Segurança no Desenvolvimento, Implantação e Operação de Sistemas de Informação Baseado na ISO 15408

Palestrante:	Alexandre Sieira, CISSP
Autores:	Alexandre Correia Pinto, CISSP
	Alexandre Sieira, CISSP
Local:	GTS 02.2003 - Praia Vermelha, RJ
Data:	03/12/2003

Agenda

- Introdução
- Apresentação da Common Criteria
  - Escopo
  - Áreas Não Abordadas
  - Contexto de Segurança
  - Requerimentos de Segurança
- Estudo de caso: Segurança em PHP
- Conclusões

Introdução

- Organizações utilizam-se de sistemas desenvolvidos internamente ou por terceiros para uma variedade de objetivos.
- Estes sistemas podem ser críticos, ou estar intimamente ligados a sistemas críticos, à operação da organização.
- Estes sistemas muitas vezes estão expostos externamente para parceiros e clientes, como *extranets* ou serviços Internet.
- Não é usual a incorporação de requisitos de segurança na funcionalidade e no processo de desenvolvimento, por um dos seguintes motivos:

  - Falta de cultura de Segurança da Informação.
  - Falta de preparo da equipe de desenvolvimento.
  - Escassez de recursos que leva a uma priorização dos requisitos funcionais.

Introdução

A questão do elo mais fraco:

Segurança de rede não basta se as aplicações tiverem vulnerabilidades.

Apresentação da Common Criteria

CIPHER
SEGURANÇA DA INFORMAÇÃO

Escopo

- Define critérios para a avaliação de segurança de produtos e sistemas de TI.
- Permite a comparação de avaliações independentes de diferentes produtos ou sistemas (EAL1-EAL7).
- Auxilia análise de risco e tomada de decisão de consumidores.
- Guia o desenvolvimento de produtos e sistemas seguros.
- É aplicável a medidas de segurança em hardware, firmware ou software.
- Abrange, entre outros:
    - Sistemas operacionais
    - Redes
    - Sistemas distribuídos
    - Aplicações

Apresentação da Common Criteria

CIPHER
SEGURANÇA DA INFORMAÇÃO

Escopo

- Analisa a preservação de confidencialidade, integridade e disponibilidade da informação.
- Foco em ameaças originadas por seres humanos, maliciosas ou não.
- Analisa o *design* e processo de desenvolvimento, além de informações coletadas no processo de avaliação.

Apresentação da Common Criteria

CIPHER
SEGURANÇA DA INFORMAÇÃO

Áreas Não Abordadas

- Medidas administrativas de segurança do ambiente e usuários do produto ou solução.
- Controle de emissões eletromagnéticas.
- Metodologia de avaliação de produtos e sistemas (abordado no CEM – Common Evaluation Methodology).
- Critérios de avaliação de algoritmos criptográficos.

