**Segurança em sistema Operacional**

**INTRODUÇÃO:**

A informação nos dias atuais é de grande valia para as pessoas comuns e para empresas, que além de manipular, se faz necessário armazenar e proteger os dados e recupera-los de forma segura quando necessário. E dai surge a necessidade da segurança nos sistemas.

Todo sistema operacional se preocupa em proteger a informação contra o uso não autorizado.

Anteriormente havia poucos computadores e estes eram de múltiplos usuários e em sua maioria eram isolados ( sem rede) a segurança era intrínseca em apenas manter os usuários isolados um do outro. Os modelos se softwares eram simples. Certificava que nenhum usuário tivesse o direito de acesso aos quais ele não era credenciado.

Com o surgimento dos variados dispositivos de massa e a globalização da internet o cenário mudou. Um desktop tem apenas um usuário, mas apresenta outras ameaças.

Sistemas operacionais e softwares cada vez maiores e complexos abrem a possibilidades para defeitos/falhas. Quando há um defeito de segurança se diz defeito de vulnerabilidade.

Softwares vulneráveis podem permitir invasões através de uma “exploração”. Onde o invasor de alguma forma alimenta o software vulnerável a fim de desencadear o defeito e assim conseguir invadir e assumir o controle da maquina.

O atacante pode fazer a exploração de modo manual ou automática, por meio de vírus ou worms.

* Vírus 🡪 precisa da interação do usuário para se propagar
* Worms 🡪 se propagam de maneira independente do usuário.
* Cavalo de troia 🡪 é embutido em algum programa funcionalidades a mais de forma imperceptível ao usuário, que ao ser instalado se passa o acesso e controle do pc ao atacante.

**9.1- Ambiente de segurança:**

Diferença tênue entre segurança e proteção:

* Segurança🡪 refere-se aos problemas gerais envolvidos em garantir que os arquivos não sejam lidos ou modificados por pessoas não autorizadas.
* Proteção🡪 são mecanismos específicos do sistema operacional usados para fornecer segurança.

**9.1.1. Ameaças**

A literatura de compõem a segurança em 3 propriedades: CIA



1- confidencialidade 🡪 O sistema deve garantir que não haja liberação dos dados para quem não é autorizado. O dado só poderá estar disponível pra quem estiver autorizado.

2- Integridade🡪 somente poderá alterar ou excluir quem esteja autorizado para tal. Usuários não autorizados não devem ser capazes de alterar sem a permissão do proprietário.

3- disponibilidade🡪Ninguém poderá perturbar o sistema a fim de torna-lo inutilizável.

Há outras propriedades: autenticidade, responsabilidade, não repudiação, privacidade, entre outras.

Os sistemas estão sobre constantes ameaças: (exemplos)

* Não encriptação pelo protocolo da rede pode violar a confidencialidade. Permitindo que o atacante acesse os dados.
* O intruso ao invadir um BD ao alterar e apagar viola a integridade.
* Ataque de recusa de serviço 🡪 enviar muitas solicitações de serviço a um servidor poderá paralisa-lo deixando consumir seu tempo de cpu.

Antes algumas definições:

* Hacker 🡪 termos de honra dado a grandes programadores
* Chapéus brancos 🡪 hackers éticos, especialistas em comprometer sistemas de segurança de computadores e utilizam suas habilidades para o bem.
* Crackers ou chapéus pretos 🡪Invasor de sistemas que não lhe dizem respeito ( ganho pessoal)

Muitos dos ataques são apoiados por ferramentas e serviços avançados. As ferramentas a seguir são usadas tanto por atacantes como defensores.

