Operating Systems: Internals and Design Principles

Capítulo 15 Segurança em Sistemas Operacionais

Ninth Edition
By William Stallings

Ameaças ao Sistema

Invasores

Masquerader

Um indivíduo que não é autorizado usar o computador e invadiu o sistema de controle de acesso para explorar uma conta de um usuário legítimo.

Misfeasor

Um usuário legítimo que acessa recursos que não possui direito de acesso, ou usuário com direito de acesso que abusa do uso dos recursos.

Clandestine user

Um indivíduo que obtém acesso do controle de supervisor do sistema e aproveita-se para evadir mecanismos de auditoria e controles de acessos.

Software Maliciosos

- Programas que exploram vulnerabilidades em sistemas de computacionais.
- Também denominados de **malware**.
- Podem ser divididos em duas categorias:
  - Parasíticos
    - Fragmentos de programas que não podem existir independentemente de algum programa.
    - Exemplos: vírus, bombas lógicas e backdoors.
  - Independentes
    - Programas independentes que podem ser escalonados e executados pelo SO.
    - Exemplos: worms and bots.

Contramedidas

- RFC 4949 (*Internet Security Glossary)* define detecção de intrusão como um serviço de segurança que monitora e analisa eventos do sistema para o propósito de encontrar e prover alertas em tempo real ou mais rápido possível contra acessos dos recursos do sistema de forma não autorizada.
- Sistemas de Detecção de Intrusão (IDSs) são classificados:
  - Host-based IDS
    - Monitoram eventos e atividades suspeitas em um único host.
  - Network-based IDS
    - Monitoram tráfego de rede para um segmento ou dispositivos de rede, e analisa o tráfego para identificar atividades suspeitas.

Componentes IDS

Autenticação

- Autenticação de usuários é um bloco fundamental e a primeira linha de defesa de muitos sistemas.
- RFC 4949 define autenticação de usuário como o processo de verificar uma identidade afirmada pelo usuário ou por uma entidade do sistema.
- O processo de autenticação consiste de dois passos:
  - Identificação
    - Apresentar um identificador para o sistema de segurança.
  - Verificação
    - Apresentar ou gerar uma informação de autenticação que verifica a associação entre a entidade e o identificador.

Formas de Autenticação

- Algo que o indivíduo sabe
  - Exemplos: senha, número pessoal de identificação (PIN), ou respostas para um conjunto de questões.
- Algo que o indivíduo possui
  - Exemplos: cartão eletrônico, chaves físicas.
  - Referenciado com ***token***
- Algo que o indivíduo é (biometria)
  - Exemplos: impressão digital, retina, face.
- Algo que o indivíduo faz (biometria dinâmica)
  - Exemplos: padrão de voz, características de escrita, ritmo de digitação.

Controle de Acesso

- Implementar uma política de segurança que especifica quem ou o que tem acesso a cada recurso do sistema e o tipo de acesso que é permitido em cada instância.
- Atua entre um usuário e os recursos do sistema, tais como aplicações, SO, firewalls, roteadores, arquivos e banco de dados.
- Um administrador de segurança mantém uma base de autorização que especifica quais os tipos de acessso são permitidos para um usuário
  - O controle de acesso consulta a base para determinar o direito de acesso.
- Uma função de auditoria monitora e registra os acessos do usuário aos recursos do sistema.

Firewalls

Objetivos de projeto:

1) Todo tráfego, de dentro para fora e vice-versa, deve passar pelo firewall.

2) Somente tráfego autorizado, especificado por uma política de segurança local, deve ser permitido passar pelo firewall.

3) O firewall deve ser imune a invasões.

Ataque de Buffer Overflow

- Também conhecido como ***buffer overrun***
- Definido pelo NIST (National Institute of Standards and Technology) como:

> "Uma condição em uma interface que mais informações d entrada podem ser colocadas em um buffer ou área de dados que a capacidade permitida, sobrescrevendo outros informações. Atacantes exploram tais condições para derrubar um sistema ou inserir código que possibilite ganhar controle do sistema."

- Um dos mais comuns e perigosos tipos de ataques.

```
int main(int argc, char *argv[]) {
    int valid = FALSE;
    char str1[8];
    char str2[8];

    next_tag(str1);
    gets(str2);
    if (strncmp(str1, str2, 8) == 0)
        valid = TRUE;
    printf("buffer1: str1(%s), str2(%s), valid(%d)\n", str1, str2, valid);
}
```

(a) Basic buffer overflow C code

```
$ cc -g -o buffer1 buffer1.c
$ ./buffer1
START
buffer1: str1(START), str2(START), valid(1)
$ ./buffer1
EVILINPUTVALUE
buffer1: str1(TVALUE), str2(EVILINPUTVALUE), valid(0)
$ ./buffer1
BADINPUTBADINPUT
buffer1: str1(BADINPUT), str2(BADINPUTBADINPUT), valid(1)
```

