OFFENSIVE SECURITY WEB EXPLOITATION PT.2

JOAS ANTONIO

SOBRE O LIVRO

Livro pouco prático e bastante teórico, apenas apresentando os ataques e alguns exemplos;

Recomendo que tenha conhecimentos fundamentais sobre aplicação web, pois não é bastante aprofundado;

Técnicas de exploração de vulnerabilidades em aplicações web geralmente usada em Bug Bountys;

SOBRE O AUTOR

Joas Antonio;

Apaixonado por Segurança da Informação;

https://www.linkedin.com/in/joas-antonio-dos-santos/

O QUE VOCÊ ENCONTRARÁ?

Conceitos básicos de aplicação web;

OWASP-TOP 10;

Conceitos de ataques em aplicações web;

Vulnerabilidades em Aplicações Web;

Conclusão;

CONCEITOS DE APLICAÇÕES WEB

HTTP E HTTPS

HTTP:

Hypertext Transfer Protocol, ou simplesmente HTTP, é um protocolo de comunicação, utilizado pela internet para transferir dados entre o computador do usuário e servidores de hipermedia. Ou seja, é através deste protocolo, que cada byte de informação navega entre seu computador/smartphone e os servidores de internet. Normalmente o protocolo HTTP usa a porta 80 do seu dispositivo para transferir os dados.

HTTP E HTTPS

HTTPS:

Hypertext Transfer Protocol Secure, ou simplesmente HTTPS, é uma versão idêntica do protocolo HTTP sobre uma camada SSL. Essa camada adicional permite que os dados sejam transmitidos através de uma conexão criptografada e que se verifique a autenticidade do servidor e do cliente através de certificados digitais. A porta TCP usada por norma para o protocolo HTTPS é a 443.

https://www.oficinadanet.com.br/post/15657-diferencas-entre-http-e-https

SSL E TLS

- **SSL** significa **Secure Sockets Layer**, um tipo de segurança digital que permite a comunicação criptografada entre um site e um navegador. Atualmente a tecnologia se encontra **depreciada** e está sendo completamente substituída pelo TLS.
- **TLS** é uma sigla que representa **Transport Layer Security** e certifica a proteção de dados de maneira semelhante ao SSL. Como o SSL não está mais de fato em uso, esse é o termo correto que deveria ser utilizado.
- Certificados SSL/TLS funcionam por unir digitalmente uma chave criptográfica à informação de identificação de uma companhia. Isso permite que dados possam ser transferidos de maneira que não podem ser descobertos por terceiros.
- O SSL/TLS funciona através de chaves públicas e privadas, além de chaves de sessão para cada conexão segura. Quando o visitante coloca uma URL com SSL no navegador e navega pela página segura, o navegador e o servidor fazer uma conexão.

CÓDIGO DE STATUS

Os códigos de status das respostas HTTP indicam se uma requisição HTTP foi corretamente concluída. As respostas são agrupadas em cinco classes:

- **Respostas de informação (100-199),**
- **Respostas de sucesso (200-299),**
- **Redirecionamentos (300-399),**
- **Erros do cliente (400-499),**
- **Erros do servidor (500-599).**

https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status

MÉTODOS DE REQUISIÇÃO

GET

O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.

HEAD

O método HEAD solicita uma resposta de forma idêntica ao método GET, porém sem conter o corpo da resposta.

POST

O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.

MÉTODOS DE REQUISIÇÃO

PUT

O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.

DELETE

O método DELETE remove um recurso específico.

CONNECT

O método CONNECT estabelece um túnel para o servidor identificado pelo recurso de destino.

MÉTODOS DE REQUISIÇÃO

OPTIONS

O método OPTIONS é usado para descrever as opções de comunicação com o recurso de destino.

TRACE

O método TRACE executa um teste de chamada loop-back junto com o caminho para o recurso de destino.

PATCH

O método PATCH é utilizado para aplicar modificações parciais em um recurso.

FRONT AND BACK-END

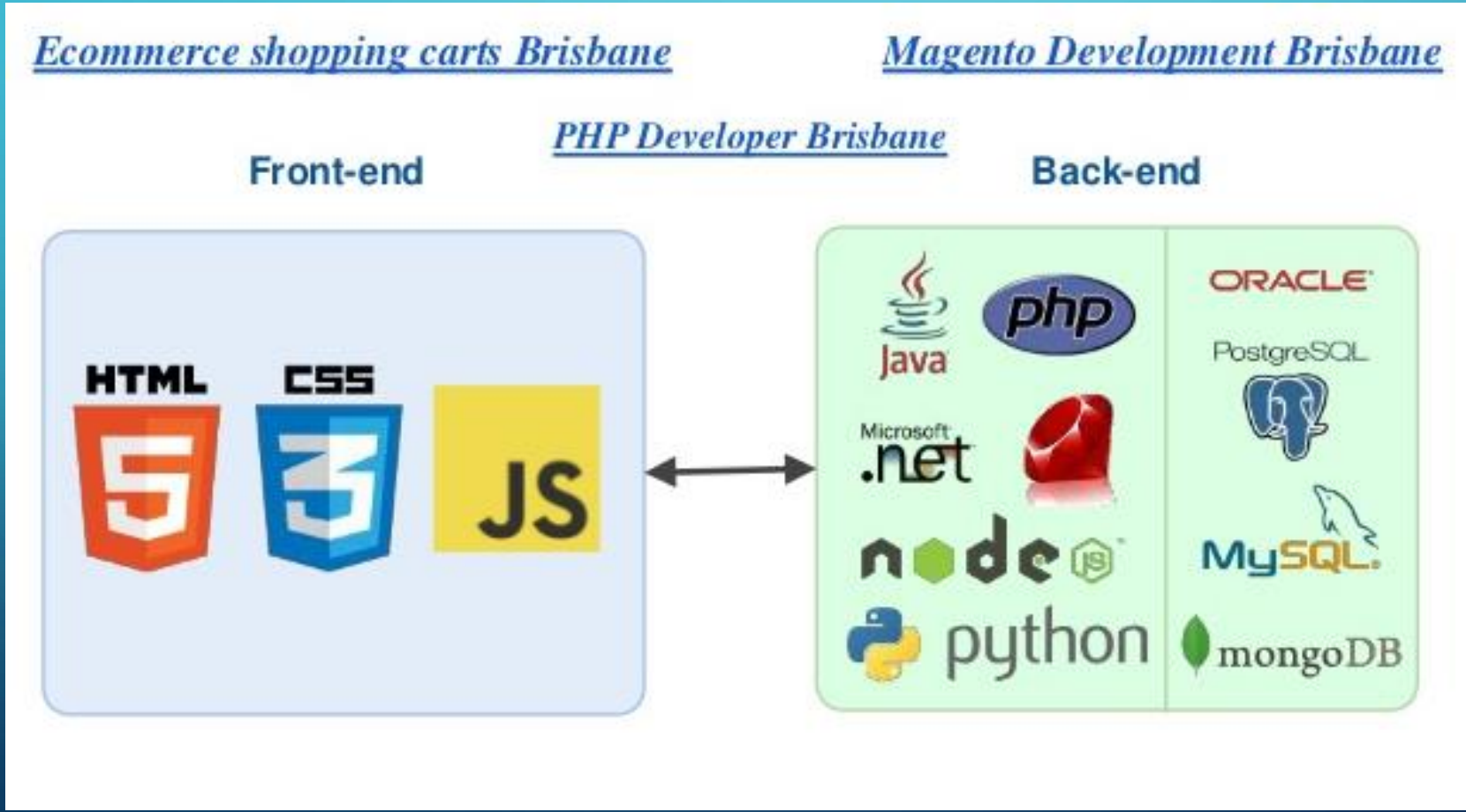
Em ciência da computação, front-end, interface frontal ou parte frontal e back-end, parte secundária, parte de suporte ou parte de retaguarda são termos generalizados que se referem às etapas inicial e final de um processo. O front-end é responsável por coligir a entrada do usuário em várias formas e processá-la para adequá-la a uma especificação em que o back-end a possa utilizar.

Em resumo o Front é a parte visual que o usuário interage;

Já o Back-end processa tudo que o usuário faz na aplicação, seja um login ou um cadastro;

https://pt.wikipedia.org/wiki/Front-end_e_back-end

FRONT AND BACK-END



FRONT AND BACK-END

CLIENT AND SERVER-SIDE

- *Server-side* diz respeito ao lado do servidor (seu servidor de aplicação, ex: IIS). *Client-side* diz respeito ao lado do cliente (ex: um web-browser).
- A interação funciona da seguinte forma: **Servidor ← → Cliente ← → Usuário**
  - O **servidor** fornece para o **cliente** uma *saída*: ele serve a página desejada ou arquivos (download).
  - O **servidor** interpreta a *entrada* do **cliente**: o cliente envia informações (formulário) para o servidor e arquivos (upload).
  - O **cliente** fornece para o **usuário** uma *saída*: a página renderizada que ele obteve do servidor (html).
  - O **cliente** interpreta a *entrada* do **usuário**: o usuário entra com diferentes dados no cliente (e este eventualmente envia-os ao servidor).

APLICAÇÃO WEB E SERVIDOR DE APLICAÇÃO

- Em computação, aplicação web designa, de forma geral, sistemas de informática projetados para utilização através de um navegador, através da internet ou aplicativos desenvolvidos utilizando tecnologias web HTML, JavaScript e CSS.Pode ser executado a partir de um servidor HTTP ou localmente, no dispositivo do usuário.
- Uma aplicação web também é definida em tudo que se é processado em algum servidor, exemplo: quando você entra em um e-commerce a página que você acessa antes de vir até seu navegador é processada em um computador ligado a internet que retorna o processamento das regras de negócio nele contido. Por isso se chama aplicação e não simplesmente site web ou um browser para permitir cookies de terceiros.

APLICAÇÃO WEB E SERVIDOR DE APLICAÇÃO

- Um Servidor de Aplicações (em inglês Applications Server), é um servidor que disponibiliza um ambiente para a instalação e execução de certas aplicações, centralizando e dispensando a instalação nos computadores clientes. Os servidores de aplicação também são conhecidos por middleware.
- O objetivo do servidor de aplicações é disponibilizar uma plataforma que separe do desenvolvedor de software algumas das complexidades de um sistema computacional. No desenvolvimento de aplicações comerciais, por exemplo, o foco dos desenvolvedores deve ser a resolução de problemas relacionados ao negócio da empresa, e não de questões de infraestrutura da aplicação. O servidor de aplicações responde a algumas questões comuns a todas as aplicações, como segurança, garantia de disponibilidade, balanceamento de carga e tratamento de exceções.

OWASP-TOP 10

O QUE É OWASP

- O Open Web Application Security Project (OWASP) é uma fundação sem fins lucrativos que trabalha para melhorar a segurança de aplicações, que auxiliam profissional da área de Desenvolvimento e Segurança, nos testes de segurança das suas aplicações, seja utilizando ferramentas ou por meio de frameworks e metodologias criadas por outros pesquisadores da área.

OWASP-TOP 10

OWASP Top 10 2013	±	OWASP Top 10 2017
A1 – Injection	→	A1:2017 – Injection
A2 – Broken Authentication and Session Management	→	A2:2017 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)	↘	A3:2013 – Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	∪	A4:2017 – XML External Entity (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017 – Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017 – Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	∪	A7:2017 – Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017 – Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017 – Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017 – Insufficient Logging & Monitoring [NEW, Comm.]

A1 - INJECTION

- Falhas de injeção, como SQL, NoSQL, OS/RCE e LDAP, ocorrem quando dados não confiáveis por meio de campos de entradas de dados, são enviados para um intérprete (Ex: Banco de dados MySQL) como parte de um comando ou consulta.

- Os dados hostis do invasor podem induzir o intérprete a executar comandos não intencionais ou acessar dados sem a devida autorização.

A2 – BROKEN AUTHENTICATION

- As funções de aplicativos relacionadas à autenticação e ao gerenciamento de sessões são frequentemente implementadas incorretamente
- Isso permite que os invasores comprometam senhas, chaves ou tokens de sessão ou explorem outras falhas de implementação para assumir a identidade de outros usuários temporária ou permanentemente.

A3 – SENSITIVE DATA EXPOSURE

- Muitos aplicativos da Web e APIs não protegem adequadamente dados confidenciais, como financeiro, assistência médica e PII (Dados pessoais identificáveis);
- Os invasores podem roubar ou modificar esses dados com pouca proteção para realizar fraudes no cartão de crédito, roubo de identidade ou outros crimes;
- Os dados confidenciais podem ser comprometidos sem proteção extra, como criptografia em rest ou em transit, e requer precauções especiais quando trocados com o navegador;

A4 – XML EXTERNAL ENTITY INJECTION XXE

- Muitos aplicativos modernos continuam usando o XML como uma forma de transferência e representação de dados devido à natureza dinâmica e multiplataforma do mesmo;

- No entanto, se o analisador XML não estiver configurado corretamente, ele poderá introduzir vulnerabilidades conhecidas como injeção de entidade externa XML, o que pode resultar na execução remota completa de código, resultando em comprometimento completo do servidor da web back-end.

A5 – BROKEN ACCESS CONTROL

- Muitas vezes, restrições sobre o que os usuários autenticados têm permissão para fazer geralmente não são aplicadas corretamente.
- Os invasores podem explorar essas falhas para acessar funcionalidades e / ou dados não autorizados, como acessar contas de outros usuários, visualizar arquivos confidenciais, modificar dados de outros usuários, alterar direitos de acesso etc.

A6 – FALHAS DE CONFIGURAÇÃO DE SEGURANÇA

- A configuração incorreta da segurança é o problema mais comum;
- Isso geralmente resulta em:
  - Configurações padrão inseguras, incompletas ou ad hoc, Armazenamento em nuvem aberta;
  - Cabeçalhos HTTP configurados incorretamente
  - Mensagens de erro detalhadas que contêm informações confidenciais;
- Não apenas todos os sistemas operacionais, estruturas, bibliotecas e aplicativos devem ser configurados com segurança, mas devem ser corrigidos / atualizados em tempo hábil.

