Conceitos

António Casimiro
Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa

1

Sumário

- Propriedades de segurança
- Falhas de segurança e confiabilidade
- Ataques
  - Hacker
    - Motivações de hackers
    - Percurso de um hacker
  - Tipos de ataques
    - Ataques passivos
    - Ataques activos
  - Ataques mais relevantes
- Vulnerabilidades
- Defesa contra falhas de segurança
- Riscos
- Sumário de mecanismos de segurança

2

O que se deve garantir para se ter “segurança”?

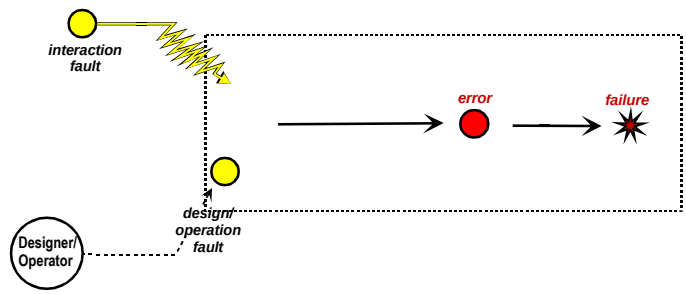
Propriedades de segurança:

- **Confidencialidade**
  - *Confidencialidade de dados*: proteção contra acesso a dados guardados num sistema
  - *Privacidade*: assegurar controlo na informação que é recolhida relativa a indivíduos
- **Integridade**
  - *Integridade de dados*: proteção contra alteração de informação/dados
  - *Integridade do sistema*: assegurar que o sistema executa a sua função
- **Disponibilidade**
  - Proteção contra recusa de provisão/acesso a dados ou sistemas
    - utilização exagerada ou abusiva de recursos
    - vandalismo
- **Autenticidade**
  - Proteção contra personificação
- **Prestação de contas (Accountability)**
  - Associar as ações de uma entidade a essa entidade de forma unívoca

3

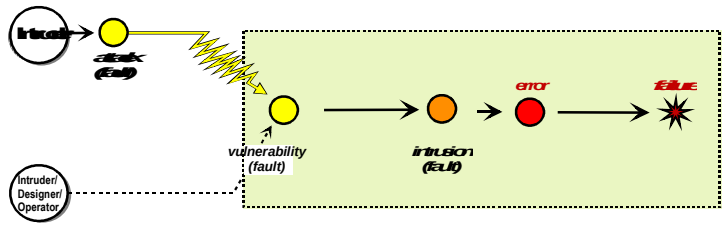
Visão sistemática da **falha de confiabilidade** um sistema

Sequência: falta → erro → falha



4

Visão sistemática de uma **falha de segurança** num sistema

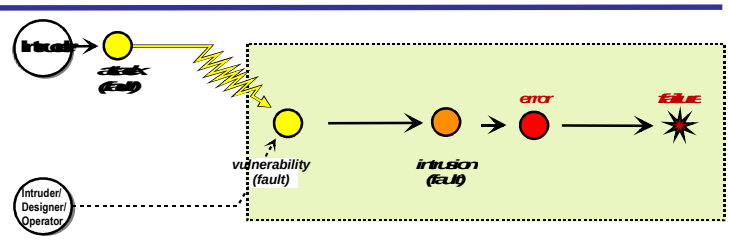


Sequência **AVI** : ***A**taque* + ***V**ulnerabilidade* → ***I**ntrusão*
intrusão → erro → falha

 5

5

Ataque + *Vulnerabilidade* → *Intrusão*



- Ataque
  - Falta intencional maliciosa introduzida no sistema com a intenção de explorar vulnerabilidades
- Vulnerabilidade
  - Fraqueza do sistema que o torna sensível a ataques
  - Normalmente não maliciosa
  - Sem ataques, as vulnerabilidades são inofensivas
  - Sem vulnerabilidades não há ataques bem sucedidos
- Intrusão
  - Falta operacional induzida por meio externo e intencionalmente maliciosa que provoca um estado erróneo no sistema
  - Pode ou não causar uma falha de segurança

São capazes de se lembrarem de exemplos de vulnerabilidades?

