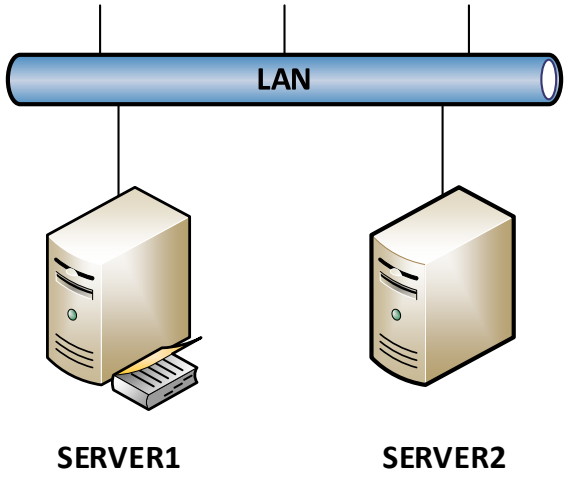
O IPsec e o IKE podem ser usados em conjunto com o IPv4 e o IPv6.

Fonte: RFC 6071 - IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap

IPsec – demonstração
Cenário

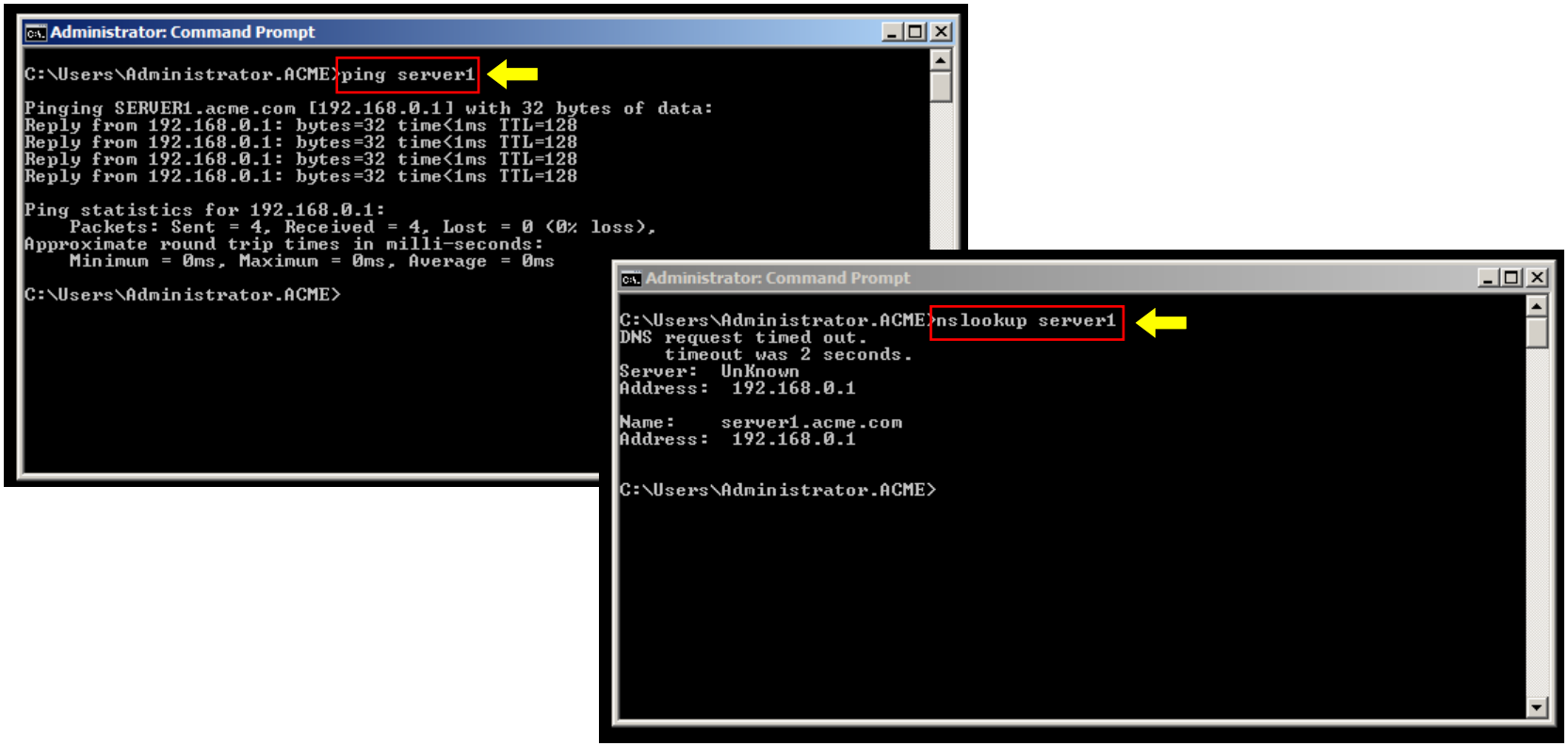
Nesta demonstração serão usados dois servidores Windows 2008 R2 Standard, com a seguinte configuração:

- **SERVER1**: IP 192.168.0.1, com os serviços de DNS e Active Directory;
- **SERVER2**: IP 192.168.0.2, sem nenhum serviço específico.



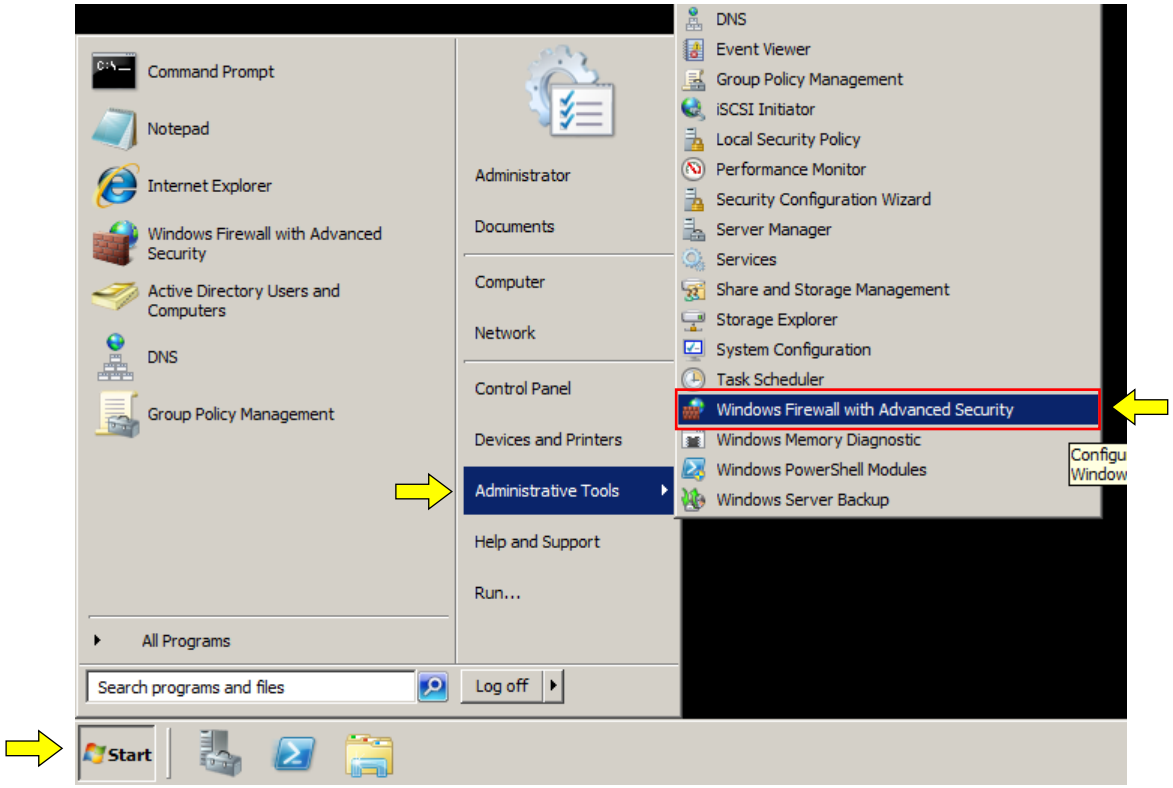
IPsec – demonstração
Verificando a conectividade

A partir do SERVER2 é possível verificar que o SERVER1 aceita comandos ICMP (ping) e consultas DNS (nslookup).



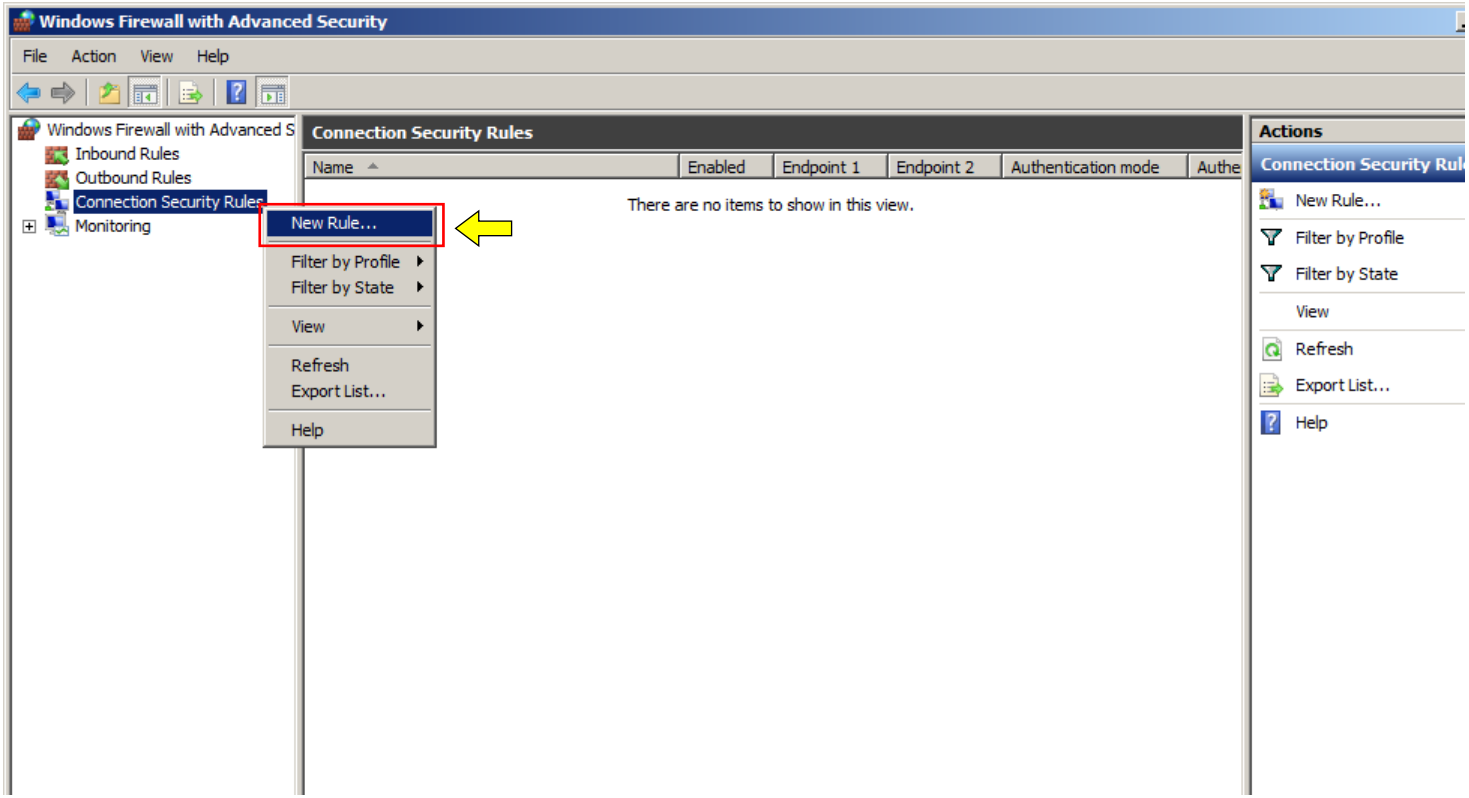
IPsec – demonstração
Configurando o IPsec no SERVER1

No Windows o IPsec é configurado a partir do Windows Firewall with Advanced Security.



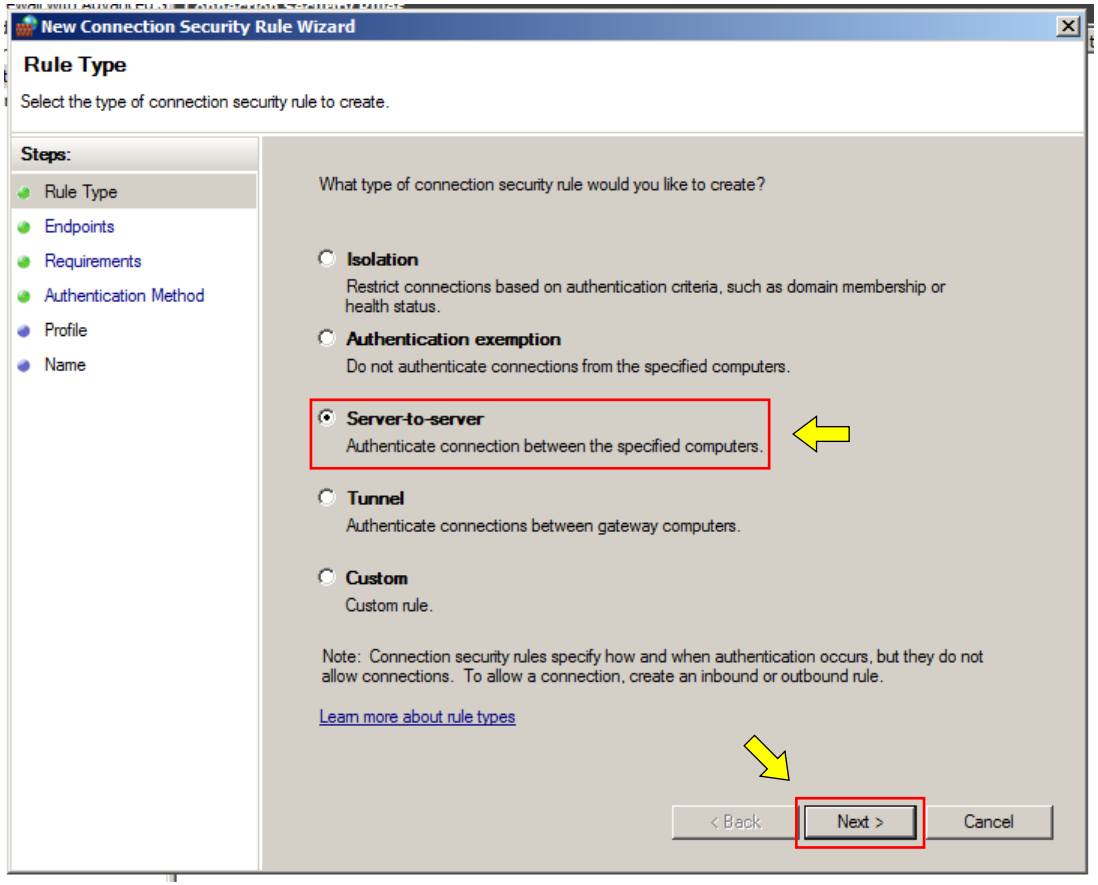
IPsec – demonstração
Configurando o IPsec no SERVER1

Uma vez na console do Windows Firewall, em Connection Security Rules criar uma nova regra (New Rule...).



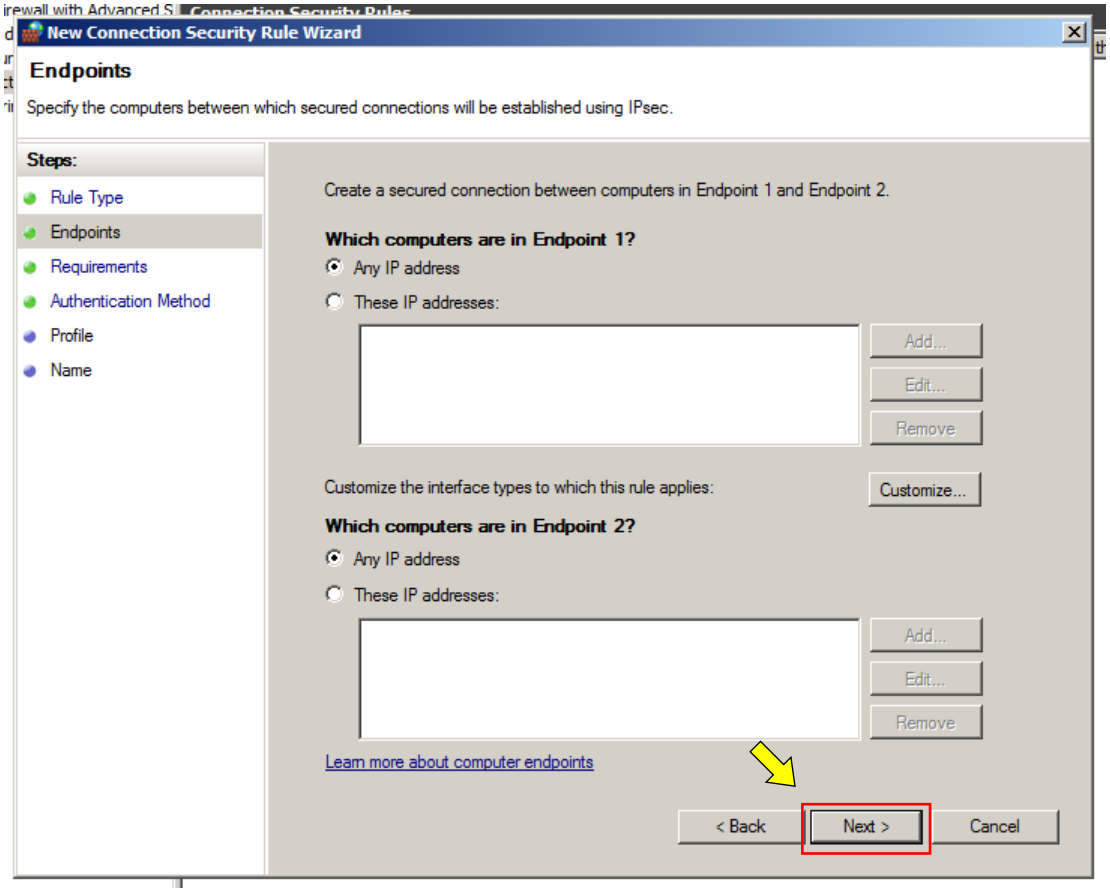
IPsec – demonstração
Configurando o IPsec no SERVER1

No assistente de criação da nova regra, em Rule Type selecionar Server-to-server.