(b) Basic buffer overflow example runs

Figure 15.1 Basic Buffer Overflow Example

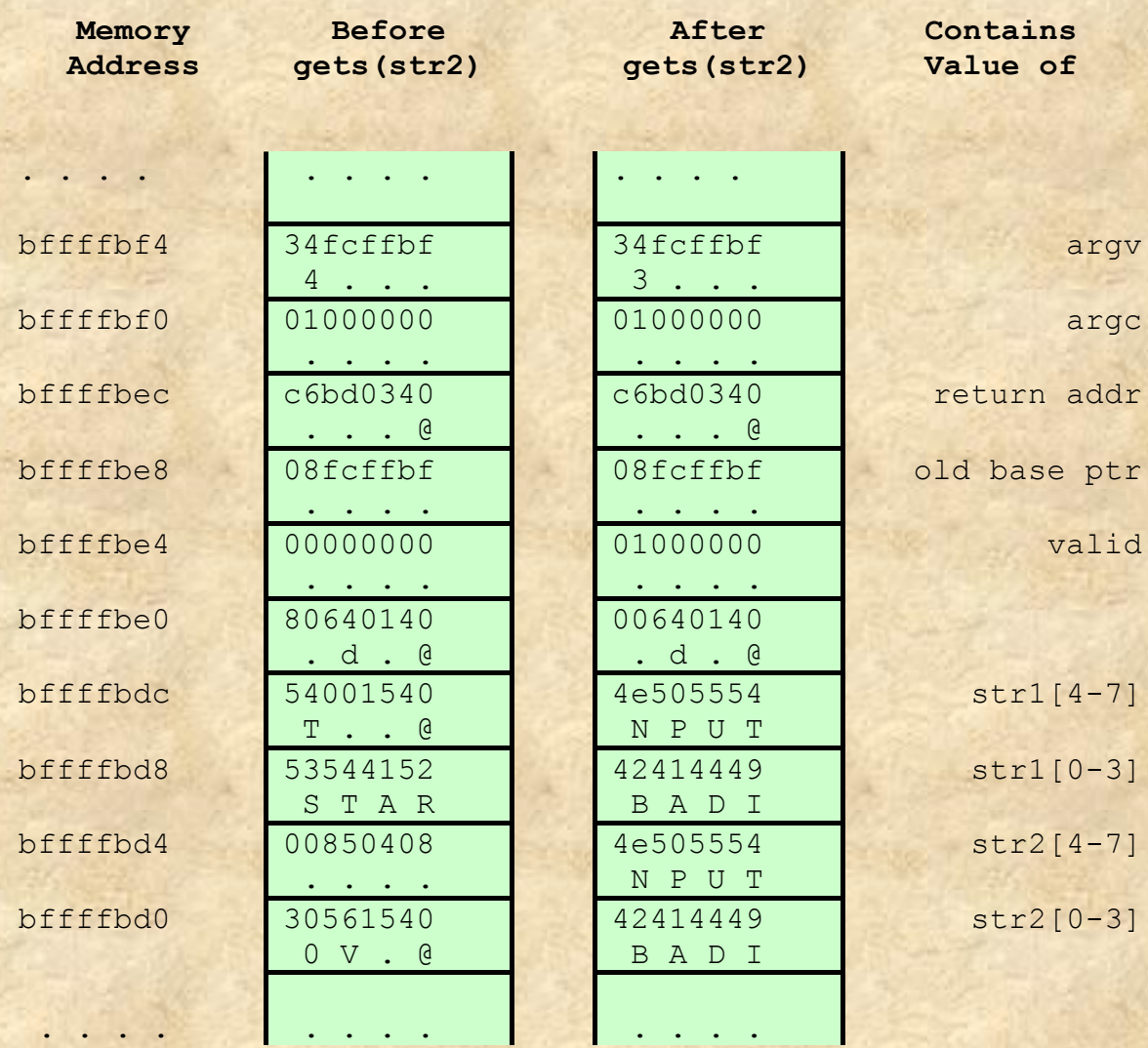


Figure 15.2 Basic Buffer Overflow Stack Values

Explorando Buffer Overflow

- Para explorar qualquer tipo de buffer overflow o atacante precisa:
  - Identificar um vulnerabilidade de buffer overflow em algum programa e que pode ser explorado usando dados de origem externa.
  - Entender como o buffer armanezará na memória, para corromper memória adjacente e, potencialmente alterar o fluxo da execução do programa.

Defesas

- Contramedidas podem ser classificadas em duas categorias:

1) Defesas em tempo de compilação, que objetivam "blindar" programas para resistir a ataques.

2) Defesas em tempo de execução, que objetivam detectar e abortar ataques em programas que estão executando.

Técnicas em Tempo de Compilação

- **Escolha da linguagem de programação**
  - Selecionar linguagens modernas com ênfase na proteção de tipos de variáveis.
  - Flexibilidade e segurança implicam em custos de desempenho.
- **Técnicas de codificação segura**
  - Inspecionar o código e reescrever código não seguro.
  - Exemplo: OpenBSD é um SO construído pensando em segurança.
  - Programadores devem auditar seus códigos antigos, SOs, bibliotecas e utilitários comuns constantemente.
- **Extensões de linguagem e uso de bibliotecas seguras**
  - Propostas de compiladores que inserem pontos de referência.
  - Exemplo: Libsafe inclui checagens para garantir que as operações de cópia não extendam o espaço na pilha.
- **Mecanismos de proteção de pilha**
  - Checar a pilha por evidência de corrupção.
  - Stackguard, extensão do GCC que insere código adicional na entrada e saída de funções.

Técnicas em Tempo de Execução

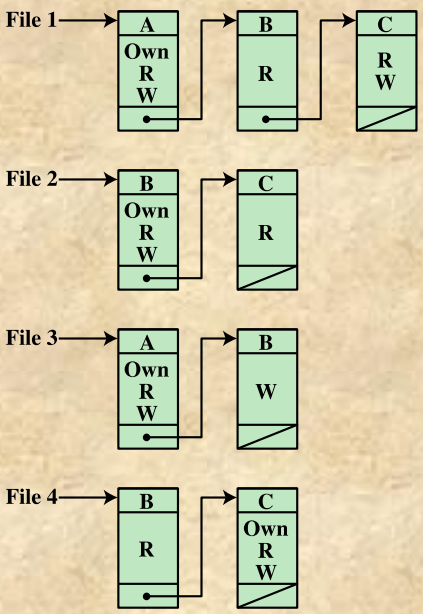
- **Proteção do espaço de execução**
  - Bloquear a execução de código na pilha.
  - Extensões para diversos SO.
- **Aleatorização do espaço de endereçamento**
  - Evita que atacantes identifiquem a região de memória de estruturas-chave dos processos alvos.
  - Usar extensões que mudam aleatoriamente a ordem de carregamento de bibliotecas padrões por programas e localizações dos endereços de sua memória virtual.
- **Guard pages**
  - Inserir guard pages (espaços ilegais) entre faixas de endereçamento.
  - As guard pages são setadas como endereços ilegais pela MMU.
  - Poderiam ser colocadas guard pages entre a pilha ou diferentes alocações na heap.