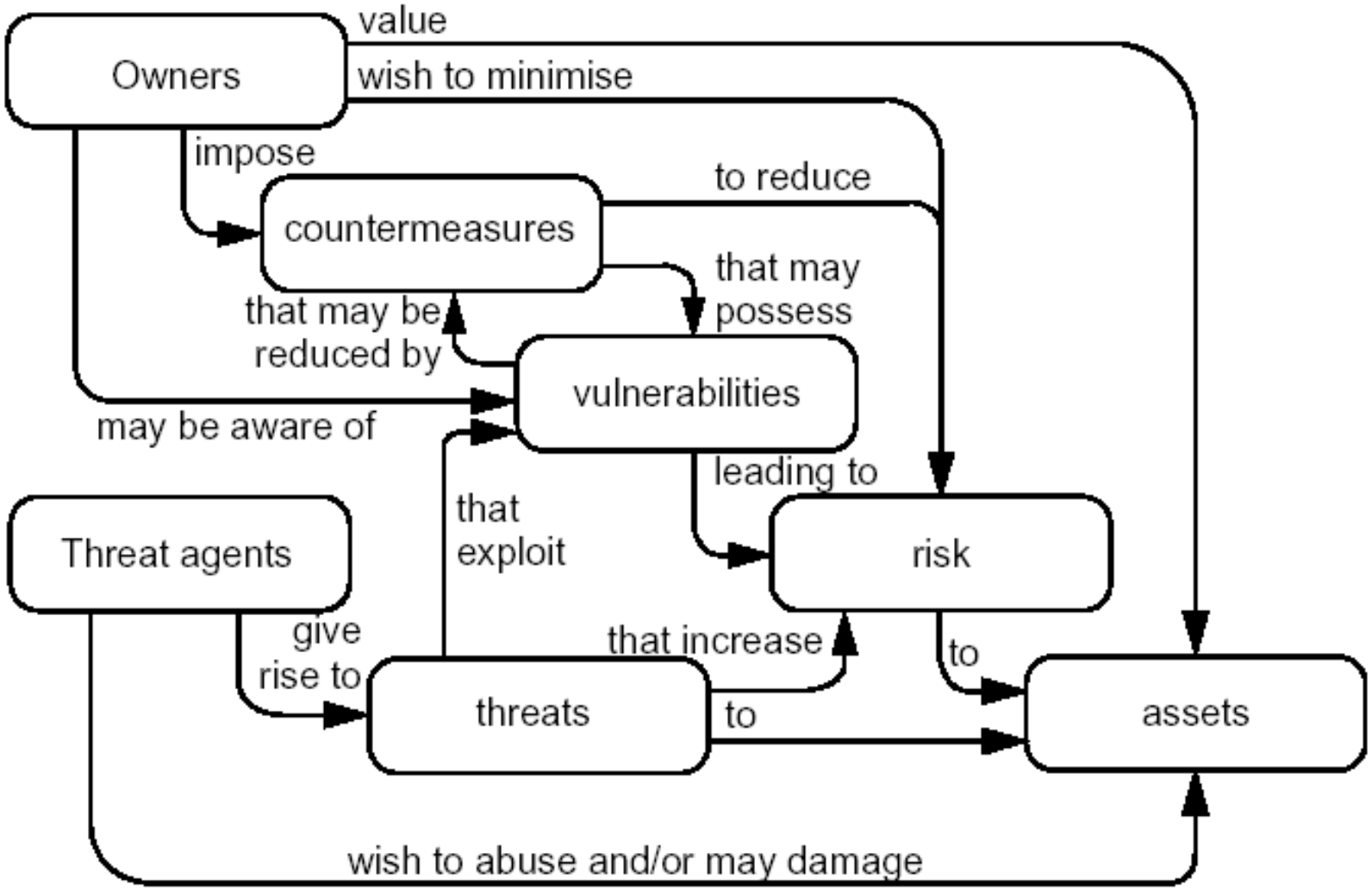
Apresentação da Common Criteria

CIPHER
SEGURANÇA DA INFORMAÇÃO

Contexto de Segurança

- O objetivo da segurança é proteger os **ativos** (*assets*) das **ameaças** (*threats*), que são usos indevidos em potencial dos mesmos.
- Os **proprietários** (*owners*) dos **ativos** associam às **ameaças** presentes em seu ambiente os **riscos** (*risks*) correspondentes.
- **Contramedidas** (*countermeasures*) são instaladas para anular **vulnerabilidades** (*vulnerabilities*) e mitigar **riscos**.