* Nmap é um [software](https://pt.wikipedia.org/wiki/Software) livre que realiza [port scan](https://pt.wikipedia.org/wiki/Port_scan" \o "Port scan) desenvolvido pelo [Gordon Lyon](https://pt.wikipedia.org/wiki/Gordon_Lyon).  É utilizado para avaliar a segurança dos computadores, e para descobrir serviços ou [servidores](https://pt.wikipedia.org/wiki/Servidor) em uma [rede de computadores](https://pt.wikipedia.org/wiki/Rede_de_computadores). Faz uma varredura de portas. Testa e estabelece conexão TCP.
* O Dsniff é um canivete suíço do administrador de redes. Oferece uma coleção de ferramentas que é útil para testes de penetração e auditoria de rede. O Dsniff pode monitorar de forma rápida uma rede de dados, e-mail, senhas, arquivos etc.
* LOIC é um [programa de computador](https://pt.wikipedia.org/wiki/Programa_de_computador) de [código aberto](https://pt.wikipedia.org/wiki/C%C3%B3digo_aberto) escrito em [C#](https://pt.wikipedia.org/wiki/C_Sharp) que tem como objetivo executar um [ataque de negação de serviço](https://pt.wikipedia.org/wiki/Ataque_de_nega%C3%A7%C3%A3o_de_servi%C3%A7o). O programa foi desenvolvido pela Praetox Technologies em [2006](https://pt.wikipedia.org/wiki/2006) com o intuito de avaliar e testar redes, sendo depois disponibilizado para [domínio público](https://pt.wikipedia.org/wiki/Dom%C3%ADnio_p%C3%BAblico).
* Metasploit é um projeto de [segurança de informação](https://pt.wikipedia.org/wiki/Seguran%C3%A7a_de_informa%C3%A7%C3%A3o) com o objetivo de análise de vulnerabilidades de segurança e facilitar [testes de penetração](https://pt.wikipedia.org/wiki/Teste_de_penetra%C3%A7%C3%A3o) e no desenvolvimento de assinaturas para sistemas de detecção de intrusos.
* Criminosos cibernéticos também oferecem serviços on line disseminando malwares , lavando dinheiro, redirecionar trafego, hospedagem de qualquer conteúdo, etc.
* A atividade que vem crescendo usa Infraestrutura Botnet  que designa um grupo de computadores conectados à [Internet](https://pt.wikipedia.org/wiki/Internet), cada um deles rodando um ou mais [bots](https://pt.wikipedia.org/wiki/Bots" \o "Bots) e se comunicando com outros dispositivos, a fim de executar determinada tarefa em conjunto.
* Os ataques podem ir além do sistema. Ataque no tratamento de esgoto maroochy Shire, em Queens – Austrália que permitiu um crime ambiental.
* Stuxnet é um [worm de computador](https://pt.wikipedia.org/wiki/Worm" \o "Worm) projetado especificamente para atacar o [sistema operacional](https://pt.wikipedia.org/wiki/Sistema_operacional) [SCADA](https://pt.wikipedia.org/wiki/SCADA) desenvolvido pela [Siemens](https://pt.wikipedia.org/wiki/Siemens) e usado para controlar as centrífugas de enriquecimento de urânio [iranianas](https://pt.wikipedia.org/wiki/Ir%C3%A3). Foi descoberto em junho de [2010](https://pt.wikipedia.org/wiki/2010) pela empresa [bielorrussa](https://pt.wikipedia.org/wiki/Bielorrussa) desenvolvedora de [antivírus](https://pt.wikipedia.org/wiki/Antiv%C3%ADrus) [VirusBlokAda](https://pt.wikipedia.org/w/index.php?title=VirusBlokAda&action=edit&redlink=1" \o "VirusBlokAda (página não existe)). É o primeiro worm descoberto que espiona e reprograma sistemas industriais.
* Um aspecto importante do problema de segurança e relacionado a confidencialidade que vem sendo motivo de amplas discussão atuais pelos governos é a privacidade: a privacidade individual, um direito individual que pode estrar em conflito com o direito coletivo, há aspecto éticos a cerca disso.

**9.1.2 – Atacantes** ( intrusos ou aversarios)

Há mais variados motivos para invasão: roubo, ativismo, vandalismo, terrorismo, guerra cibernética, espionagem, spam, extorsão, fraude, exibição, ou apenas expor a fragilidade da segurança de sistema.

Existem dos mais variados dos aspirantes aos crackers. (Script-kiddies 🡪 crianças que seguem um roteiro). Podem ser profissionais trabalhando para criminosos, governos, empresas de segurança, ou ate mesmo pessoas como passatempo.

O esforço necessário para segurança e proteção depende de quem seja o inimigo.