Controle de Acesso de Sistemas de Arquivos

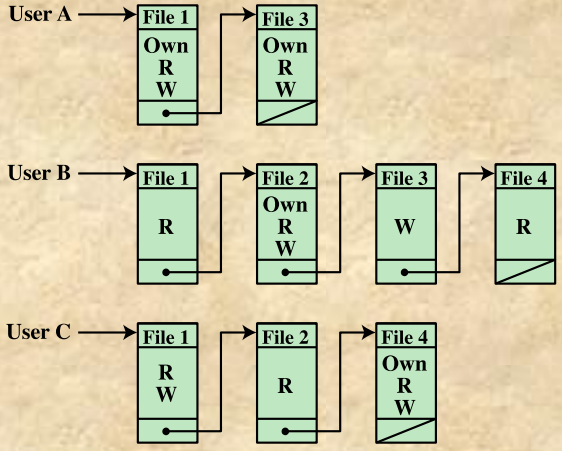
- Identifica um usuário no sistema.
- Associado com cada usuário um perfil (profile) que especifica operações de permissão e acessos de arquivos.
- SO pode impôr regras baseadas no perfil do usuário.

	File 1	File 2	File 3	File 4	Account 1	Account 2
User A	Own R W		Own R W		Inquiry Credit	
User B	R	Own R W	W	R	Inquiry Debit	Inquiry Credit
User C	R W	R		Own R W		Inquiry Debit

(a) Access matrix



(b) Access control lists for files of part (a)



(c) Capability lists for files of part (a)

Figure 15.3 Example of Access Control Structures

Políticas de Controle de Acesso

- Uma política de controle de acesso especifica quais os tipos de acesso são permitidos, em determinadas circunstâncias, e por quem
- Políticas de controle de acesso são geralmente agrupadas em :
  - Discretionary access control (DAC)
    - Controla o acesso baseado na identidade do requerente e nas regras de acesso que especificam o que é permitido fazer.
  - Mandatory access control (MAC)
    - Controla o acesso baseado na comparação de rótulos de segurança com permissões de segurança.
  - Role-based access control (RBAC)
    - Controla o acesso baseado em papéis que usuários tem dentro do sistema, e em regras que especificam quais são os acessos permitidos para dado papel.
  - Attribute-based access control (ABAC)
    - Controla o acesso baseado em atributos do usuário, do recurso a ser acessado, e condições atuais do ambiente.