 6

6

ATAQUES

7

Motivações dos hackers

- curiosidade
- coleccionar troféus
- acesso grátis a recursos computacionais e de comunicação
- ponte para outras máquinas num sistema distribuído
- efectuar danos e sabotagem em sistemas por razões criminais ou políticas
- obter informações confidenciais para uso particular ou venda, como segredos de software, comerciais, industriais ou informação pedagógica
  - Lucro
- E ainda: cyber-guerra

8

Percurso de um Hacker (1/3)

- Reconhecimento e descoberta de potenciais vulnerabilidades
  - conhecer o alvo
    - que computadores estão acessíveis (e.g., scanning)
    - pessoas relevantes e os seus endereços de email
    - olhar para informação pública sobre o alvo
  - conhecer os sistemas que são usados pelo alvo e como podem ser atacados
  - iniciar procura de pontos fracos
    - contas sem *password* ou com *default password;* configurações vulneráveis
- Acesso inicial ao sistema
  - fazer um plano de ataque
    - vulnerabilidades de software descritas nas bases de dados públicas
    - testar palavras de um dicionário com entradas no ficheiro de *passwords*
    - ataques direcionados por email (spearfishing)
  - formas mais comuns de acesso: email / web site / dispositivo de memória

Percurso de um Hacker (2/3)

- Controlar o sistema e estabelecer persistência
  - controlar todos os recursos do sistema, através da obtenção de privilégios de administrador
    - explorando vulnerabilidades de programas instalados que usem permissões de *root*
    - utilizando *shell scripts* com *suid* para *root*
    - usando cavalos de Tróia previamente instalados
- Apagar os seus rastros
  - esconder a sua actividade durante a campanha de intrusão
    - disfarçando-se enquanto está a aceder o sistema
    - apagando os *logs* do sistema após sair

Percurso de um Hacker (3/3)

- Instalar ferramentas
  - que asseguram/facilitam acessos futuros
    - cavalos de Tróia ou *backdoors* que podem ser activados por códigos ou sequências especiais
  - que permitem atacar outros sistemas na rede
- Comprometer outros sistemas da organização
  - procurar "troféus" e outros sistemas que permitam lá chegar
    - através da abertura de canais para outras máquinas
    - procurar por informações de acesso em ficheiros pessoais
    - escutar a partir da máquina invadida
- Obter informação e explorar alvo
  - retirar a informação secreta ….
  - alterar os dados ou apagá-los

Alguns princípios iniciais para proteger os sistemas

- *Know your system!*
  - se não conhecer o sistema (e.g., dispositivos que estão ligados e o software que está a correr) é impossível garantir proteção contra todas as vulnerabilidades
- *Continuous defensive work!*
  - assegurar e testar que no sistema apenas está presente o que é suposto e que tudo está atualizado e configurado sob o ponto de vista de segurança

- Diretor da NSA TAO (Tailored Access Operations)

https://youtu.be/bDJb8WOJYdA

Classes de Ataques

- *Ataques passivos*: tenta-se obter informação existente no sistema sem afetar os seus recursos
- *Ataques ativos*: tenta-se alterar o funcionamento correto do sistema

- *Ataques externos*: realizados por entidades fora do perímetro de segurança, por um utilizador não autorizado
- *Ataques internos*: realizados por uma entidade dentro do perímetro de segurança, possivelmente com alguns privilégios

Ameaças: Consequências e Ações (ou Ataques)