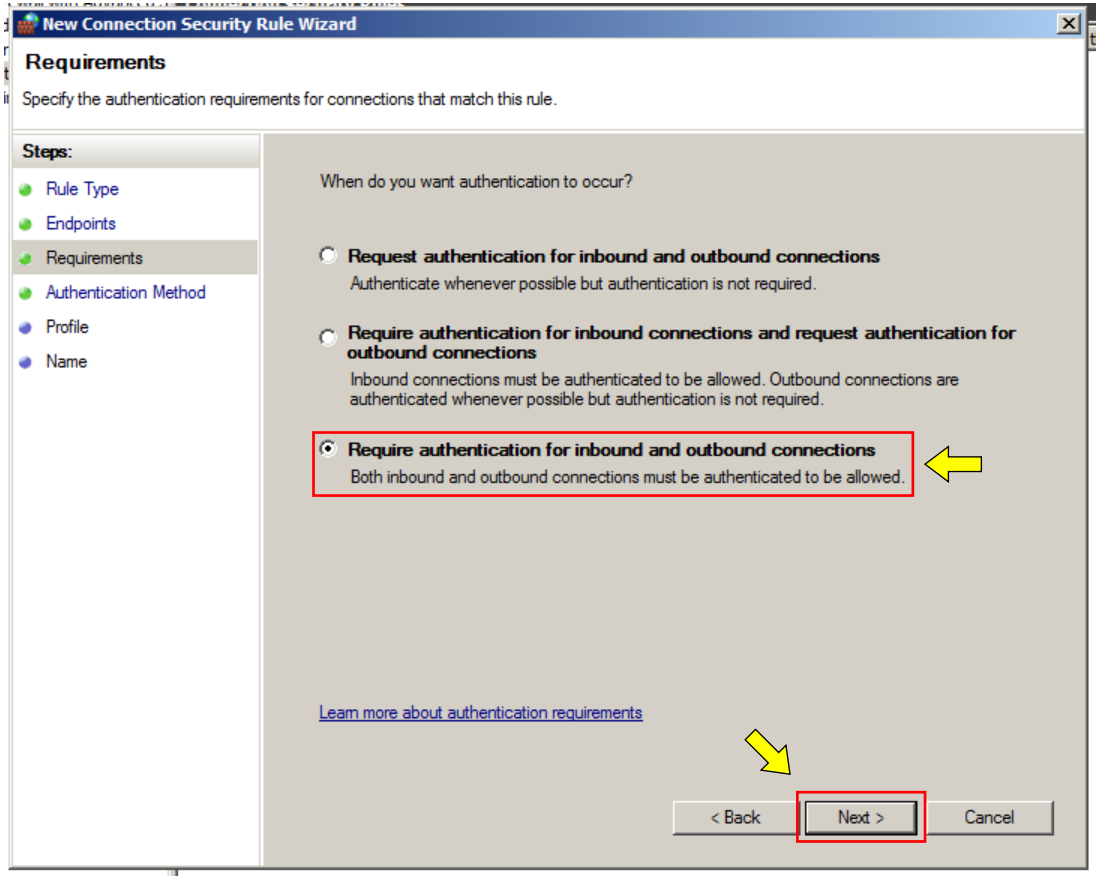
IPsec – demonstração
Configurando o IPsec no SERVER1

Em Endpoints, manter o padrão e clicar em Next.



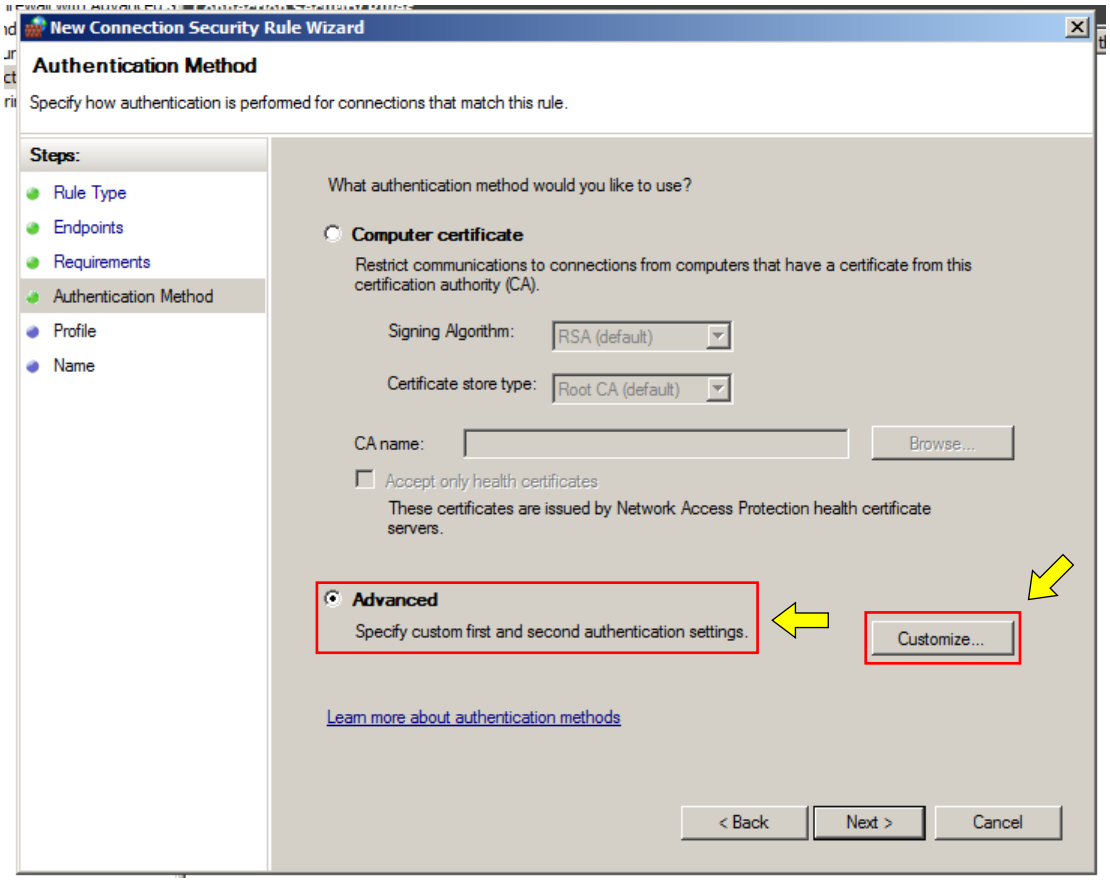
IPsec – demonstração
Configurando o IPsec no SERVER1

Em Requirements, selecionar a terceira opção.



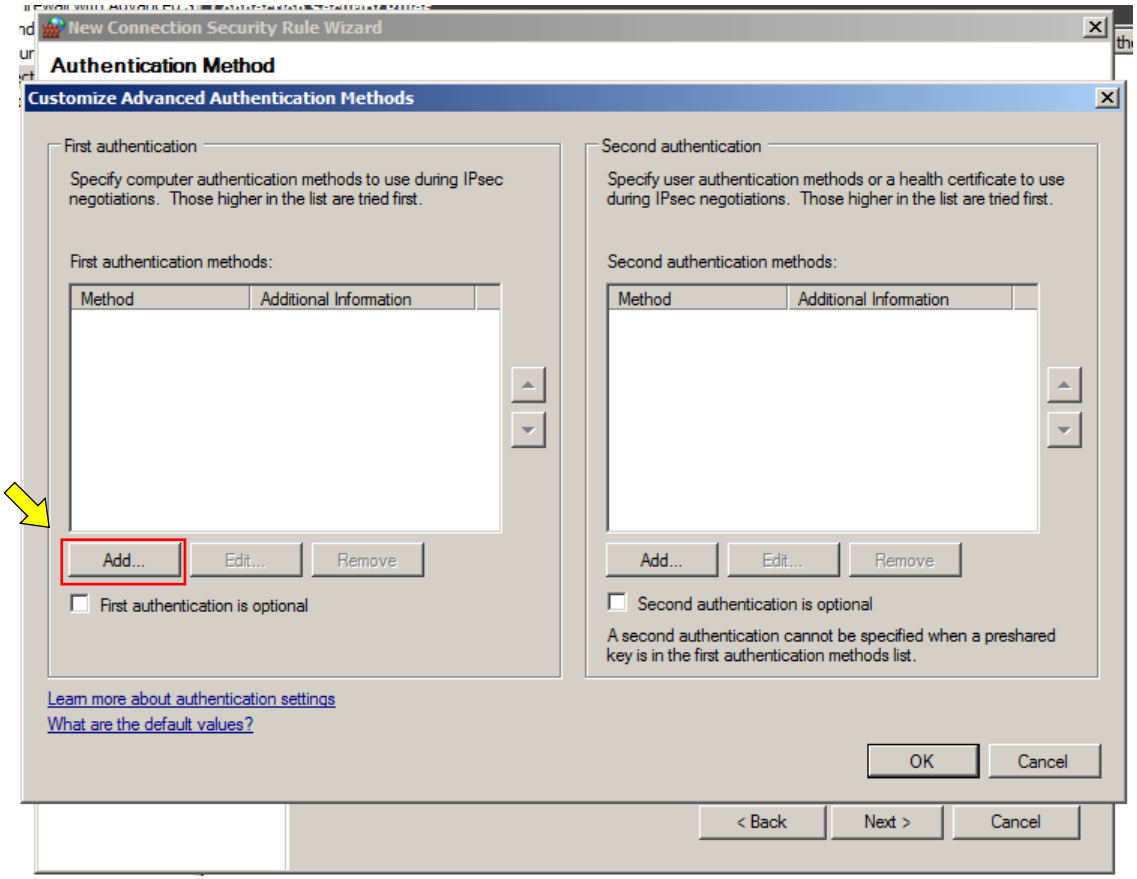
IPsec – demonstração
Configurando o IPsec no SERVER1

Em Authentication Method, selecionar Advanced e clicar em Customize...



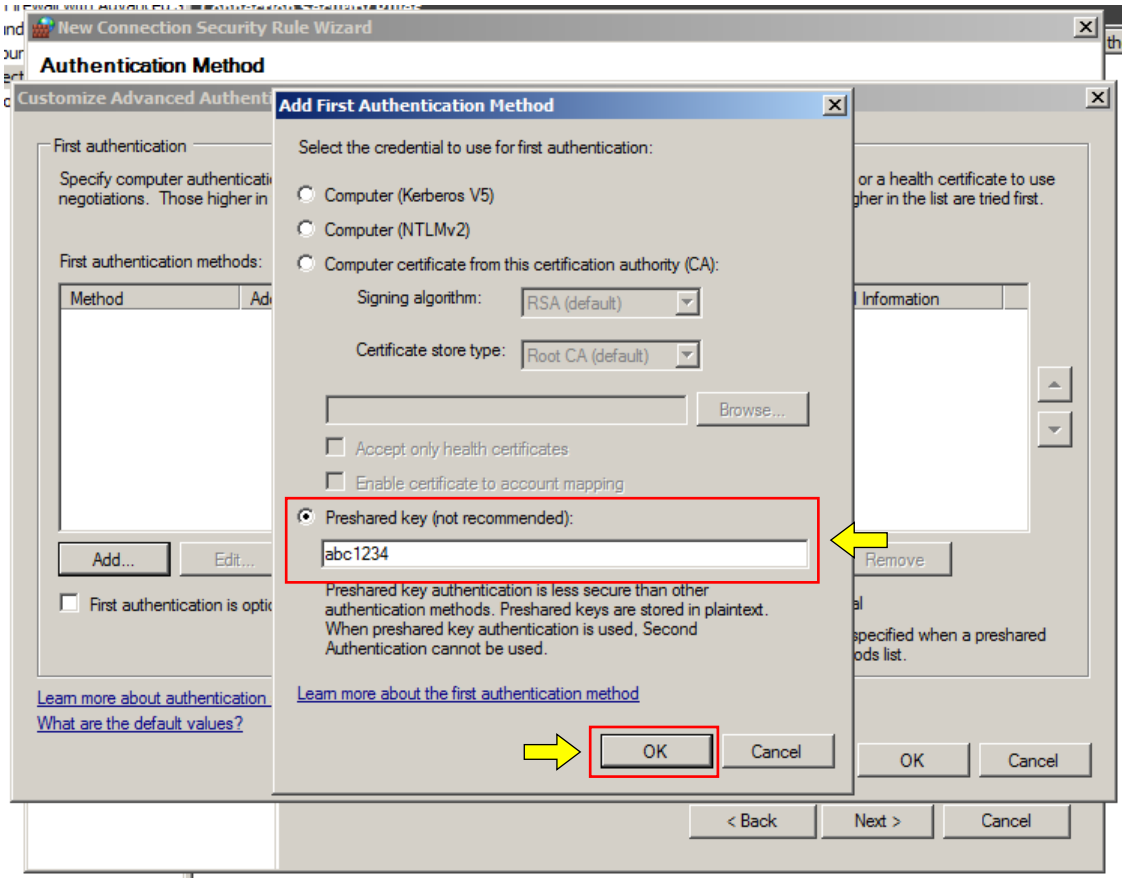
IPsec – demonstração
Configurando o IPsec no SERVER1

Em First authentication, clicar em Add...



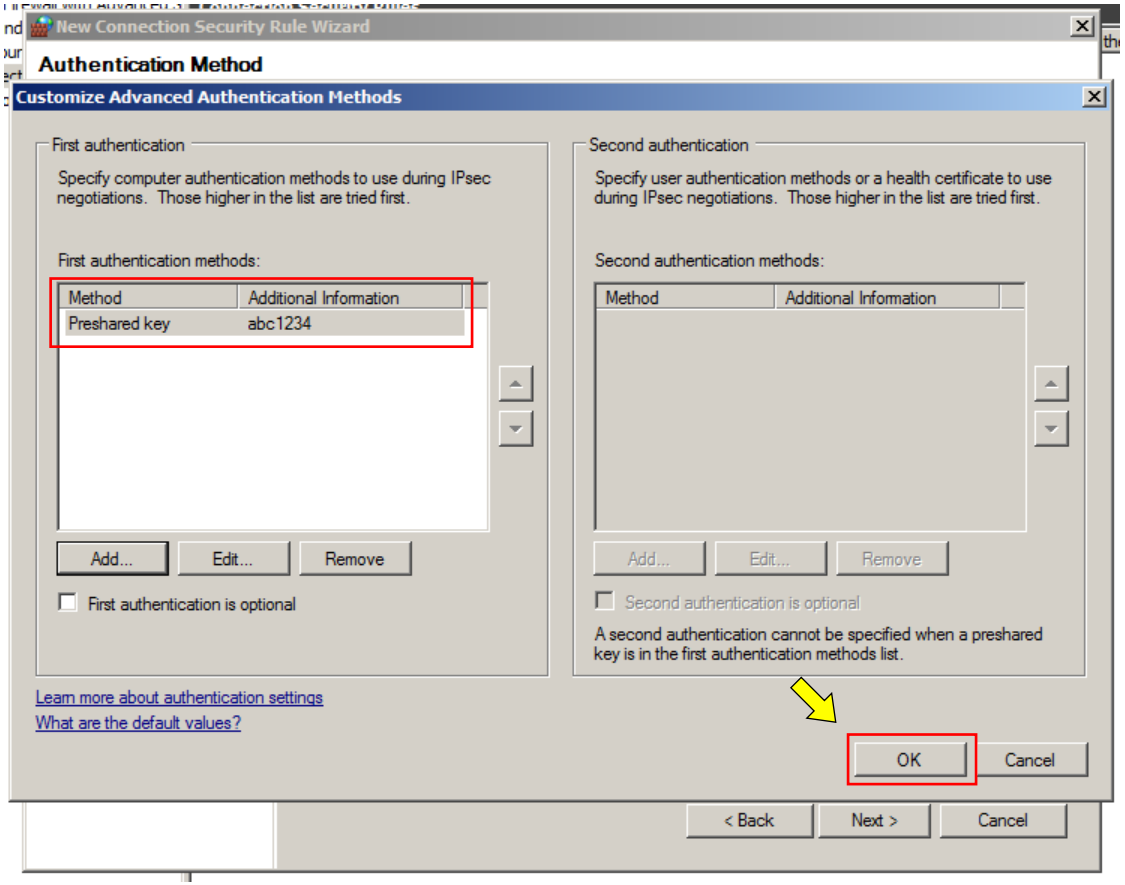
IPsec – demonstração
Configurando o IPsec no SERVER1

Em Preshared key, digitar uma senha a ser compartilhada (abc1234).



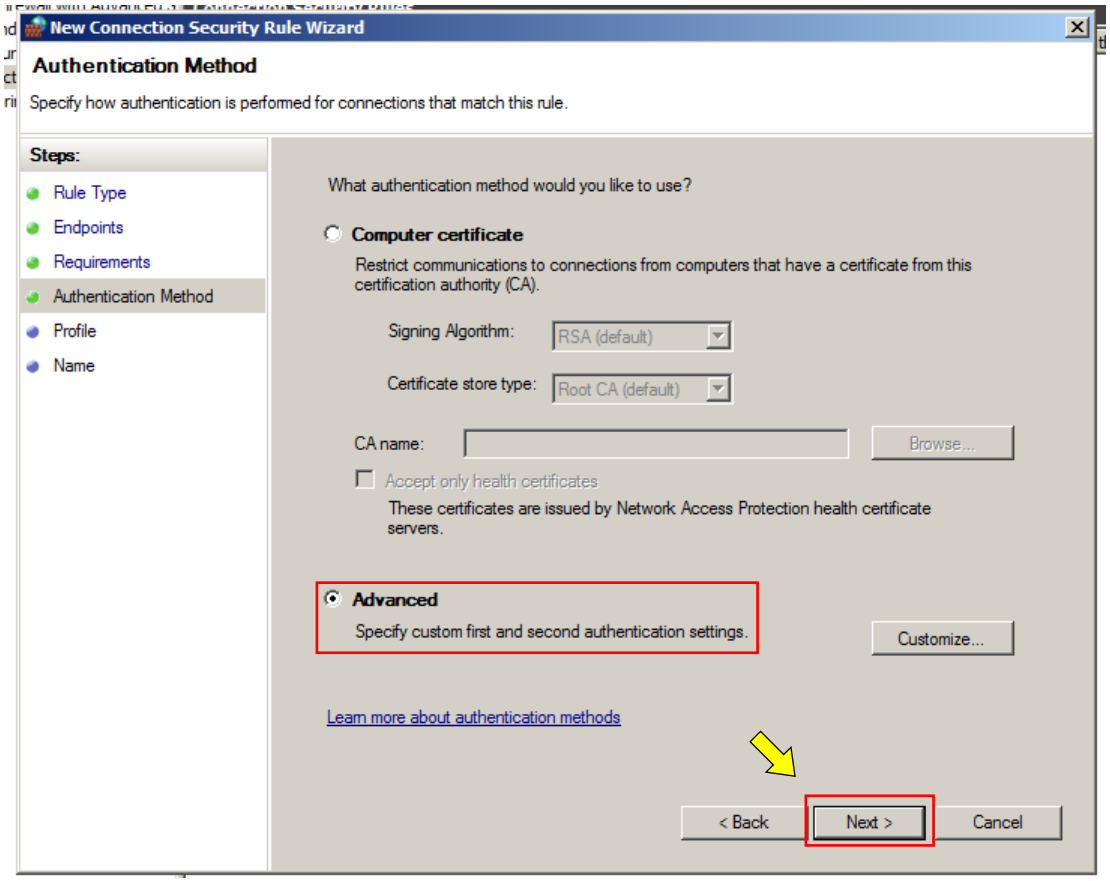
IPsec – demonstração
Configurando o IPsec no SERVER1

Clicar em OK para fechar a janela Customize Advanced Authentication Methods.



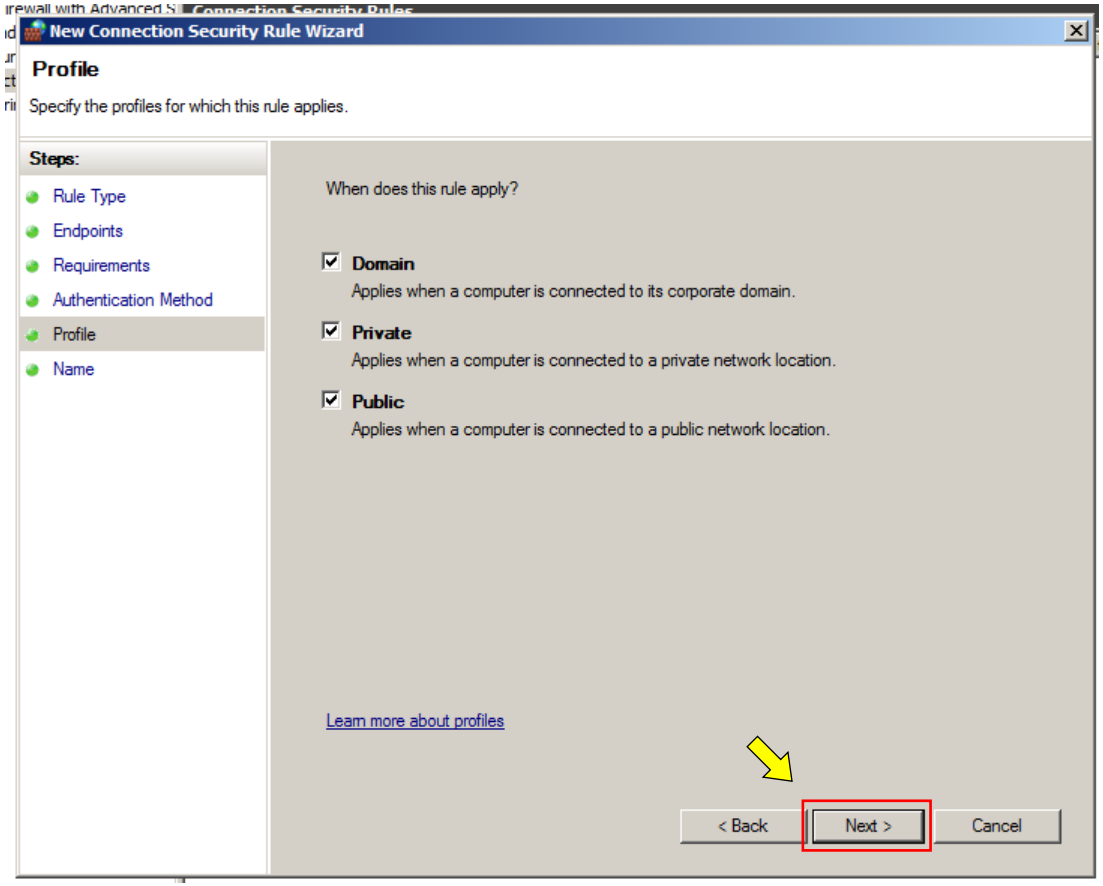
IPsec – demonstração
Configurando o IPsec no SERVER1

Clicar em Next para avançar.