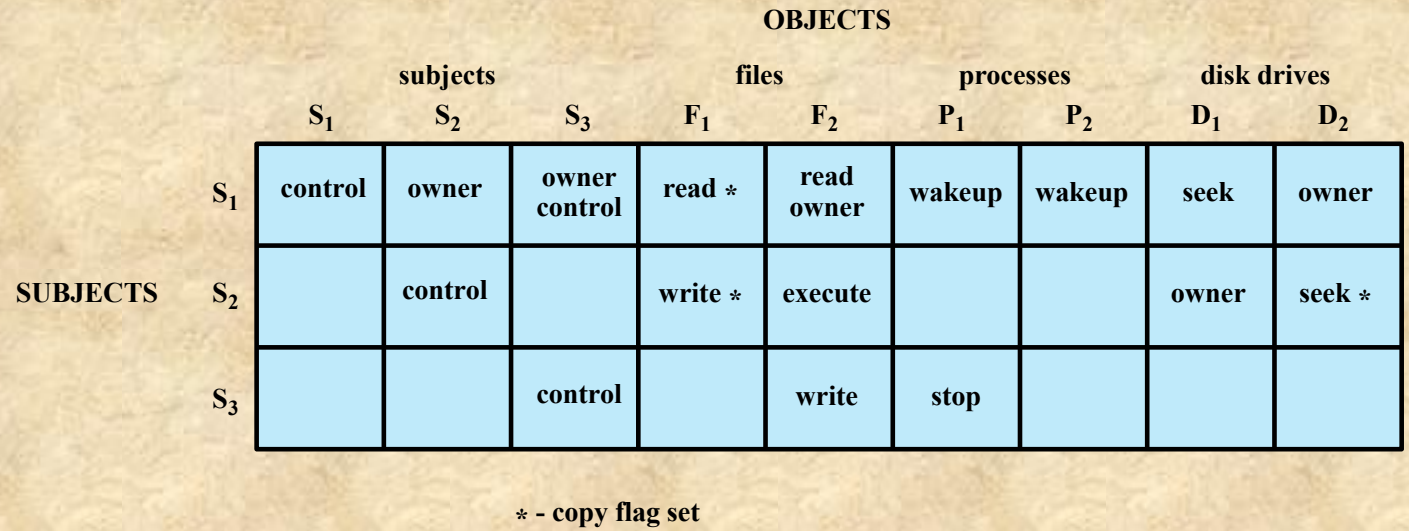


Figure 15.4 Extended Access Control Matrix

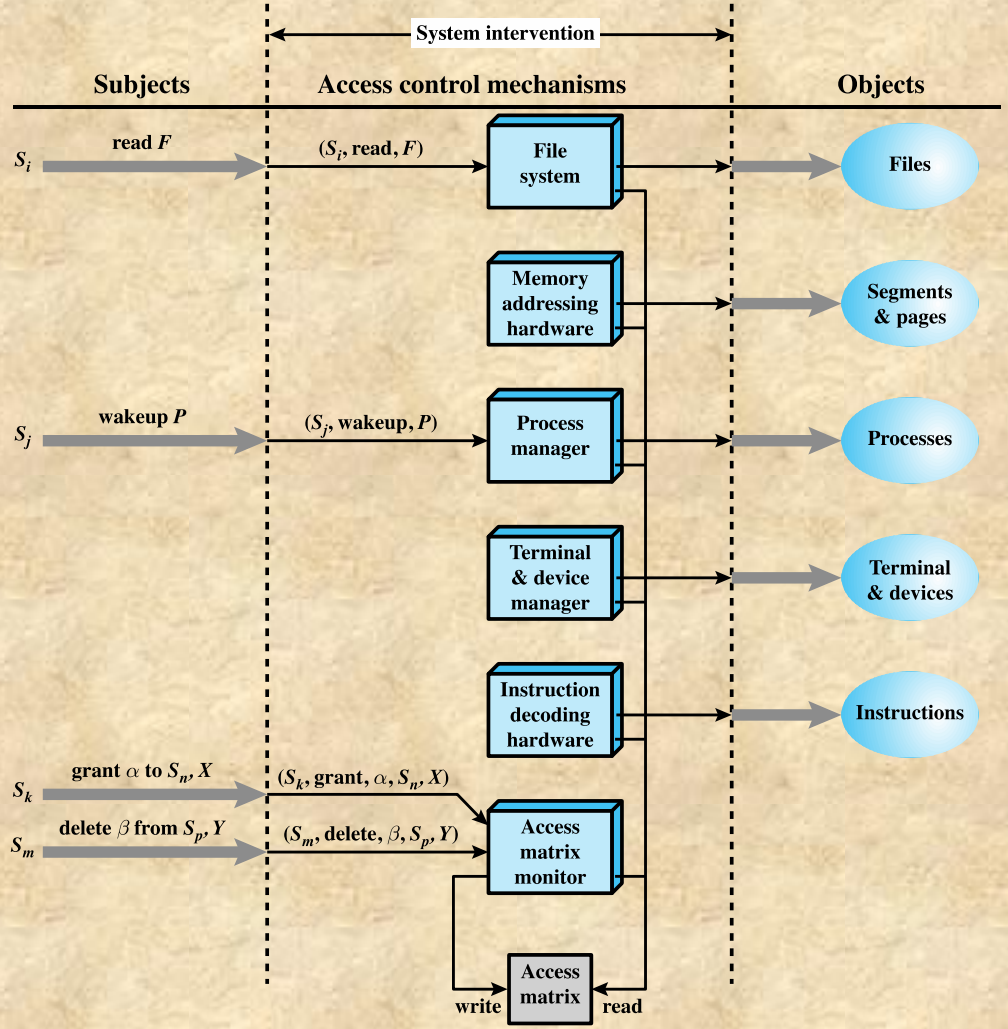


Figure 15.5 An Organization of the Access Control Function

Table 15.1

Access Control System Commands

Rule	Command (by S_o)	Authorization	Operation
R1	**transfer** $\left\{ \begin{matrix} \alpha* \\ \alpha \end{matrix} \right\}$ **to** S, X	'α*' in $A[S_o, X]$	store $\left\{ \begin{matrix} \alpha* \\ \alpha \end{matrix} \right\}$ in $A[S, X]$
R2	**grant** $\left\{ \begin{matrix} \alpha* \\ \alpha \end{matrix} \right\}$ **to** S, X	'owner' in $A[S_o, X]$	store $\left\{ \begin{matrix} \alpha* \\ \alpha \end{matrix} \right\}$ in $A[S, X]$
R3	**delete** α **from** S, X	'control' in $A[S_o, S]$ or 'owner' in $A[S_o, X]$	delete α from $A[S, X]$
R4	$w \leftarrow$ **read** S, X	'control' in $A[S_o, S]$ or 'owner' in $A[S_o, X]$	copy $A[S, X]$ into w
R5	**create object** X	None	add column for X to A; store 'owner' in $A[S_o, X]$
R6	**destroy object** X	'owner' in $A[S_o, X]$	delete column for X from A
R7	**create subject** S	none	add row for S to A; execute **create object** S; store 'control' in $A[S, S]$
R8	**destroy subject** S	'owner' in $A[S_o, S]$	delete row for S from A; execute **destroy object** S