**9.2 segurança de sistemas operacionais:**

Em geral distingue-se o ataque em: Tentam roubar informações passivamente e que tentam ativamente fazer com que o programa comporte-se mal. Exemplo:

* ataque passivo🡪 um adversário fareja o trafego de rede e tenta violar a codificação para adquirir os dados.
* Ataque ativo🡪o intruso pode assumir o controle do navegador e fazer com que o usuário execute um código malicioso a fim de roubar dados.

O sistema operacional faz uso de criptografia e endurecimento para aumentar a segurança:

* Criptografia-🡪 diz respeito ao embaralhamento da mensagem ou arquivo original que pode sr recuperado via chave. Usa na transmissão em rede, armazenar arquivos, embaralha senhas.
* Endurecimento🡪 diz respeito ao aumento de mecanismos de proteção nos softwares a fim de dificultar a ação de atacantes, assim os atacantes não conseguem injetar códigos novos em softwares em execução e garante q cada processo tem exatamente os devidos privilégios.

**9.2.1- temos condições de construís sistemas seguros?**

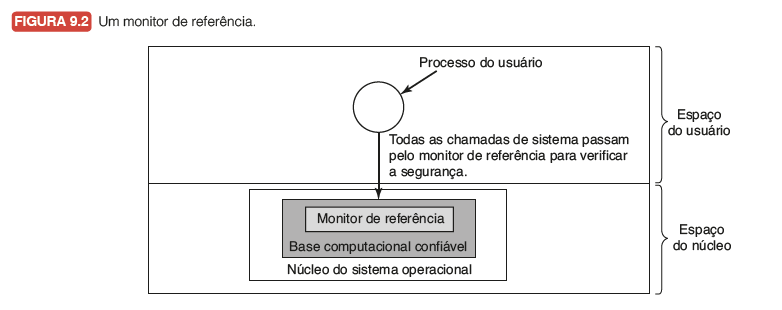
Em teoria sim, softwares podem ser livres de defeitos e podem ser certificados que são seguros. Porem ele não pode ser grande de mais ou completo.

Porque não é feito seguro? Dois motivos:

* 1º os usuários não querer trocar por um seguro. (Por motivos financeiros e não haver portabilidade.)
* 2º a única maneira de construir um software seguro e mantê-lo simples. Porém usuários querem cada vez maiores e melhores funcionalidades. Ou seja mais complexo, mais código, mais defeito, mais erro de segurança.

**9.2.2- Base computacional**

* No cerne de cada sistema confiável (sistema q possui exigências de segurança declaradas e as atende) existe o TCB – base computacional confiável consistindo de hardware e software necessários para fazer valer as regras de segurança. Se o TCB estiver conforme as especificações , a segurança do sistema não poderá ser comprometida.
* A TCB é formada pelo hardware, uma porção do núcleo do s.o e pela maioria/todos os programas de usuário que tenham poder de super-usuário.
* Funções do sistema que fazem parte da TCB: Criação de processos, a troca de processos, o gerenciamento de memoria e parte do gerenciamento de arquivo e E/S.
* Parte importante TCB é o monitor de referencia 🡪 aceita todas as chamadas de sistema que envolve segurança. Decide se uma chamada pode ou não ser realizada.



**9.3 Controlando o acesso aos recursos:**

A segurança é muito mais fácil de ser conquistada quando existe um modelo claro do que dever ser protegido e quem tem permissão pra fazer o quê.

**9.3.1- Domínios de proteção:**

Um so contém muitos recursos ou objetos ( hardware, software) que precisam ser protegidos. Onde Cada objeto tem um nome único e os processos tem um conjunto finito de operações.

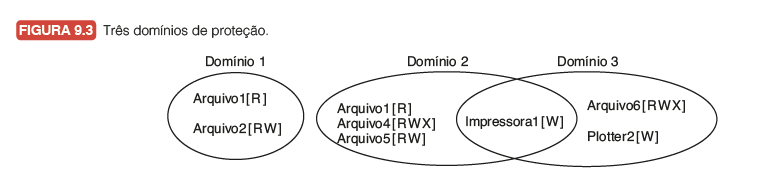
Os processos devem acessar apenas os objetos que são autorizados, proibindo de acessar os objetos que não tem autorização de acesso e deve se tornar possível restringir os processos a um subconjunto de operações legais quando necessário.

