SEGURANÇA EM REDES E COMPUTADORES | 21181

Data e hora de realização

12 de fevereiro de 2021, às 15h00 de Portugal Continental

Duração da prova

90 minutos, a que acrescem 60 minutos de tolerância.

Conteúdos

- Introdução à Segurança Informática
- Noções básicas de Criptografia
- Segurança em redes

Objetivos

- Identificar conceitos e consolidar o conhecimento sobre a segurança informática.
- Conhecer o processo da criptografia e explicar os diferentes tipos de métodos criptográficos.
- Identificar as principais ameaças à segurança de redes informáticas e conhecer os mecanismos de defesa contra as mesmas.

Trabalho a desenvolver

Leia atentamente as seguintes questões e procure responder com o máximo possível de detalhe, explicando de forma detalhada todos os passos do seu raciocínio. O enquadramento teórico para a sua resposta será importante para compreender a mesma.

1. O conceito de CIA Triad não representa os requisitos completos de segurança para dados e serviços. Indique quais são os dois conceitos adicionais necessários adicionar. (2 valores)

A tríade representada pela sigla CIA refere os 3 pilares da segurança informática e que são: C – Confidencialidade, I – Integridade e A – *Availability* (Disponibilidade). A confidencialidade está relacionada com os princípios que garantem que a informação trocada entre dois pontos, mesmo se obtida por quem não tem autorização para tal, que continua a manter-se segura, portanto incompreensível para quem a intercepta. A integridade garante que o conteúdo do que se transmite não é modificado/destruído. A disponibilidade garante que a informação pode ser acedida no exato momento em que é necessária.

Muitos autores defendem que podem ser acrescentados 2 conceitos adicionais para se ter uma definição mais completa e geral dos objectivos de um sistema de segurança. Esses dois conceitos são Autenticidade (*Authenticity*) e Responsabilidade (*Accountability*). A autenticidade refere-se à propriedade de ser genuíno e capaz de ser verificado. Ou seja, permite assegurar que a mensagem provém da origem de onde é suposto provir (autenticidade da fonte) e que o seu conteúdo não foi alterado durante a transmissão (autenticidade dos dados). Note-se que a alteração de conteúdo pode ser feita, sem comprometer a integridade da mensagem. A responsabilidade ou *accountability* permite determinar responsabilidade de uma entidade pelas suas ações. Tap propriedade garante a existência de um registo dos eventos de atividades/operações num sistema (*logs*) permitindo

posteriormente determinar quem fez que ação sobre que objeto, quando e a partir de onde.

2. Explique como funciona, de forma genérica, um modelo de cifra assimétrica. (3 valores)

No modelo de cifra assimétrica, ou cifra por chave-pública, são utilizadas duas chaves separadas no processo de encriptação e decriptação da mensagem e que se denominam por chave pública e chave privada. No caso da chave pública, esta é de conhecimento tanto do emissor como do recetor da mensagem. Já a chave privada, como o próprio nome indica, só é conhecida pelo seu proprietário. Note-se que não é possível determinar a chave privada, mesmo conhecendo a chave pública, apesar de matematicamente relacionadas.

O modelo de cifra assimétrica funciona da seguinte forma:

- Cada utilizador gera duas chaves.
- Depois coloca a chave pública num registo público, onde são mantidas chaves públicas de múltiplos utilizadores. Associada a esta chave estará a chave privada que, como mencionado, apenas o próprio utilizador conhece.

Exemplo: consideremos os utilizadores Alice e Bob. Se a Alice quiser enviar uma mensagem a Bob, terá de começar por obter a chave pública de Bob caso não a tenha já e cifrar a mensagem original (ou *plaintext*) com essa chave pública. Quando o Bob recebe a mensagem cifrada com a sua chave pública (ou *ciphertext*), consegue decifrá-la com a sua chave privada. Ou seja, apenas o Bob consegue decifrar a mensagem, porque é o único que conhece a sua chave privada, que faz com a sua chave pública.