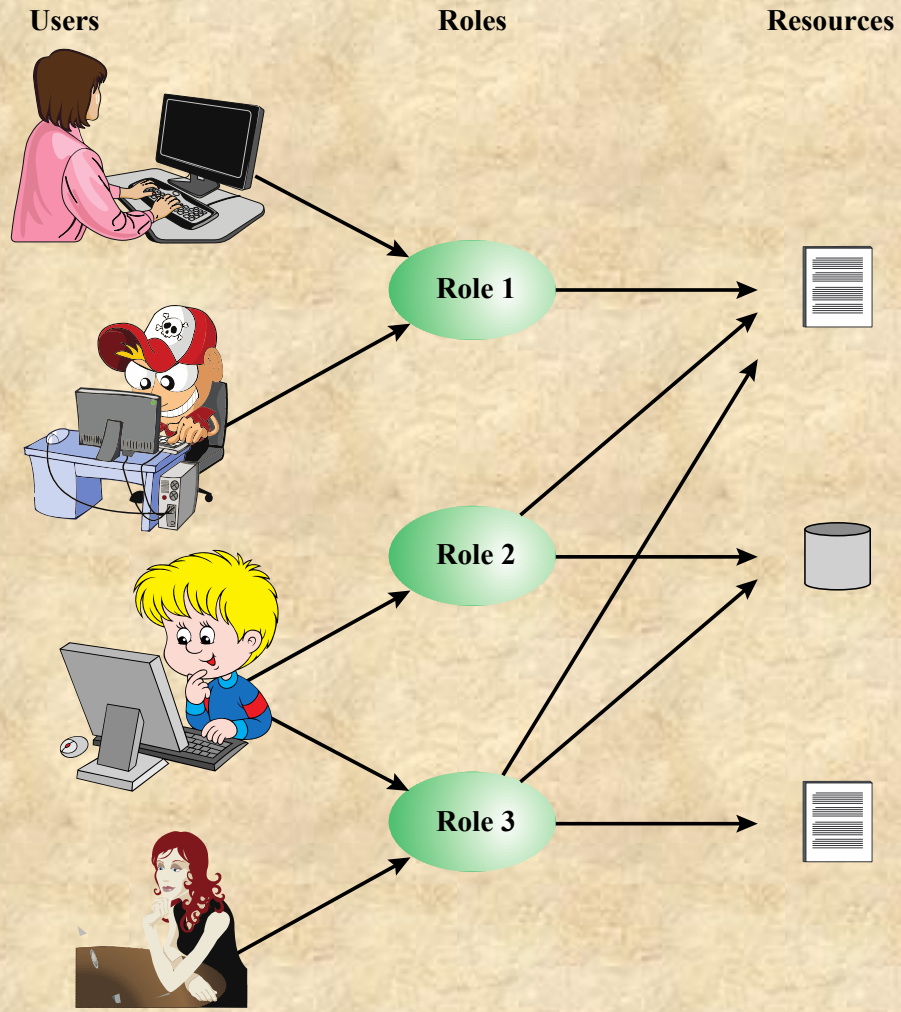
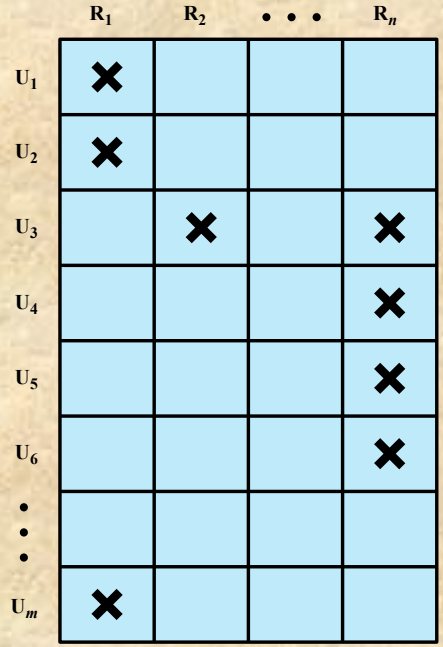
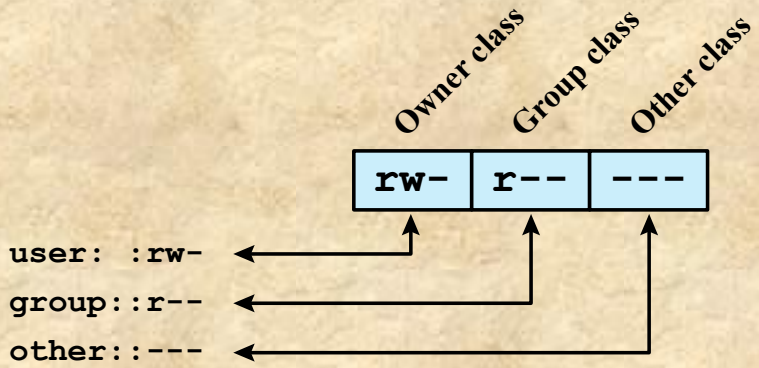


Figure 15.6 Users, Roles, and Resources

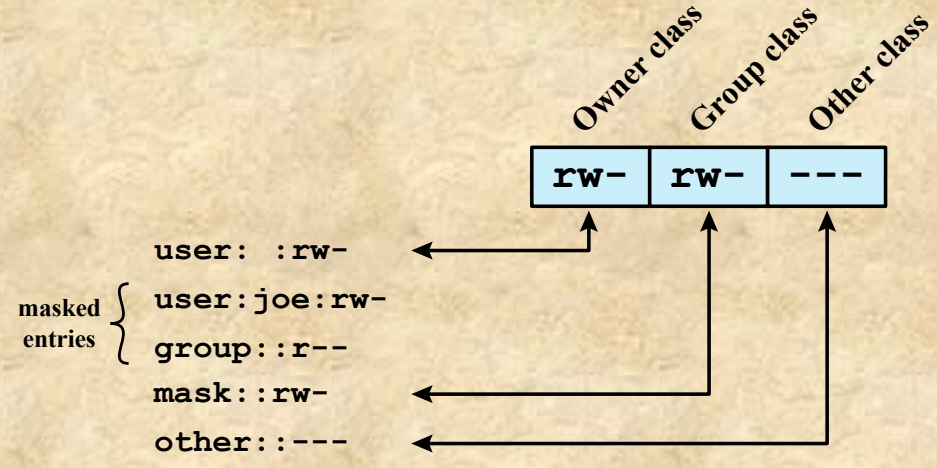


	OBJECTS								
ROLES	R_1	R_2	R_n	F_1	F_1	P_1	P_2	D_1	D_2
R_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
R_2		control		write *	execute			owner	seek *
• • •									
R_n			control		write	stop			

Figure 15.7 Access Control Matrix Representation of RBAC



(a) Traditional UNIX approach (minimal access control list)



(b) Extended access control list

Figure 15.8 UNIX File Access Control

Operating Systems Hardening

- Passos básicos para prover segurança em SO:
  - Instalar e atualizar (patch) o SO
  - Blindar e configurar o SO para endereçar as necessidades de segurança do sistema:
    - Remover serviços, aplicações e protocolos não necessários.
    - Configurar usuários, grupos e permissões.
    - Configurar controle de recursos.
  - Instalar e configurar controles de segurança adicionais:antivirus, host-based firewalls, and intrusion detection systems (IDS), se necessário.
  - Testar a segurança do SO para garantir que os passos anteriores foram executados corretamente.