Threat Consequence	Threat Action (Attack)
Unauthorized Disclosure A circumstance or event whereby an entity gains access to data for which the entity is not authorized.	**Exposure:** Sensitive data are directly released to an unauthorized entity. **Interception:** An unauthorized entity directly accesses sensitive data traveling between authorized sources and destinations. **Inference:** A threat action whereby an unauthorized entity indirectly accesses sensitive data (but not necessarily the data contained in the communication) by reasoning from characteristics or by-products of communications. **Intrusion:** An unauthorized entity gains access to sensitive data by circumventing a system's security protections.
Deception A circumstance or event that may result in an authorized entity receiving false data and believing it to be true.	**Masquerade:** An unauthorized entity gains access to a system or performs a malicious act by posing as an authorized entity. **Falsification:** False data deceive an authorized entity. **Repudiation:** An entity deceives another by falsely denying responsibility for an act.
Disruption A circumstance or event that interrupts or prevents the correct operation of system services and functions.	**Incapacitation:** Prevents or interrupts system operation by disabling a system component. **Corruption:** Undesirably alters system operation by adversely modifying system functions or data. **Obstruction:** A threat action that interrupts delivery of system services by hindering system operation.
Usurpation A circumstance or event that results in control of system services or functions by an unauthorized entity.	**Misappropriation:** An entity assumes unauthorized logical or physical control of a system resource. **Misuse:** Causes a system component to perform a function or service that is detrimental to system security.

TPC: Estudar no livro

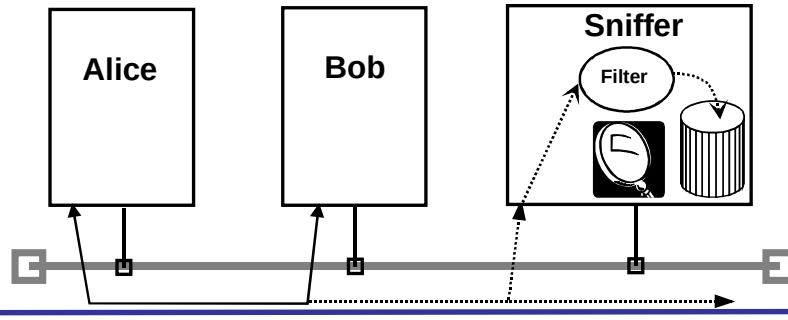
Ataques Passivos

- Ataques passivos
  - Não requer uma acção explicita contra os mecanismos de protecção ou a integridade dos dados, focando-se na confidencialidade ☹
- Exemplos:
  - Escutar *(sniffing):* sondas passivas apenas escutam o tráfego numa rede com o objetivo de o ler
  - Análise de tráfego *(traffic analysis):* escuta-se o tráfego, e embora não se consiga ler, obtém-se informação sobre o que está a ser enviado
  - Vasculhar *(snooping):* vasculhar o interior de sistemas e repositórios de dados em busca de informação relativa a *passwords*, configurações, etc.
  - Sondar *(probing):* sondas pesquisam sistemas em busca de informações e vulnerabilidades (ex., *portscan*, *doorknob rattling*)
- Observação: ataques passivos são menos "destrutivos" que ataques activos, no entanto são muito mais difíceis de serem detectados

15

Sniffing

- Modo de operação:
  - O adaptador de rede local da máquina do *sniffer* é configurado no modo promíscuo (aceita todos as frames *Ethernet*)
  - A informação recebida é filtrada
  - O que for de interesse é armazenado em disco, para uso posterior
- Observação: é muito difícil detectar um *sniffer*, já que se trata de um ataque completamente passivo