O algoritmo assimétrico tão mais robusto, quanto maior for o comprimento das chaves utilizadas. E o algoritmo utilizado irá também ajudar a tornar ainda mais robusto todo o processo, sendo que atualmente o considerado mais seguro é o denominado por curvas elípticas.

Este processo de cifra impõe muito maior carga de processamento, porque ao trabalhar com chaves de grande comprimento (ex.: 1024 bits) necessita de muito trabalho de processador quer para a cifra, quer para decifrar as mensagens, pela quantidade de operações lógicas que terão de ser executadas. A adicionar a isto, temos ainda a carga de processamento necessário para a geração das próprias chaves. Estes algoritmos realizam-se pela utilização de funções matemáticas *trapdoor one-way*, ou seja, funções nas quais é computacionalmente fácil obter a mensagem original a partir da cifra se se possuir a chave necessária; no entanto será computacionalmente irrealizável determinar a mensagem original a partir da cifra sem se possuir essa chave. Além disso, como já referimos, será também computacionalmente irrealizável a obtenção da chave privada de um utilizador a partir da sua chave pública.

3. Explique o que é um *message digest*, como se pode criar e as diferentes finalidades de utilização. (2 valores)

Message digest (ou *hash*) é o resultado que se obtém ao aplicar uma função de *hash* criptográfica a uma mensagem ou conjunto de dados.

Uma função hash H é uma função que aceita como input uma mensagem M de dimensão variável e produz como output um valor h de tamanho fixo (ex.: 256 bits) que é designado de código *hash* ou *message digest*: h = H(M).

As funções de *hash* são chamadas de "one-way function", já que para as mesmas não existe função inversa. Desta forma, na posse do *hash* resultante não se conseguem obter os dados originais. De resto, o *hash* será, normalmente, mais pequeno que os dados iniciais, pelo que seria impossível recuperar os dados.

As funções hash têm diversas aplicações na área de segurança informática, entre as quais podemos salientar: (1) a autenticação de mensagens; 2) produção de assinaturas digitais; 3) criação de ficheiros de palavras-passe seguros; 4) detecção de vírus ou intrusões nos ficheiros de um sistema; e, 5) geração de números pseudoaleatórios.

O *message digest* pode ser visto como uma assinatura única gerada a partir dos dados originais, pelo que, se os dados forem alterados, o cálculo de uma nova assinatura, utilizando a mesma função de *hash* resultará numa assinatura totalmente diferente. Desta forma, esta abordagem é utilizada primariamente para verificação de integridade de dados.

Alguns exemplos de algoritmos de *message digest* são: o MD5 e o SHA. O MD5 produz um *digest* de 128 bits (16 carateres) e o SHA1 produz apenas 20 carateres. Para evitar colisões (i.e. o mesmo resultado para dados diferentes) aumentou-se o número de caracteres do *hash/digest*. Assim surgiram o SHA-2 e SHA-3, que produzem *digests* de 64 e 512, respetivamente, no máximo.

Um exemplo muito concreto e frequente de utilização deste tipo de algoritmos é na garantia da autenticidade das mensagens. Desta forma, um determinado utilizador utiliza uma função hash H para produzir um *message diges*t h da mensagem M. O valor h tem que ser enviado de forma segura de tal forma a que se um atacante conseguir interceptar a mensagem e alterar o seu conteúdo, deve ser incapaz de conseguir alterar o valor hash da mensagem para que seja

possível o destinatário detetar a existência de violação da integridade do conteúdo da mensagem. Desta forma, o *message digest* deve ser encriptado com um esquema de cifra, p.e. simétrica, utilizando uma chave secreta K_s e posteriormente concatenado com a mensagem M (enviando-se então Y = M||E(K_s, h)), ou, se se pretender também confidencialidade na transmissão da mensagem, pode-se encriptar o conjunto M || h (enviando-se Y = E(M||h)).