Instalação do SO: Configuração Inicial

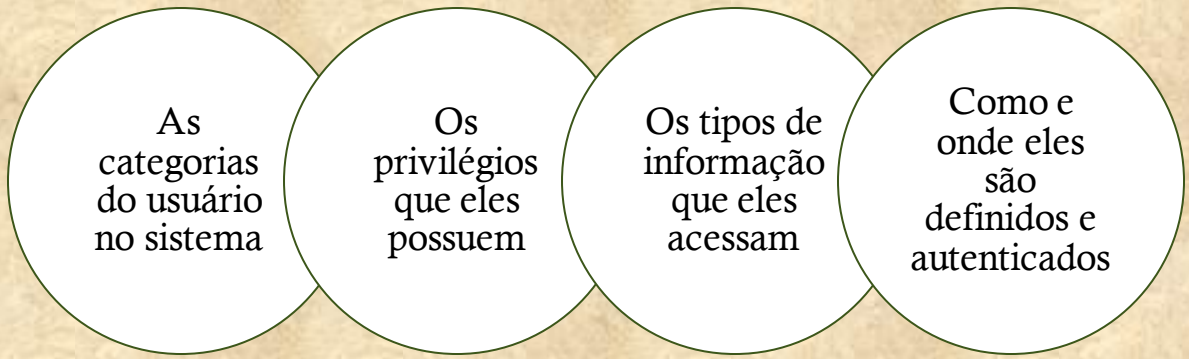
- Instalação do SO
  - Idealmente novos sistemas deveriam estar em redes protegidas
    - Realizar a instalação mínima e adicionar pacotes somente se forem necessários
      - O processo de boot também deve ser protegido
        - Cuidados adicionais na instalação de novos drivers.

Remover serviços, aplicações e protocolos não necessários

- Manter somente o que é necessário para a finalidade do sistema.
- Fazer a instalação mínima do SO, evitar instalações padrões.
- Muitos guias de security-hardening proveem listas de serviços, aplicações e protocolos que não deveriam ser instalados. Por exemplo, https://www.cyberciti.biz/tips/linux-security.html
- Preferencialmente não instalar software não desejado, pois desabilitar ou desistalar pode deixar componentes ativos ainda.

Configurar Usuários, Grupos e Autenticação

O que deve ser considerado no planejamento:



- Dar privilégios elevados somente aos usuários que realmente precisam.
- Contas padrões incluídas na instalação deverim ser verificadas quanto a segurança.
- Remover ou desabilitar contas não necessárias.
- Contas do sistema que gerenciam serviços devem restringir acesso para sessões interativas.
- Qualquer senha instalada por padrão deve ser modificada para valores apropriados.
- Poíticas aplicadas a credenciais de autenciação e senhas devem ser configuradas.

Configurar Controles de Recursos

- Após configuração de usuários e grupos, permissões devem ser atribuídas nos dados e recursos para estarem de acordo com a política.
- Limitar os usuários que podem executar algum programa ou limitar quais usuários podem ler ou escrever dados em árvores de diretório.
- Guias security-hardening guides proveem listas de alterações recomendadas para configuração de acesso padrão para aumentar o nível de segurança.

Instalar Controle de Segurança Adicional

- Instalar software adicional de segurança: antivirus software, host-based firewall, IDS or IPS software, entre outros.
- Algumas são providas em conjunto com SO, mas é importante modificar as configurações padrões.