Apresentação da Common Criteria

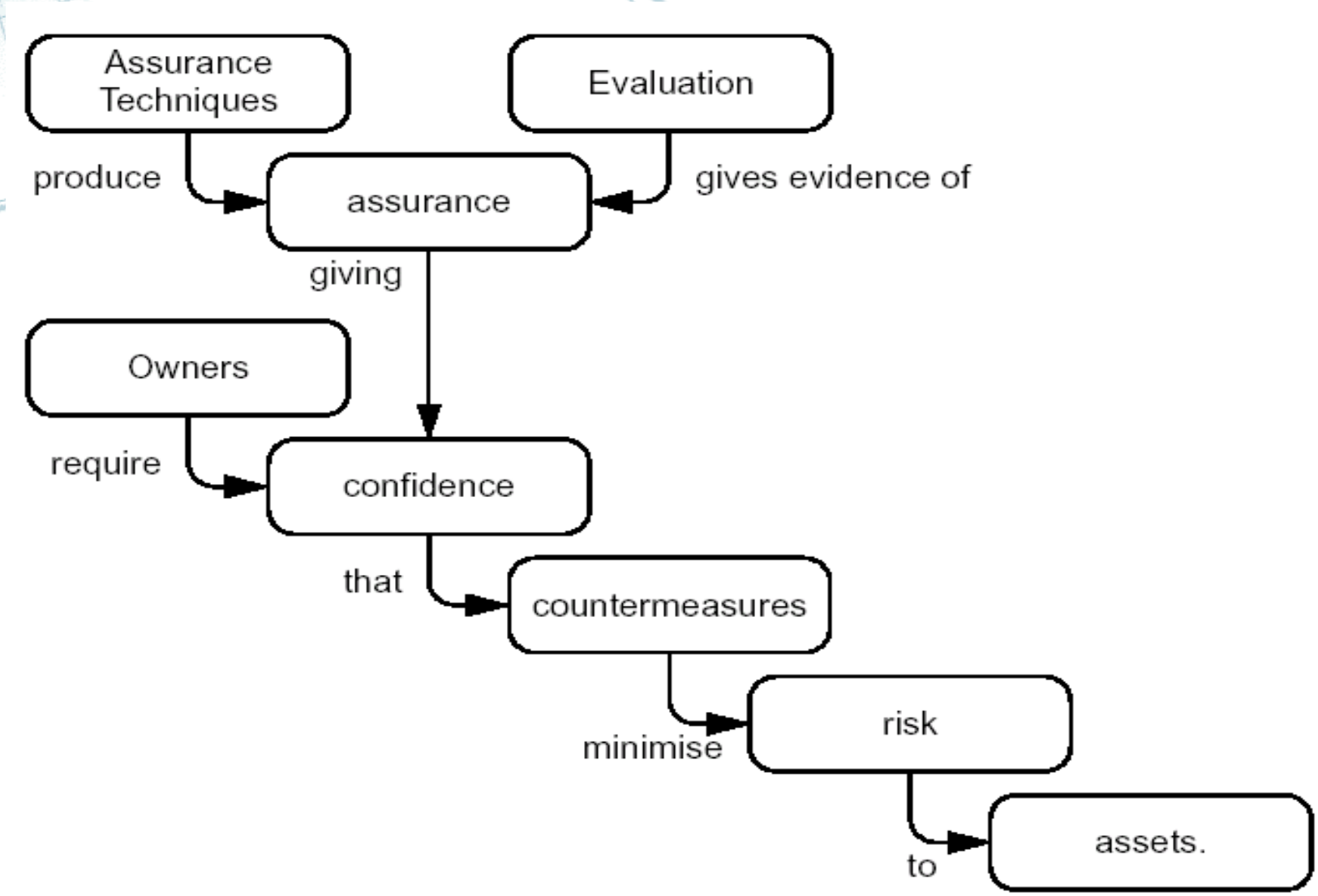


Apresentação da Common Criteria

Contexto de Segurança

- Os **proprietários** de **ativos** devem ter **confiança** (*confidence*) na eficácia das **contramedidas**, mas podem não possuir a habilidade de avaliá-la.
- Assim, os **proprietários** utilizam-se de uma **avaliação** (*evaluation*), que gera uma declaração do nível de **validação** (*assurance*) de sua capacidade de mitigar **riscos**.

Apresentação da Common Criteria



Apresentação da Common Criteria

CIPHER
SEGURANÇA DA INFORMAÇÃO

Requerimentos de Segurança

- São agrupados em famílias, que por sua vez são agrupadas em classes.
- Se dividem em dois tipos:
  - Requerimentos Funcionais de Segurança
  - Requerimentos de Validação de Segurança

Apresentação da Common Criteria

Requerimentos de Segurança

- Classes de Requerimentos Funcionais de Segurança:

  - Auditoria de segurança (FAU) – validação da funcionalidade de geração de trilha de auditoria de eventos relevantes do ponto de vista de segurança;

  - Comunicação (FCO) – validação dos controles aplicados à comunicação entre o sistema e outras entidades através de não repúdio de recebimento e envio;

  - Suporte criptográfico (FCS) – validação dos controles aplicados ao suporte criptográfico, como gerenciamento de chaves e escolha dos algoritmos;

  - Proteção a dados do usuário (FDP) – validação dos controles de proteção aos dados do usuário na importação, exportação, armazenamento ou comunicação interna entre componentes do sistema;

Apresentação da Common Criteria

Requerimentos de Segurança

- Classes de Requerimentos Funcionais de Segurança:

  - Identificação e autenticação (FIA) – validação dos controles aplicados à correta identificação dos usuários do sistema, bem como sua associação a perfis de permissões;

  - Gerenciamento de segurança (FMT) – validação dos controles aplicados ao gerenciamento de dados de segurança como perfis de permissões, atributos de segurança e outros dados internos;

  - Privacidade (FPR) – validação dos controles utilizados para impedir que usuários do sistema comprometam a segurança dos dados uns dos outros;

  - Proteção do sistema (FPT) – validação dos controles utilizados para garantir a integridade dos dados e processamento do sistema diretamente associados à sua segurança;