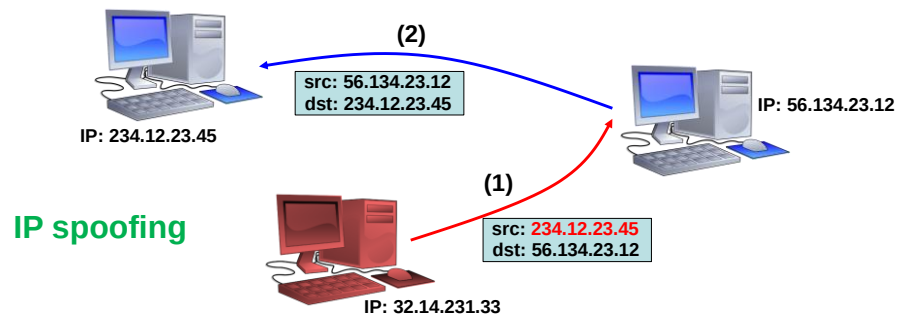
16

Ataques Activos

- Ataques activos
  - Tentativas agressivas de entrar no sistema, para corromper a sua operação e/ou roubar, modificar ou mesmo destruir dados
- Exemplos:
  - Personificação (autenticidade ☹)
    - Endereços de mail, endereços IP, … (*spoofing*)
  - Interposição e alteração (integridade ☹)
    - Alteração: inserir, apagar, repetir, atrasar
  - Negação de serviço (disponibilidade ☹)
- Ataques podem ser executados através de software malicioso
  - Vírus
  - *Worms*
  - Bombas lógicas
  - Cavalos de Tróia
  - Zombies / Bots

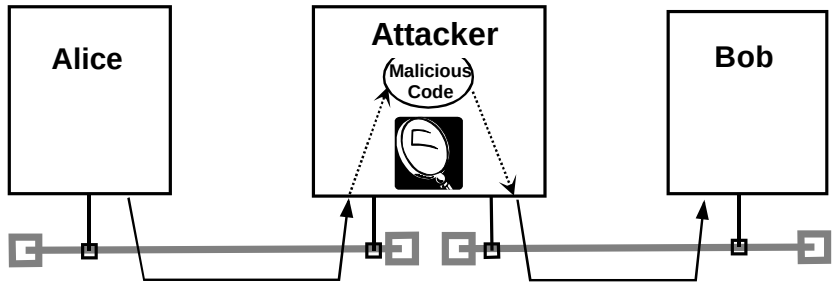
Spoofing

- Situação em que uma pessoa ou sistema personifica outra entidade, podendo atuar em seu nome, conseguindo por exemplo falsificar dados
- Spoofing aparece frequentemente ligado a email, IP, MAC



Homen-no-Meio *(Man-in-the-Middle)*

- Modo de operação:
  - Uma máquina maliciosa intercepta a comunicação entre dois participantes, depois lê e/ou muda o seu conteúdo dinamicamente
- Alguns exemplos:
  - Inserção/remoção de dados ou reenvio (*replay*) de mensagens inteiras
  - Modificação em tempo de execução do conteúdo da mensagem
  - Adição de código malicioso a mensagens



Vírus de Computador

- Programas que se inserem dentro de um ou mais ficheiros e executam alguma ação maliciosa
  - em geral propaga-se por outros ficheiro e depois inicia a execução
- Pedaço de código **auto-replicável** com algum outro código (usualmente malicioso) associado
  - Carrega código para fazer cópias de si mesmo
  - E também código para executar alguma tarefa "nefasta"
- Ciclo de vida de um vírus:
  - *adormecido* – à espera pelo evento de ativação
  - *propagação* – replicação para novos ficheiros
  - *execução* – do código malicioso, causado por algum evento

Worm de Computador

- Programa que se **copia de um computador para outro**
- Replica-se espalhando-se pela rede -> consome recursos, mas tipicamente não infecta ficheiros
- Ciclo de vida de um *worm*:
  - *adormecido* – à espera pelo evento de activação
  - *propagação* – replicação para novos sistemas
  - *execução* – do código malicioso
- Observação: *worms* replicam-se por diferentes máquinas enquanto vírus replicam-se por ficheiros numa mesma máquina

Bomba Lógica (Logic Bomb)

- Programa que executa uma ação que impõe uma falha de segurança no sistema **quando um evento externo ocorre**
  - Ex.: programa que apaga a base de dados de uma organização quando alguma condição é satisfeita, ex., uma data de aniversário
- O código das bombas lógicas é geralmente embutido em programas legítimos
- Tipicamente causam danos ao sistema
- Condições de ativação podem variar:
  - ação em uma base de dados, hora e data, a receção de uma mensagem, *login* de um utilizador, etc.
- Observação: algumas vezes as bombas lógicas vem dentro de vírus (já ouviram falar “sexta-feira 13”?)