A7 – CROSS SITE SCRIPTING (XSS)

- Falhas do XSS ocorrem sempre que um aplicativo inclui dados não confiáveis em uma nova página da Web sem validação ou escape adequados ou atualiza uma página da Web existente com dados fornecidos pelo usuário usando uma API do navegador que pode criar HTML ou JavaScript.
- O XSS permite que os atacantes executem scripts no navegador da vítima, que podem seqüestrar sessões do usuário, desfigurar sites ou redirecionar o usuário para sites maliciosos.

A8 – INSECURE DESERIALIZATION

- A desserialização insegura geralmente leva à execução remota de código. Mesmo que as falhas de desserialização não resultem em execução remota de código, elas podem ser usadas para executar ataques, incluindo ataques de repetição (Interceptação de tráfego e redirecionamento), ataques de injeção e ataques de escalonamento de privilégios.

A9 – USING COMPONENTES WITH KNOW VULNERABILITIES

- O uso de componentes como; bibliotecas, estruturas e outros módulos de software, são executados com os mesmos privilégios que o aplicativo.
- Se um componente vulnerável for explorado, esse ataque poderá facilitar a perda séria de dados ou a aquisição de servidores.
- Aplicativos e APIs que usam componentes com vulnerabilidades conhecidas podem minar as defesas de aplicativos e permitir vários ataques e impactos.

A10 – INSUFICIENTE LOGGING & MONITORING

- O registro e o monitoramento insuficientes, juntamente com a integração ausente ou ineficaz com a resposta a incidentes, permitem que os invasores continuem atacando os sistemas, mantenham a persistência, façam o giro para mais sistemas e violem, extraiam ou destruam dados.

- A maioria dos estudos de violação mostra que o tempo para detectar uma violação é superior a 200 dias, geralmente detectados por partes externas, em vez de processos ou monitoramento interno.

CONCEITO DE ATAQUES DE APLICAÇÕES WEB

PROCESSO DO PENTEST USANDO PTES

- O padrão de execução do teste de penetração consiste em sete (7) seções principais. Elas abrangem tudo relacionado a um teste de penetração - desde a comunicação inicial e o raciocínio por trás de um teste, até as fases de coleta de inteligência e modelagem de ameaças, nas quais os testadores estão trabalhando nos bastidores para entender melhor a organização testada, através da pesquisa de vulnerabilidades, exploração e pós-exploração, onde os conhecimentos técnicos de segurança dos testadores passam a ser combinados com o entendimento comercial do trabalho e, finalmente, com os relatórios, que capturam todo o processo, de uma maneira que faça sentido para o cliente e forneça o mais valor para isso.

- Esta versão pode ser considerada uma v1.0, pois os elementos principais do padrão são solidificados e são "testados na estrada" há mais de um ano na indústria. Um v2.0 está em andamento em breve e fornecerá um trabalho mais granular em termos de "níveis" - como nos níveis de intensidade nos quais cada um dos elementos de um teste de penetração pode ser realizado. Como nenhum pentest é como outro, e os testes vão desde o aplicativo da web ou o teste de rede mais mundanos até o engajamento completo da equipe em vermelho, esses níveis permitirão que uma organização defina quanta sofisticação eles esperam que seu adversário exiba e permita o testador para intensificar a intensidade nas áreas em que a organização mais precisa deles. Alguns dos trabalhos iniciais sobre "níveis" podem ser vistos na seção de coleta de informações.

PROCESSO DO PENTEST USANDO PTES

A seguir, estão as principais seções definidas pelo padrão como base para a execução do PenTest:

Interações pré-engajamento
Coleta de informações
Modelagem de ameaças
Análise de vulnerabilidade
Exploração
Pós-Exploração
Report

ESTRUTURA DE UM RELATÓRIO

- **Sumário Executivo:** O **sumário executivo** é a parte inicial do plano de negócios, que tem como objetivo resumir os principais tópicos do documento. Trata-se, portanto, de uma breve contextualização de cada seção do plano de negócios, de modo a proporcionar uma visão geral da empresa e de sua viabilidade;

- **Vulnerability Report:** No relatório de vulnerabilidade você tem uma descrição detalhada de vulnerabilidade, URL completo, uma prova de conceito ou detalhes de como é feito à exploração;

- **Remediation Report:** Na remediação você coloca instruções de como corrigir à vulnerabilidade, quais mecanismos de controle e segurança são essências e quais passos tomar para eliminar esse risco;

BASE64

- **Base64 é um método para codificação de dados para transferência na Internet. É utilizado frequentemente para transmitir dados binários por meios de transmissão que lidam apenas com texto, como por exemplo para enviar arquivos anexos por e-mail. É constituído por 64 caracteres que deram origem ao seu nome;**

- **O esquema de codificação Base64 é composto por dígitos de [0-9] e letras latinas, maiúsculas e minúsculas [a-z e A-Z], para dar um total de 62. Para concluir o conjunto de caracteres para 64, existem os caracteres de mais (+) e de barra (/). No entanto, implementações diferentes podem usar outros valores para os dois caracteres mais recentes e o usado para preenchimento (=);**

COOKIE

- Um ***cookie***, no âmbito do protocolo de comunicação HTTP usado na Internet, é um pequeno arquivo de computador ou pacote de dados enviados por um sítio de Internet para o navegador do usuário, quando o utilizador visita o site. Cada vez que o usuário visita o site novamente, o navegador envia o *cookie* de volta para o servidor para notificar atividades prévias do usuário. Os *cookies* foram concebidos para serem um mecanismo confiável para que sítios se lembrem de informações da atividade do usuário, como senhas gravadas, itens adicionados no carrinho de compras em uma loja online, hiperligações que foram clicadas anteriormente, entre outros. Assim, melhoram a navegação, aumentando a eficiência da busca.

COOKIE - FUNCIONAMENTO

- Quando o servidor deseja activar um cookie no cliente, envia uma linha no cabeçalho HTTP iniciada por Set-Cookie: ...
- A partir desse momento, consoante as opções especificadas pelo cookie, o cliente irá enviar no seu cabeçalho HTTP dos pedidos uma linha contendo os cookies relevantes, iniciada por Cookie: ....
- Entre os parâmetros dos cookies estão: o tempo de vida (a data para o cookie "expirar a validade") e o domínio, ou grupo de páginas a que o cookie se aplica. Por exemplo, é possível fazer com que um cookie seja aplicado apenas a endereços iniciados por de maneira que esse mesmo cookie já não se aplique para skins, por exemplo.

COOKIE – FUNCIONAMENTO 2

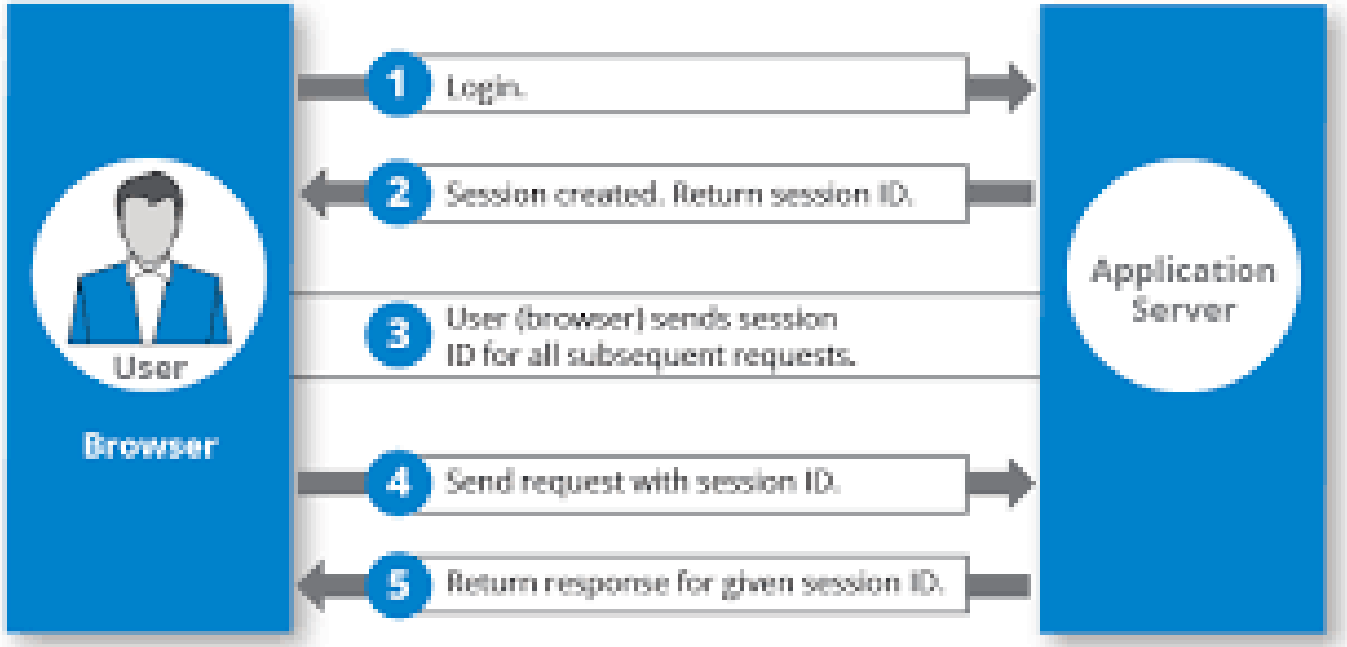
- Se não especificada a data de validade para o cookie, ele irá expirar assim que o usuário fechar o navegador.

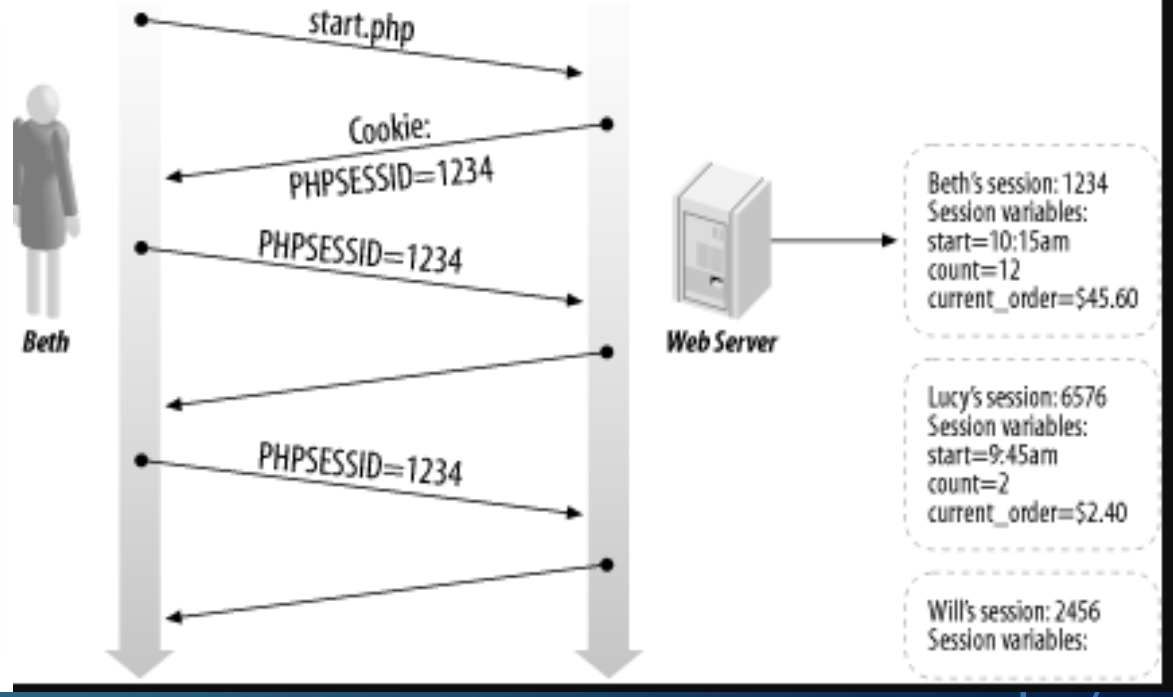
- Em JavaScript (embutido no HTML da página acessada), podemos criar um script para manipulá-los. Utilizamos "document.cookie" (sem aspas). Em ASP, podemos utilizar cookies por meio dos objetos Response e Request. Em PHP, os cookies são tratados por meio da função setcookie(). Esta deverá vir antes de qualquer dado ser enviado ao navegador, devido ao fato de os cookies fazerem parte do cabeçalho HTTP.