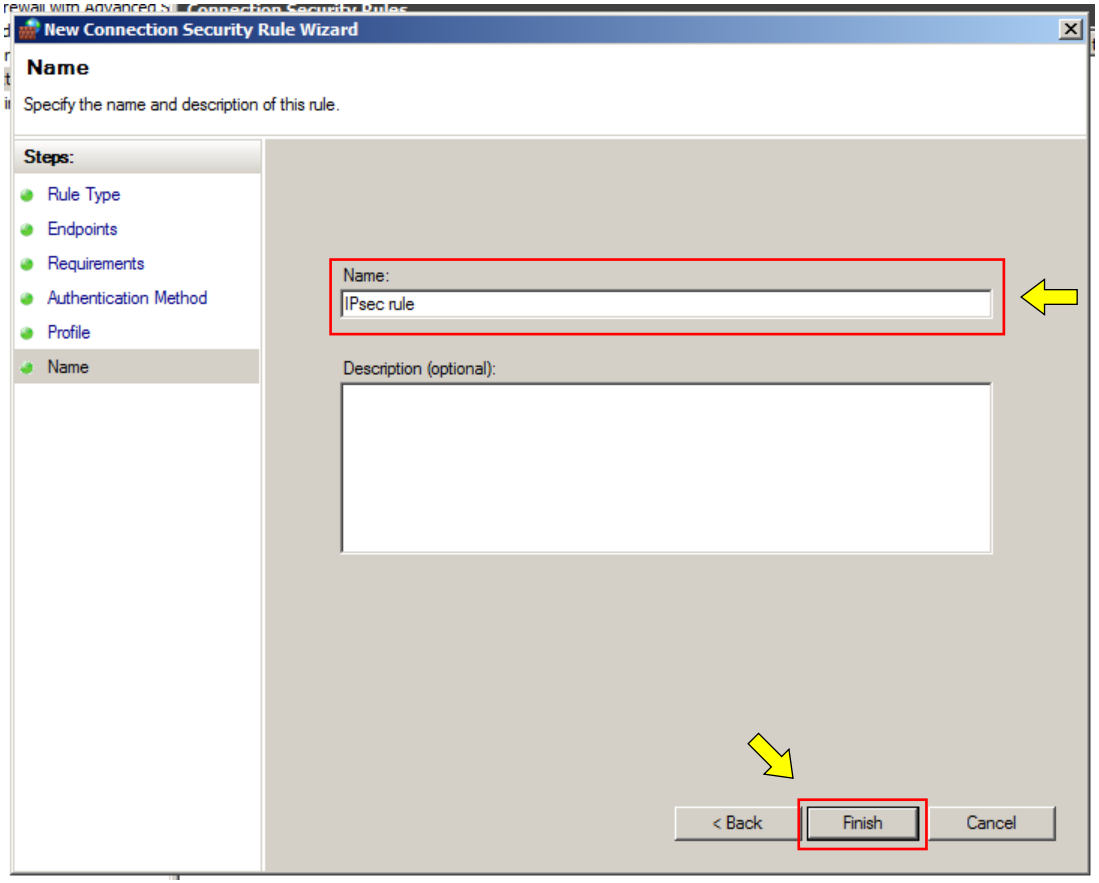
IPsec – demonstração
Configurando o IPsec no SERVER1

Em Profile, manter o padrão e clicar em Next.



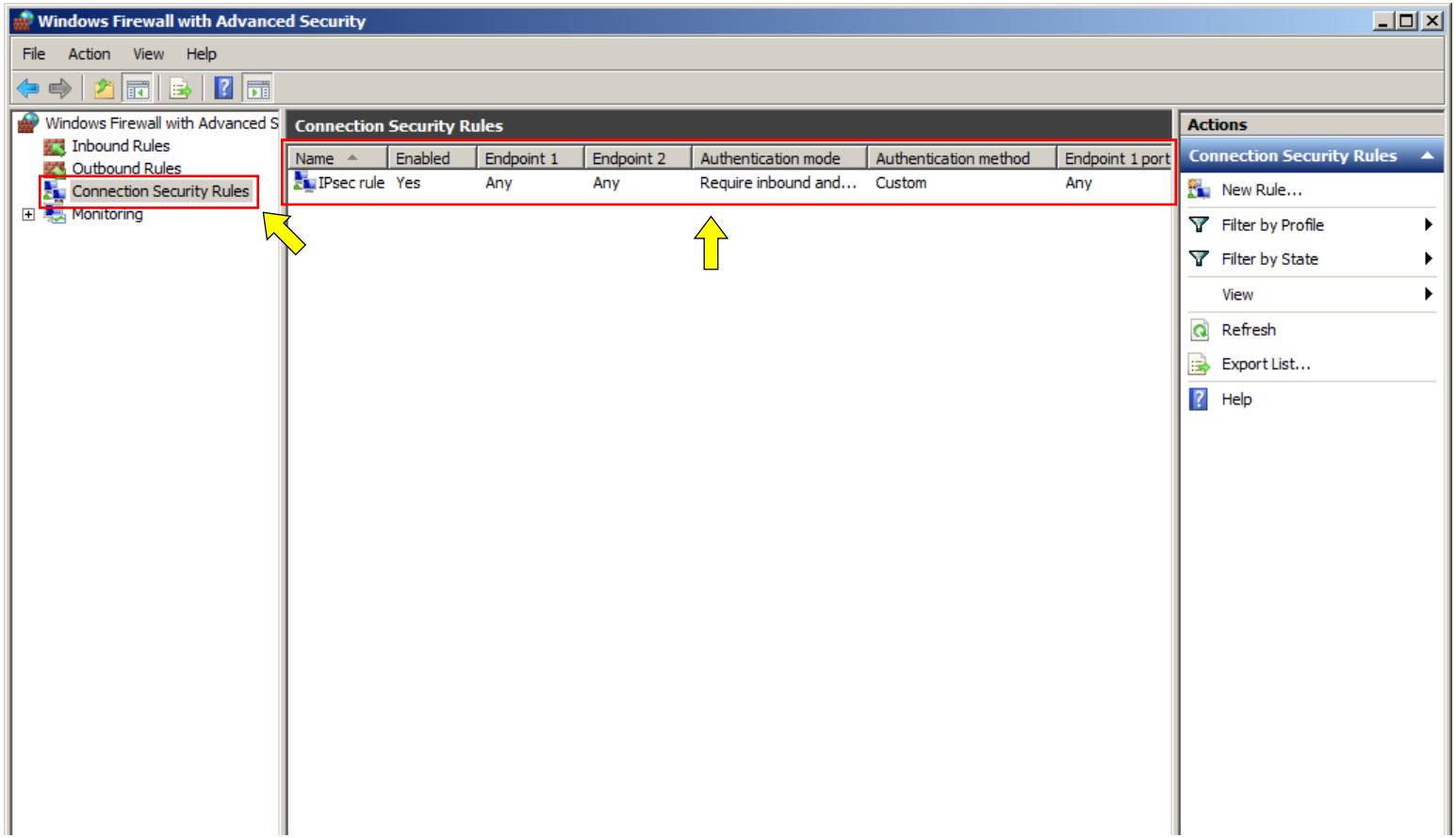
IPsec – demonstração
Configurando o IPsec no SERVER1

No campo Name, digitar o nome IPsec rule.



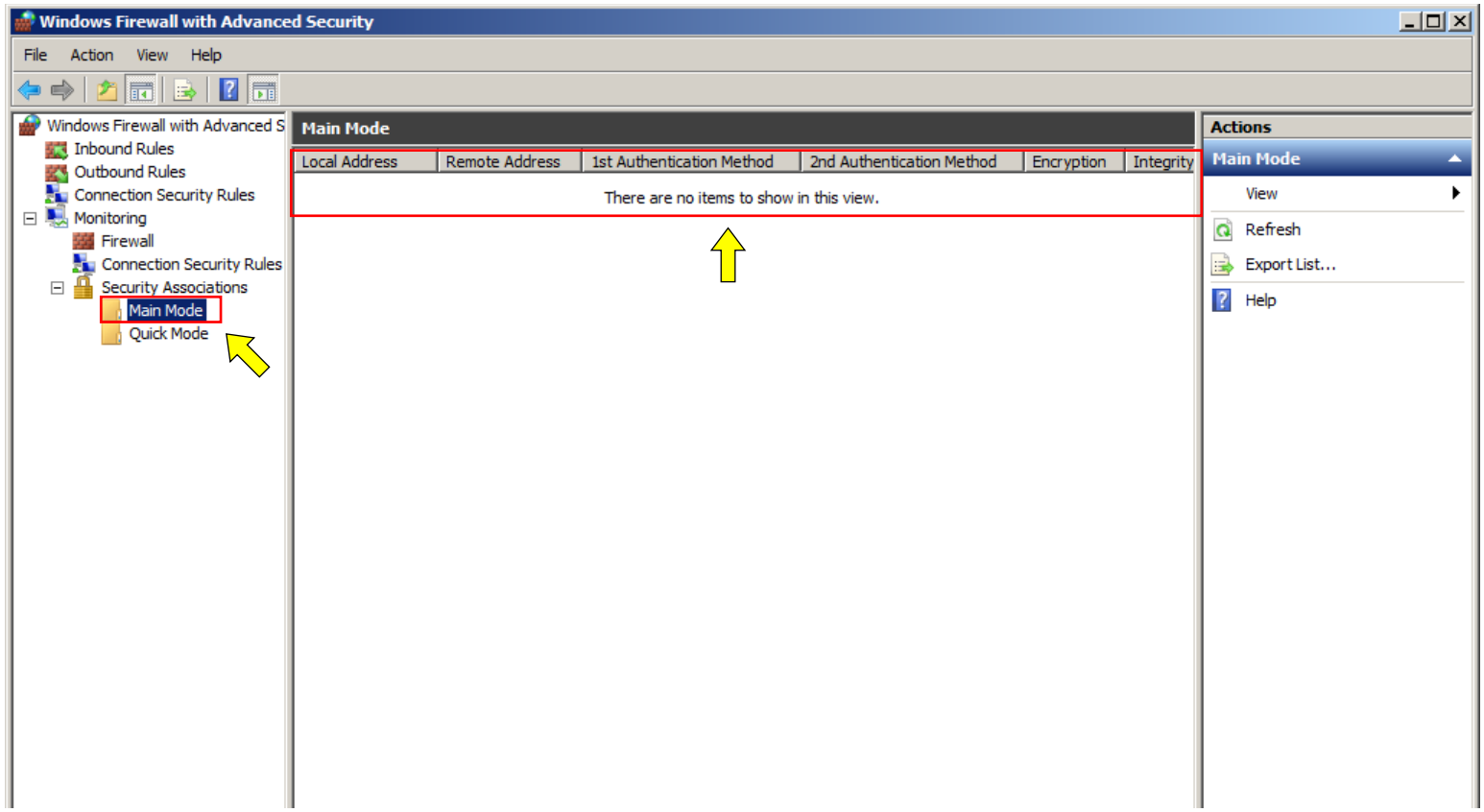
IPsec – demonstração
Configurando o IPsec no SERVER1

Em Connection Security Rules é possível verificar que a regra foi criada.



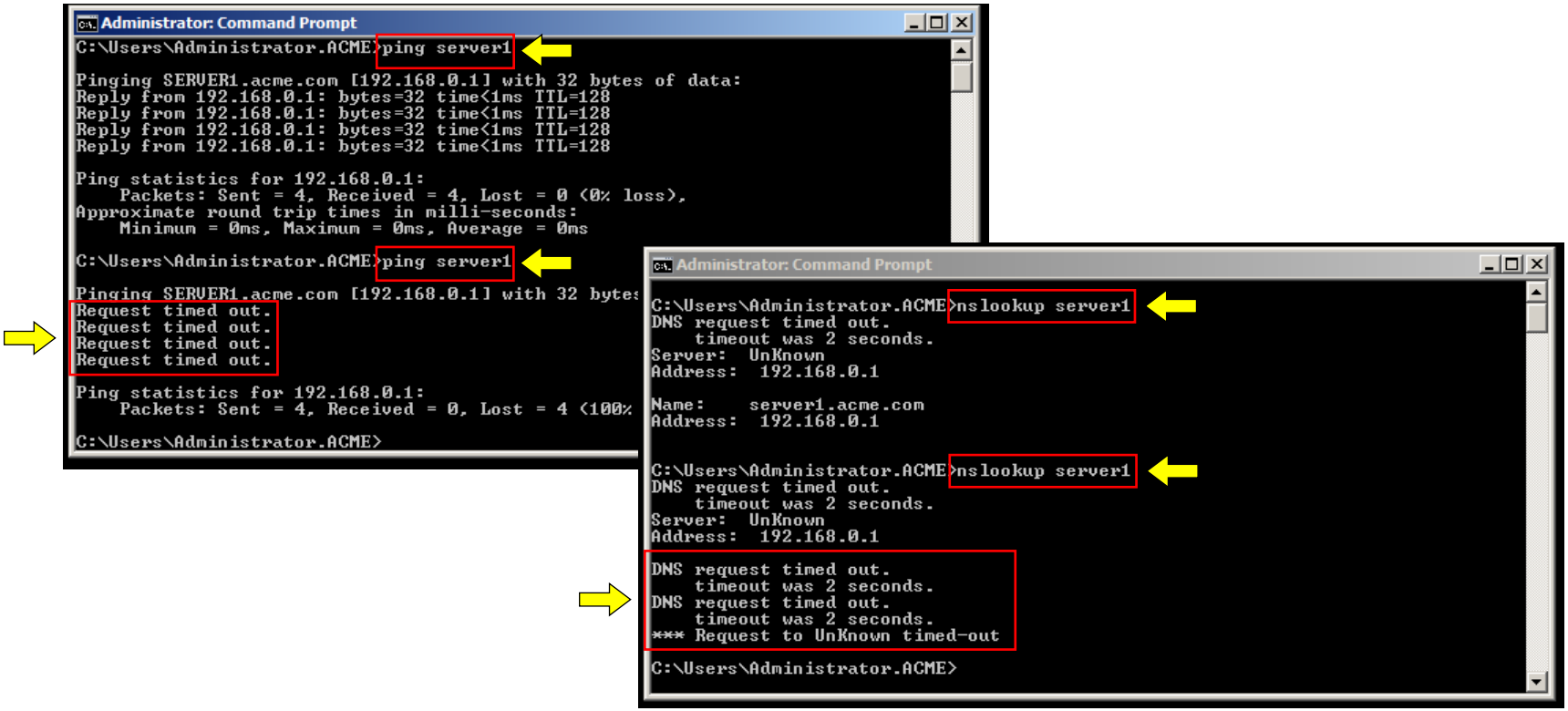
IPsec – demonstração
Configurando o IPsec no SERVER1

Em Monitoring, Security Associations, Main Mode, é possível ver que a regra ainda não está sendo usada.



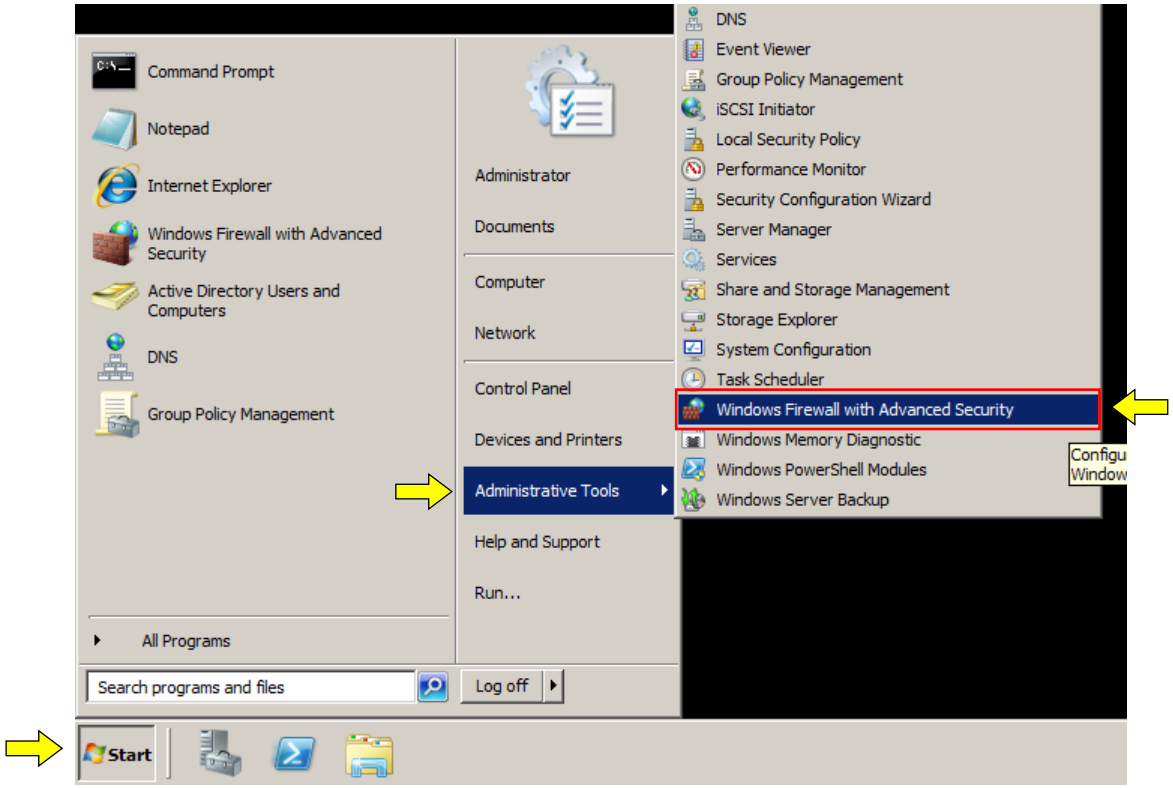
IPsec – demonstração
Verificando a conectividade novamente

A partir do SERVER2 agora é possível verificar que o SERVER1 não aceita mais comandos ICMP (ping) e consultas DNS (nslookup).