Cavalo de Tróia (Trojan)

- Programa com um objectivo aberto (conhecido pelo utilizador) e um outro objectivo escondido (desconhecido pelo utilizador)
- Exemplo: cavalo de Tróia *login*
  - Propósito aberto: permite o acesso de um utilizador, aceitando seu *login* e *password*
  - Propósito escondido: armazenar *passwords* num ficheiro escondido para uso posterior
- Basicamente o cavalo de Tróia é um programa que aparentemente é atractivo mas que esconde funcionalidades escondidas

- Observação: usualmente são usados para propagar vírus e *worms*, instalar uma *backdoor* ou apenas destruir dados

Zombie / Bot

- Programa que secretamente tem controle de um computador ligado à rede
  - Exemplo: através da exploração de uma vulnerabilidade no sistema
- Permanece adormecido até ser activado
  - Exemplo: por uma mensagem ou pedido de conexão remota
- A partir daí usa a máquina vítima para realizar tarefas de interesse de seu controlador:
  - Lançar ataques indirectos
  - Enviar *spam*

- Observação: colecções de zombies conhecidas como ***BotNets*** têm sido usadas para lançar ataques de negação de serviço distribuídos (*DDoS*)
  - Estima-se que há mais de **1.000.000 de computadores** em *BotNets.*

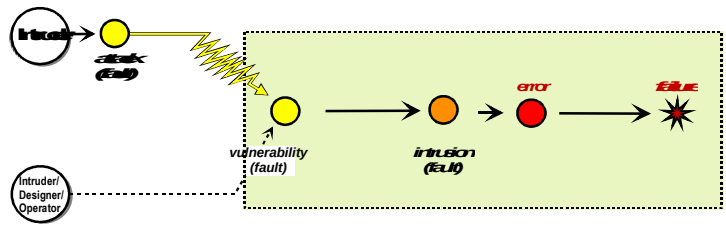
Ataques mais relevantes

- Mais informação por exemplo
  - Symantec's Global Intelligence Network
    http://www.symantec.com/security_response/publications/threatreport.jsp
- Ataques mais relevantes
  - Negação de serviço
  - Código malicioso
    - botnets / worms
    - dispositivos móveis
    - ransomware
  - …
- Alvos
  - Cyber-guerra
  - Infraestruturas críticas
- Democratização dos ataques
  - Scripty kid

VULNERABILIDADES E MECANISMOS DE PROTECÇÃO

Vulnerabilidades

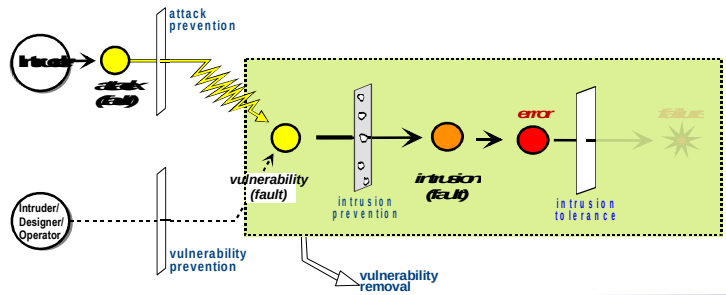
- Vulnerabilidades
  - Deficiências técnicas
  - Atitude das pessoas