SESSION ID

- Na ciência da computação , um **identificador de sessão** , **ID de sessão** ou **token de sessão** é um dado usado em comunicações de rede (geralmente por HTTP) para identificar uma sessão, uma série de trocas de mensagens relacionadas. Os identificadores de sessão tornam-se necessários nos casos em que a infraestrutura de comunicações usa um protocolo sem estado, como HTTP. Por exemplo, um comprador que visita o site de um vendedor deseja coletar vários artigos em um carrinho de compras virtual e finalizar a compra acessando a página de checkout do site. Isso geralmente envolve uma comunicação contínua, em que várias páginas da web são solicitadas pelo cliente e enviadas a eles pelo servidor. Em tal situação, é vital acompanhar o estado atual do carrinho do comprador, e um ID de sessão é uma maneira de atingir esse objetivo;
- Um ID de sessão geralmente é concedido a um visitante em sua primeira visita a um site. É diferente do ID do usuário, pois as sessões geralmente duram pouco (elas expiram após um tempo predefinido de inatividade que pode ser de minutos ou horas) e podem se tornar inválidas após o cumprimento de uma determinada meta (por exemplo, uma vez que o comprador finalizou o pedido, ele não pode usar o mesmo ID da sessão para adicionar mais itens);

SESSION ID - EXAMPLE





WHOIS

- WHOIS é um protocolo TCP específico para consultar informações de contato e DNS sobre entidades na internet. Uma entidade na internet pode ser um nome de domínio, um endereço IP ou um AS;
- Para cada entidade, o protocolo WHOIS apresenta três tipos de contato: Contato Administrativo (*Admin Contact*), Contato Técnico (*Technical Contact*) e Contato de Cobrança (*Billing Contact*). Estes contatos são informações de responsabilidade do provedor de internet, que as nomeia de acordo com as políticas internas de sua rede;
- Para os registros de domínios, os usuarios tem a opção de optar por um Whois privado, que esconde os dados do dono do domínio;

WHOIS

Command Prompt

```
C:\temp>
C:\temp>whoiscl -r microsoft.com

WHOIS Server: whois.opensrs.net

Registrant:
 Microsoft Corporation
 One Microsoft Way
 Redmond, WA 98052
 US

 Domain name: MICROSOFT.COM

 Administrative Contact:
    Administrator, Domain  domains@microsoft.com
    One Microsoft Way
    Redmond, WA 98052
    US
    +1.4258828080
 Technical Contact:
    Hostmaster, MSN  msnhst@microsoft.com
    One Microsoft Way
    Redmond, WA 98052
    US
    +1.4258828080
```

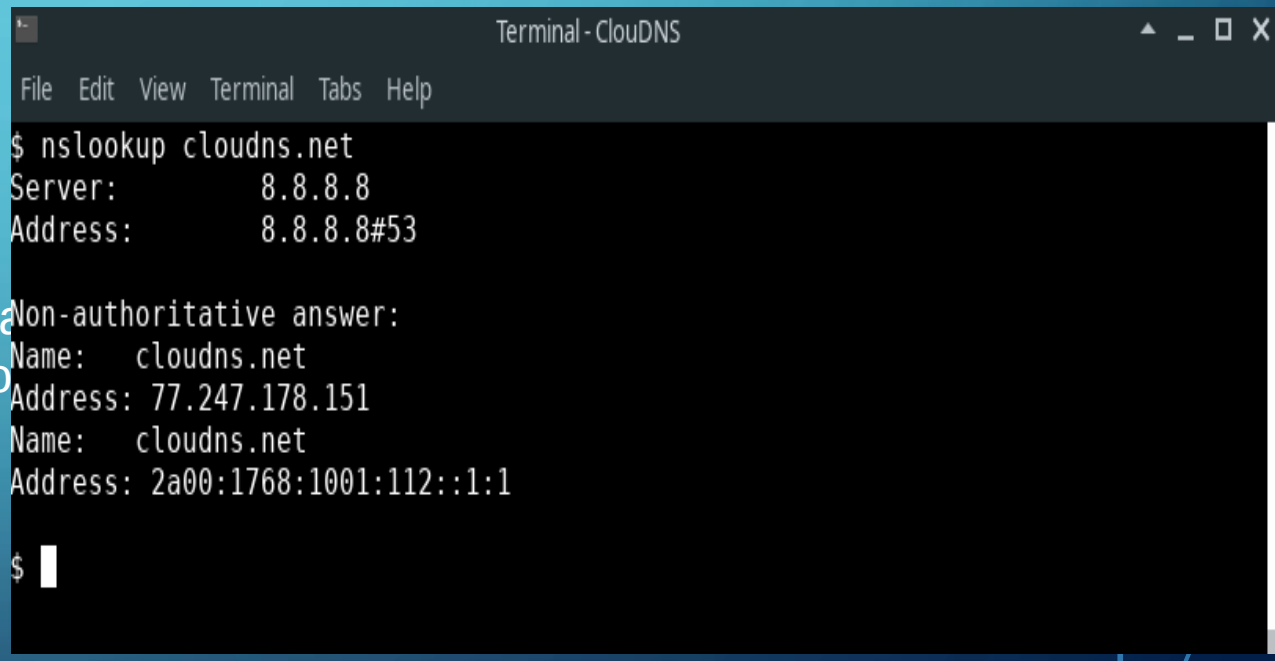
```
root@fwhwin:~# whois flashwebhost.com
The program 'whois' is currently not installed. You can install it by typing:
apt-get install whois
root@fwhwin:~# apt-get install whois
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  whois
0 upgraded, 1 newly installed, 0 to remove and 267 not upgraded.
Need to get 29.5 kB of archives.
After this operation, 152 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu/ trusty/main whois i386 5.1.1 [29.5 kB]
Fetched 29.5 kB in 0s (34.0 kB/s)
Selecting previously unselected package whois.
(Reading database ... 116005 files and directories currently installed.)
Preparing to unpack .../archives/whois_5.1.1_i386.deb ...
Unpacking whois (5.1.1) ...
Processing triggers for man-db (2.6.7.1-1) ...
Setting up whois (5.1.1) ...
root@fwhwin:~# whois flashwebhost.com

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: FLASHWEBHOST.COM
   Registrar: PDR LTD. D/B/A PUBLICDOMAINREGISTRY.COM
   Whois Server: whois.PublicDomainRegistry.com
   Referral URL: http://www.PublicDomainRegistry.com
   Name Server: NS58.HOSTHAT.COM
   Name Server: NS59.HOSTHAT.COM
   Status: clientTransferProhibited
   Updated Date: 11-oct-2013
   Creation Date: 01-nov-2001
   Expiration Date: 01-nov-2015
```

DNS

- DNS significa Domain Name Server, em português, Servidor de Nomes de Domínios. Esse sistema nasceu para facilitar a nossa navegação. Dessa forma, ele traduz domínios de sites em IP's (Internet Protocol), que são sequências numéricas de identificação de um domínio em seu servidor. Sempre que fazemos um registro de domínio, precisamos em seguida configurar o servidor DNS;
- Graças a esse sistema, você encontra facilmente um site usando o seu nome amigável. Por exemplo, o sistema DNS permite um usuário acessar um site como www.exemplo.com.br, e não uma sequência numérica, como 200.123.123.15;
- Sem o DNS, nossa navegação seria baseada em números gigantescos! Em suma, não teríamos sites com nomes amigáveis como Google.com , Homehost.com.br , Yahoo.com e etc;
- Em resumo, o servidor DNS traduz um nome de domínio para seu endereço IP. Por exemplo, o cliente acessa www.homehost.com.br , e o servidor DNS traduz este nome para um IP na internet;

NSLOOKUP

- **nslookup** é uma ferramenta, comum ao Windows e ao Linux, utilizada para se obter informações sobre registros de DNS de um determinado domínio, host ou IP.
- Em uma busca **nslookup** padrão, o servidor DNS do provedor de acesso é consultado, e retorna as informações sobre o domínio ou host pesquisado.
- A informação "Non-authoritative answer" (Não é resposta de autorização) significa que o servidor DNS do provedor de acesso não responde por este domínio, em outras palavras, isto significa que uma consulta externa foi realizada, aos servidores DNS do domínio WIKIPEDIA.ORG.



https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/nslookup

NETCAT

```
root@kali:~# nc 192.168.179.146 80
HEAD / HTTP/1.0
HTTP/1.1 400 Bad Request
Date: Tue, 01 Aug 2017 16:26:23 GMT
Server: Apache/2.4.25 (Debian)
Content-Length: 301
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.25 (Debian) Server at 127.0.1.1 Port 80</address>
</body></html>
```

- Com netcat, você consegue coletar o Banner do servidor de aplicação e mais outros detalhes de comunicação, como o tipo de aplicação sendo utilizada, seja PHP, ASP, JSP, JBOSS e etc. Além de ser bem útil para testar em outras portas também, como servidor FTP e SMTP.

https://www.sans.org/security-resources/sec560/netcat_cheat_sheet_v1.pdf

WHATWEB

```
File  Edit  View  Search  Terminal  Help
root@kali:~# whatweb www.facebook.com
/usr/lib/ruby/1.9.1/rubygems/custom_require.rb:36:in `require': icon
v will be deprecated in the future, use String#encode instead.
http://www.facebook.com [302] Country[IRELAND][IE], IP[31.13.79.246]
, RedirectLocation[https://www.facebook.com/], UncommonHeaders[x-fb-
debug]
https://www.facebook.com/ [200] Country[IRELAND][IE], HTML5, IP[31.1
3.79.246], Meta-Refresh-Redirect[/?_fb_noscript=1], PasswordField[pa
ss,reg_passwd__], Script, UncommonHeaders[strict-transport-security,
x-frame-options,x-xss-protection,x-content-type-options,x-fb-debug],
 X-Frame-Options[DENY], X-XSS-Protection[0]
https://www.facebook.com/?_fb_noscript=1 [200] Cookies[noscript], Co
untry[IRELAND][IE], HTML5, IP[31.13.79.246], PasswordField[pass,reg_
passwd__], Script, UncommonHeaders[strict-transport-security,x-frame
-options,x-xss-protection,x-content-type-options,x-fb-debug], X-Fram
e-Options[DENY], X-XSS-Protection[0]
root@kali:~#
```

WhatWeb identifica sites. Seu objetivo é responder à pergunta "O que é esse site?". O WhatWeb reconhece tecnologias da Web, incluindo sistemas de gerenciamento de conteúdo (CMS), plataformas de blogs, pacotes de estatística / análise, bibliotecas JavaScript, servidores da Web e dispositivos incorporados. WhatWeb tem mais de 1700 plugins, cada um para reconhecer algo diferente. O WhatWeb também identifica números de versão, endereços de email, IDs de conta, módulos de estrutura da web, erros de SQL e muito mais.

https://tools.kali.org/web-applications/whatweb

ENUMERAÇÃO E SCANNING WEB

- A enumeração sempre começa com levantamento de todos subdominios disponíveis, páginas do websites, paineis de administrativos e etc;
- E o Scanner Auxilia na detecção de serviços, vulnerabilidades, tecnologias utilizadas em uma aplicação e etc;
- Essas duas etapas são essências em um teste de vulnerabilidades e quanto mais informações você levanter, melhor será;

ENUMERAÇÃO E SCANNING WEB - CONTEÚDOS

- https://resources.infosecinstitute.com/process-scanning-and-enumeration/#gref
- https://booksite.elsevier.com/samplechapters/9781597496278/Chapter_3.pdf
- https://thebe0vlksaga.com/2019/03/31/ethical-hacking-101-web-server-enumeration/
- https://www.greycampus.com/opencampus/ethical-hacking/enumeration-and-its-types
- https://hkh4cks.com/blog/2018/01/22/common-enumeration-tools/
- https://www.hackingtutorials.org/scanning-tutorials/enumerate-webserver-directories-with-nmap/
- https://attackd0gz.com/2019/11/04/web-application-enumeration/
- https://www.youtube.com/watch?v=ngOmkZ3U2e4

ENUMERAÇÃO E SCANNING WEB – CONTEÚDOS 2

- https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/
- https://owasp.org/www-chapter-belgium/assets/2012/2012-09-12/OWASP-Modern_Information_Gathering.pdf
- https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v3.pdf
- https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf
- https://pt.slideshare.net/KZAbv/owasp-modern-information-gathering
- https://www.futurelearn.com/courses/ethical-hacking-an-introduction/0/steps/71525

ENUMERAÇÃO E SCANNING WEB

- A enumeração sempre começa com levantamento de todos subdominios disponíveis, páginas do websites, paineis de administrativos e etc;
- E o Scanner Auxilia na detecção de serviços, vulnerabilidades, tecnologias utilizadas em uma aplicação e etc;
- Essas duas etapas são essências em um teste de vulnerabilidades e quanto mais informações você levanter, melhor será;

ENUMERAÇÃO E SCANNING WEB

- Recon-ng
- DNSRecon
- Knock Knock (https://github.com/guelfoweb/knock)
- Dirb (https://tools.kali.org/web-applications/dirb)
- GHDB https://www.exploit-db.com/google-hacking-database
- ReconOfJaah (https://github.com/OfJAAH/ReconOfJAAAH)
- W3af
- Wfuzz
- Burp Suite
- Apache-users
- Nmap
- Fierce
- Netcraft (https://www.netcraft.com/)
- Shodan.io e Censys
- TheHaverster

ENUMERAÇÃO E SCANNING WEB

- WPSCAN
- JOOMSCAN
- NIKTO
- https://geekflare.com/online-scan-website-security-vulnerabilities/
- https://owasp.org/www-community/Vulnerability_Scanning_Tools

PROXY WEB

- O **proxy web** (também conhecido como *filtro de conteúdo* e *web filter*) surgiu a partir da necessidade de conectar uma rede local a internet por meio de um equipamento que compartilha conexões com as demais máquinas. Assim, o proxy web é um serviço que atua como intermediário entre um dispositivo e os serviços de internet. No momento em que o endereço de um site é digitado no navegador, a solicitação é enviada ao proxy, que então realiza esta solicitação ao servidor no qual o site é hospedado e devolve o resultado para o usuário.
- Desta forma, é possível ter controle absoluto sobre o tráfego da internet e realizar bloqueios (ou liberações) de acordo com as políticas estabelecidas pela empresa.

https://ostec.blog/seguranca-perimetro/proxy-web-tipos-e-terminologias

BURP SUITE

- Burp Suite é um software desenvolvido em Java pela PostWigger, para a realização de testes de segurança em aplicações web. O Burp Suite é dividido em diversos componentes:
- https://portswigger.net/burp/documentation/desktop/getting-started
- https://portswigger.net/burp/documentation
- https://portswigger.net/burp/documentation/desktop/tools
- https://portswigger.net/training
- https://portswigger.net/web-security
- https://www.udemy.com/course/burp-suite/