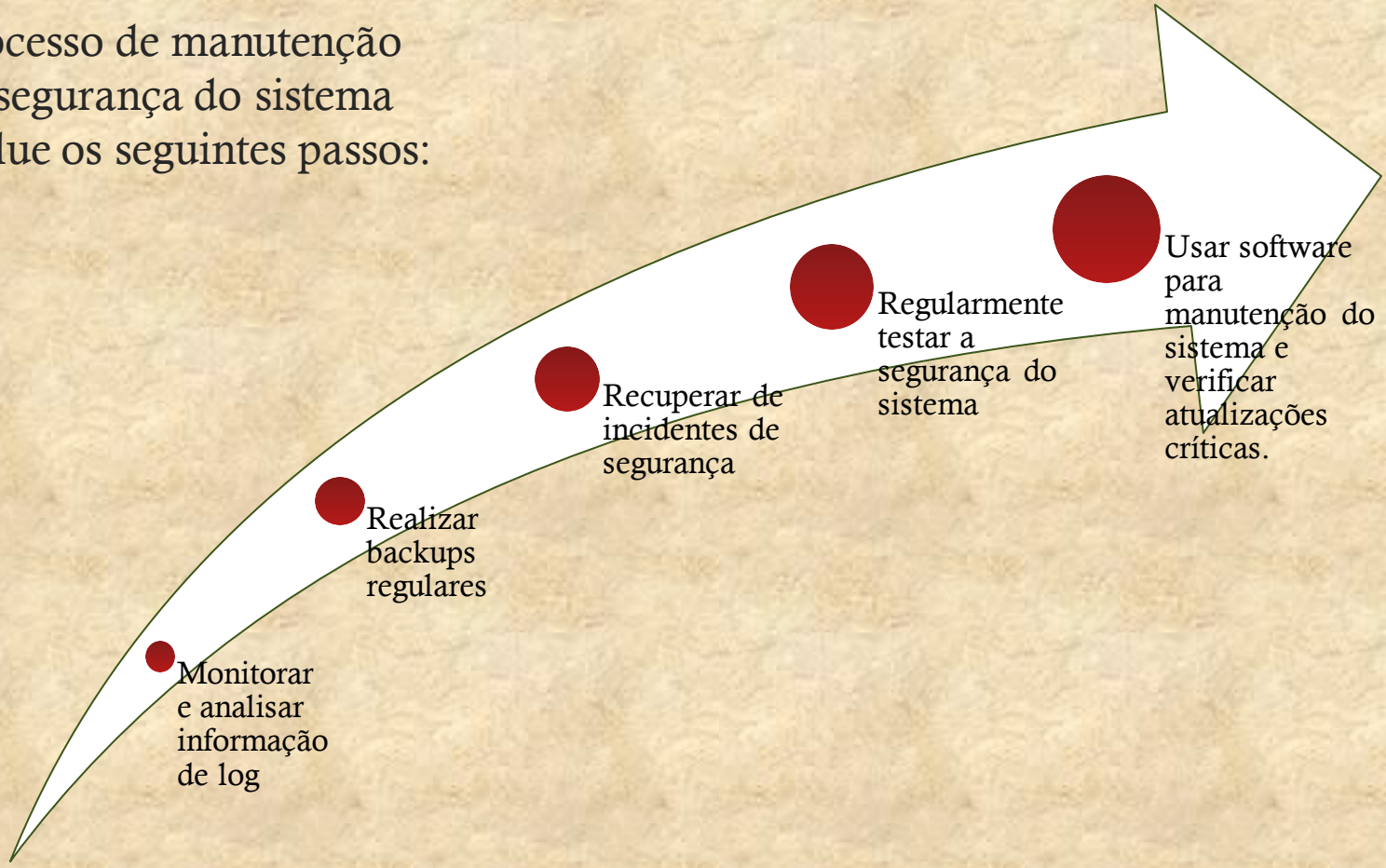
Testar o Sistema de Segurança

- Passo final é testar a segurança do SO.
- Garantir que os mecanismos e serviços de segurança foram instalados e configurados corretamente.

- Checklists são incluídos em muitos guias de security-hardening.
- Há programas que chegam configurações e procuram por vulnerabilidades e configurações impróprias de serviços.
- O processo de teste deve ser realizado ao final do processo de *hardening* e também periodicamente.

Manutenção de Segurança

- Processo de manutenção da segurança do sistema inclue os seguintes passos:



Logging

- Realizar o registro de atividades do sistema ajuda a identificar brechas de segurança, recuperar de falhas e incidentes mais rapidamente.
- Informações de log são geradas pelo SO, rede e aplicações.
- Os dados e a forma de log devem ser definidos no estágio de planejamento da segurança do sistema.
- Cuidados: logs podem gerar grandes volumes de informação.
- Usar mecanismos de rotação de logs e arquivamento.
- Usar mecanismos de análise automática para identificar comportamentos anômalos.