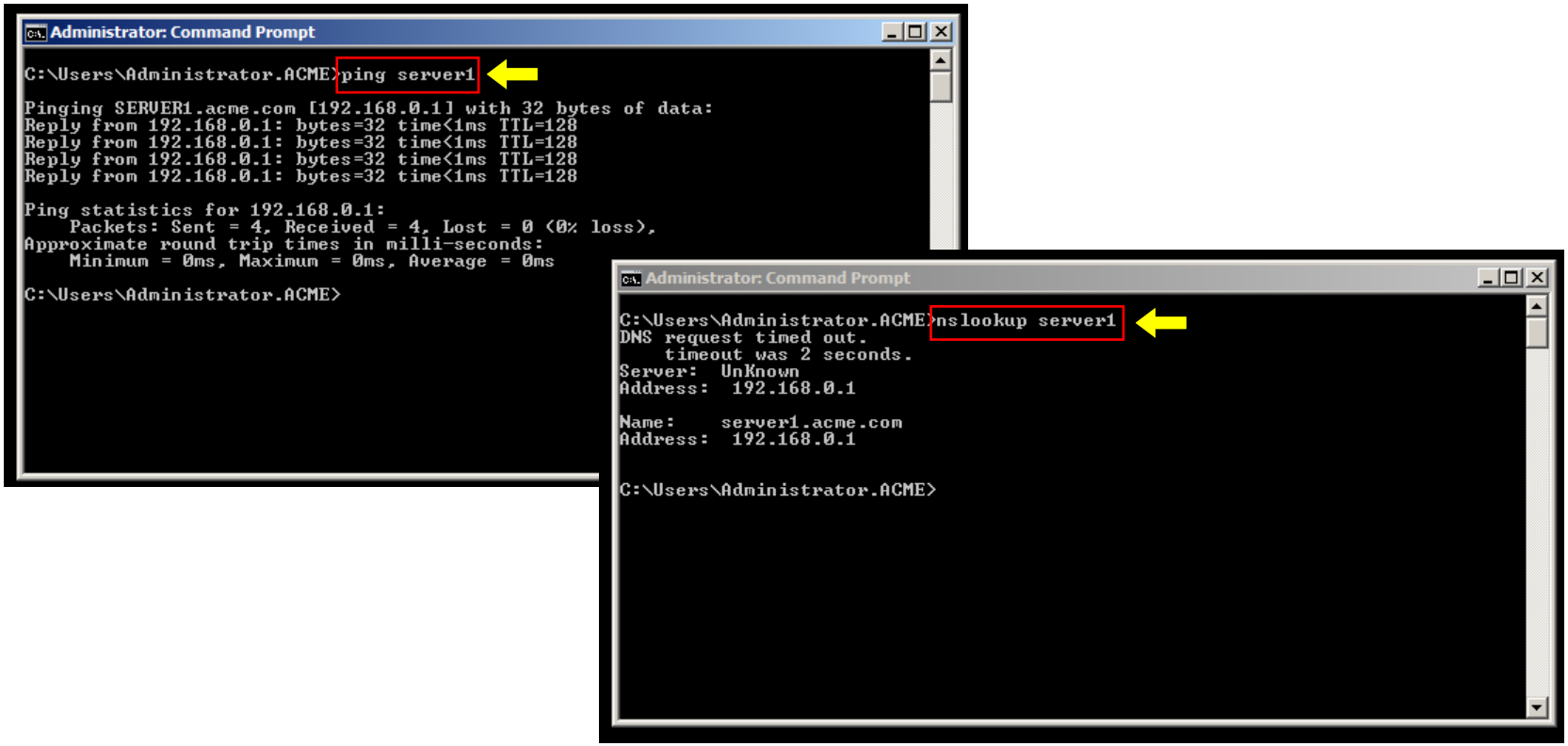
IPsec – demonstração
Configurando o IPsec no SERVER2

Na máquina SERVER2, repetir os mesmos procedimentos para configurar o IPsec a partir do Windows Firewall with Advanced Security.



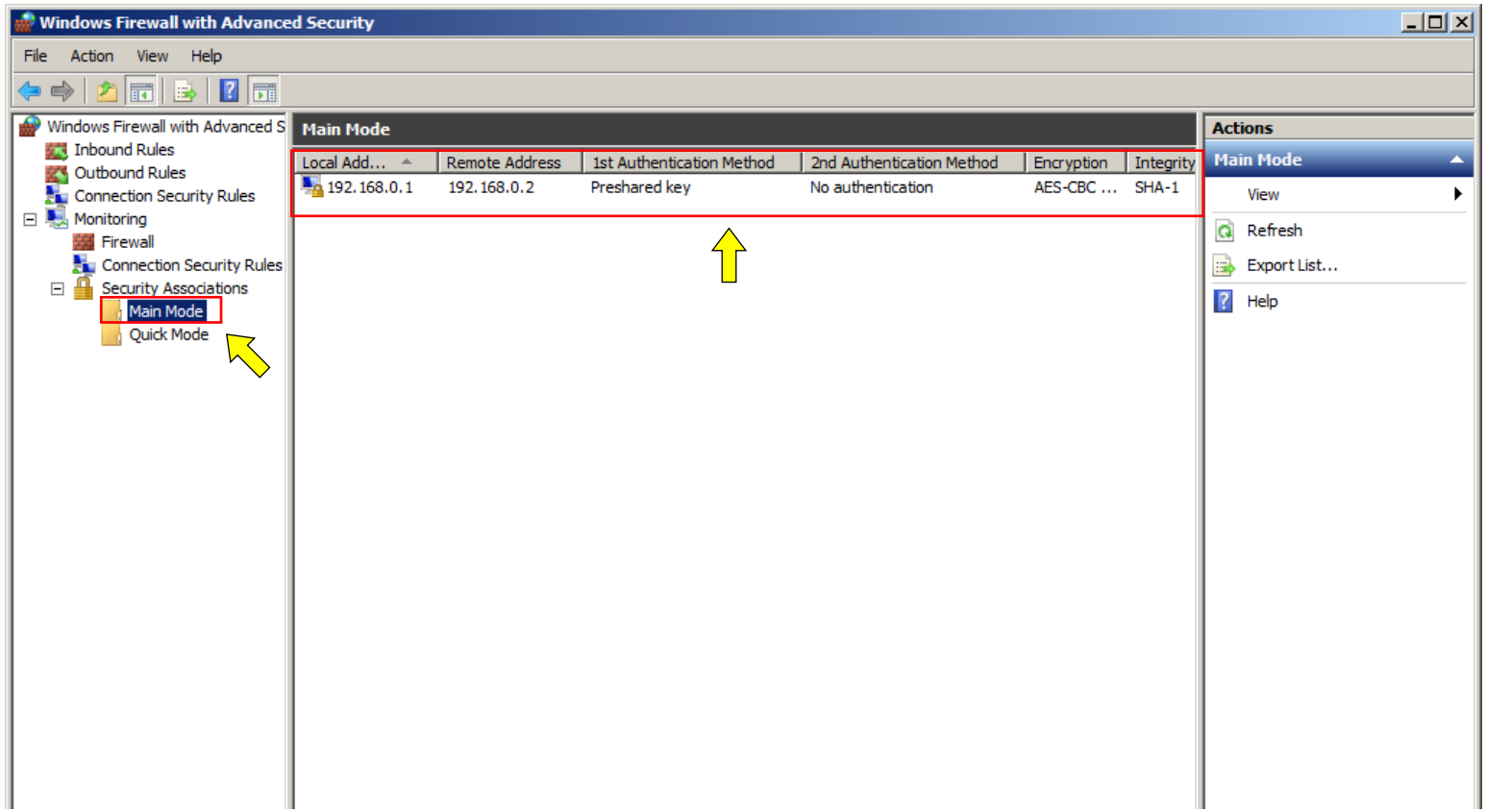
IPsec – demonstração
Verificando a conectividade pela terceira vez

A partir do SERVER2 agora é possível verificar que o SERVER1 aceita novamente comandos ICMP (ping) e consultas DNS (nslookup).



IPsec – demonstração
Configurando o IPsec no SERVER1

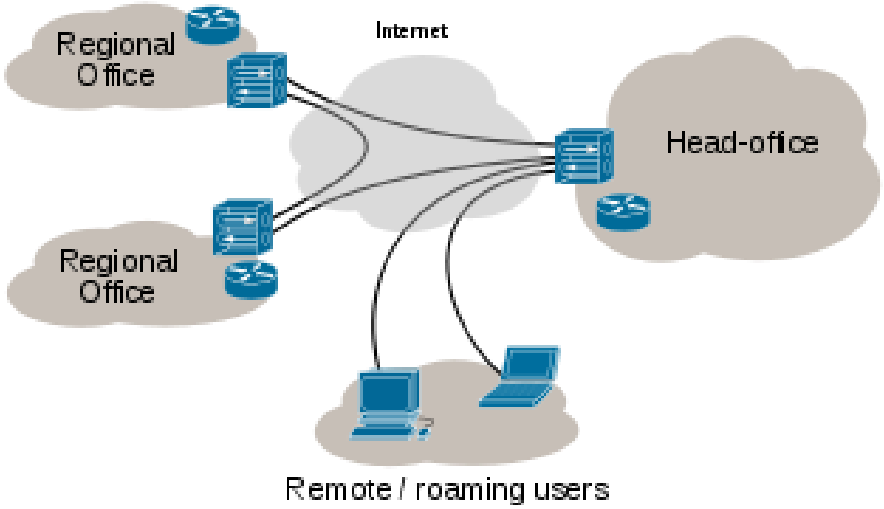
Em Monitoring, Security Associations, Main Mode, agora é possível ver que a regra está sendo usada.



VPN

A VPN (Virtual Private Network), ou Rede Privada Virtual, é uma rede de comunicações privada construída sobre uma rede de comunicações pública (como por exemplo, a Internet).

O tráfego de dados é levado pela rede pública utilizando protocolos padrões, não necessariamente seguros. Em resumo, cria uma conexão segura e criptografada, que pode ser considerada como um túnel, entre o seu computador e um servidor operado pelo serviço VPN.



Fonte: wikipedia.org

VPN – protocolos

Os protocolos VPN definem como o serviço lida com a transmissão de dados através de um túnel VPN.

Os protocolos mais comuns são PPTP, L2TP, SSTP, IKEV2 e OpenVPN.

- PPTP (protocolo de encapsulamento ponto a ponto), é um dos protocolos mais antigos em uso, originalmente projetado pela Microsoft. Prós: funciona em computadores antigos, faz parte do sistema operacional Windows e é fácil de configurar. Contras: pelos padrões de hoje, é pouco seguro;
- L2TP/IPsec (protocolo de encapsulamento de camada 2), é uma combinação do PPTP e do protocolo L2F da Cisco. O conceito deste protocolo é bom: ele usa chaves para estabelecer uma conexão segura em cada extremidade do seu túnel de dados, mas a execução não é muito segura. A adição do protocolo IPsec melhora um pouco a segurança, mas há relatos da suposta capacidade da NSA de quebrar esse protocolo e ver o que está sendo transmitido. Não importa se isso é verdade, o fato de haver um debate é talvez o suficiente para evitar isso também;

Fonte: thebestvpn.com

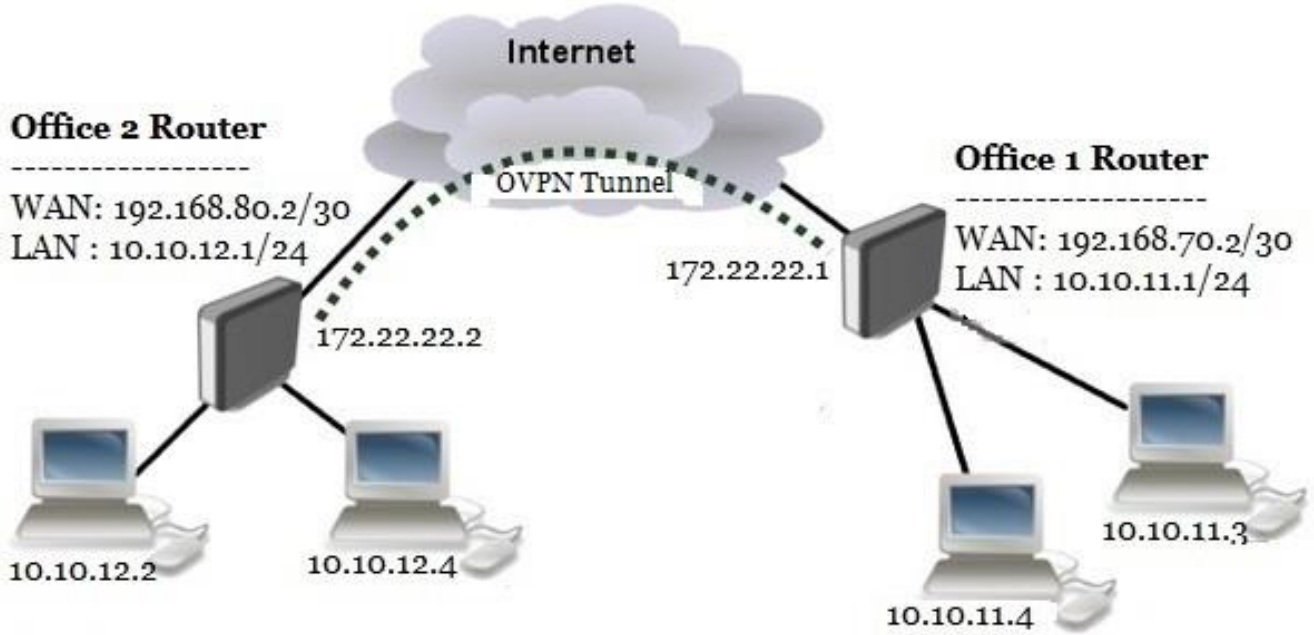
VPN – protocolos

- SSTP (Secure Socket Tunneling Protocol) é outro protocolo criado pela Microsoft. A conexão é estabelecida com alguma criptografia SSL/TLS (o padrão de fato para criptografia da web atualmente). A força do SSL e do TLS é baseada na criptografia de chave simétrica, uma configuração na qual apenas as duas partes envolvidas na transferência podem decodificar os dados. No geral, o SSTP é uma solução muito segura;
- IKEv2 (troca de chaves da Internet, versão 2) é mais um protocolo da Microsoft. É uma iteração dos protocolos anteriores da Microsoft e muito mais segura. Ele fornece algumas das melhores seguranças;
- OpenVPN, é o que há de melhor nos protocolos acima e elimina a maioria das falhas. É baseado em SSL/TLS e é um projeto de código aberto, o que significa que está sendo constantemente aprimorado por centenas de desenvolvedores. Ele protege a conexão usando chaves conhecidas apenas pelas duas partes participantes no final da transmissão. No geral, é o protocolo mais versátil e seguro que existe.

Fonte: thebestvpn.com

VPN – protocolos

De um modo geral, a maioria das VPNs permite que você selecione o protocolo que você usa. Quanto mais seguro o protocolo que você conectar através de (OpenVPN, IKEv2), mais segura será toda a sua sessão.



Fonte: thebestvpn.com

Para saber mais...

... leia o trabalho Implementação de um Firewall baseado em Software Livre para acesso ao SIAFI e ao SIDOR, de Wallace Rodrigues de Santana

… veja a animação online do funcionamento do NAT, da Cisco Systems, Inc.

… veja a animação online do funcionamento do NAT, de Greg Tomsho e Angela Poland

... leia o capítulo 1 do livro Virtual Private Network – Aprenda a construir redes privadas virtuais em plataformas Linux e Windows, de Lino Sarlo da Silva

Módulo 16

Análise de Logs e Ferramentas de Monitoração

Registro de eventos

Registro de eventos ou *log* refere-se à coleta de eventos relevantes em um sistema computacional.

Um arquivo de *log* pode ser utilizado para auditoria e diagnóstico de problemas em sistemas computacionais, bem como pode servir também para alimentar estatísticas que irão prover informações sobre a utilização e o desempenho de um sistema.

Os registros de eventos estabelecem o fundamento para os sistemas de monitoramento automáticos, os quais são capazes de gerar relatórios consolidados e alertas na segurança do sistema.

Desta forma, os *logs* proveem mecanismos para:

- Analisar o estado de segurança de um sistema;
- Determinar uma sequência de eventos que possam ter comprometido o sistema.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil; ISO/IEC 27002:2013

Registro de eventos – categorias

Os registros de *log*, em geral, podem ser dos seguintes tipos:

- de aplicativos;
- de segurança;
- de configuração;
- de sistema;
- etc.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Registro de eventos – tipos

Em ambientes Windows, são comuns os seguintes tipos de eventos:

- **Erro:** um evento que indica um problema significativo, como perda de dados ou perda de funcionalidade. Por exemplo, se um serviço não for carregado durante a inicialização, um evento de erro será registrado;

- **Aviso:** um evento que não é necessariamente significativo, mas pode indicar um possível problema futuro. Por exemplo, quando o espaço em disco é baixo, um evento de aviso é registrado. Se um aplicativo puder se recuperar de um evento sem perda de funcionalidade ou dados, ele geralmente poderá classificar o evento como um evento de Aviso;

- **Informações:** um evento que descreve a operação bem-sucedida de um aplicativo, driver ou serviço. Por exemplo, quando um driver de rede é carregado com êxito, pode ser apropriado registrar um evento Information. Observe que geralmente é inadequado para um aplicativo de desktop registrar um evento toda vez que ele é iniciado;

continua...

Fonte: Microsoft Windows Dev Center

Registro de eventos – tipos

...continuação

- **Auditoria de sucesso:** um evento que registra uma tentativa de acesso de segurança auditada que seja bem-sucedida. Por exemplo, a tentativa bem-sucedida de um usuário de efetuar logon no sistema é registrada como um evento de Auditoria de Sucesso.

- **Auditoria de falha:** um evento que registra uma tentativa de acesso de segurança auditada que falha. Por exemplo, se um usuário tentar acessar uma unidade de rede e falhar, a tentativa será registrada como um evento de auditoria de falha.

Fonte: Microsoft Windows Dev Center

Registro de eventos – conteúdo

De acordo com o item 12.4.1 da ISO/IEC 27002:2013, quando for relevante, um registro de *log* deve conter:

- identificação dos usuários (ID);
- atividades do sistema;
- datas, horários e detalhes de eventos-chave, como, por exemplo, horário de entrada (*log-on*) e saída (*log-off*) no sistema;
- identidade do dispositivo ou sua localização, quando possível, e o identificador do sistema;
- registros das tentativas de acesso ao sistema, aceitas e rejeitadas;
- registros das tentativas de acesso a outros recursos e dados, aceitas e rejeitadas;
- alterações na configuração do sistema;

continua...

Fonte: ISO/IEC 27002:2013

Registro de eventos – conteúdo

... continuação

- uso de privilégios;
- uso de aplicações e utilitários do sistema;
- arquivos acessados e o tipo de acesso;
- endereços e protocolos de rede;
- alarmes provocados pelo sistema de controle de acesso;
- ativação e desativação dos sistemas de proteção, como sistemas de antivírus e sistemas de detecção de intrusos;
- registros de transações executadas pelos usuários nas aplicações.

Fonte: ISO/IEC 27002:2013

Registro de eventos – proteção

De acordo com o item 12.4.2 da ISO/IEC 27002:2013, convém que as informações dos registros de eventos (*log*) e os seus recursos sejam protegidos contra acesso não autorizado e adulteração.

Convém que os controles implementados objetivem a proteção contra modificações não autorizadas às informações dos *logs* e problemas operacionais com os recursos dos registros (*log*), como:

- alterações dos tipos de mensagens que são gravadas;
- arquivos de registros (*log*) sendo editados ou excluídos;
- capacidade de armazenamento da mídia magnética do arquivo de registros (*log*) excedida, resultando em falhas no registro de eventos ou sobreposição do registro de evento anterior.