Defesa contra Falhas de Segurança

- Prevenção
  - Diminuir/remover vulnerabilidades
  - Impedir o ataque com sucesso
- Deteção : Sistemas de deteção de intrusões
- Recuperação
  - Minimizar os riscos decorrentes de ataques bem sucedidos
  - Reposição do estado antes do ataque
  - Tolerância a intrusões

Segurança por ocultação ☹



Risco da intrusão

- Risco: métrica composta que leva em consideração o nível de **ameaça** (do ataque) que um sistema esta exposto, e o seu grau de **vulnerabilidade** e o **impacto** financeiro do ataque

Probabilidade de Ataque com Sucesso = *Ameaça x Vulnerabilid.*
RISCO = *Probabilidade de Ataque com Sucesso x Impacto*

- A medida correcta de quão potencialmente inseguro um sistema é (ou, quão difícil é torná-lo seguro) depende de:
  - O número e a severidade das falhas do sistema *(vulnerabilidades)*
  - As potenciais ameaças a que ele pode ser submetido *(ataques)*
- *Custo vs benefício*

31

(Alguns) Mecanismos de segurança

- Criptografia
- Mecanismos de confinamento
  - Sandbox
  - Firewalls
  - Zonas desmilitarizadas (DMZ)
- Mecanismos de controlo de acesso
- Mecanismos de execução privilegiada
  - Setuid
- Mecanismos de filtragem
  - firewall
- Mecanismos de inspecção
  - Sistemas de detecção de intrusões
- Mecanismos de auditoria

32

Alguns Princípios de Desenho

- *Desenho aberto (Open design)*
  - não depender de segurança por obscuridade
  - Motivação: mais tarde ou mais cedo o desenho é divulgado
- *Economia do mecanismo (Economy of the mechanism)*
  - usar medidas de segurança simples para assegurar a correção
  - Motivação: mais fácil de implementar, validar e testar
- *Menor mecanismo comum (Least common mechanism)*
  - os utilizadores devem partilhar o menor número de mecanismos (e.g., para interagirem)
  - Motivação: simplificar a validação destes mecanismos
- *Por omissão usar modo seguro (Fail-safe default)*
  - dar permissões em vez de definir exclusões
  - por omissão, o acesso deve ser negado

33

Alguns Princípios de Desenho (cont.)

- *Interposição (Complete mediation)*
  - todos os acessos devem ser verificados face à política de segurança
- *Separação de privilégios (Separation of privilege)*
  - dividir os privilégios em várias partes e
    - requerer vários desses privilégios para executar uma ação (e.g., autenticação)
    - as tarefas específicas necessitam apenas de um subconjunto desses privilégios
- *Privilégio mínimo (Least privilege)*
  - cada processo/utilizador deve ter atribuídos os menores privilégios possíveis mas que ainda lhes permite realizar as suas tarefas
- *Usabilidade (Psychological acceptability)*
  - os mecanismos de segurança não devem interferir (ou devem minimizar a interferência) com as tarefas a realizar pelos utilizadores
- *Menor espanto (Least astonishment)*
  - o programa/mecanismo funciona da maneira como o utilizador esperaria

34

Alguns Princípios de Desenho (cont.)

- *Isolamento (Isolation)*
  1. serviços públicos devem estar isolados (?fisicamente?) dos privados;
  2. processos/ficheiros de diferentes utilizadores estão isolados;
  3. os mecanismos de segurança estão isolados dos utilizadores
- *Encapsulamento (Encapsulation)*
  - caso específico do isolamento para programação orientada a objetos
- *Modularidade (Modularity)*
  - separar módulos que implementam mecanismos de segurança
  - o sistema deve ser dividido em módulos diferentes, permitindo a sua alteração/substituição sem afetar os restantes
- *Camadas de proteção (Layering ou Defense in depth)*
  - usar várias camadas de proteção, em que se uma delas for quebrada, não coloca a segurança do sistema em causa

35

Bibliografia

- Stallings 2014
  - Cap 1, 6 e 8.1

36