WEB EXPLOITATION - TOOLS

- Reconhecimento https://github.com/OfJAAH/ReconOfJAAAH
- Scanner de Vulnerabilidades: https://github.com/sullo/nikto
- Whois: https://registro.br/tecnologia/ferramentas/whois/ / https://whois.net/ / https://who.is/
- Burp Suite: https://portswigger.net/burp
- cURL: https://curl.haxx.se/
- Hydra: https://github.com/vanhauser-thc/thc-hydra
- Dirb Web Crawler: https://tools.kali.org/web-applications/dirb
- Wfuzz Fuzzing: https://github.com/xmendez/wfuzz
- Whatweb Info Collection: https://tools.kali.org/web-applications/whatweb
- OSINT Framework: https://osintframework.com/

WEB EXPLOITATION - TOOLS 2

- Reconhecimento 2: https://github.com/guelfoweb/knock
- XSS Audit: https://github.com/s0md3v/XSStrike
- SQLMap SQL Audit: http://sqlmap.org/
- Nmap: https://nmap.org/
- Shodan: https://www.shodan.io/
- Censys: https://censys.io/
- HTTP Basic Honeypot: https://github.com/bjeborn/basic-auth-pot
- WPScan Wordpress Audit: https://github.com/wpscanteam/wpscan
- DNSRecon: https://tools.kali.org/information-gathering/dnsrecon
- Reconhecimento 3: http://virustotal.com/
- Webshells: https://github.com/xl7dev/WebShell (Warning: Revise o código antes)
- Google Hacking and GHDB: https://www.exploit-db.com/google-hacking-database

VULNERABILIDADES WEB

HTML INJECTION

- A injeção de HTML é um ataque semelhante ao XSS (Cross-site Scripting). Enquanto na vulnerabilidade XSS, o invasor pode injetar e executar o código Javascript, o ataque por injeção de HTML permite apenas a injeção de determinadas tags HTML. Quando um aplicativo não manipula adequadamente os dados fornecidos pelo usuário, um invasor pode fornecer código HTML válido, geralmente por meio de um valor de parâmetro, e injetar seu próprio conteúdo na página. Esse ataque geralmente é usado em conjunto com alguma forma de engenharia social, pois o ataque explora uma vulnerabilidade baseada em código e a confiança do usuário.

Cenáriode ataque (OWASP)

Um possível cenário de ataque é demonstrado abaixo:

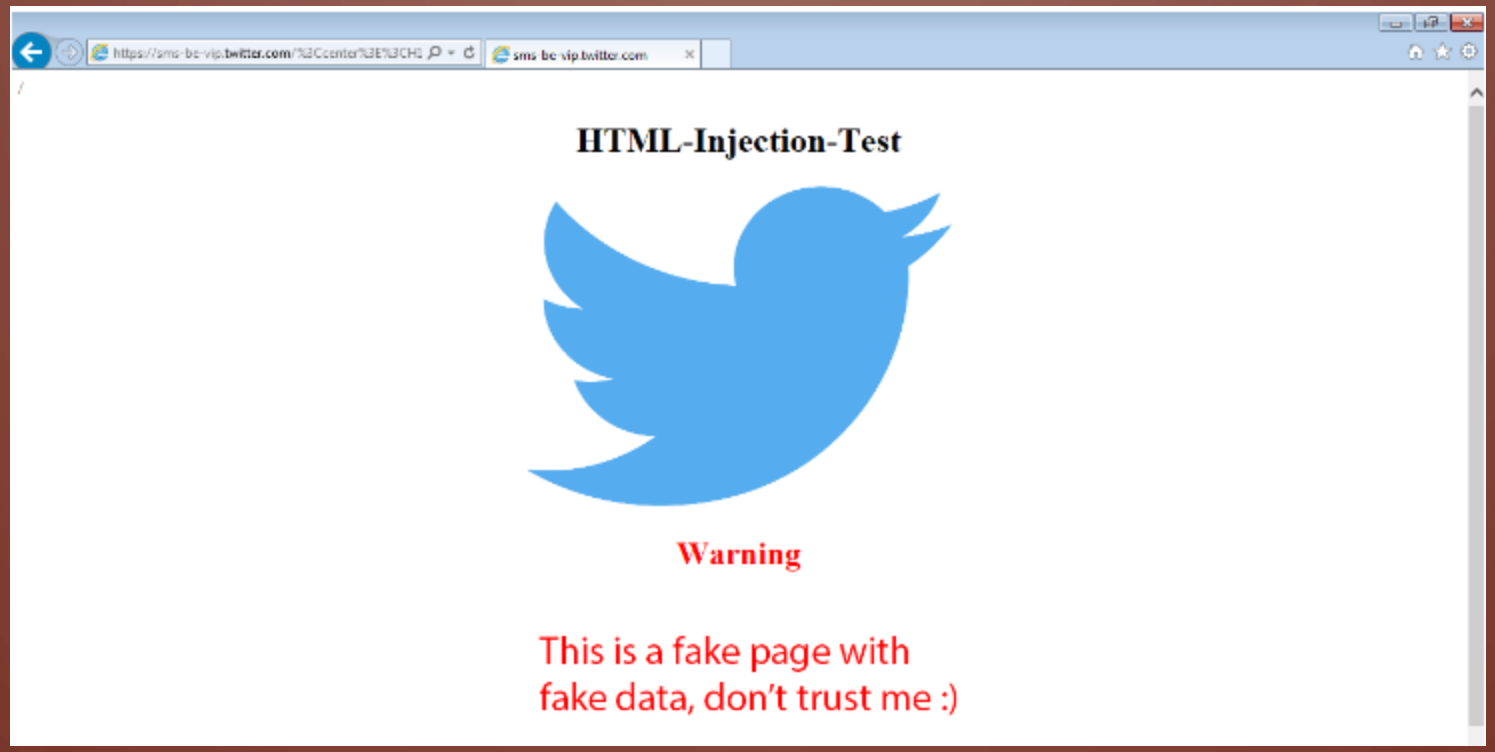
- O invasor descobre a vulnerabilidade da injeção e decide usar um ataque de injeção HTML
- O atacante cria um link malicioso, incluindo o conteúdo HTML injetado, e envia para um usuário por email
- O usuário visita a página devido à localização de um domínio confiável
- O HTML injetado pelo atacante é renderizado e apresentado ao usuário solicitando um nome de usuário e senha
- O usuário digita um nome de usuário e senha, que são enviados ao servidor do invasor

https://www.acunetix.com/vulnerabilities/web/html-injection/

HTML INJECTION - EXEMPLOS

Outros Exemplos: https://medium.com/@elberandre/bugbounty-types-html-injection-via-email-8409b6dc4d18

HTML INJECTION - EXEMPLOS



Outros Exemplos: https://hackerone.com/reports/150179

CROSS SITE SCRIPTING

- Os ataques de cross-site scripting (XSS) são um tipo de injeção, na qual os scripts maliciosos são injetados em sites benignos e confiáveis. Os ataques XSS ocorrem quando um invasor usa um aplicativo da Web para enviar código malicioso, geralmente na forma de um script do lado do navegador, para um usuário final diferente. As falhas que permitem que esses ataques sejam bem-sucedidos são bastante difundidas e ocorrem em qualquer lugar em que um aplicativo Web use entrada de um usuário na saída gerada sem validá-lo ou codificá-lo.
- Um invasor pode usar o XSS para enviar um script mal-intencionado a um usuário inocente. O navegador do usuário final não tem como saber que o script não deve ser confiável e o executará. Como ele acredita que o script veio de uma fonte confiável, o script mal-intencionado pode acessar todos os cookies, tokens de sessão ou outras informações confidenciais retidas pelo navegador e usadas com esse site. Esses scripts podem até reescrever o conteúdo da página HTML.

CROSS SITE SCRIPTING

Os ataques de script entre sites (XSS) ocorrem quando:

1. Os dados entram em um aplicativo da Web por meio de uma fonte não confiável, com mais freqüência uma solicitação da web.
2. Os dados são incluídos no conteúdo dinâmico enviado a um usuário da Web sem ser validado para conteúdo malicioso.

- O conteúdo malicioso enviado ao navegador da Web geralmente assume a forma de um segmento de JavaScript, mas também pode incluir HTML, Flash ou qualquer outro tipo de código que o navegador possa executar. A variedade de ataques baseados no XSS é quase ilimitada, mas geralmente incluem a transmissão de dados privados, como cookies ou outras informações da sessão, ao invasor, redirecionando a vítima para o conteúdo da Web controlado pelo invasor ou executando outras operações maliciosas na máquina do usuário sob o disfarce do site vulnerável.

CROSS SITE SCRIPTING - TIPOS

Ataques XSS armazenados

- Ataques armazenados são aqueles em que o script injetado é armazenado permanentemente nos servidores de destino, como em um banco de dados, em um fórum de mensagens, log de visitantes, campo de comentários etc. A vítima recupera o script malicioso do servidor quando solicita o armazenamento. em formação. O XSS armazenado também é conhecido como XSS Persistente ou Tipo I.

Ataques XSS refletidos

- Ataques refletidos são aqueles em que o script injetado é refletido no servidor da Web, como em uma mensagem de erro, resultado da pesquisa ou qualquer outra resposta que inclua parte ou toda a entrada enviada ao servidor como parte da solicitação. Ataques refletidos são entregues às vítimas por outra rota, como em uma mensagem de email ou em outro site. Quando um usuário é enganado a clicar em um link malicioso, enviar um formulário especialmente criado ou até mesmo navegar em um site malicioso, o código injetado viaja para o site vulnerável, o que reflete o ataque ao navegador do usuário. O navegador então executa o código porque veio de um servidor "confiável". O XSS refletido também é conhecido como XSS não persistente ou tipo II.