Fonte: ISO/IEC 27002:2013

Registro de eventos – proteção

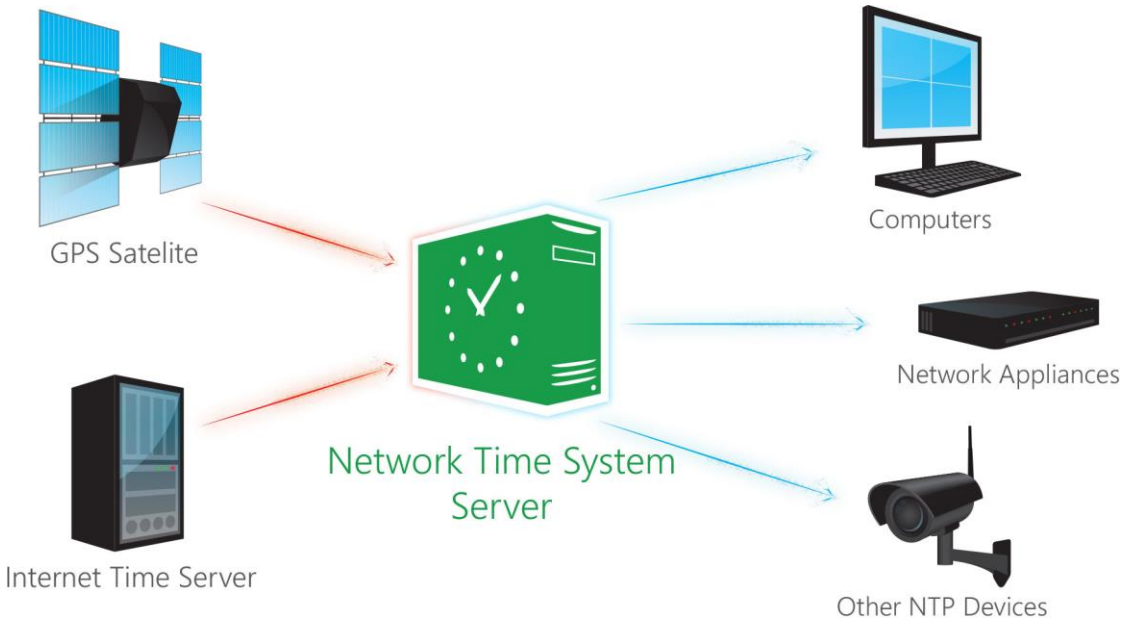
Já o item 12.4.3 da ISO/IEC 27002:2013 alerta que as atividades dos administradores e operadores do sistema devem ser registrados e os registros (*logs*) protegidos e analisados criticamente, a intervalos regulares.

Isso se faz importante pois as pessoas que possuem conta de usuário privilegiado podem ser capazes de manipular os registros (*logs*) nos recursos de processamento da informação que estão sob o seu controle direto, sendo portanto, necessário proteger e analisar criticamente os registros (*logs*) para manter o controle dos usuários privilegiados.

Fonte: ISO/IEC 27002:2013

Registro de eventos – sincronismo

De acordo com o item 12.4.4 da ISO/IEC 27002:2013, convém que os relógios de todos os sistemas de processamento de informações relevantes, dentro da organização ou do domínio de segurança, sejam sincronizados com uma única fonte de tempo precisa, como um servidor NTP, por exemplo.



Fonte: ISO/IEC 27002:2013

Registro de eventos – gerenciamento

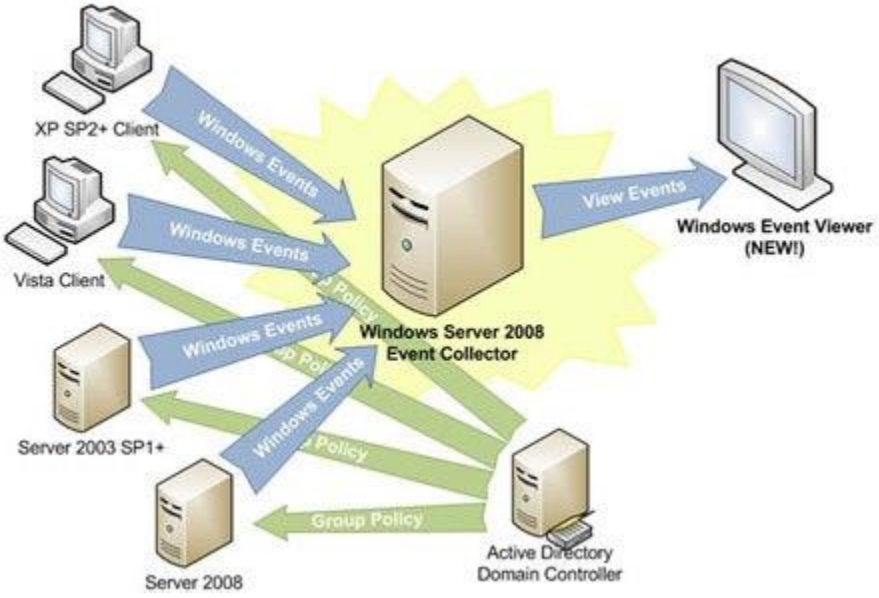
O gerenciamento dos registros de *log* se dá em três etapas:

- Coleta dos dados;
- Análise dos dados;
- Notificação.

Fonte: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Registro de eventos – Windows

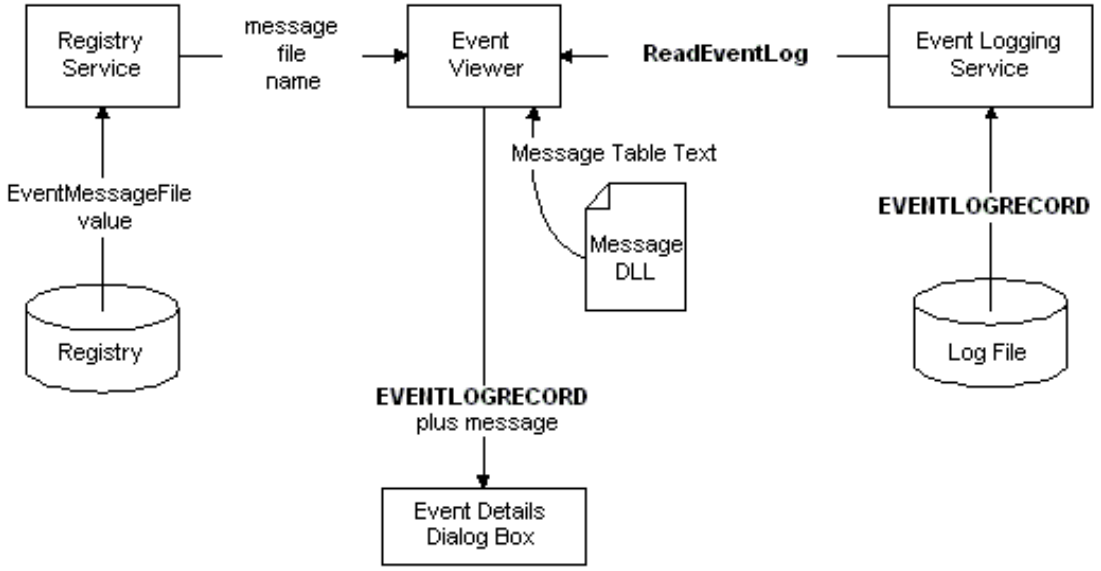
O log de eventos do Windows fornece uma maneira padrão e centralizada para os aplicativos e o sistema operacional registrarem eventos importantes de software e hardware. O serviço de registro de eventos registra eventos de várias origens e os armazena em uma única coleção chamada log de eventos, que pode ser visualizada por meio do aplicativo nativo Event Viewer.



Fonte: Microsoft Windows Dev Center

Registro de eventos – Windows

O aplicativo nativo Event Viewer usa a função OpenEventLog para abrir o log de eventos de uma fonte qualquer, como um serviço ou outro aplicativo de sistema, por exemplo. O Event Viewer pode então usar a função ReadEventLog para ler registros de eventos do log. A função ReadEventLog retorna um buffer contendo uma estrutura EVENTLOGRECORD e informações adicionais que descrevem um evento registrado, conforme ilustra a figura.



Fonte: Microsoft Windows Dev Center

Registro de eventos – Linux

O *syslog* é um padrão para o registro de eventos. Ele permite separar o software que gera as mensagens do sistema que as armazena e do software que as reporta e analisa.

Cada mensagem é rotulada com um código de recurso, indicando o tipo de software que gerou a mensagem e atribuindo um nível de severidade.

O *syslog* pode ser usado para gerenciamento de sistemas e auditoria de segurança, bem como mensagens informativas gerais, análise e depuração.

Uma grande variedade de dispositivos, como impressoras, roteadores e receptores de mensagens em muitas plataformas usam o padrão *syslog*.

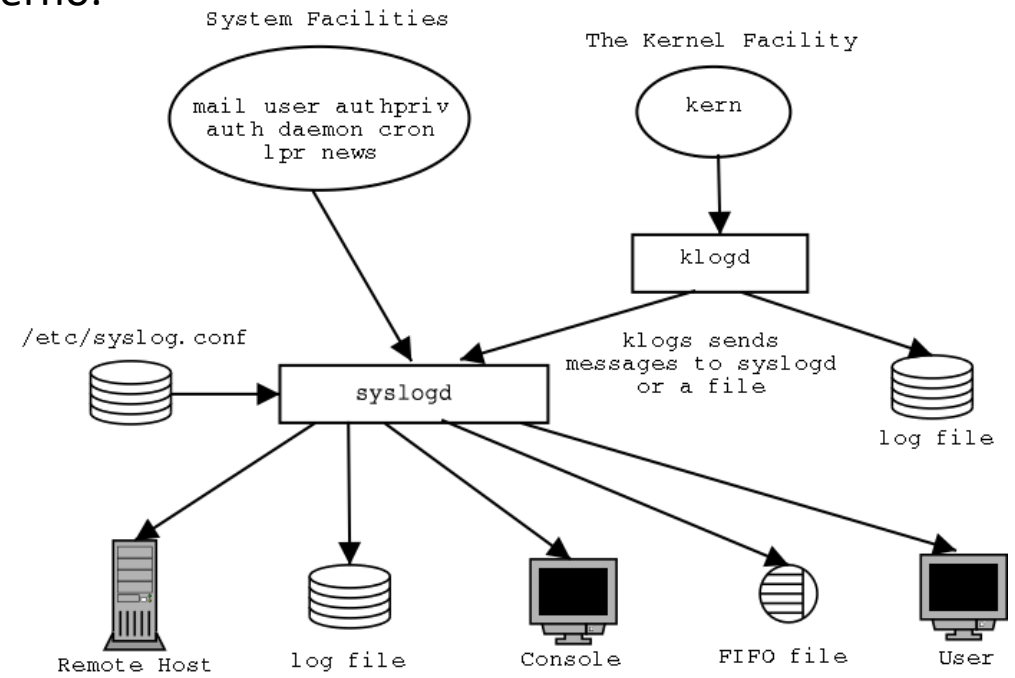
Isso permite a consolidação de dados de registro de diferentes tipos de sistemas em um repositório central. Implementações do *syslog* existem para muitos sistemas operacionais.

Fonte: wikipedia.org

Registro de eventos – Linux

O *daemon* (serviço) *syslogd* é um processo que registra as mensagens de aplicativos e demais processos do sistema. Ele pode receber esse registros de uma infinidade de outros serviços e também pode exportá-los para um servidor de registros de eventos externo.

Para se comunicar com outros servidores, o *syslogd* usa a porta UDP 514 ou a porta TCP 6514 em conjunto com o protocolo TLS.

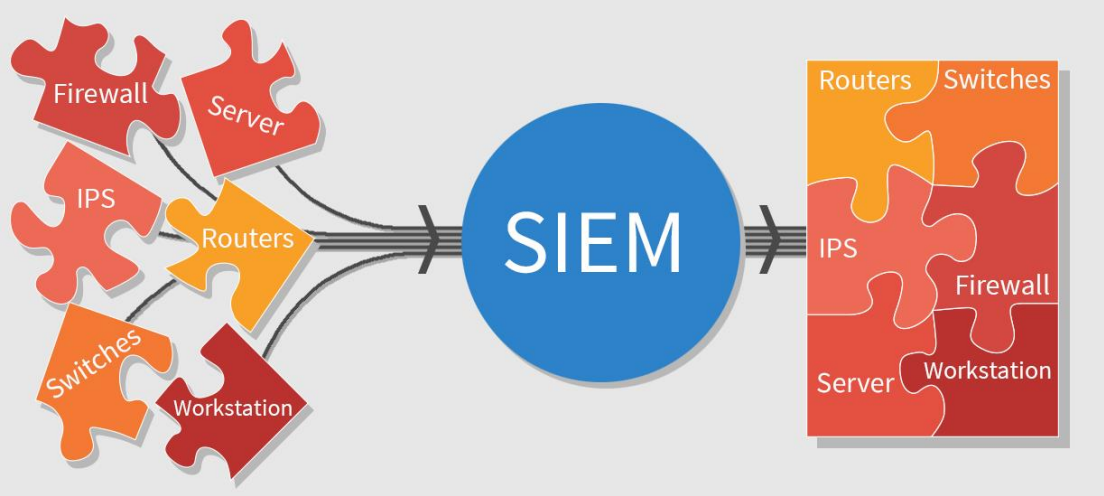


Fonte: RFC 5425 – Transport Layer Security (TLS) Transport Mapping for Syslog

SIEM - Gerenciamento e Correlação de Eventos de Segurança

O SIEM (Security Information and Event Management) é uma solução de software que combina SIM (Security Information Management) com SEM (Security Event Manager).

Uma solução SIEM permite que os eventos gerados por diversas aplicações de segurança, tais como firewalls, proxies, sistemas de prevenção a intrusão e antivírus, sejam coletados, normalizados, armazenados e correlacionados. Isso possibilita uma rápida identificação e resposta aos incidentes.



Fonte: wikipedia.org

SIEM - Gerenciamento e Correlação de Eventos de Segurança

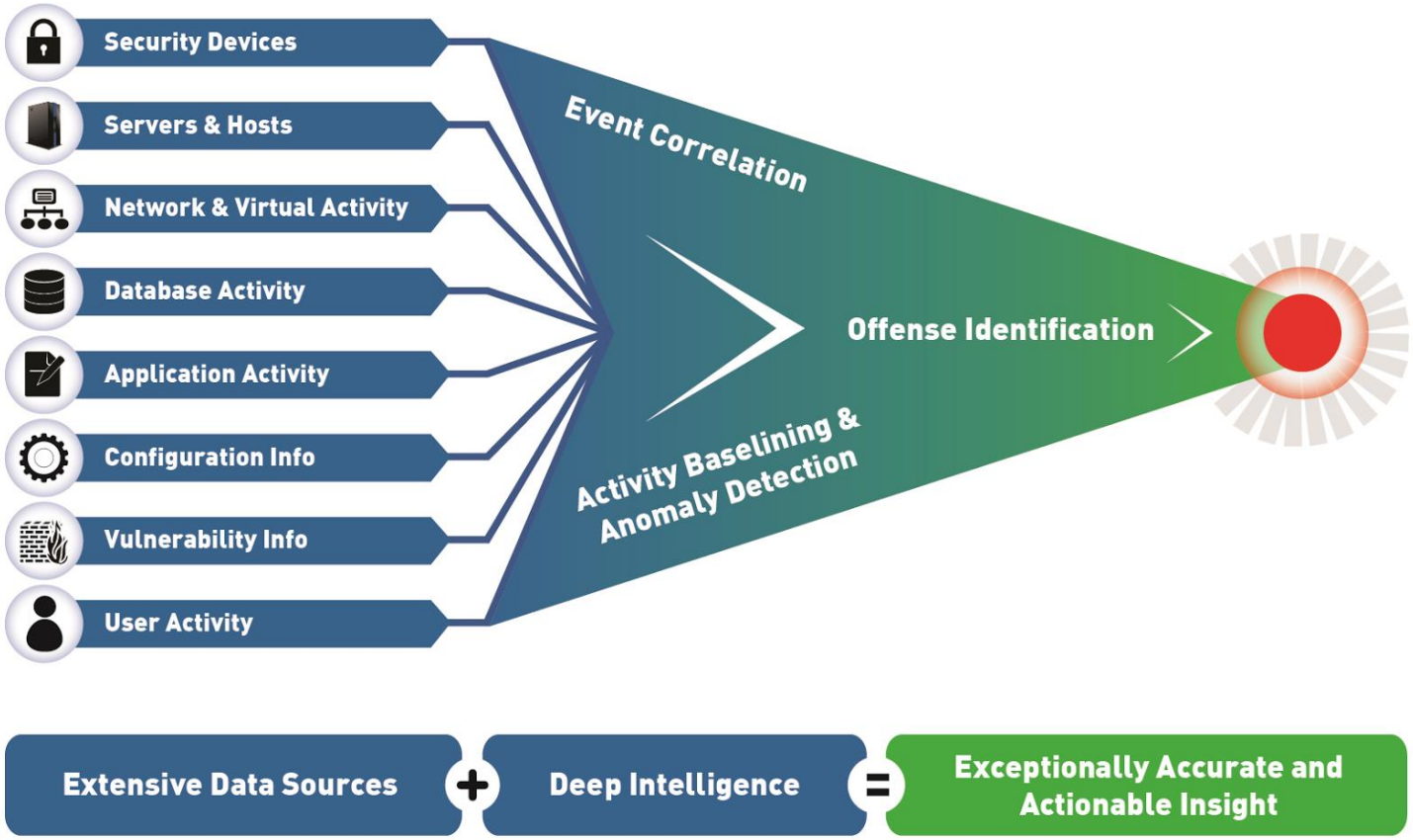
Enquanto ferramentas SEM oferecem monitoramento em tempo real dos eventos de segurança, coletando e agregando os dados (com resposta automática em alguns casos), uma ferramenta SIM oferece análise histórica dos eventos de segurança, também coletando e correlacionando os eventos, porém não em tempo real, o que permite consultas mais complexas ao repositório.

As soluções SIEM combinam os recursos oferecidos em ambas as tecnologias (SIM e SEM).

O termo SIEM foi cunhado em 2005 por Mark Nicolett e Amrit Williams da Gartner, descrevendo um produto capaz de coletar, analisar e apresentar informações dos dispositivos de segurança de rede; softwares de controle de acesso; gerenciamento de vulnerabilidades; ferramentas de conformidade; logs de sistema operacional, banco de dados e aplicações; e por último, dados de ameaças externas.

Fonte: wikipedia.org

SIEM - Gerenciamento e Correlação de Eventos de Segurança



Fonte: Malta Information Technology Agency

SIEM - Gerenciamento e Correlação de Eventos de Segurança – Exemplo

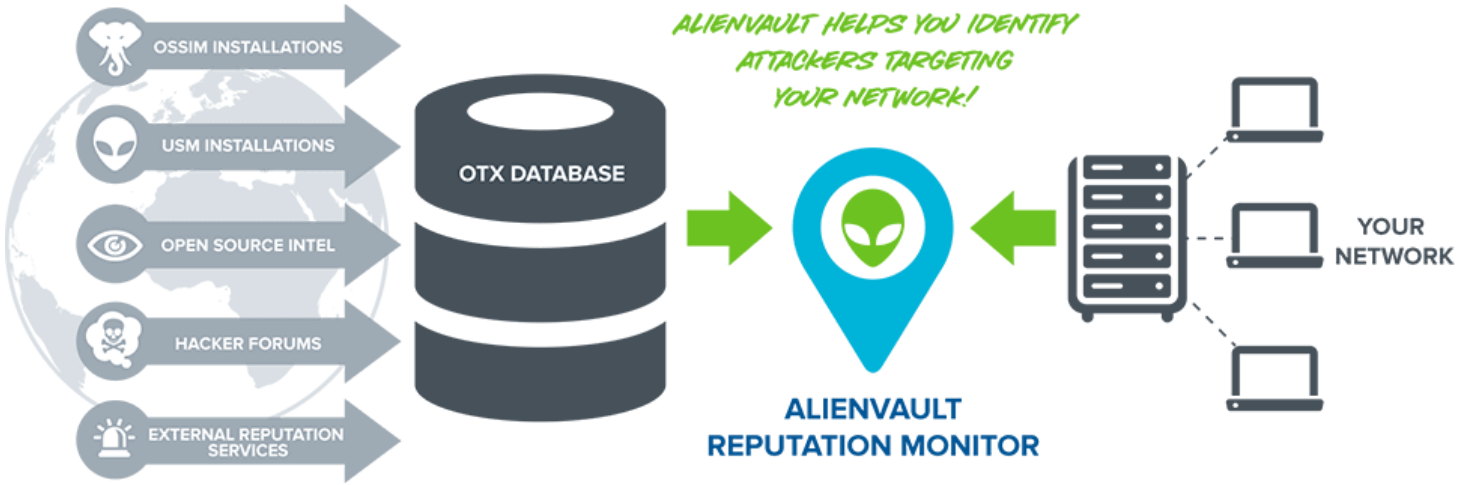
O OSSIM (Open Source Security Information Management) é um sistema de gerenciamento de eventos e informações de segurança de código aberto, integrando uma seleção de ferramentas projetadas para auxiliar administradores de rede em segurança de computadores, detecção e prevenção de intrusões.