Quando recebe a mensagem, caso esta venha encriptada, o destinatário utiliza a chave secreta partilhada K_s para desencriptar e obter o conjunto M e o *message digest* h. Caso apenas o valor hash (*message digest*) seja encriptado, é utilizada a chave secreta K_s para o desencriptar. Posteriormente, o destinatário aplica a mesma função *hash* que o remetente usou sobre a mensagem que recebeu e calcula um valor h'. Se o novo valor *hash* calculado for diferente do *message digest* recebido, o destinatário sabe que a houve um problema na transmissão que levou à alteração do conteúdo da mensagem. Para que o ataque passe despercebido, o atacante teria de ser capaz de alterar o conteúdo da mensagem e gerar um novo *message digest* encriptando-o com a chave secreta K_s. No entanto como só o emissor e o destinatário conhecem a chave secreta K_s, qualquer alteração na mensagem irá fazer com que o *message digest* calculado no destino seja diferente daquele que é enviado em conjunto com a mensagem.

4. Indique quais as quatro abordagens de contramedidas de *malware*, bem como detalhe do modo de implementação preventiva das mesmas. (3 valores)

As contramedidas para prevenir ataques com *malware* passam por 4 diferentes aspetos, nomeadamente: (1) políticas de segurança; (2) consciencialização dos utilizadores para as ameaças pela via da formação; (3) mitigação de vulnerabilidades do sistema e (4) mitigação de ameaças.

A existência de uma política de segurança na organização é a base para implementar outras medidas. Por exemplo, uma empresa deve assegurar que todos os computadores que se ligam na sua rede tenham os respetivos sistemas operativos na sua versão mais atualizada, na medida em que grande parte do software malicioso aproveita-se de vulnerabilidades a esse nível para "infetar" um sistema. Por isso, a implementação de uma política que vise a constante atualização dos sistemas operativos ajudará na segurança do sistema. Outro exemplo que se pode dar é a nível de controlos de acesso adequados, permitindo apenas que o administrador do sistema (ou grupo de administradores) tenha acesso a certos ficheiros, de forma a assegurar a manutenção das propriedades de segurança do sistema (CIA).

A consciencialização dos utilizadores pela via da formação sobre as ameaças, ajuda a prevenir ataques por *malware* que utiliza mecanismos de propagação baseados em engenharia social (esquemas de *phishing*, emails de spam, *trojans*). Desta forma, estando atentos às implicações de determinadas ações , estarão menos propensos a executá-las, minimizando o risco de comprometimento do sistema.

Contudo, sabemos que a prevenção pode falhar. Um utilizador pode, inadvertidamente, utilizar uma *pen drive* e infetar (com ou sem intenção) um computador e, a partir daí, toda a rede. Por isso, é necessário recorrer a mecanismos que limitem ou mitiguem as ameaças. Esses mecanismos deverão ser capazes de: (a)

detetar a infeção, localizando-a; (b) identificar o tipo de *malware* que infetou o sistema; (c) remover o *malware* identificado, eliminando-o completamente do sistema para que o mesmo não se continue a propagar.

A detecção do *malware* no sistema, pode ser feita a diferentes níveis: (i) A nível do *host*; (ii) a nível da rede; (iii) a nível distribuído.

Um sistema de deteção que esteja no *host* é o que tem o maior grau de informação pois consegue ver como é que o *malware* interage com a sua aplicação alvo. Estes sistemas podem ser por exemplo: software antivírus; software *behavior-blocking*; software *anti-spyware* e contramedidas para *rootkits*, muitas das quais se baseiam na detecção de padrões de código ou de comportamento típicos dos *rootkits* e que estão frequentemente incluídas em softwares antivírus.