https://owasp.org/www-community/attacks/xss/

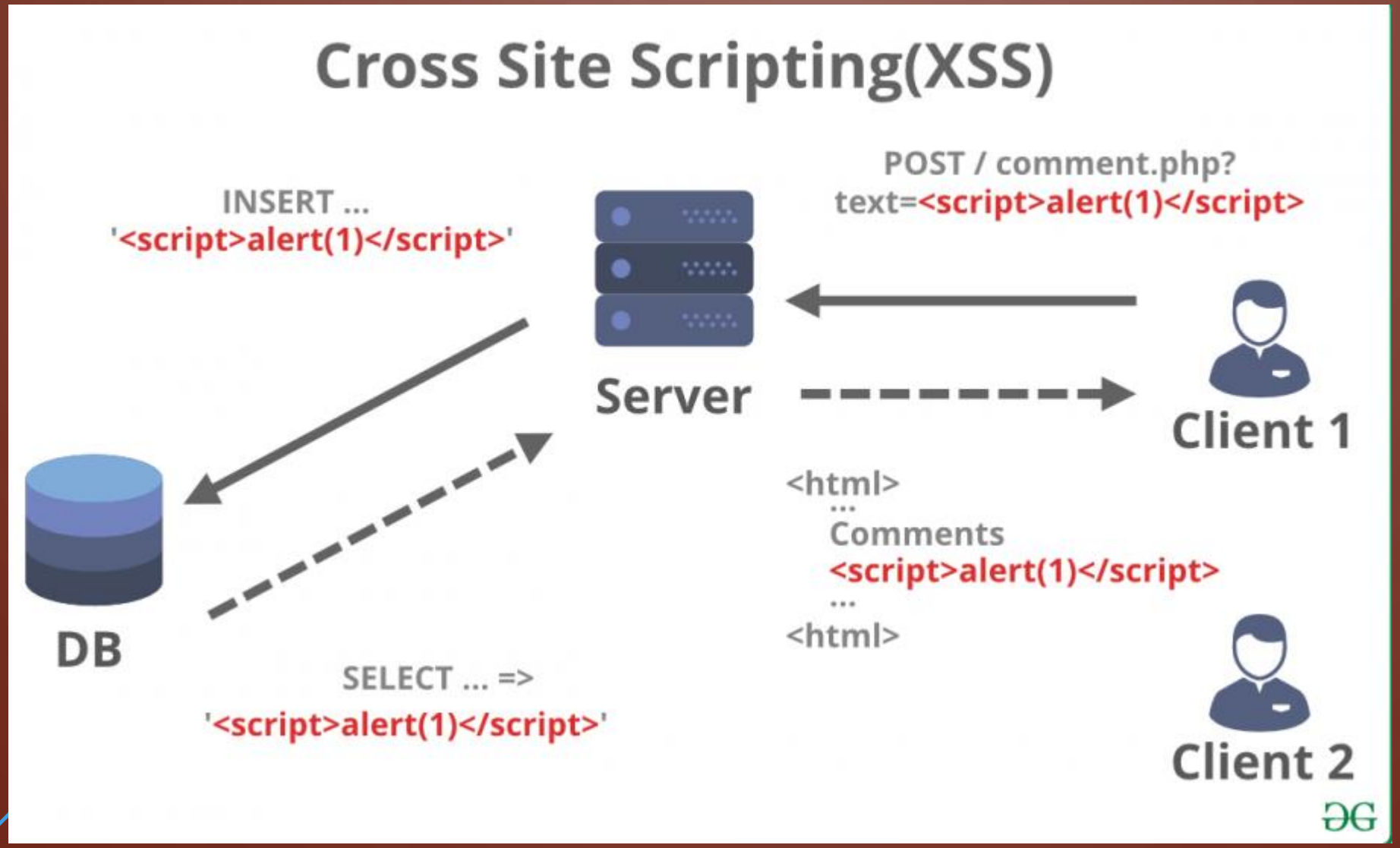
CROSS SITE SCRIPTING - TIPOS

Ataques XSS baseado em DOM

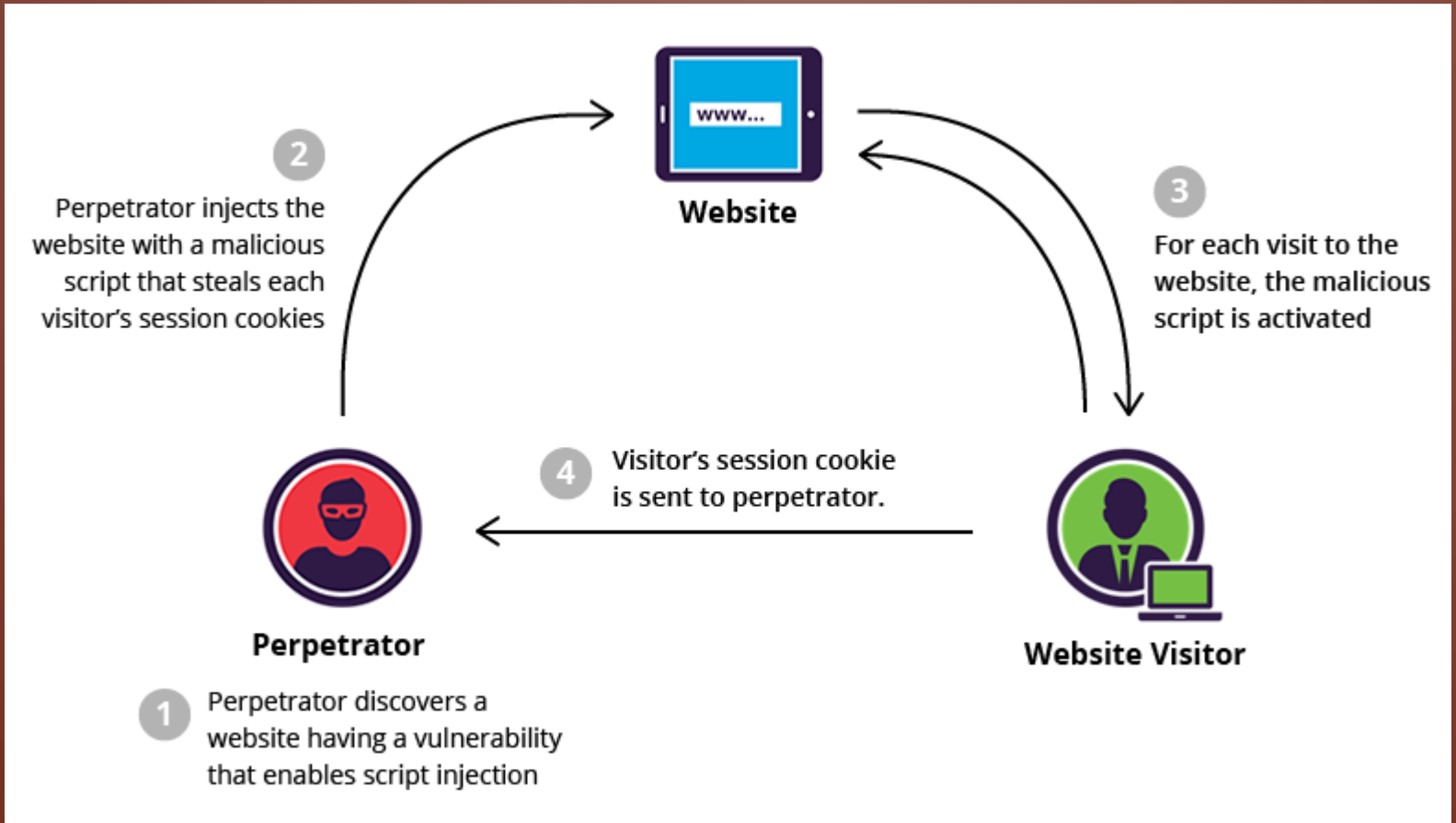
- O XSS baseado em DOM (ou como é chamado em alguns textos, "tipo 0 XSS") é um ataque XSS em que a carga útil do ataque é executada como resultado da modificação do "ambiente" DOM no navegador da vítima usado pelo lado do cliente original script, para que o código do lado do cliente seja executado de maneira "inesperada". Ou seja, a própria página (que é a resposta HTTP) não é alterada, mas o código do lado do cliente contido na página é executado de maneira diferente devido às modificações maliciosas que ocorreram no ambiente DOM.

https://owasp.org/www-community/attacks/DOM_Based_XSS

CROSS SITE SCRIPTING - EXEMPLOS



CROSS SITE SCRIPTING - EXEMPLOS



CROSS SITE SCRIPTING - EXEMPLOS

https://pentester.land/list-of-bug-bounty-writeups.html

https://medium.com/bugbountywriteup/stored-xss-in-bug-bounty-13c08e6f5636

https://pethuraj.com/blog/google-bug-bounty-writeup/

https://medium.com/@corneacristian/top-25-xss-bug-bounty-reports-b3c90e2288c8

https://www.youtube.com/watch?v=YMhhmLCefnw

https://www.youtube.com/watch?v=XHf3y-4H_AU

https://www.youtube.com/watch?v=lhPsBMBDFcg

https://www.youtube.com/watch?v=aybCeXhEjsA

https://www.youtube.com/watch?v=FqNxYDSjovc

https://www.youtube.com/watch?v=hb_qENFUdOk

CROSS SITE REQUEST FORGERY

- A falsificação de solicitação entre sites (CSRF) é um ataque que força um usuário final a executar ações indesejadas em um aplicativo Web no qual eles estão atualmente autenticados. Com uma pequena ajuda da engenharia social (como o envio de um link por email ou bate-papo), um invasor pode induzir os usuários de um aplicativo da Web a executar ações de sua escolha. Se a vítima for um usuário normal, um ataque CSRF bem-sucedido pode forçar o usuário a executar solicitações de alteração de estado, como transferência de fundos, alteração de endereço de email e assim por diante. Se a vítima for uma conta administrativa, o CSRF poderá comprometer todo o aplicativo da web.

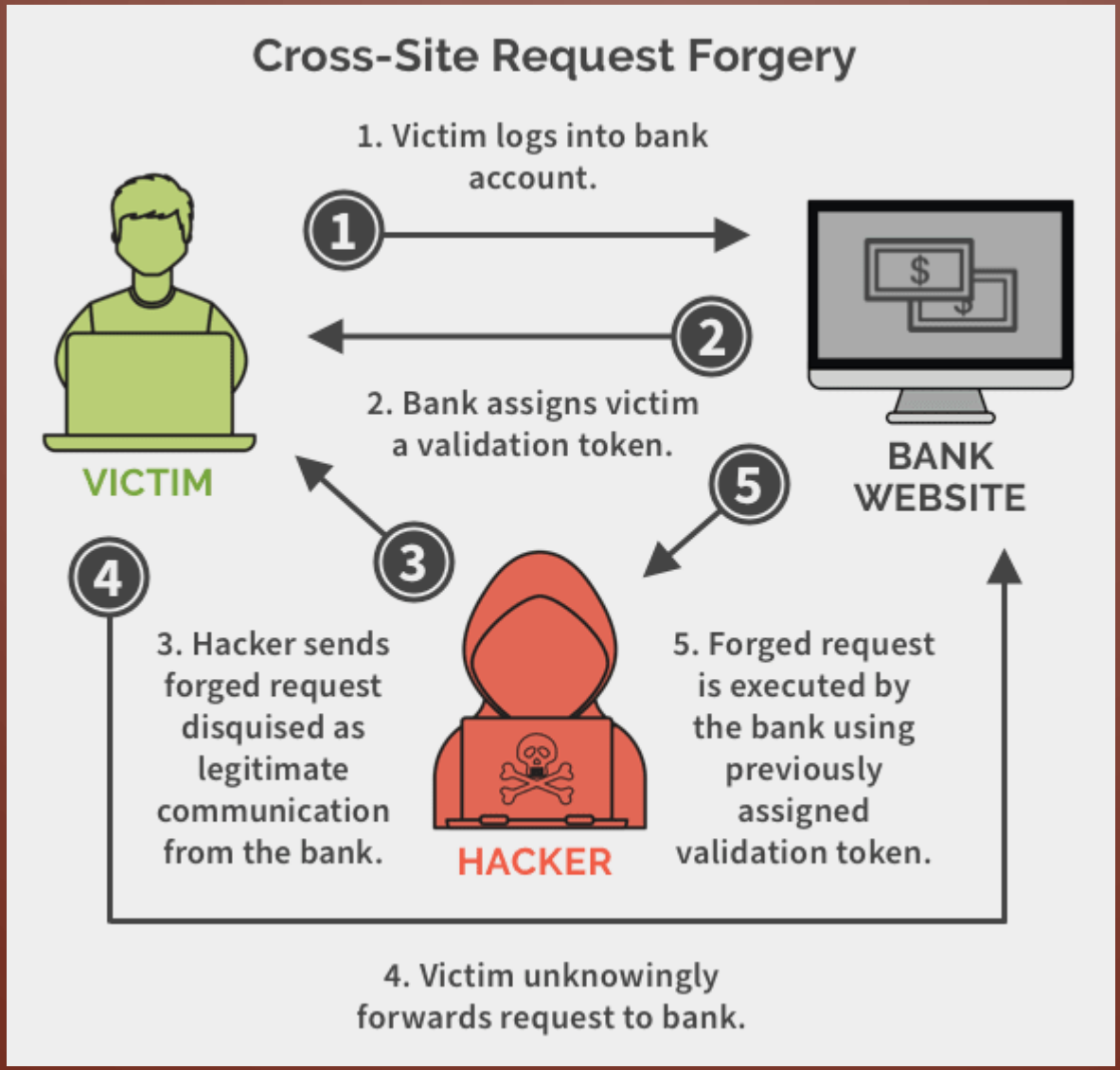
https://owasp.org/www-community/attacks/csrf

CROSS SITE REQUEST FORGERY

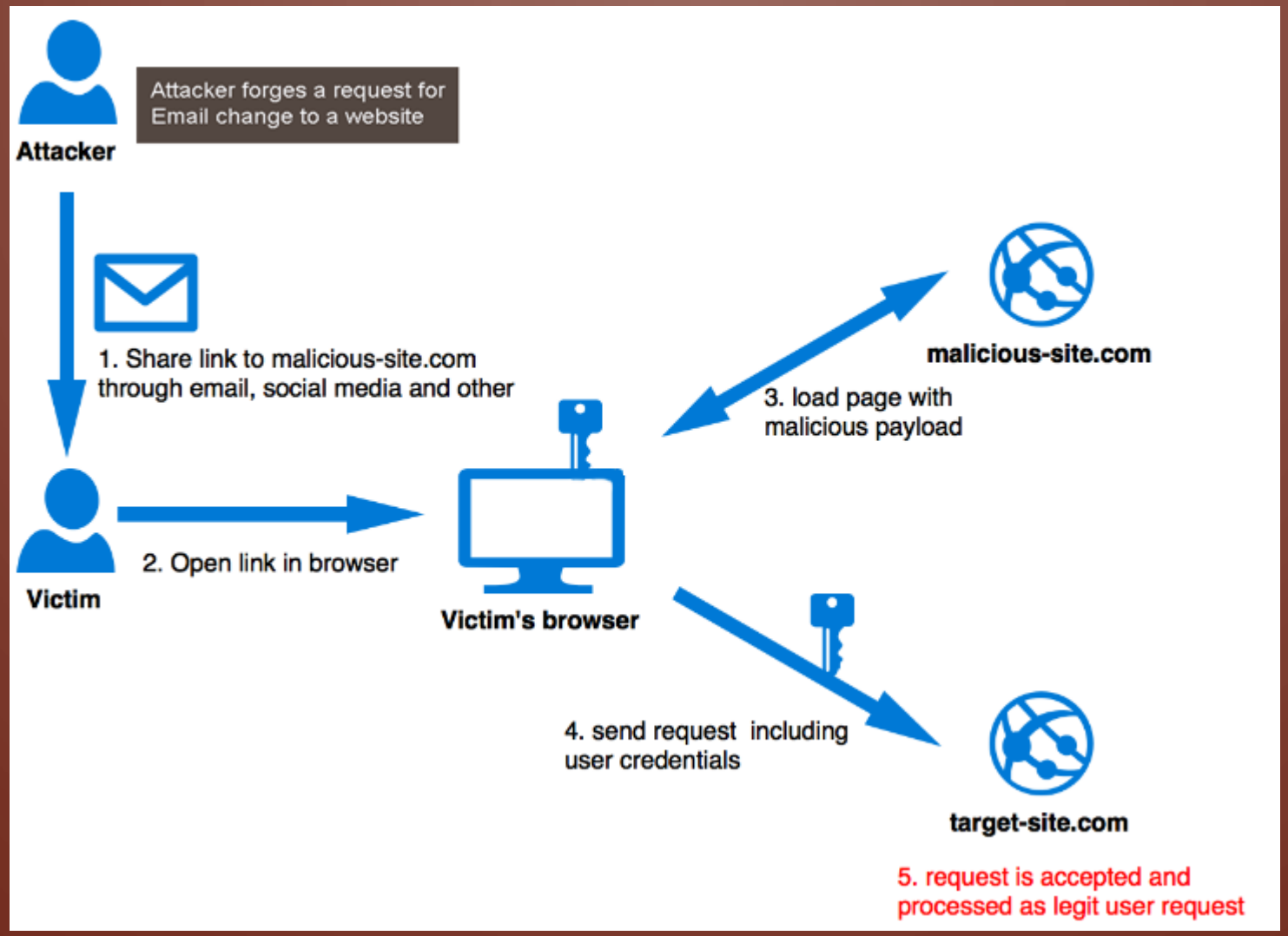
- Existem várias maneiras pelas quais um usuário final pode ser enganado para carregar informações ou enviar informações para um aplicativo da web. Para executar um ataque, precisamos primeiro entender como gerar uma solicitação maliciosa válida para nossa vítima executar. Vamos considerar o seguinte exemplo: Alice deseja transferir US $ 100 para Bob usando o aplicativo da web *bank.com* vulnerável ao CSRF. Maria, uma atacante, quer convencer Alice a enviar o dinheiro para Maria. O ataque compreenderá as seguintes etapas:

1. construindo um URL ou script de exploração
2. enganando Alice para executar a ação com engenharia social

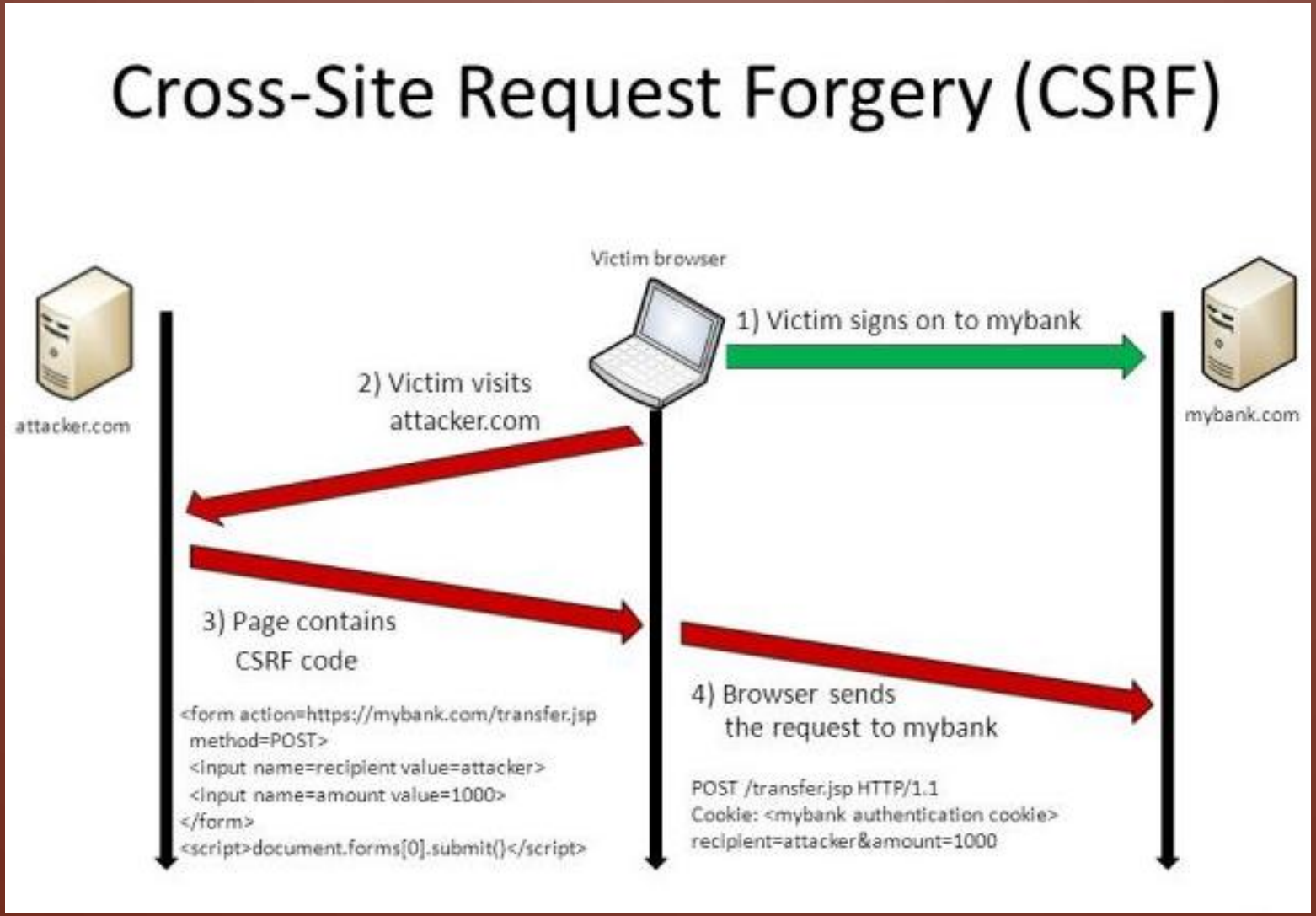
CROSS SITE REQUEST FORGERY - EXEMPLOS



CROSS SITE REQUEST FORGERY - EXEMPLOS



CROSS SITE REQUEST FORGERY - EXEMPLOS



CROSS SITE REQUEST FORGERY - EXEMPLOS

- https://www.youtube.com/watch?v=13QPmRuhbhU
- https://www.youtube.com/watch?v=vRBihr41JTo
- https://www.youtube.com/watch?v=eWEgUcHPle0
- https://www.youtube.com/watch?v=1NO4I28J-0s
- https://www.youtube.com/watch?v=jnNa4i01aok
- https://www.youtube.com/watch?v=lqhyx6-ky1k
- https://hackerone.com/reports/419891
- https://www.imperva.com/learn/application-security/csrf-cross-site-request-forgery/
- https://www.infosec.com.br/cross-site-request-forgery/

UNRESTRICTED FILE UPLOAD

- Os arquivos enviados representam um risco significativo para os aplicativos. O primeiro passo em muitos ataques é obter algum código no sistema a ser atacado. Então o ataque precisa apenas encontrar uma maneira de executar o código. O uso de um upload de arquivo ajuda o invasor a executar a primeira etapa.
- As consequências do upload irrestrito de arquivos podem variar, incluindo controle completo do sistema, um sistema de arquivos ou banco de dados sobrecarregado, encaminhamento de ataques para sistemas de back-end, ataques do lado do cliente ou desfiguração simples. Depende do que o aplicativo faz com o arquivo carregado e, principalmente, de onde está armazenado.
- Existem realmente duas classes de problemas aqui. O primeiro é com os metadados do arquivo, como o caminho e o nome do arquivo. Eles geralmente são fornecidos pelo transporte, como codificação HTTP com várias partes. Esses dados podem induzir o aplicativo a substituir um arquivo crítico ou a armazená-lo em um local incorreto. Você deve validar os metadados com muito cuidado antes de usá-los.
- A outra classe de problemas está no tamanho ou no conteúdo do arquivo. A variedade de problemas aqui depende inteiramente do uso do arquivo. Veja os exemplos abaixo para obter algumas idéias sobre como os arquivos podem ser mal utilizados. Para se proteger contra esse tipo de ataque, você deve analisar tudo o que seu aplicativo faz com os arquivos e pensar cuidadosamente sobre o processamento e os intérpretes envolvidos.

UNRESTRICTED FILE UPLOAD

- **Ataques na plataforma de aplicativos:**
- Carregar arquivo .jsp na árvore da web - código jsp executado como usuário da web
- Carregar arquivo .gif para ser redimensionado - falha na biblioteca de imagens explorada
- Carregar arquivos enormes - negação de serviço do espaço no arquivo
- Carregar arquivo usando nome ou caminho malicioso - substitua um arquivo crítico
- Carregar arquivo contendo dados pessoais - outros usuários acessam
- Carregar arquivo contendo "tags" - as tags são executadas como parte de serem "incluídas" em uma página da web
- Carregar arquivo .rar a ser verificado pelo antivírus - comando executado em um servidor executando o software antivírus vulnerável
- **Ataques em outros sistemas:**
- Carregar arquivo .exe na árvore da web - as vítimas baixam o executável trojaned
- Carregar arquivo infectado por vírus - máquinas das vítimas infectadas
- Carregar arquivo .html contendo script - experiências da vítima Script entre sites (XSS)
- Carregue o arquivo .jpg que contém um objeto Flash - a vítima experimenta o seqüestro de conteúdo entre sites.
- Carregar arquivo .rar a ser verificado pelo antivírus - comando executado em um cliente executando o software antivírus vulnerável

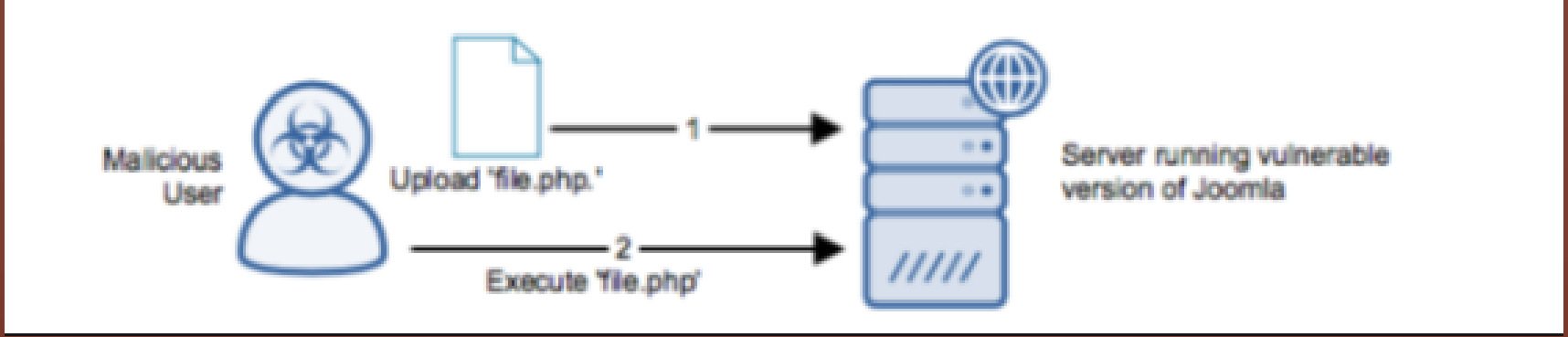
UNRESTRICTED FILE UPLOAD - EXEMPLOS

https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

UNRESTRICTED FILE UPLOAD - EXEMPLOS

https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

UNRESTRICTED FILE UPLOAD - EXEMPLOS



https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

FILE UPLOAD UNRESTRICTED - EXEMPLOS

- https://medium.com/@shayboy123/how-i-gain-unrestricted-file-upload-remote-code-execution-bug-bounty-381d0aab0dad
- https://www.youtube.com/watch?v=4g2uwJ7H7zM
- https://www.youtube.com/watch?v=xpCLMz3efUw
- https://www.youtube.com/watch?v=YwZwxfGTpVc
- https://nileshsapariya.blogspot.com/2015/11/linkedin-unrestricted-file-upload.html
- https://blog.securitybreached.org/2017/12/19/unrestricted-file-upload-to-rce-bug-bounty-poc/
- https://0x00sec.org/t/unrestricted-cv-file-upload/20325
- https://github.com/modzero/mod0BurpUploadScanner (PLUGIN - BURP SUITE)
- https://www.youtube.com/watch?v=CmF9sEyKZNo

SQL INJECTION

- Um ataque de injeção SQL consiste na inserção ou "injeção" de uma consulta SQL através dos dados de entrada do cliente para o aplicativo. Uma exploração bem-sucedida da injeção SQL pode ler dados confidenciais do banco de dados, modificar dados (Inserir / Atualizar / Excluir), executar operações de administração no banco de dados (como desligar o DBMS), recuperar o conteúdo de um determinado arquivo presente no arquivo DBMS sistema e, em alguns casos, emita comandos para o sistema operacional. Os ataques de injeção SQL são um tipo de ataque de injeção, no qual comandos SQL são injetados na entrada do plano de dados para efetuar a execução de comandos SQL predefinidos.

https://owasp.org/www-community/attacks/SQL_Injection

SQL INJECTION

- Erros de injeção SQL ocorrem quando:

1. Os dados entram no programa a partir de uma fonte não confiável.
2. Os dados usados para construir dinamicamente uma consulta SQL

- As principais conseqüências são:
- **Confidencialidade** : como os bancos de dados SQL geralmente mantêm dados confidenciais, a perda de confidencialidade é um problema frequente nas vulnerabilidades da Injeção SQL.
- **Autenticação** : se comandos SQL ruins forem usados para verificar nomes de usuário e senhas, talvez seja possível conectar-se a um sistema como outro usuário sem conhecimento prévio da senha.
- **Autorização** : se as informações de autorização estiverem em um banco de dados SQL, talvez seja possível alterá-las através da exploração bem-sucedida de uma vulnerabilidade de Injeção SQL.
- **Integridade** : Assim como pode ser possível ler informações confidenciais, também é possível fazer alterações ou até mesmo excluir essas informações com um ataque de injeção de SQL.

SQL INJECTION - TIPOS

- A injeção de SQL pode ser usada de várias maneiras para causar problemas sérios. Ao alavancar a injeção de SQL, um invasor pode ignorar a autenticação, acessar, modificar e excluir dados em um banco de dados. Em alguns casos, o SQL Injection pode até ser usado para executar comandos no sistema operacional, potencialmente permitindo que um invasor passe a ataques mais prejudiciais dentro de uma rede que fica atrás de um firewall.
- A injeção de SQL pode ser classificada em três categorias principais - *SQLi em banda* , *SQLi inferencial* e *SQLi fora de banda* .

SQL INJECTION - TIPOS

SQLi em banda (SQLi clássico)

- O SQL Injection em banda é o ataque mais comum e fácil de explorar dos SQL Injection. A injeção de SQL em banda ocorre quando um invasor pode usar o mesmo canal de comunicação para iniciar o ataque e coletar resultados.
- Os dois tipos mais comuns de injeção SQL em banda são *SQLi baseado em erro* e *SQLi baseado em união* .

SQLi baseado em erro

- O SQLi baseado em erro é uma técnica de injeção de SQL em banda que se baseia em mensagens de erro lançadas pelo servidor de banco de dados para obter informações sobre a estrutura do banco de dados. Em alguns casos, apenas a injeção SQL baseada em erro é suficiente para um invasor enumerar um banco de dados inteiro. Embora os erros sejam muito úteis durante a fase de desenvolvimento de um aplicativo Web, eles devem ser desativados em um site ativo ou registrados em um arquivo com acesso restrito.