Fonte: wikipedia.org

SIEM - Gerenciamento e Correlação de Eventos de Segurança – Exemplo

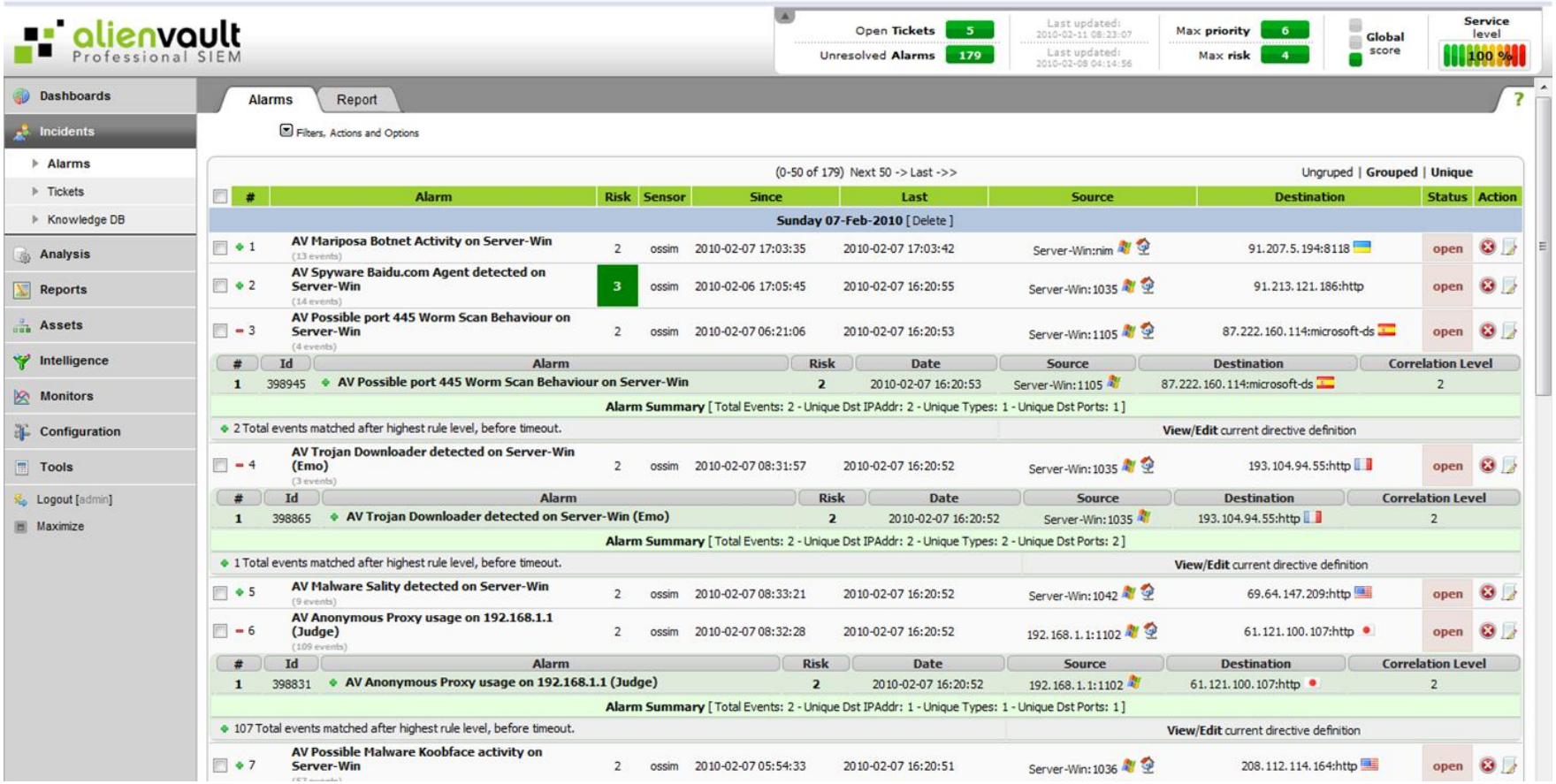
A AlienVault, mantenedora do OSSIM, também mantém um serviço colaborativo (crowdsource) de coleta de informações de reputação de IP, gerado pelas instalações OSSIM ativas espalhadas pelo mundo.

Este serviço, denominado OTX (Open Threat Exchange), usa informações codificadas das instalações OSSIM participantes para identificar endereços da Internet envolvidos em atividades maliciosas e compartilhar essas informações nessas mesmas instalações do OSSIM. Este serviço foi lançado em 2012.



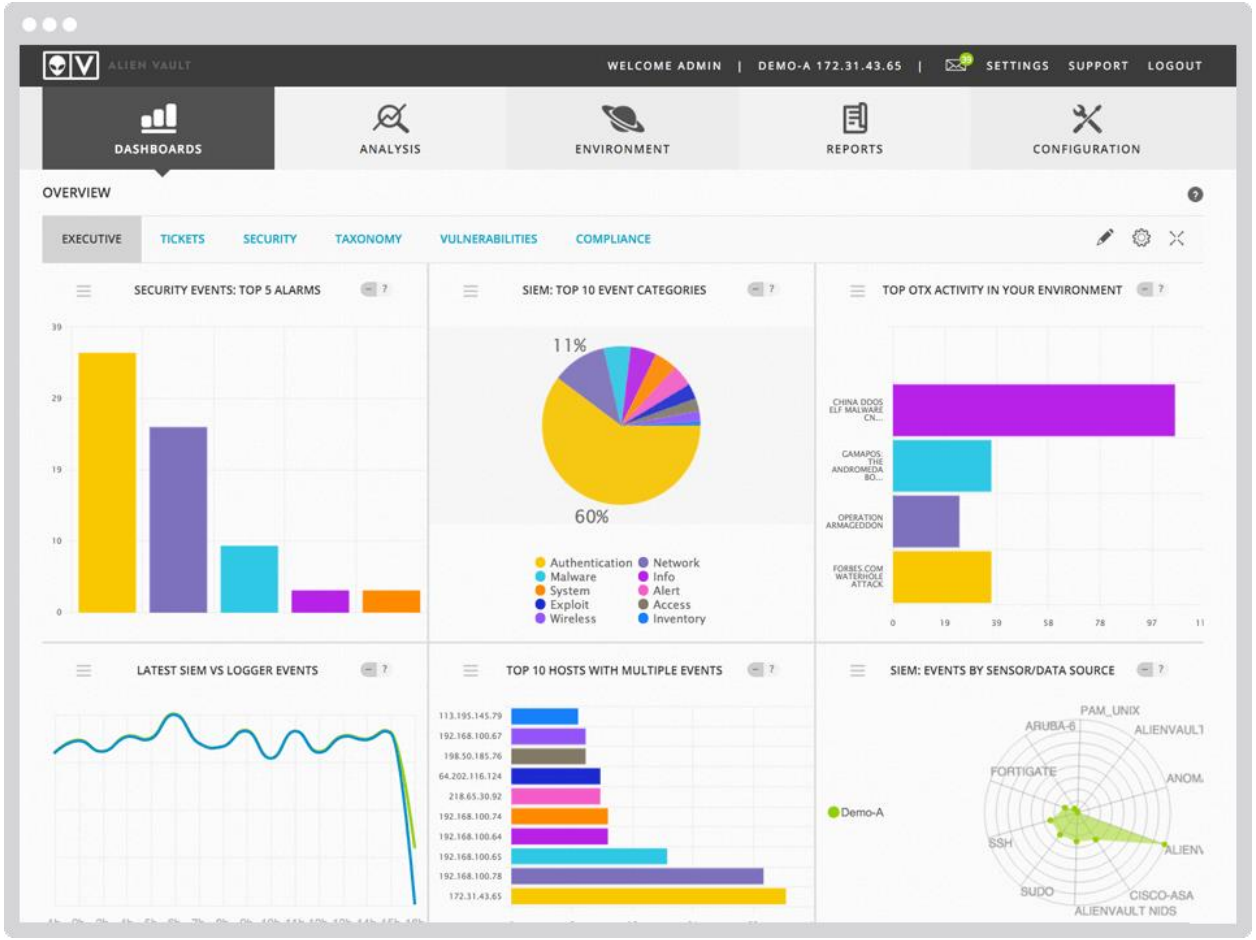
Fonte: wikipedia.org

SIEM - Gerenciamento e Correlação de Eventos de Segurança – Exemplo



Fonte: wikipedia.org

SIEM - Gerenciamento e Correlação de Eventos de Segurança – Exemplo



Fonte: alienvault.com

Para saber mais...

… leia o tutorial Análise e Interpretação de logs, do Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

Módulo 17

Auditoria

Introdução

"Exame ou verificação de uma dada matéria, tendente a analisar a conformidade da mesma com determinadas regras, normas ou objetivos, conduzido por uma pessoa idônea, tecnicamente preparada, realizado com observância de certos princípios, métodos e técnicas geralmente aceites, com vista a possibilitar ao auditor formar uma opinião e emitir um parecer sobre a matéria analisada."

Organização das Instituições Superiores de Controle (ISC) da Comunidade dos Países de Língua Portuguesa (CPLP)

"A auditoria é uma atividade que engloba o exame das operações, processos, sistemas e responsabilidades gerenciais de uma determinada entidade, com o objetivo de verificar sua conformidade com certos objetivos e políticas institucionais, orçamentos, regras, normas ou padrões."

Abílio Bueno Neto e Davi Solonca

Conceitos

Alguns conceitos básicos relacionados com a auditoria são: campo, âmbito e área de verificação.

- **Campo da auditoria:** É a definição do objeto e do período a fiscalizar, bem como da natureza da auditoria a realizar (por exemplo, auditoria da legalidade e/ou regularidade de determinadas operações em determinado mês, ano ou, em alguns casos, ao período de gestão do administrador da instituição). O seu objeto pode ser uma entidade completa (organismo público, empresa ou projeto, etc.), uma parte ou uma função dessa entidade;
- **Âmbito da auditoria:** Tem por finalidade determinar a amplitude e a exaustão dos processos de auditoria preconizados, o que inclui uma limitação racional dos trabalhos a executar, de modo a tornar aceitável para o auditor o risco de serem errôneas as suas conclusões de auditoria. Em outras palavras, determina o escopo;
- **Área de verificação:** É a área determinada pelo campo da auditoria e pelo seu âmbito (escopo), quando considerados em conjunto. A área de auditoria delimita de modo muito preciso os temas da auditoria, em função, por um lado, da entidade a fiscalizar e, por outro, da natureza da auditoria preconizada.

Fonte: Organização das Instituições Superiores de Controle (ISC) da Comunidade dos Países de Língua Portuguesa (CPLP)

Conceitos

Algumas técnicas usadas em auditorias de sistemas são comuns a outros tipos de auditoria, como:

- **Entrevista:** reunião realizada com os envolvidos com o ponto auditado, que deve ser documentada;
- **Questionário:** conjunto de perguntas que podem ser aplicadas a muitas pessoas simultaneamente, sem a presença do auditor;
- **Verificação in loco**: observação direta de instalações, atividades ou produtos auditados.

Fonte: Auditoria de Sistemas Informatizados, Abílio Bueno Neto e Davi Solonca.

Auditoria de sistemas

Uma auditoria de sistemas de informação pode abranger desde o exame de dados registrados em sistemas informatizados, até a avaliação do próprio sistema operacional e demais aplicativos, bancos de dados, bem como incluir a avaliação do ambiente de desenvolvimento, do ambiente de operação, do ambiente de gerenciamento da rede e todos os demais elementos associados a um ou mais sistemas de informação corporativos.

A auditoria de sistemas de informação inclui também o exame detalhado das políticas de segurança da informação e dos controles adotados para mitiga-los.

Fonte: Auditoria de Sistemas Informatizados, Abílio Bueno Neto e Davi Solonca.

Auditoria de sistemas – demonstração Cenário

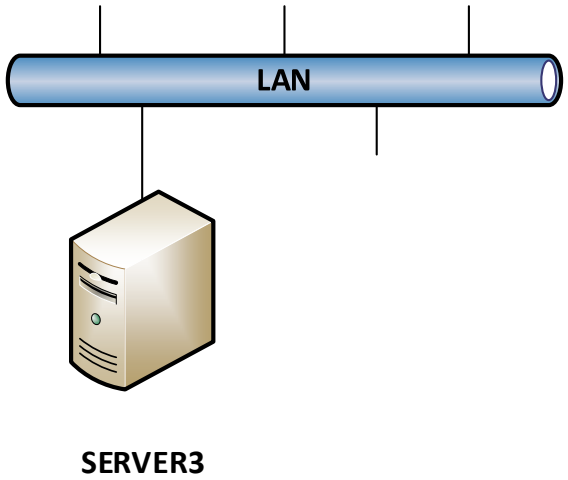
Nesta demonstração será usado um servidor Windows 2008 R2 Standard, com a seguinte configuração:

- **SERVER3**: IP 192.168.0.3, sem nenhum serviço específico.

Iremos demonstrar como auditar a criação e exclusão de arquivos em uma dada pasta.

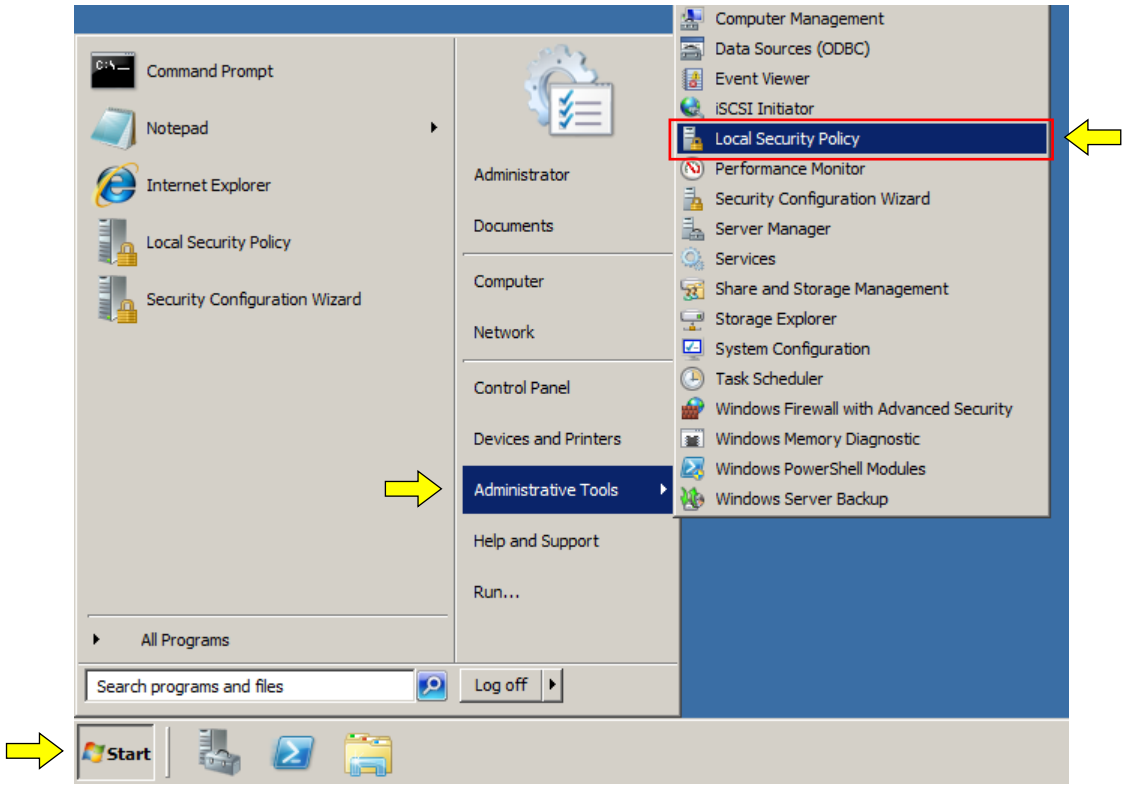
Para isso são necessários dois passos:

- Editar a política de segurança local (que também pode ser feita usando-se políticas de grupo (GPO) a serem distribuídas pelo Active Directory);
- Habilitar a auditoria na pasta desejada.



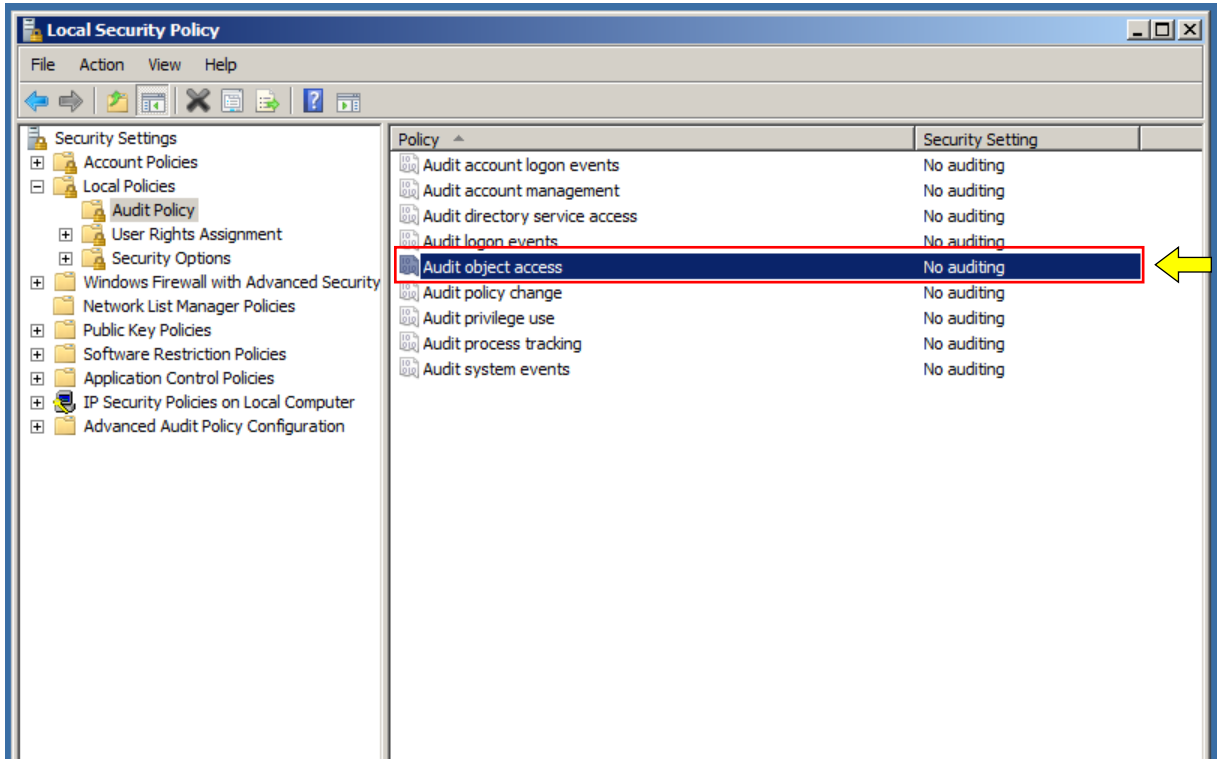
Auditoria de sistemas – demonstração Editando a política de segurança local

No Windows a Política de Segurança Local é configurada a partir do Local Security Policy.