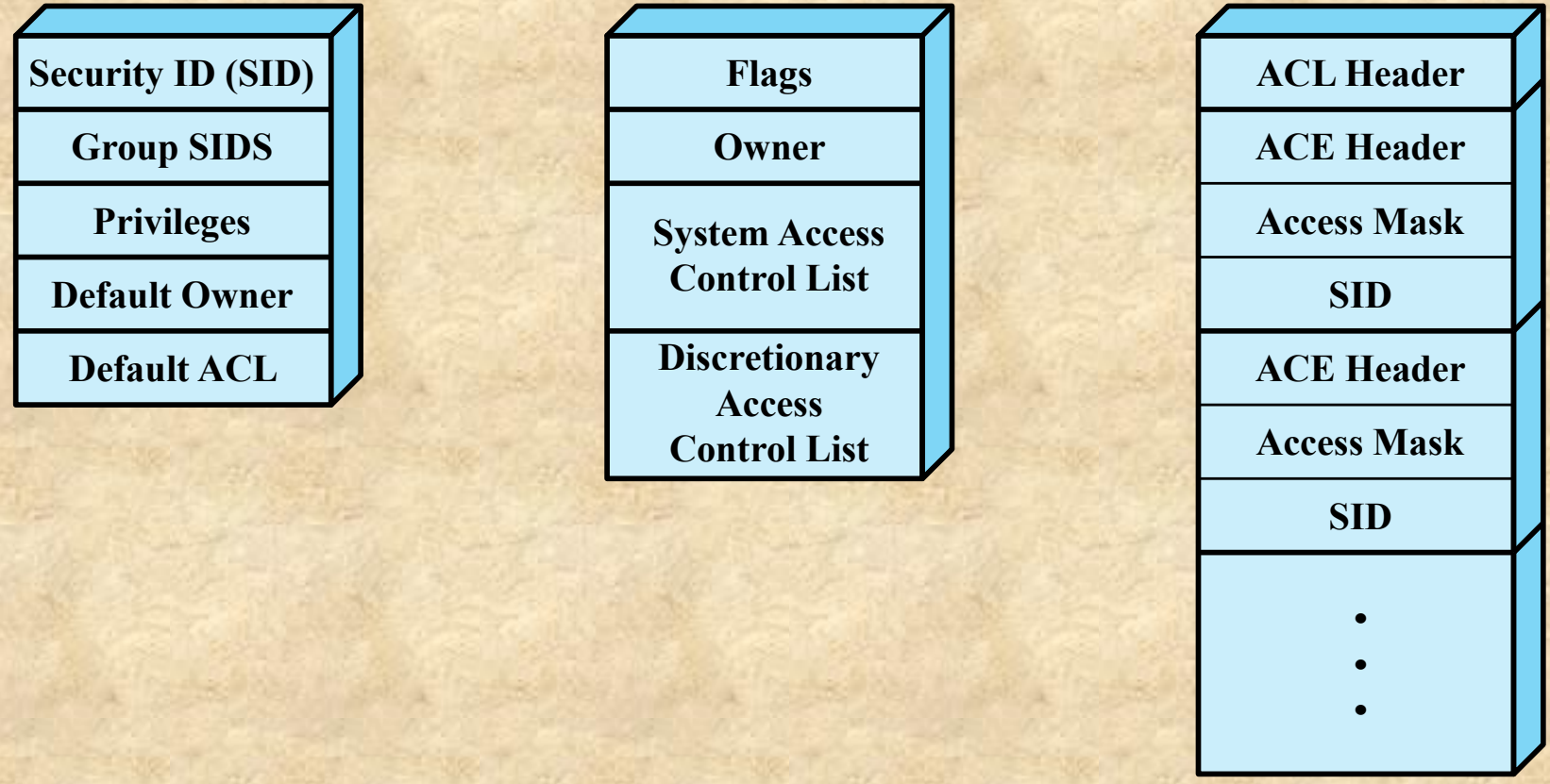
Backup de Dados

- As necessidades e políticas para *backup* e *archive* deveriam ser determinadas no estágio de planejamento.
  - Decisões importantes: cópias devem ser mantidas online ou offline, localmente ou remotamente, quais dados devem ser armazenados, quanto tempo devem ser mantidos.
- Backup
  - O processo de fazer cópias de dados em intervalos regulares, permitindo recuperação de dados perdidos ou corrompidos em períodos relativamente curtos.
- Archive
  - O processo de manter cópias de dados durante longos períodos de tempo (meses, anos), com finalidade de atender requisitos legais ou operacionais para acesso de dados passados.

Esquema de Controle de Acesso

- No Windows, um esquema de usuário/senha é usado para autenticar o usuário.
- No caso de sucesso, um processo é criado para o usuário e um token de acesso é associado com o processo.
  - O token de acesso inclui security ID (SID) que identifica que o usuário é reconhecido pelo sistema.
  - O token também contêm SIDs para grupos de segurança para os grupos para que o usuário pertence.

O token de acesso serve para dois propósitos:	Manter todas informação de segurança para rápido acesso.
	Permitir que cada processo modifique suas características de segurança de forma limitada e independente.



(a) Access token (b) Security descriptor (c) Access control list

Figure 15.9 Windows Security Structures

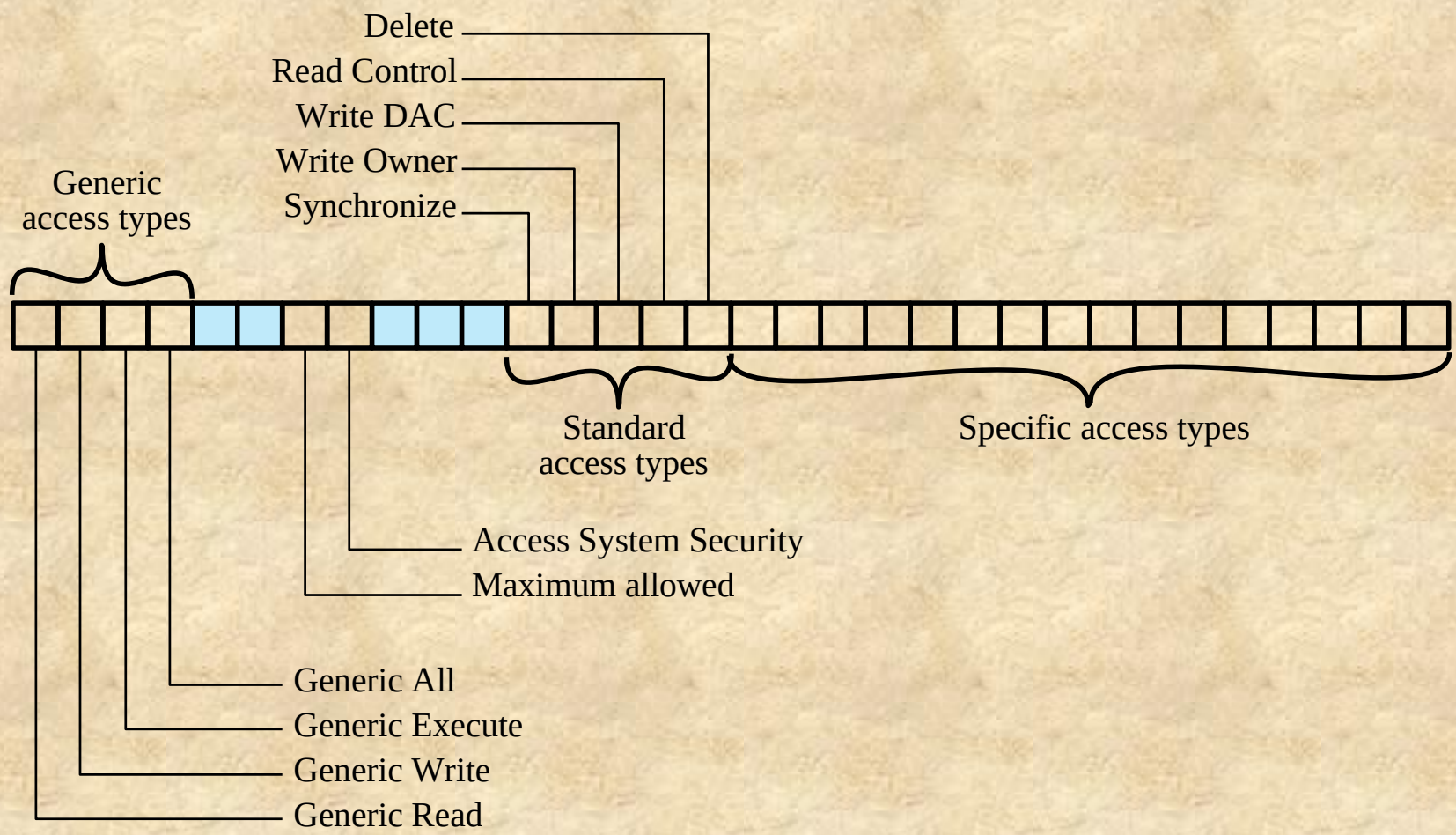


Figure 15.10 Access Mask

Referências

- Stallings, William. **Operating Systems: Internals and Design Principles**. 9th Edition, Pearson, 2017.