* Domínio 🡪 é um conjunto de pares, formado por D(objetos, direitos), onde cada par especifica um objeto e algum subconjunto de operações que podem ser desempenhadas no objeto.
* Direito🡪 permissão para desempenhar uma das operações.

Um domínio pode ser mais geral do que apenas um usuário.

Como os objetos são alocados para os domínios depende das questões especificas relativas a quem precisa saber o que.

Um domínio apresenta um ou mais objetos, onde cada objeto apresenta um conjunto de direitos.



Examinando o UNIX:

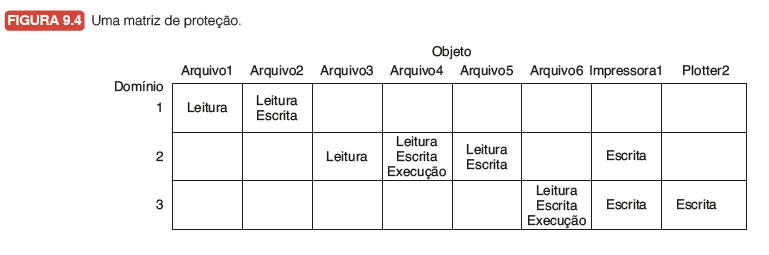
O domínio de um processo e definido por sua UID e GID que são herdados por seus filhos. Dada qualquer combinação UID e GID é possível listar todos os objetos que podem ser acessados e as permissões (r,w,x). Dois processos com a mesma UID e GID terão acesso aos mesmo conjuntos de dados. Processo com diferentes valores terão acesso a um conjunto diferentes de arquivo.

Lembrando que: Como cada processo tem 2 metades uma usuário e uma núcleo. Quando é realizada uma chamada do sistema ele troca da parte usuário para núcleo, assim ele passa ter acesso a um conjunto diferente de objetos, trocando assim de domínio.

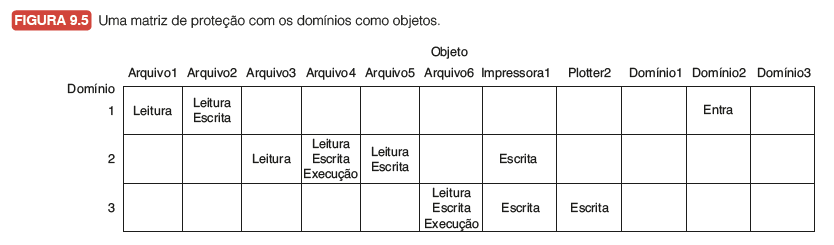
Outra troca de domínio é quando o processo realiza um exec em um arquivo com o bit setuid ou SETGID adquirindo um novo UID e GID. Passando a ter um conjunto diferente de arquivo e combinações.

Como o sistema controla quais objetos pertencem a qual domínio?

Conceitualmente por uma matriz em que a linhas são domínios e os objetos colunas. Assim dada a matriz, o numero do domínio , o sistema pode dizer se o acesso e permissão a um objeto em particular é permitido.



A troca de domínio em si pode ser incluída no modelo matriz ao tratarmos o domínio como um objeto, com a operação enter. A figura mostrar que o processo do domínio 1 podem trocar para o 2, mas uma vez ali não podem voltar ao 1. Representando assim uma modelo de execução do SETUID.

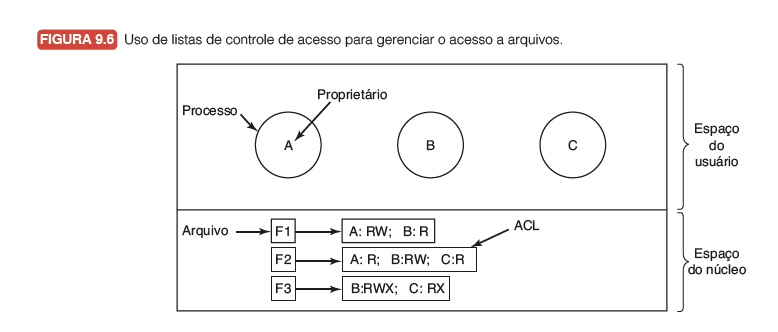


**9.3.2 lista de controle de acesso**

Problemas da matriz: grande, esparsa com muitas células vazia (desperdício de espaço do disco). Então é proposto dois métodos práticos : armazenar a matriz por linhas ou por colunas , e somente os elementos não vazios.