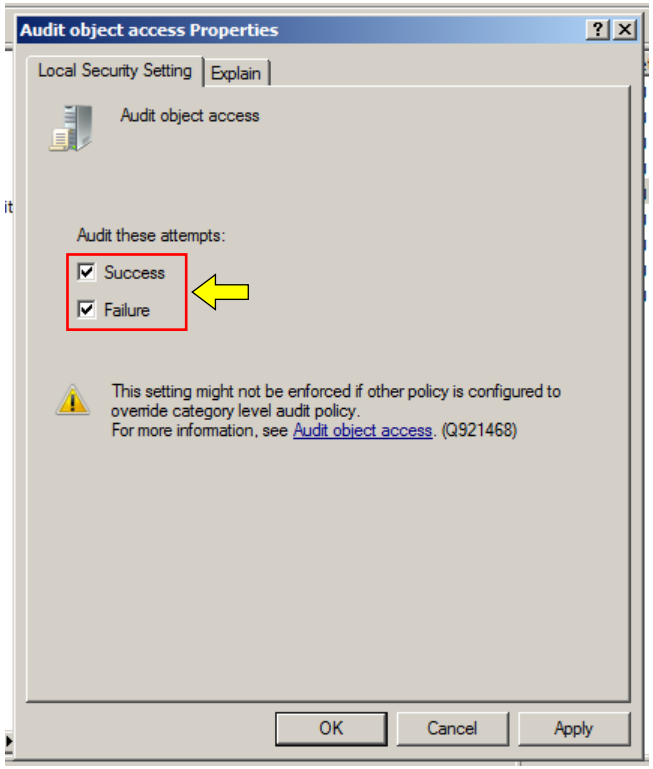
Auditoria de sistemas – demonstração
Editando a política de segurança local

Na console do Local Security Policy, em Security Settings, clicar em Local Policies, Audit Policy e por fim dar duplo clique em Audit object acsess.



Auditoria de sistemas – demonstração
Editando a política de segurança local

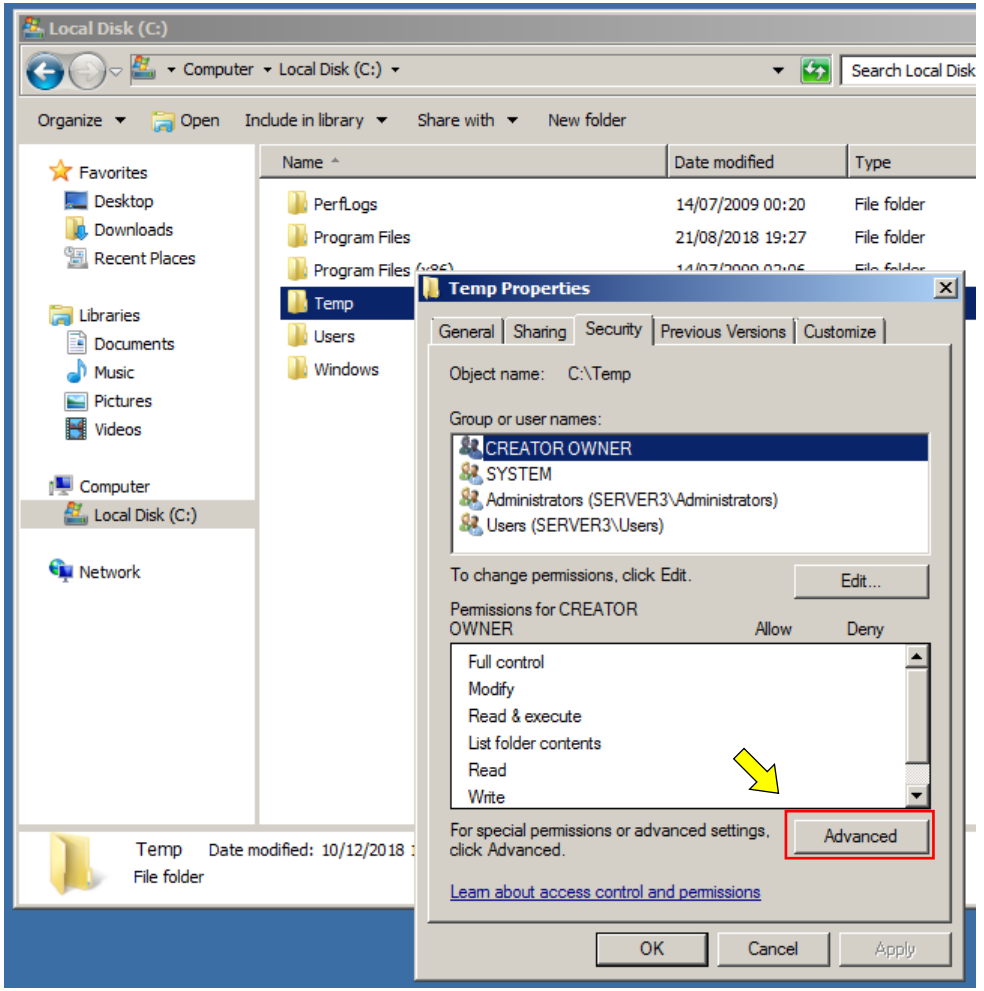
Na janela Audit object access Properties, selecionar as caixas de seleção Success e Failure.



Auditoria de sistemas – demonstração
Habilitando a auditoria

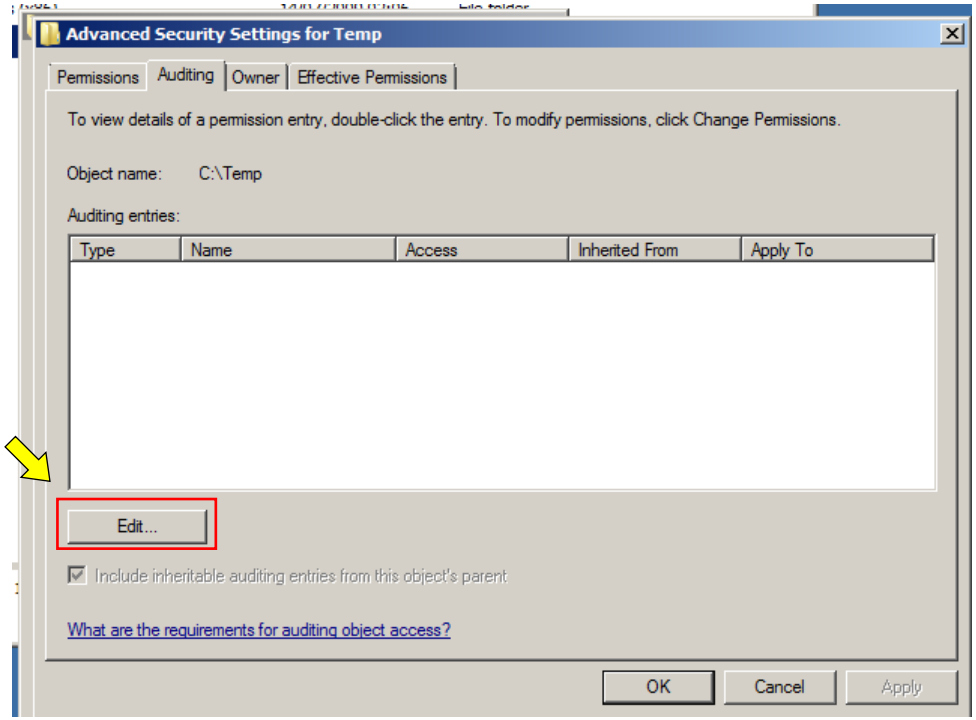
Para habilitar a auditoria, seguir os seguintes passos:

- Criar uma pasta Temp em C:;
- A partir do Windows Explorer, clicar com o botão direito na pasta criada e selecionar Properties;
- Na janela de propriedades, selecionar a aba Security e clicar em Advanced.



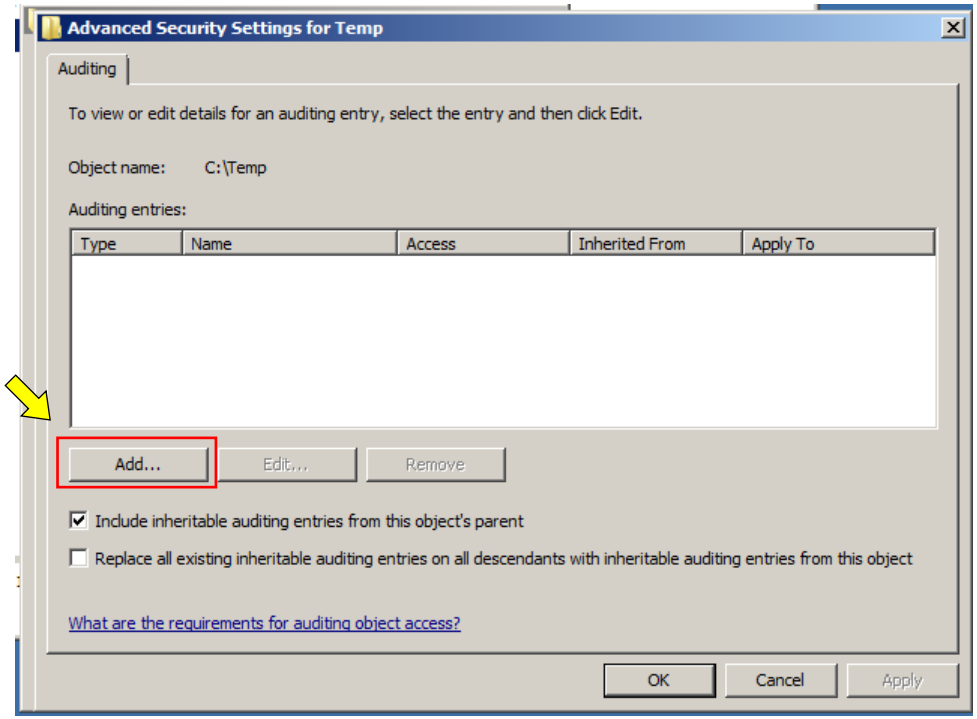
Auditoria de sistemas – demonstração
Habilitando a auditoria

Na janela Advanced Security Settings selecionar a aba Auditing e clicar em Edit...



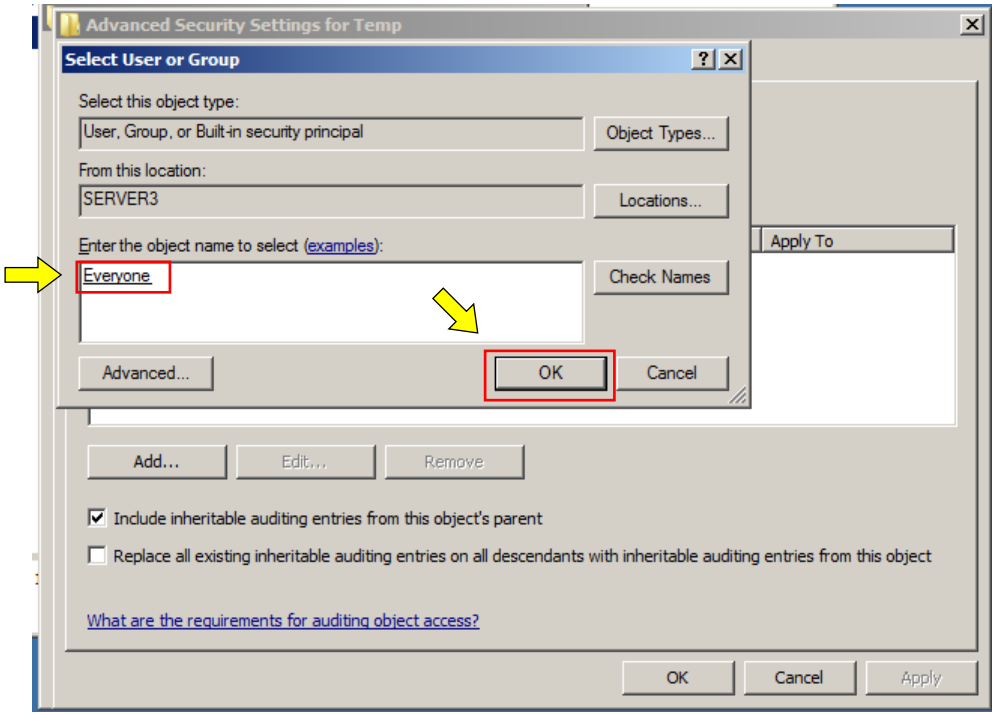
Auditoria de sistemas – demonstração
Habilitando a auditoria

Em Auditing, clicar em Add...



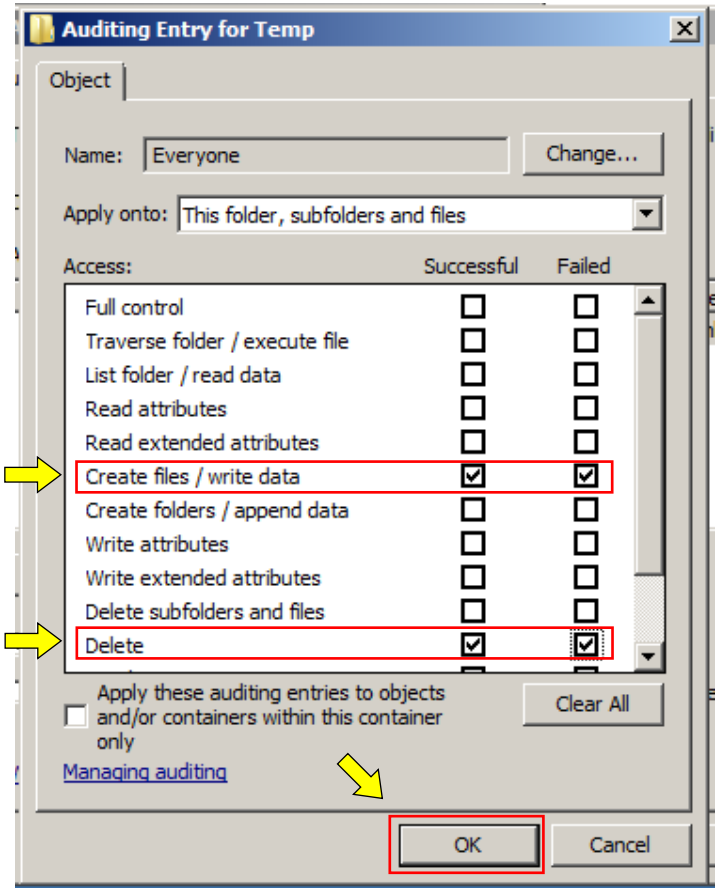
Auditoria de sistemas – demonstração
Habilitando a auditoria

Na janela Select User or Group, adicionar o grupo Everyone (Todos) e clicar em OK.



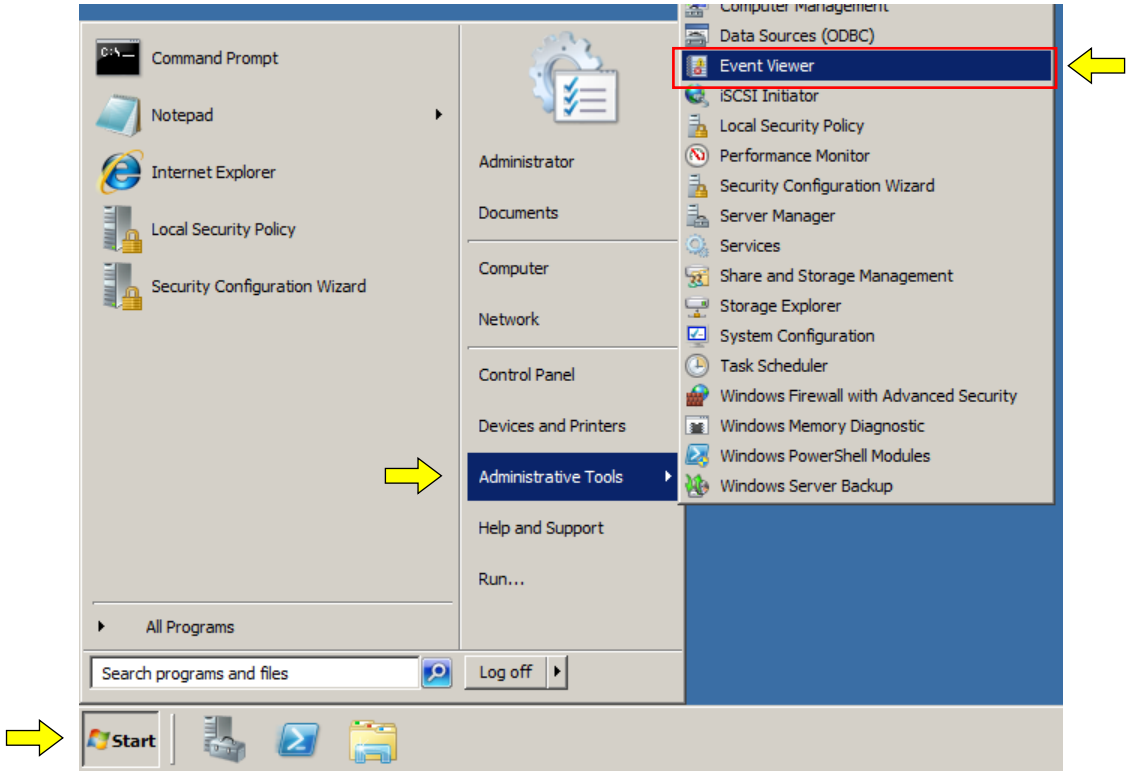
Auditoria de sistemas – demonstração
Habilitando a auditoria

Na janela Auditing Entry, selecionar as caixas de seleção para “Create files / write data” e “Delete”, tanto para eventos executados com sucesso (Successful) quanto para eventos executados com falha (Failed).



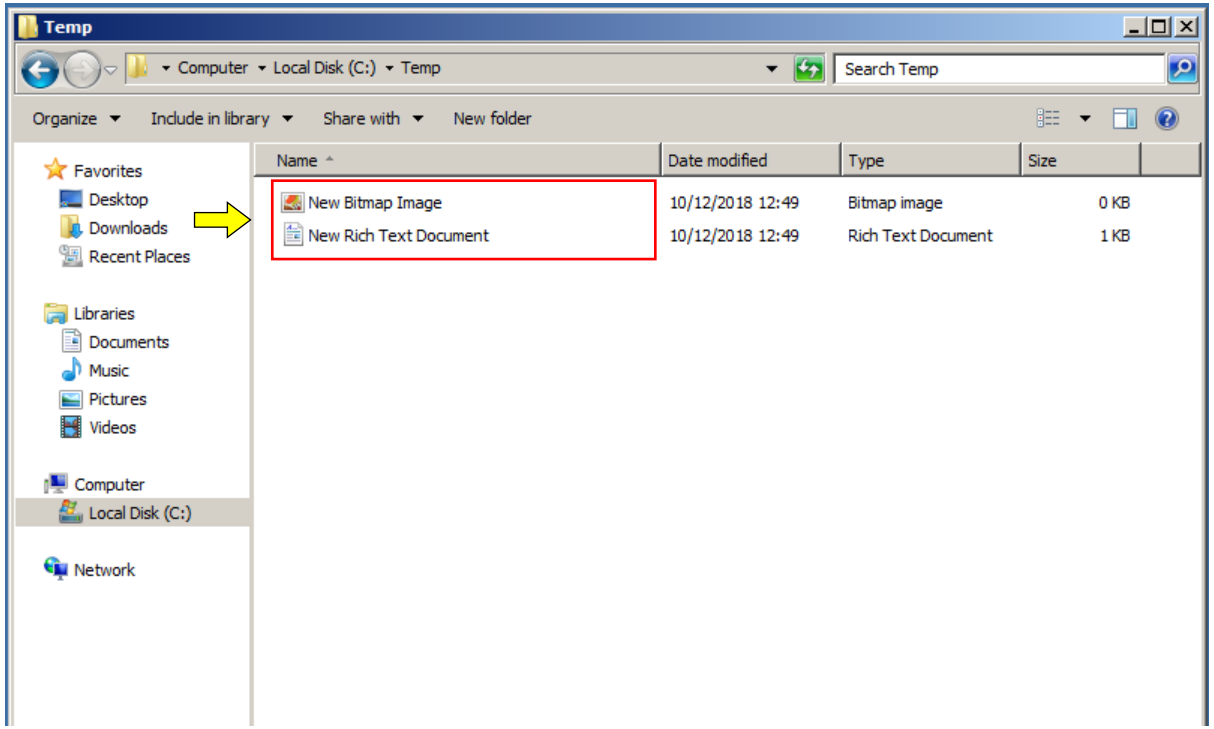
Auditoria de sistemas – demonstração Verificando os eventos de auditoria

No Windows o Registro de Eventos é visualizado a partir do Event Viewer.