SQL INJECTION - TIPOS

SQLi baseado em união

- O SQLi baseado em união é uma técnica de injeção de SQL em banda que utiliza o operador UNION SQL para combinar os resultados de duas ou mais instruções SELECT em um único resultado que é retornado como parte da resposta HTTP.

SQLi Inferencial (SQL Cego)

- A injeção inferencial de SQL, diferentemente do SQLi em banda, pode levar mais tempo para um invasor explorar, no entanto, é tão perigoso quanto qualquer outra forma de injeção de SQL. Em um ataque inferencial do SQLi, nenhum dado é realmente transferido pelo aplicativo Web e o invasor não seria capaz de ver o resultado de um ataque dentro da banda (é por isso que esses ataques são comumente referidos como " ataques cegos de injeção de SQL ") . Em vez disso, um invasor é capaz de reconstruir a estrutura do banco de dados enviando cargas, observando a resposta do aplicativo Web e o comportamento resultante do servidor de banco de dados.
- Os dois tipos de injeção SQL inferencial são *-boolean baseado em Cegos SQLi* e *SQLi baseado em Cego-time* .

SQL INJECTION - TIPOS

SQLi cego baseado em booleano (baseado em conteúdo)

- A Injeção SQL baseada em booleano é uma técnica inferencial de Injeção SQL que se baseia no envio de uma consulta SQL ao banco de dados, o que força o aplicativo a retornar um resultado diferente, dependendo se a consulta retorna um resultado VERDADEIRO ou FALSO.
- Dependendo do resultado, o conteúdo da resposta HTTP será alterado ou permanecerá o mesmo. Isso permite que um invasor deduza se a carga útil usada retornou verdadeiro ou falso, mesmo que nenhum dado do banco de dados seja retornado. Esse ataque geralmente é lento (especialmente em bancos de dados grandes), pois um invasor precisa enumerar um banco de dados, caractere por caractere.

SQLi cego baseado em tempo

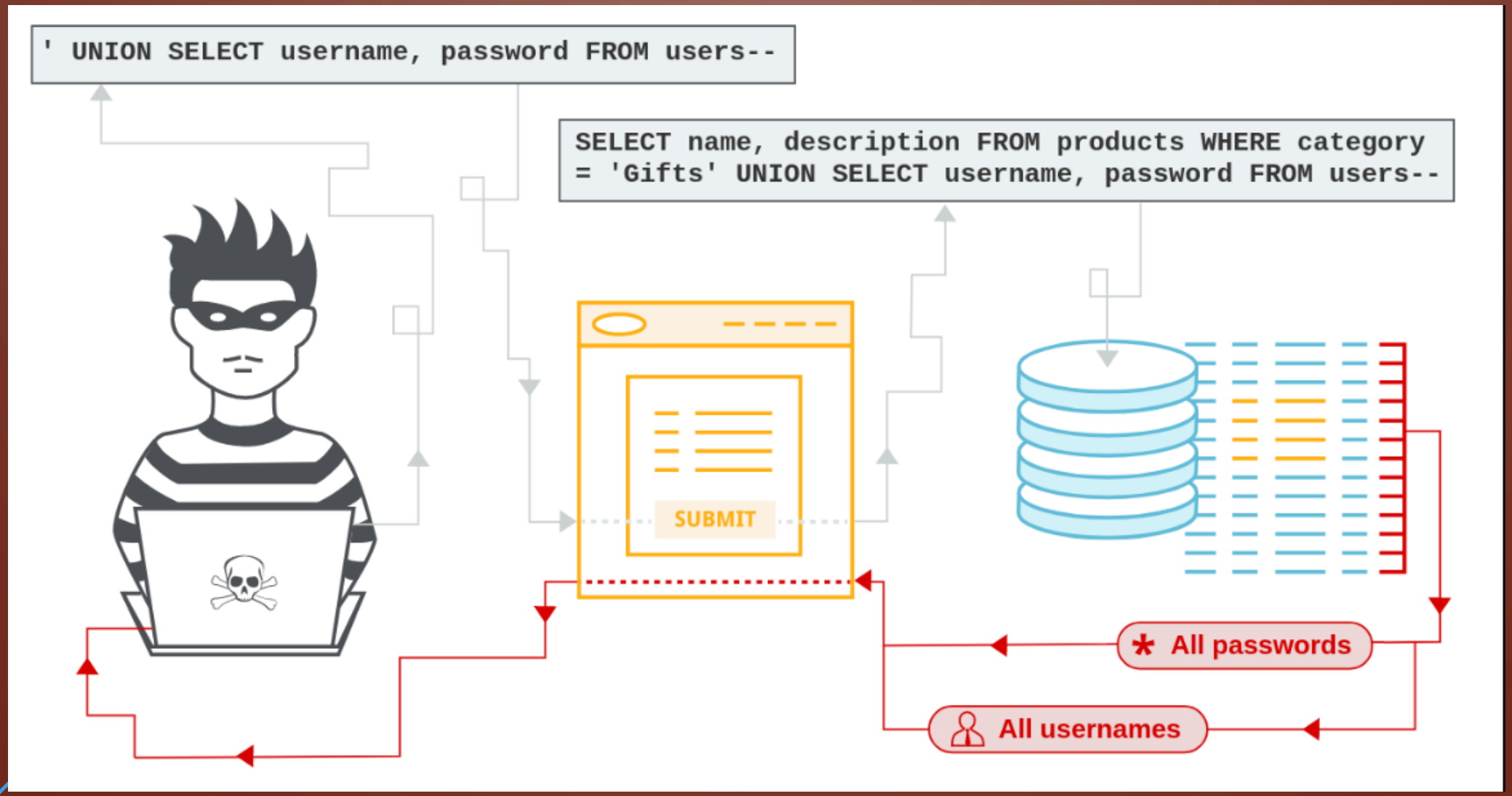
- A injeção de SQL baseada em tempo é uma técnica inferencial de injeção de SQL que depende do envio de uma consulta SQL ao banco de dados, o que força o banco de dados a aguardar um período de tempo especificado (em segundos) antes de responder. O tempo de resposta indicará ao invasor se o resultado da consulta é VERDADEIRO ou FALSO.
- Dependendo do resultado, uma resposta HTTP será retornada com um atraso ou retornada imediatamente. Isso permite que um invasor deduza se a carga útil usada retornou verdadeiro ou falso, mesmo que nenhum dado do banco de dados seja retornado. Esse ataque geralmente é lento (especialmente em bancos de dados grandes), pois um invasor precisará enumerar um caractere de banco de dados por caractere.

SQL INJECTION - TIPOS

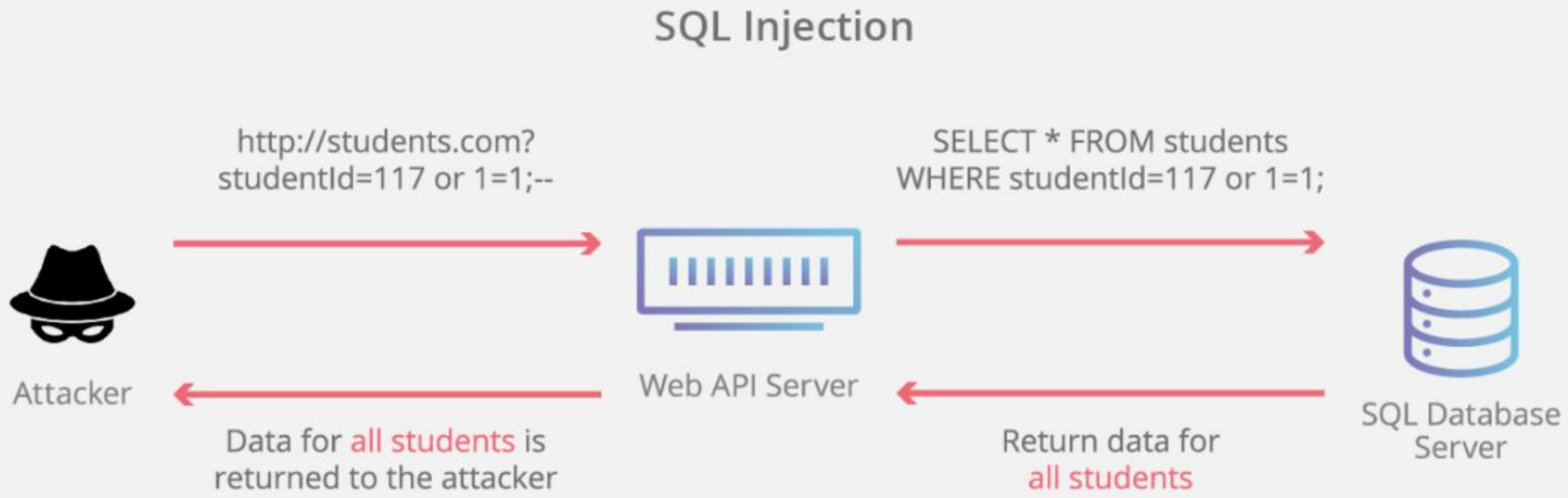
SQLi fora da banda

- A injeção de SQL fora de banda não é muito comum, principalmente porque depende dos recursos ativados no servidor de banco de dados que está sendo usado pelo aplicativo da web. A injeção SQL fora de banda ocorre quando um invasor não consegue usar o mesmo canal para iniciar o ataque e coletar resultados.
- As técnicas fora da banda oferecem ao invasor uma alternativa às técnicas inferenciais baseadas em tempo, especialmente se as respostas do servidor não forem muito estáveis (tornando um ataque inferencial baseado em tempo não confiável).
- As técnicas de SQLi fora de banda dependeriam da capacidade do servidor de banco de dados de fazer solicitações de DNS ou HTTP para fornecer dados a um invasor. É o caso do xp_dirtreecomando do Microsoft SQL Server , que pode ser usado para fazer solicitações de DNS para um servidor que um invasor controla; bem como o pacote UTL_HTTP do Oracle Database, que pode ser usado para enviar solicitações HTTP de SQL e PL / SQL para um servidor que um invasor controla.
- https://www.acunetix.com/websitesecurity/sql-injection2/

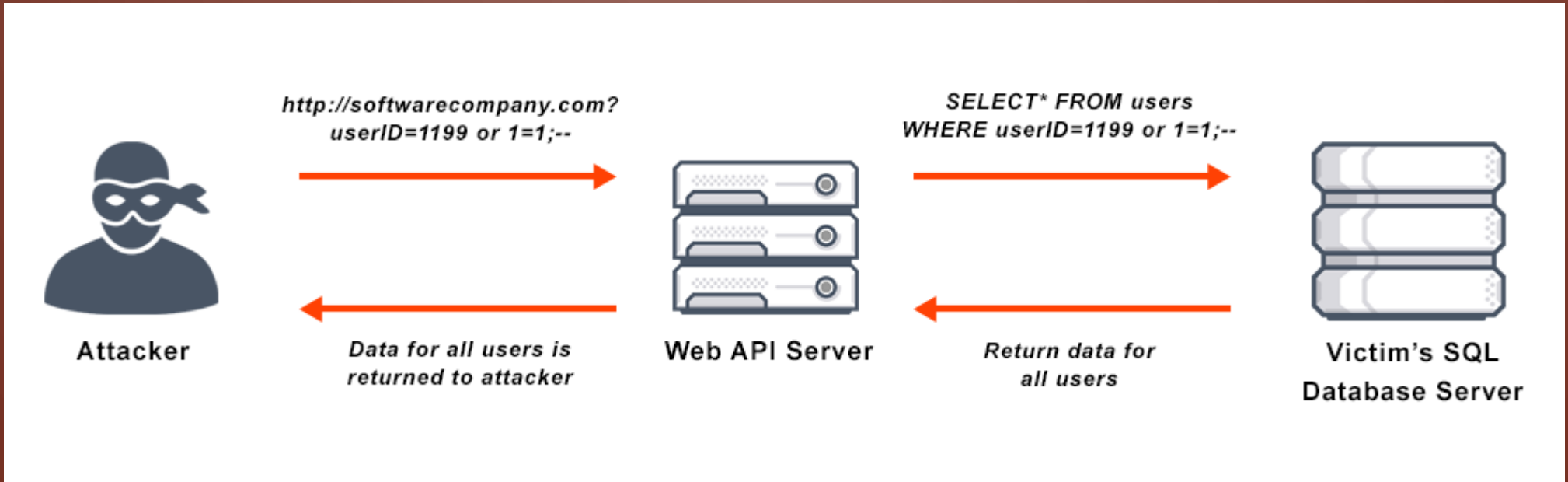
SQL INJECTION - EXEMPLOS



SQL INJECTION - EXEMPLOS



SQL INJECTION - EXEMPLOS



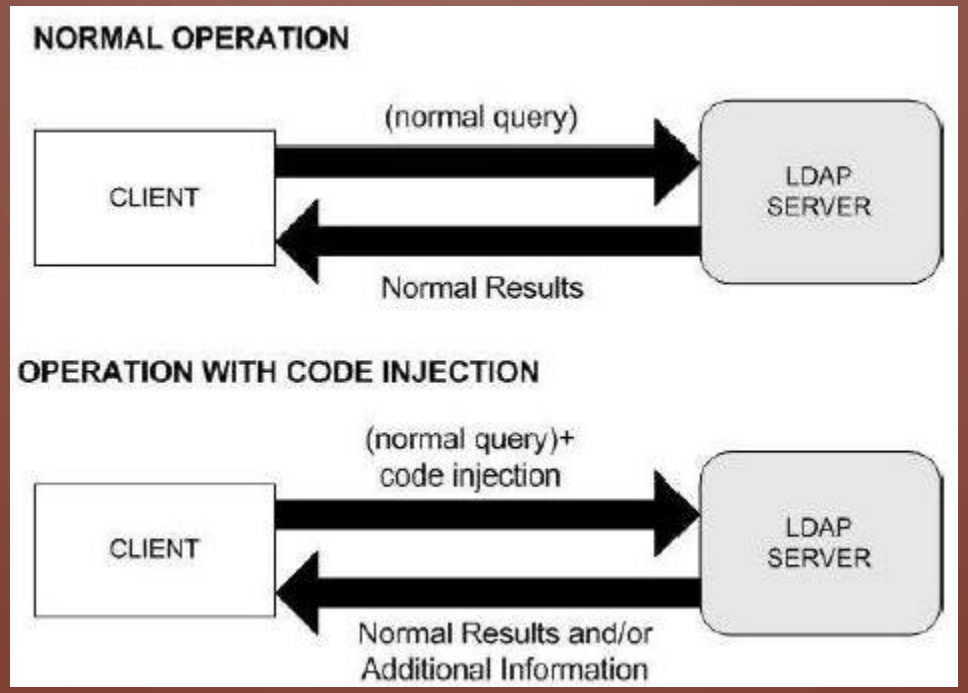
SQL INJECTION - EXEMPLOS

- https://www.youtube.com/watch?v=7DlrFns3200
- https://www.youtube.com/watch?v=bIB3Hi6KeZU
- https://www.youtube.com/watch?v=U3Qzc2YUNIU
- https://www.hackerone.com/blog/8-high-impact-bugs-and-how-hackerone-customers-avoided-breach-sql-injection
- https://medium.com/sud0root/bug-bounty-writeups-exploiting-sql-injection-vulnerability-20b019553716
- https://www.youtube.com/watch?v=w0k82G-q5BI
- https://www.youtube.com/watch?v=KS5F20i1kvU
- https://www.youtube.com/watch?v=IsuCEGsrVPc
- https://www.youtube.com/watch?v=_i9u8O2ehlg
- https://www.youtube.com/watch?v=zfIGYk3rmq0
- https://www.youtube.com/watch?v=UMJV3OpjsoM