Um sistema de detecção a nível da rede (ou ao nível do perímetro), por outro lado, dá informação acerca do *malware* que circula na rede, podendo estar por exemplo localizado numa firewall externa, num IDS (*Intrusion Detection System*) ou num *honeypot* da organização. Estes sistemas podem monitorizar tráfego que vem do exterior da rede para o interior ou tráfego que vai do interior da rede para o exterior. Neste último caso, pode ser possível detetar processos de *scanning/fingerprinting* que podem indicar a existência de um *worm* ou de um *bot* no interior da rede.

Um sistema a nível distribuído consegue recolher dados a partir de sistemas de mitigação de *malware* localizados no *host* e em diferentes pontos da rede, transferindo esses dados para uma instância central que os irá analisar e pode, posteriormente, indicar padrões de comportamento e assinaturas do *malware*. Dependendo da complexidade do sistema, essa instância central pode ainda ser capaz de testar o *malware* encontrado num ambiente seguro e, conforme os resultados obtidos, produzir *patches* de segurança e

fazer o *deploy* desses *patches* para as máquinas da rede de forma a protege-las contra futuros ataques. Desta forma, uma abordagem distribuída fornece uma imagem mais abrangente da evolução das diferentes ameaças e pode ajudar a contribuir para desenvolver novas estratégias de segurança.

Se, por alguma razão, não for possível eliminar o *malware*, uma alternativa é destruir os ficheiros infetados e substituir os mesmos com versões anteriores de *backup*, à partida num estado de integridade anterior à infeção (esta é normalmente a opção em casos de ataque tipo *ransomware*). Portanto, manter *backups* de ficheiros importantes atualizados e em localizações separadas, constitui um mecanismo muito importante na segurança da informação e dos sistemas.

5. Explique em que consiste a engenharia social, enquanto mecanismo de comprometimento de medidas de segurança informática e quais as contramedidas que podem ser acionadas. (2 valores)

No contexto da segurança informática, a engenharia social é usada por atacantes para obter acesso a um sistema ou informações que podem levar a esse mesmo acesso. Existem várias técnicas de engenharia social, que geralmente envolvem um perpetrador fazendo-se passar por uma pessoa que está num nível mais alto na hierarquia organizacional do que a vítima. Para se preparar para essa falsa representação, o perpetrador já pode ter usado táticas de engenharia social contra outras pessoas na organização para coletar informações aparentemente não relacionadas que, quando usadas em conjunto, tornam a falsa representação mais credível. Por exemplo, qualquer um pode verificar o site de uma empresa ou até mesmo ligar para a receção principal para obter o nome do CIO; um invasor pode obter

ainda mais informações ligando para outras pessoas na empresa e afirmando falsamente sua autoridade ao mencionar o nome do CIO. Ataques de engenharia social podem envolver pessoas fazendo-se passar por novos funcionários ou como funcionários atuais, solicitando assistência. Às vezes, os invasores ameaçam, persuadem ou imploram para influenciar o alvo.

Portanto, engenharia social, no âmbito de comprometimento de medidas de segurança informática, refere-se a uma tática de "enganar" utilizadores, para que os mesmos participem no comprometimento das suas próprias informações confidenciais. Por exemplo, quando um utilizador executa um *Cavalo de Troia*, que vem anexado a um email *spam*. Normalmente, este tipo de *malware* é anexado a emails que parecem vir de alguém conhecido/confiável. Por este, e outros motivos, é que a engenharia social ainda é um grande problema na segurança de redes de computadores, na medida em que nós, seres humanos, tendemos a confiar nos outros, principalmente, se forem pessoas conhecidas ou estranhos a personificar as mesmas.