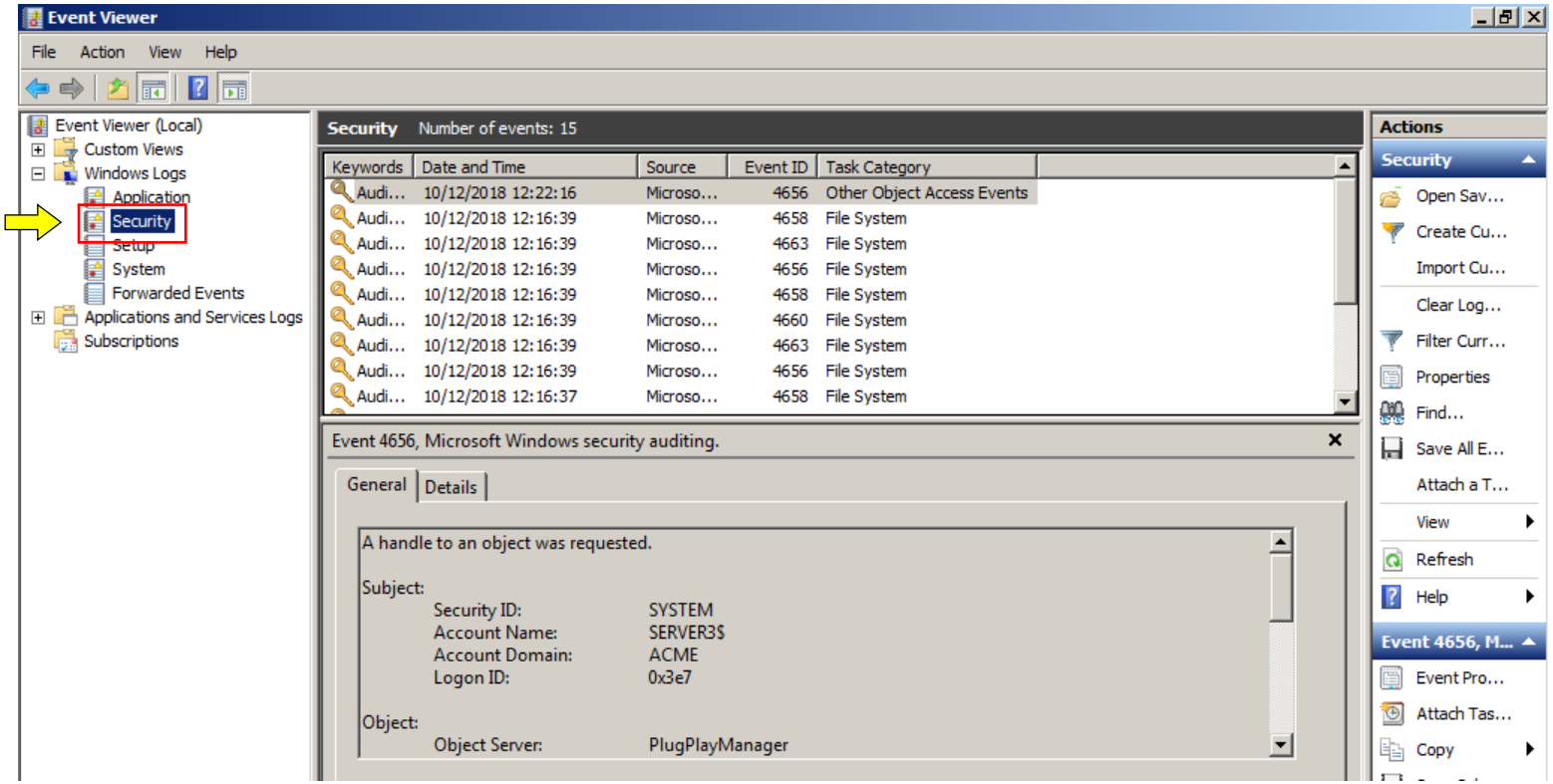
Auditoria de sistemas – demonstração
Testando a auditoria

Após habilitar a auditoria na pasta C:\Temp, experimente criar e apagar alguns arquivos nesta pasta, para que os eventos possam ser gerados e visualizados posteriormente.



Auditoria de sistemas – demonstração
Verificando os eventos de auditoria

Uma vez na console do Event Viewer, em Event Viewer (Local), clicar em Windows Logs e em seguida clicar em Security.

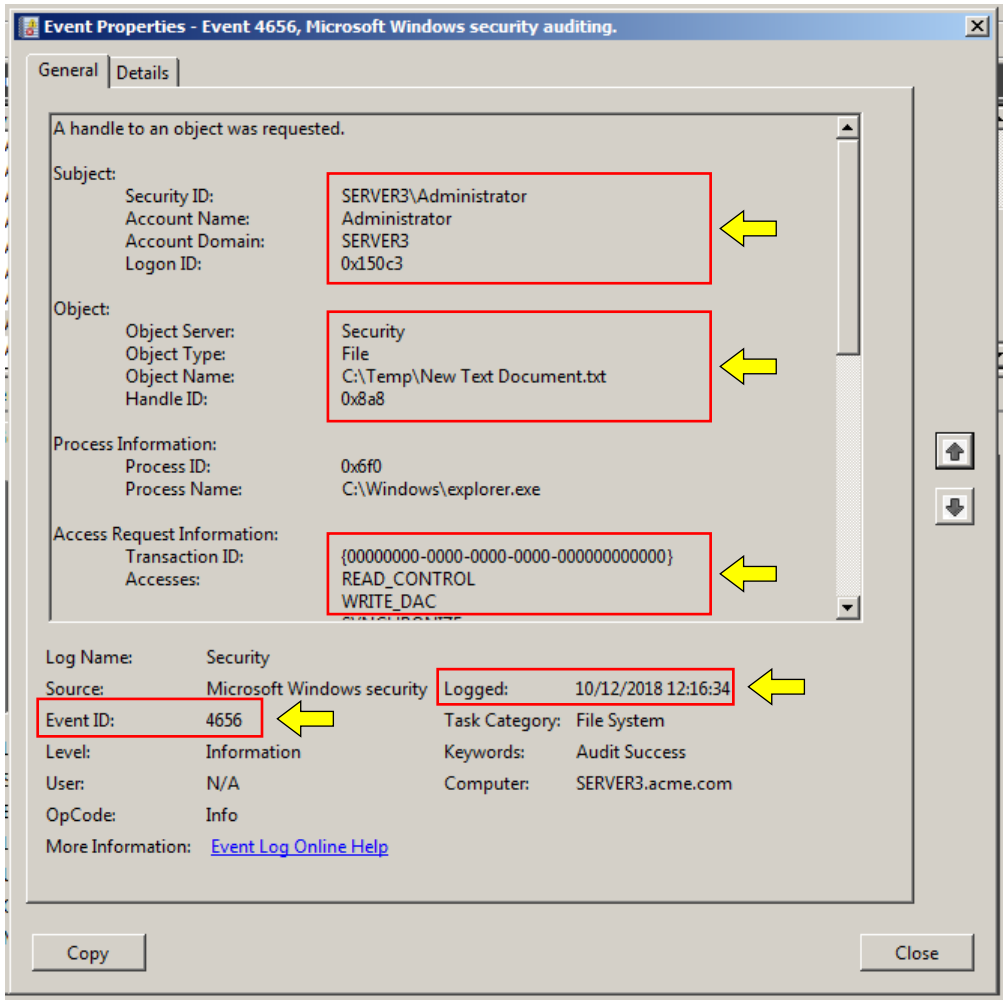


Auditoria de sistemas – demonstração
Verificando os eventos de auditoria

No registro de evento de auditoria, para os eventos de criação de arquivos, procurar pelo código de evento 4656.

Neste evento é possível ver quem foi o usuário; qual o nome do arquivo envolvido; e qual foi a operação.

Neste caso, o usuário Administrator criou (WRITE_DAC) o arquivo C:\Temp\New Text Document.txt, na data e hora indicados.

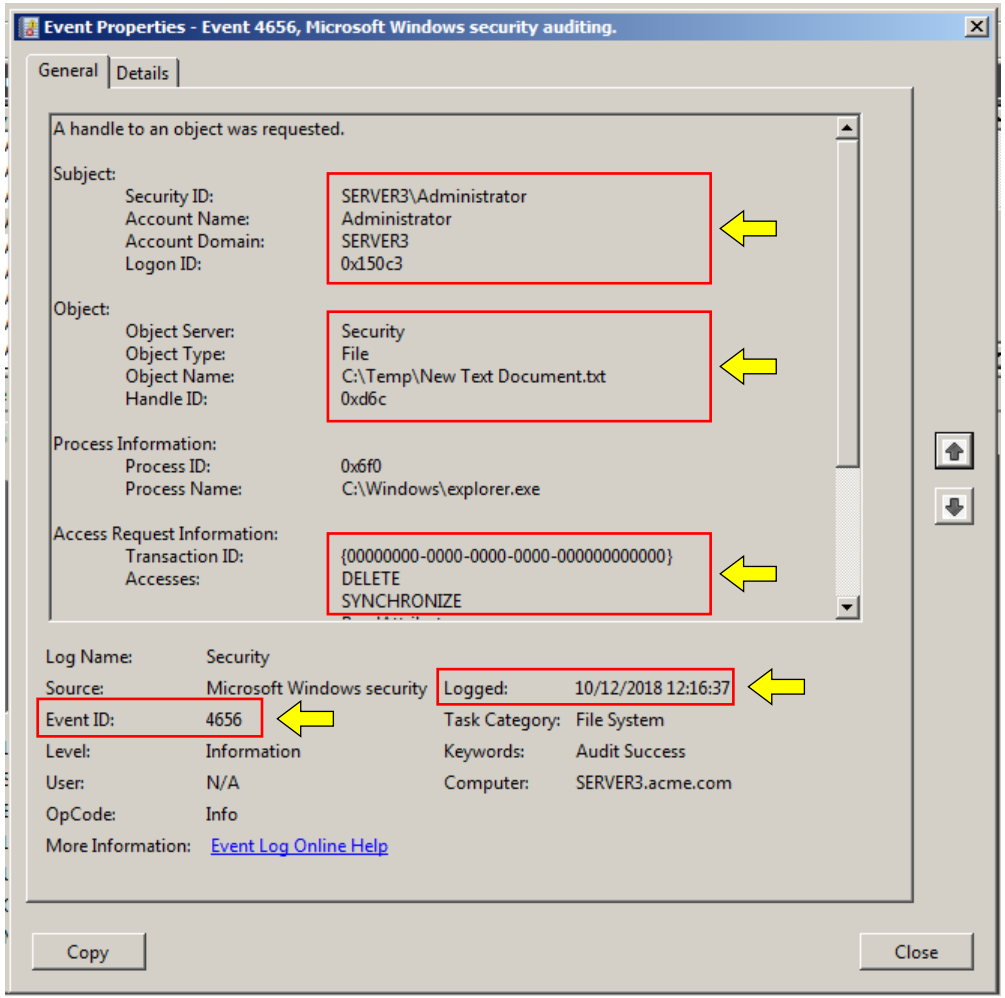


Auditoria de sistemas – demonstração
Verificando os eventos de auditoria

No registro de evento de auditoria, para os eventos de criação de arquivos, procurar pelo código de evento 4656.

Neste evento é possível ver quem foi o usuário; qual o nome do arquivo envolvido; e qual foi a operação.

Neste caso, o usuário Administrator apagou (DELETE) o arquivo C:\Temp\New Text Document.txt, na data e hora indicados.



Auditoria de sistemas – cuidados

A auditoria do Windows aumenta o processamento e os arquivos de registro de eventos (log) crescem muito rápido.

Por isso, a auditoria do Windows só deve ser ligada para averiguar comportamentos pontuais no sistema ou para investigar problemas.

Para saber mais...

… leia a apostila Auditoria de Sistemas Informatizados, de Abílio Bueno Neto e Davi Solonca

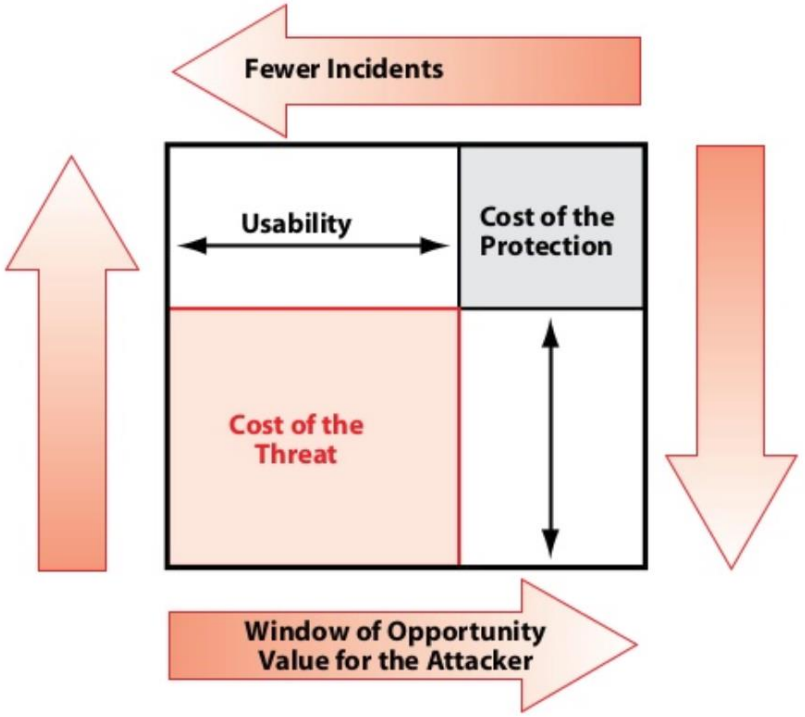
Módulo 18

Return on Security Investiment

Introdução

O Retorno do Investimento de Segurança, também conhecido como ROSI (Return On Security Investment), é um método para calcular e determinar se um conjunto de medidas de segurança que tem por objetivo mitigar ou reduzir os riscos de segurança vale ou não a pena ser implementado.

Em outras palavras, para que um conjunto de medidas de segurança seja financeiramente viável, a redução do risco deve ser maior do que o custo de implementação destas medidas.



Return on Investiment

No mundo das finanças e dos negócios, o investimento de capital deve ser medido pela sua eficácia em gerar rentabilidade para a organização. É aí que entra o cálculo do Retorno do Investimento, ou ROI (Return on Investiment), para a avaliação de um investimento.

O ROI é um índice de rentabilidade para um investimento específico. Ele ajuda a determinar se devemos fazer o investimento ou simplesmente ignorá-lo.

Para que um investimento seja justificado, deve expressar em termos quantitativos por que precisa acontecer. As propostas com o maior potencial de rentabilidade geralmente vencem, e é por isso que as propostas de segurança da informação muitas vezes perdem. O cálculo típico do retorno do investimento é dado por:

$$ROI = \frac{Ganho\ do\ Investimento - Custo\ do\ Investimento}{Custo\ do\ Investimento}$$

Fonte: How to calculate your return on security investments, Isaac Kohen

ROI vs ROSI

Por que o ROI clássico não funciona para retorno do investimento em segurança?

A equação do ROI funciona apenas para investimentos que geram resultados positivos, como redução de custos ou aumento de receita.

Mas o que é um investimento em segurança?

Um investimento em segurança não aumenta as receitas diretamente e nem fornece retorno imediato. Em vez disso, os investimentos em segurança dizem respeito ao gerenciamento de riscos que resultam em prevenção de perdas e mitigação de riscos. Em outras palavras, a segurança em geral não é um investimento que fornece lucro, mas que previne as perdas.

Assim, um cálculo do ROSI deve indicar quanta perda a organização poderia evitar devido ao investimento em segurança, por isso se faz necessária uma fórmula diferente.

Fonte: How to Calculate Return on Security Investment, Matt Middleton-Leal

Conceitos de avaliação de risco

Para quantificar o impacto da segurança da informação nos negócios, o risco precisa ser determinado. Os seguintes conceitos de risco serão a base do cálculo do ROSI.

Expectativa de Perda Única ou SLE (Single Loss Expectancy)

O SLE é a quantia esperada de dinheiro que será perdida quando ocorrer um risco. Nesta abordagem, o SLE pode ser considerado como o custo total de um incidente, assumindo sua única ocorrência.

Devido à natureza específica do incidente de segurança, a maior complexidade é levar em conta todos os ativos em que esse incidente tem impacto. Por exemplo, um laptop roubado não apenas custará a substituição do laptop em si, mas também implicará em perda de produtividade, perda de reputação, tempo de suporte de TI e, possivelmente, custo de perda de propriedade intelectual.

Fonte: Introduction to Return on Security Investment, da European Network and Information Security Agency

Conceitos de avaliação de risco

Taxa Anual de Ocorrência ou ARO (Annual Rate of Occurrence)

A ARO é uma medida da probabilidade de um risco ocorrer em um ano.

Estes dados são uma aproximação e podem depender de muitos fatores. Por exemplo, a ARO de uma inundação dependerá de fatores geográficos, a ARO de uma falha de disco é influenciado pela temperatura de operação, a ARO de um arrombamento dependerá da localização do ativo, e claro, a ARO também depende das medidas de segurança existentes. A ARO de um ataque de código malicioso bem-sucedido diminuirá significativamente após a implementação de um antivírus eficaz.

Uma forma de medir a ARO é avaliando o histórico de ocorrências.

Fonte: Introduction to Return on Security Investment, da European Network and Information Security Agency

Conceitos de avaliação de risco

Expectativa Anual de Perdas ou ALE (Annual Loss Expectancy)

A ALE é a perda monetária anual que pode ser esperada de um risco específico sobre um ativo específico.

A ALE é dada por:

$$ALE = ARO \times SLE$$

Fonte: Introduction to Return on Security Investment, da European Network and Information Security Agency

Conceitos de avaliação de risco

Expectativa Anual de Perda Modificada ou mALE (Modified Annual Loss Expectancy)

A implementação de uma solução de segurança eficaz reduz a ALE. Quanto mais eficaz é uma solução, mais reduzida a ALE será. Essa redução da perda monetária pode ser definida pela diferença entre a ALE sem a solução de segurança e a ALE modificada (mALE) que implementa a solução de segurança.

$$mALE = ALE \times (1 - Taxa\ de\ Mitigação)$$

Fonte: Introduction to Return on Security Investment, da European Network and Information Security Agency

Cálculo do ROSI

O cálculo do ROSI combina a avaliação quantitativa de riscos e o custo da implementação de medidas de segurança para esse risco. No final, compara a ALE com a perda esperada. A fórmula do ROSI é dada por:

$$ROSI = \frac{ALE - mALE - Custo\ da\ Solução}{Custo\ da\ Solução}$$

-- ou --

$$ROSI = \frac{ALE \times Taxa\ de\ Mitigação - Custo\ da\ Solução}{Custo\ da\ Solução}$$

-- ou --

$$ROSI = \frac{(ARO \times SLE) \times Taxa\ de\ Mitigação - Custo\ da\ Solução}{Custo\ da\ Solução}$$

Fonte: Introduction to Return on Security Investment, da European Network and Information Security Agency

Cálculo do ROSI – Exemplo

A empresa ACME está pensando em investir em uma solução antivírus. A cada ano a empresa sofre 5 ataques de vírus (ARO = 5). O CSO (Chief Security Officer) estima que cada ataque custa aproximadamente US$ 15.000 em perda de dados e produtividade (SLE = 15.000). Espera-se que a solução antivírus bloqueie 80% dos ataques (Taxa de Mitigação = 80%) e ela custará US$ 25.000 por ano (incluindo taxas de licença, treinamento, instalação, manutenção, etc.).

O retorno do investimento em segurança para essa solução será dado por:

$$ROSI = \frac{(ARO \times SLE) \times Taxa\ de\ Mitigação - Custo\ da\ Solução}{Custo\ da\ Solução}$$

$$ROSI = \frac{(5 \times 15000) \times 80\% - 25000}{25000} = \boxed{140\%}$$

Fonte: Introduction to Return on Security Investment, da European Network and Information Security Agency

Para saber mais...

… leia o documento Introduction to Return on Security Investment, da European Network and Information Security Agency

FIM