* Armazenar por colunas:

Consiste em associar com cada objeto uma lista ordenada contendo todos os domínios de acesso. Esta lista é denominada ACL 🡪 access sontrol list – lista de controle de acesso



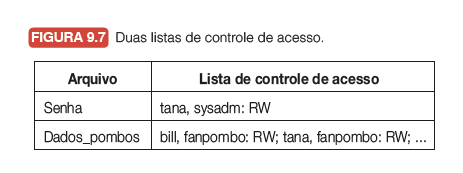
Cada arquivo tem uma ACL associada a ele. Ex. o arquiv f1 tem duas entradas na sua ACL onde a primeira diz que qualquer processo de propriedade do usuário A pode ler e escrever em F1. E a segunda entrada diz que qualquer processo de B pode ler F1. E todos os demais acesso por esses usuários e todos os acessos dos outros usuários são proibido. Os direitos de acesso são concedidos por usuário e não por processo.

Existe outros tipos de direitos genéricos como: destroy object, copy object e específicos como: append message e sort alphabetically.

* Grupos também tem nome e podem possuir ACL. Tem duas variações de semânticas:
* **1ª semântica:**

Onde cad usuário tem sua id (UID) e uma id de grupo (GID) onde: UID1,GID1: DIREITOS1; UID2,GID2:DIREITOS2;... Assim se a combinação de ambos UID E GID estiver na lista ACL será disponibilizado acesso, caso contraria não.

Usando grupo introduz o conceito de papel



Neste caso tana tem e administradora do sistema e esta no grupo sysadm e tana também é pertence ao grupo fanpombo. Assim o acesso a um arquivo vai depender de qual grupo ele esta atualmente conectada. Tana não poderá acessar o arquivo de senha quando estiver conectada como fanpombo.

Para contar isso, o usuária poderá ter acesso a um determinado arquivo independente do grupo conectado introduzindo um curinga wild-card , por exemplo tana, \*:RW, assim Tana teria acesso a senha independente do grupo. Outra alternativa é de que se o usuário não precisa especificar o grupo que tem acesso, automaticamente todos os grupos são inerentes ao usuário todo tempo. Neste caso bastaria o direito de acesso. Porem este método o encapsulamento é reduzido.

O uso de grupos e wildcars possibilita o bloquei seletivo de usuário para acesso ao arquivo. Por exemplo, virgil, \* : (nome); \* : RW . neste caso todos usuários menos o virgil teriam aceso para ler e escrever.

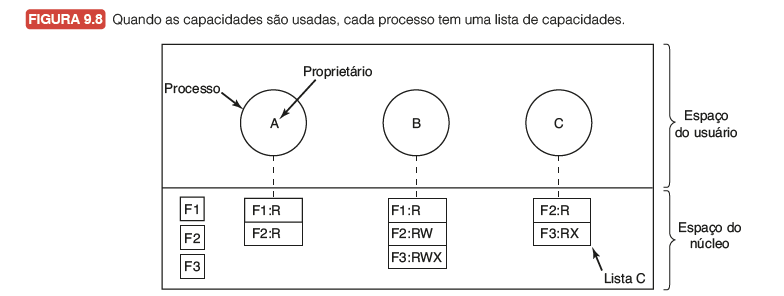
* **2ª semântica:**

Ao invés do grupo possuir entrada ACL em pares (UID,GID) ter para cada entrada apenas UID ou um GIG. Ex: debbie: RW; fapombo: RW

Com ACL revogar um direito de acesso anteriormente concedido e relativamente direto. Basta editar a ACL para fazer a mudança. No entanto a se o arquivo estiver aberto durante a edição da ACL somente a chamada futura terá efeito.

**9.3.3 capacidades:**

A divisão da matriz por linhas. Este método consiste em associar para cada processo uma lista de objetos que podem ser acessados juntamente com os respectivos domínios (diretos de acesso). Esta lista é chamada de lista de capacidade ou lista C.