Recorrendo às técnicas de engenharia social, um atacante pode promover a propagação de *malware*, por exemplo convencendo um utilizador a executar *software* malicioso ou a executar determinadas ações (p. e. ir para determinada página web) que irão comprometer a segurança do sistema e dos dados. Estas técnicas incluem, por exemplo, o envio de emails de spam tentando convencer o utilizador a fazer download de um anexo que pode esconder um cavalo-de-tróia (*trojan*), *worm* ou vírus. Esse software malicioso, quando executado pode exercer diversas ações danosas: (1) aceder a dados sensíveis do utilizador (por exemplo a ficheiros de password); (2) apagar ou modificar ficheiros ou causar danos físicos ao sistema; (3) instalar software *spyware* ou software de *keylogging* que vai monitorizar a atividade do utilizador ou monitorizar as teclas que o utilizador pressiona, permitindo roubar por exemplo credenciais de acesso a

sistemas bancários ou roubar dados sobre a identidade do utilizador ; ou, (4) abrir uma *backdoor* que permite ao atacante ter um canal de acesso ao sistema do utilizador sem ter que passar pelos mecanismos de controlo de acesso normais e que pode ser o primeiro passo para transformar o utilizador num *bot* comandado pelo atacante.

Alternativamente, os emails podem convencer o utilizador a carregar um determinado URL que o vai direcionar para uma página web falsa, como por exemplo a página de uma entidade bancária, e quando o utilizador insere as suas credenciais de acesso, estas são roubadas pelo atacante (esquemas de *phishing*).

O principal mecanismo de prevenção de ataques baseados em engenharia social passa por educar os utilizadores acerca dos perigos a que estão sujeitos quando acedem a uma rede não segura como é a Internet. Desta forma, é importante que as organizações promovam ações de formação para que os seus empregados, utilizadores do sistema computacional, possam obter conhecimentos básicos de segurança, e.g. aprender a como manter e manipular a informação confidencial segura, a reconhecer e não abrir emails spam, ou utilizar *passwords* seguras.

Além do treino que os utilizadores devem receber, é preciso educá-los, de forma quase constante para que estejam conscientes da responsabilidade e das consequências das suas ações.

Se, mesmo assim, o ataque ocorrer é importante que o utilizador tenha o sistema atualizado com todos os *patches* de segurança instalados e que possua, também, determinados softwares protectores que consigam detectar a entrada de um *malware* no sistema e removê-lo (ex: software antivírus, software *anti-spyware*). Esse software é capaz de fazer o scan do sistema e procurar por determinados padrões de bits (assinaturas) frequentemente presentes em malware ou por outro lado, interagir com o sistema operativo para bloquear ações suspeitas

e típicas de software malicioso. Além disso, é importante estabelecer mecanismos de controlo de acesso que garantam que apenas determinados processos têm acesso a ficheiros com informação sensível presentes no sistema do utilizador (o que pode, por exemplo, limitar a propagação de vírus no sistema).

Critérios de avaliação e cotação

Na avaliação do trabalho serão tidos em consideração os seguintes critérios e cotações:

1. 2 valores
2. 3 valores
3. 2 valores
4. 3 valores
5. 2 valores

Total: 12 valores

Normas a respeitar

Deve redigir o seu E-fólio Global na Folha de Resolução disponibilizada na turma e preencher todos os dados do cabeçalho. Em todo e qualquer caso, só será aceite para correção o seu E-fólio Global com respostas digitadas em processador de texto (por exemplo: MS-Word), com a exceção de algum desenho realizado à mão relacionado com a resposta. Neste caso, pode incorporá-lo como uma imagem na folha de resolução.

Todas as páginas do documento devem ser numeradas.

O seu E-fólio Global não tem limite de páginas A4 redigidas em Verdana, tamanho de letra 12. O espaçamento entre linhas deve corresponder a 1,5 linhas.

Nomeie o ficheiro com o seu número de estudante, seguido da identificação do E-fólio Global, segundo o exemplo apresentado: 000000efolioA.

Deve carregar o referido ficheiro para a plataforma no dispositivo E-fólio Global até à data e hora limite de entrega. Evite a entrega próximo da hora limite para se precaver contra eventuais problemas.

O ficheiro a enviar não deve exceder 8 MB.

Votos de bom trabalho!

Henrique S. Mamede