Apresentação da Common Criteria

Requerimentos de Segurança

- Classes de Requerimentos Funcionais de Segurança:

  - Utilização de recursos (FRU) – validação dos controles utilizados para garantir o uso devido de recursos como memória, armazenamento e banda, com a devida compartimentalização e priorização de sua alocação;

  - Acesso ao sistema (FTA) – validação dos controles aplicados ao estabelecimento, gerenciamento e encerramento de sessões entre o usuário e o sistema;

  - Trusted Paths (FTP) – validação dos controles aplicados à criação de canais de comunicação confiáveis (não-repúdio) entre usuários e o sistema, e entre este e outras entidades com as quais exista um relacionamento de confiança.

Apresentação da Common Criteria

Requerimentos de Segurança

- Classes de Requerimentos de Validação de Segurança:

  - Gerenciamento de Configuração (ACM) – manutenção da integridade do sistema do controle e dos processos de modificação do mesmo;

  - Entrega e Operação (ADO) – manutenção da segurança do sistema durante entrega, instalação, inicialização e operação;

  - Desenvolvimento (ADV) – provisões de segurança na especificação, *design* e implementação do sistema.

  - Documentação (AGD) – cobre a facilidade de compreensão, abrangência e completude da documentação operacional do sistema para usuários e administradores;

Apresentação da Common Criteria

Requerimentos de Segurança

- Classes de Requerimentos de Validação de Segurança:

  - Suporte ao Ciclo de Vida (ALC) – que inclui a segurança do ambiente, processos e ferramentas utilizadas para o desenvolvimento e manutenção do sistema;

  - Testes (ATE) – analisa os testes sistemáticos utilizados para validar a aderência do sistema aos seus requisitos funcionais de segurança;

  - Análise de Vulnerabilidades (AVA) – que cobre a identificação de vulnerabilidades exploráveis no sistema, como *covert channels*;

  - Manutenção da Validação (AMA) – que cobre a validação continuada de segurança após a avaliação inicial, quando da alteração do sistema.

Estudo de Caso – Segurança em PHP

- Ponto importante sobre sistemas web:

Sistemas web são equivalentes a *daemons*, e devem ser desenvolvidos com critérios estritos de Segurança da Informação.

Estudo de Caso – Segurança em PHP

CIPHER
SEGURANÇA DA INFORMAÇÃO

- PHP é uma linguagem simples de usar e rica em funcionalidades;
- PHP é insegura por *default*:

  - Funções de execução de scripts e abertura de arquivos aceitam URLs remotas;

  - Parâmetros de origem externa se transformam em variáveis globais automaticamente;

  - É possível acessar todos os arquivos permitidos ao usuário do processo do servidor web;

  - Uploads de arquivo são permitidos em todas as páginas;

  - Expõe informações sensíveis quando ocorre um erro de execução;

  - Identificadores de sessão são armazenados no /tmp, que é legível a todos os usuários;

Estudo de Caso – Segurança em PHP

Autenticação em Sistema Web Típica (Insegura)

```
query = "select id from usuarios where nome = '$nome' and senha = '$senha'";
$stmt = oci_parse( $conn, $query );
oci_exec( $stmt );
if (oci_fetchinto( $stmt, $result, OCI_ASSOC ) == FALSE) {
            (...) // volta usuário para tela de login com mensagem de erro
} else {
            // grava identidade do usuário em cookie
            setcookie( "user_id", $result[0] );
            (...) // redireciona usuário para tela de entrada
}
```

Estudo de Caso – Segurança em PHP

Desabilitando variáveis globais automáticas para dados externos com *register_globals*:

```
// importa para variáveis globais apenas os dados esperados
$nome = $_POST[“nome”]; $senha = $_POST[“get”];
query = “select id from usuarios where nome = ‘$nome’ and senha = ‘$senha’”;
$stmt = oci_parse( $conn, $query );
oci_exec( $stmt );
if (oci_fetchinto( $stmt, $result, OCI_ASSOC ) == FALSE) {
            // volta usuário para tela de login com mensagem de erro
            (...)
} else {
            // grava identidade do usuário em cookie
            setcookie( “user_id”, $result[0]);
            (...) // redireciona usuário para tela de entrada
}
```