Assim cada capacidade concede ao proprietário determinados direitos sobre um certo objeto.

Em geral a capacidade consiste em um objeto identificador e uma mapa de bits para vários direitos ( no unix o identificador é o numero do i-node)

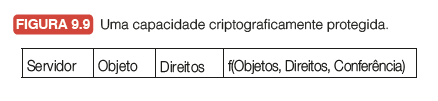
Como as listas de capacidades são objetos podem ser apontadas por outras listas, compartilhando assim subdomínios.

Listas de capacidades devem ser protegidas do usuários: 3 métodos de proteção:

1º arquitetura marcada : um projeto de hardware no qual cada palavra de memoria tem um bit (marca) extra que diz se a palavra contem capacidade ou não. Este bit so pode ser modificado pelo sistema operacional.

2º manter a lista C dentro do so. Assim as capacidades serão referenciadas por sua posição na lista de capacidades. ( ex.:leia 1 kb do arquivo apontado pela capacidade 3)

3ª manter a lista C no espaço de usuário: deverá criptografar as capacidades ( usada em sistemas distribuídos). É feita da seguinte forma: O cliente envia um mensagem de criação para o servidor de arquivo remoto. O servidor cria o objeto, gera um longo numero aleatório e o campo de conferência. Uma lacuna na tabela de arquivos do servidor é reservada para o objeto e o campo de referencia e armazenado junto do endereço dos blocos do disco ( armazenado no i-node do unix).



O servidor então gera e retorna uma capacidade para usuário conforme a figura acima. O usuário recebe a capacidade com identificados do servidor, numero do objeto (numero do i-node), e os direitos, armazenados como mapa de bits. O ultimo campo consiste na concatenação do objeto, direitos, e campo de conferencia. Concatenação esta q se utiliza de uma função de mão única criptografadamente segura.

Assim quando o usuário quiser acessar o objeto ele envia a capacidade como parte da solicitação. O servidor então extrai o numero do objeto para indexa-lo em sua tabela para encontrar o objeto. O servidor então pega os dois campo da própria capacidade e o terceiro de sua própria tabela, se o resultado for mesmo que o 4 campo a solicitação é aprovada, caso contrario rejeitada.

Direitos genéricos aplicados a objetos:

1. Copiar capacidade: criar uma nova capacidade para o mesmo objeto
2. Copiar objeto: criar um objeto duplicado para uma nova capacidade
3. Remover capacidade: apagar uma entrada da lista C ( objeto não é afetado)
4. Destruir objeto: remover objeto e capacidade permanentemente

Na versão gerenciada pelo núcleo é difícil revogar o acesso a um objeto. Uma vez que as capacidades estão armazenadas em lisa c por todo o disco. Sendo assim difícil para o so. Encontrar todas as capacidades em aberto para qualquer objeto. Uma alternativa é fazer com que cada capacidade aponte para um objeto indireto ( que aponte para o objeto em si), o sistema então rompe com a conexão invalidando assim as capacidades.

Problemas:

* Não é permitido por sistemas de capacidade revogação seletiva
* Uma copia valida da capacidade não poderia ser distribuída aos usuários.

ACL e Capacidades se complementam:

|  |  |
| --- | --- |
| **ACL - lista de controle de acesso** | **capacidade** |
| Consiste em associar com cada objeto uma lista ordenada contendo todos os domínios de acesso. | A divisão da matriz por linhas. Este método consiste em associar para cada processo uma lista de objetos que podem ser acessados juntamente com os respectivos domínios |
| Uma busca da ACL pode ser necessária . se não há suporte a grupos, então conceder a todos direito de leitura a um arquivo exige exige enumerar todos os usuários da ACL. | São eficientes. Não precisa de conferência |  |
| Não permite com tanta facilidade | Permite que um processo seja encapsulado facilmente |
| Permite revogação seletiva de direitos | Não permite revogação seletiva de direitos |
| Não apresenta problemas com remoção | Se um objeto é removido e as capacidades não , ou se a capacidade for removida e um objeto não aparecerão problemas. |
| Utilizadas por Windows e unix | Nucleo L4 ( android) e freeBSD (capsicum) |