Estudo de Caso – Segurança em PHP

Criando trilha de auditoria (FAU):

```
$nome = $_POST[“nome”]; $senha = $_POST[“get”];

query = “select id from usuarios where nome = ‘$nome’ and senha = ‘$senha’”;

$stmt = oci_parse( $conn, $query );

oci_exec( $stmt );

$quando = date("Y/m/d H:i:s");

if (oci_fetchinto( $stmt, $result, OCI_ASSOC ) == FALSE) {

            // registra falha de autenticação

            syslog(LOG_WARNING, “$quando Acesso negado: $nome $_SERVER[‘REMOTE_ADDR’] ($_SERVER
['HTTP_USER_AGENT´])");

            // volta usuário para tela de login com mensagem de erro

            (...)

} else {

            // registra login de usuário

            syslog(LOG_NOTICE, “$quando Acesso ok: $nome $_SERVER[´REMOTE_ADDR´] ($_SERVER
['HTTP_USER_AGENT’])");

            // grava identidade do usuário em cookie

            setcookie( “user_id”, $result[0]);

            (...) // redireciona usuário para tela de entrada

}
```

Estudo de Caso – Segurança em PHP

Protegendo mecanismo de autenticação contra SQL Injection:

```
$nome = $_POST[“nome”]; $senha = $_POST[“get”];

$query = “select id from usuarios where nome = :nome and senha = :senha”;

$stmt = oci_parse( $conn, $query );

ocibindbyname( $stmt, “:nome”, $nome );

ocibindbyname( $stmt, “:senha”, $senha );

oci_exec( $stmt );

$quando = date("Y/m/d H:i:s");

if (oci_fetchinto( $stmt, $result, OCI_ASSOC ) == FALSE) {

            syslog(LOG_WARNING, “$quando Acesso negado: $nome $_SERVER[‘REMOTE_ADDR’] ($_SERVER
[‘HTTP_USER_AGENT´])");

            // volta usuário para tela de login com mensagem de erro

            (...)

} else {

            syslog(LOG_NOTICE, “$quando Acesso ok: $nome $_SERVER[´REMOTE_ADDR´] ($_SERVER
[‘HTTP_USER_AGENT’])");

            // grava identidade do usuário em cookie

            setcookie( “user_id”, $result[0]);

            (...) // redireciona usuário para tela de entrada

}
```

Estudo de Caso – Segurança em PHP

Protegendo a associação da sessão ao usuário autenticado:

```
(...)
} else {
            syslog(LOG_NOTICE, "$quando Acesso ok: $nome $_SERVER[´REMOTE_ADDR´] ($_SERVER
['HTTP_USER_AGENT'])");
            // começa nova sessão, destruindo eventuais sessões pré-existentes
            session_destroy();
            session_unset();
            session_start();
            // armazena identidade do usuário em variável de sessão, junto a dados para validar
            // origem de futuras requisições e evitar session hijacking
            $_SESSION['user_id'] = $result[0];
            $_SESSION['remote_addr'] = $_SERVER['REMOTE_ADDR'];
            $_SESSION['user_agent'] = $_SERVER['HTTP_USER_AGENT'];
            (...) // redireciona usuário para tela de entrada
}
```

Estudo de Caso – Segurança em PHP

Protegendo a associação da sessão ao usuário autenticado:

- As páginas de autenticação, bem como todas as restritas a usuários autenticados, devem ser acessíveis apenas através de SSL;
- Os arquivos que armazenam as variáveis de sessão no servidor devem ser movidos para diretório com permissões restritas usando a opção *session.save_path*;
- A opção *session.use_only_cookies* deve ser usada para que o identificador de sessão seja armazenado apenas em *cookies;*
- A opção *session.cookie_secure* deve ser usada de forma que o *cookie* com o identificador de sessão trafegue apenas em conexões seguras;
- Todas as páginas restritas a usuários autenticados devem acessar as variáveis de sessão e garantir que o endereço de origem e a identificação do *browser* existem e são as mesmas da requisição atual, de forma a detectar usuários não autenticados e *session hijacking*.

Conclusões

CIPHER
SEGURANÇA DA INFORMAÇÃO

- Segurança de rede não basta se as aplicações tiverem vulnerabilidades.

- A ISO 15408 – Common Criteria é a referência internacional para desenvolvedores, administradores e auditores de segurança de sistemas de informação;

- Sistemas web são equivalentes a *daemons*, e devem ser desenvolvidos com critérios estritos de Segurança da Informação.

Contato

Cipher

Rua Marquês de São Vicente, 225 - Ed. Gênesis

Gávea - 22451-041

Rio de Janeiro - RJ

Alexandre Sieira, CISSP

Tel: 21 2529-2629

E-mail: alexandre.sieira@ciphersec.com.br

URL: www.ciphersec.com.br