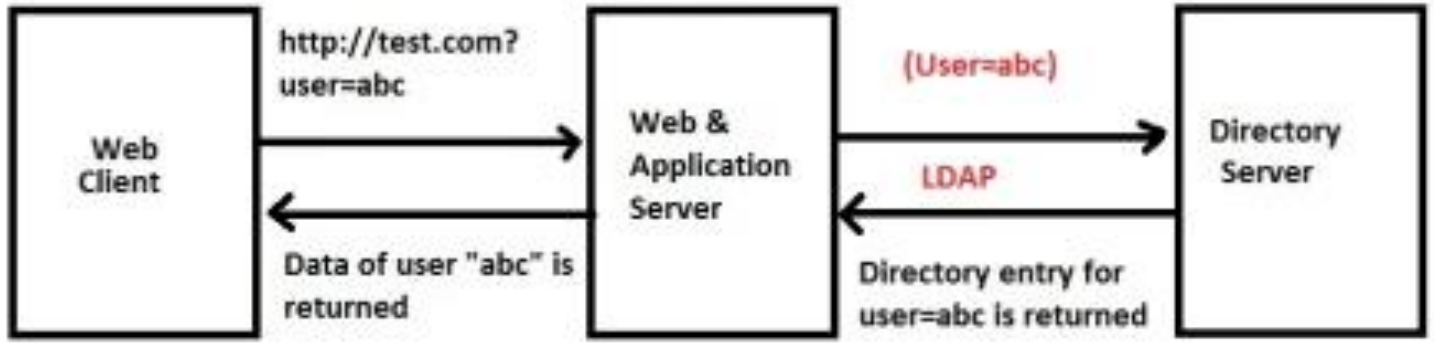
LDAP INJECTION

- Na segurança do computador, a injeção LDAP é uma técnica de injeção de código usada para explorar aplicativos da Web que podem revelar informações confidenciais do usuário ou modificar as informações representadas nos armazenamentos de dados LDAP.
- A injeção LDAP surge quando os dados controláveis pelo usuário são copiados de maneira insegura em uma consulta LDAP executada pelo aplicativo. Se um invasor puder injetar metacaracteres LDAP na consulta, ele poderá interferir na lógica da consulta. Dependendo da função para a qual a consulta é usada, o invasor poderá recuperar dados confidenciais para os quais não estão autorizados ou subverter a lógica do aplicativo para executar alguma ação não autorizada.
- Observe que testes automatizados baseados em diferenças para falhas na injeção de LDAP geralmente podem não ser confiáveis e propensos a resultados falsos positivos. Os resultados do scanner devem ser revisados manualmente para confirmar se uma vulnerabilidade está realmente presente.
- https://en.wikipedia.org/wiki/LDAP_injection
- https://portswigger.net/kb/issues/00100500_ldap-injection

LDAP INJECTION - EXEMPLOS



LDAP INJECTION - EXEMPLOS



Typical Application Server and Directory Server Communication

LDAP - EXEMPLOS

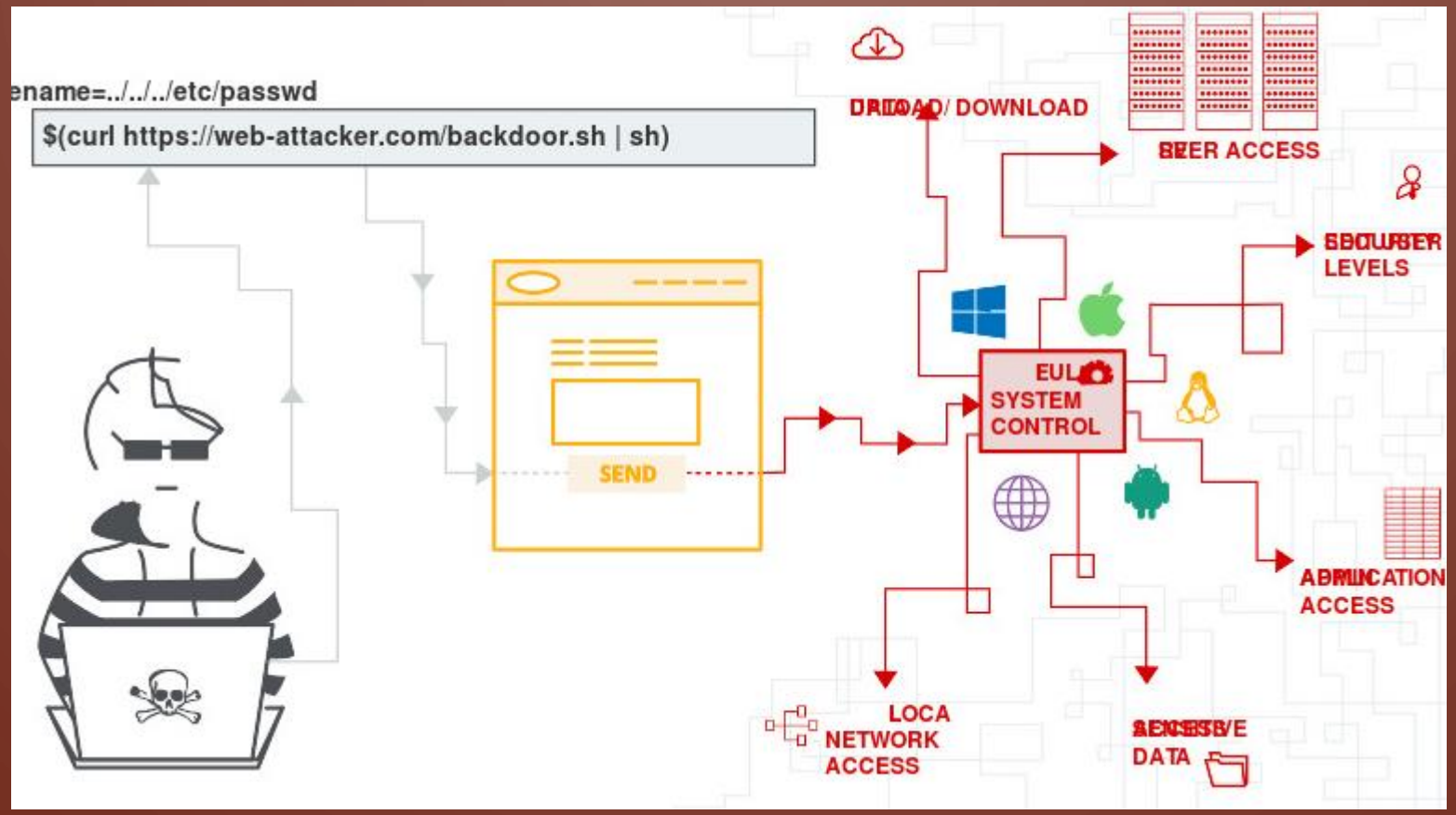
- https://www.youtube.com/watch?v=qStzSfsEQGQ
- https://www.youtube.com/watch?v=iUbqJy_MOiE
- https://www.youtube.com/watch?v=DkKUDbEt46A
- https://hackerone.com/reports/359290
- https://www.blackhat.com/presentations/bh-europe-08/Alonso-Parada/Whitepaper/bh-eu-08-alonso-parada-WP.pdf
- https://research.securitum.com/ldap-injection-vulnerability-definitions-examples-of-attacks-methods-of-protection/
- https://www.youtube.com/watch?v=K0q10q9BQAk

COMMAND INJECTION

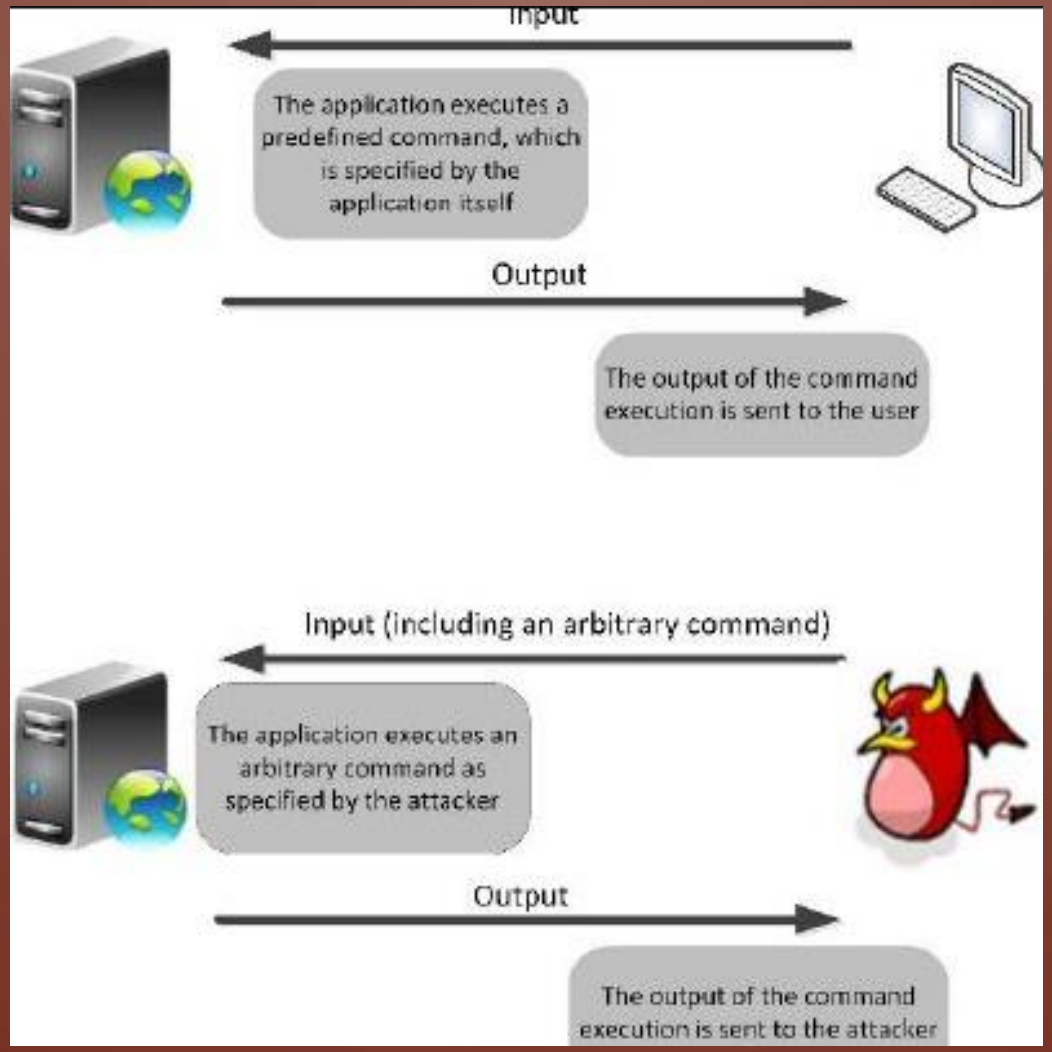
- Injeção de comando é um ataque no qual o objetivo é a execução de comandos arbitrários no sistema operacional host por meio de um aplicativo vulnerável. Os ataques de injeção de comando são possíveis quando um aplicativo passa dados inseguros fornecidos pelo usuário (formulários, cookies, cabeçalhos HTTP etc.) para um shell do sistema. Nesse ataque, os comandos do sistema operacional fornecidos pelo invasor geralmente são executados com os privilégios do aplicativo vulnerável. Os ataques de injeção de comando são possíveis em grande parte devido à validação de entrada insuficiente.
- Esse ataque difere da Injeção de código , pois a injeção de código permite que o invasor adicione seu próprio código que é executado pelo aplicativo. Na Injeção de Comando, o invasor estende a funcionalidade padrão do aplicativo, que executa comandos do sistema, sem a necessidade de injetar código.

https://owasp.org/www-community/attacks/Command_Injection

COMMAND INJECTION - EXEMPLOS



COMMAND INJECTION - EXEMPLOS



COMMAND INJECTION - EXEMPLOS

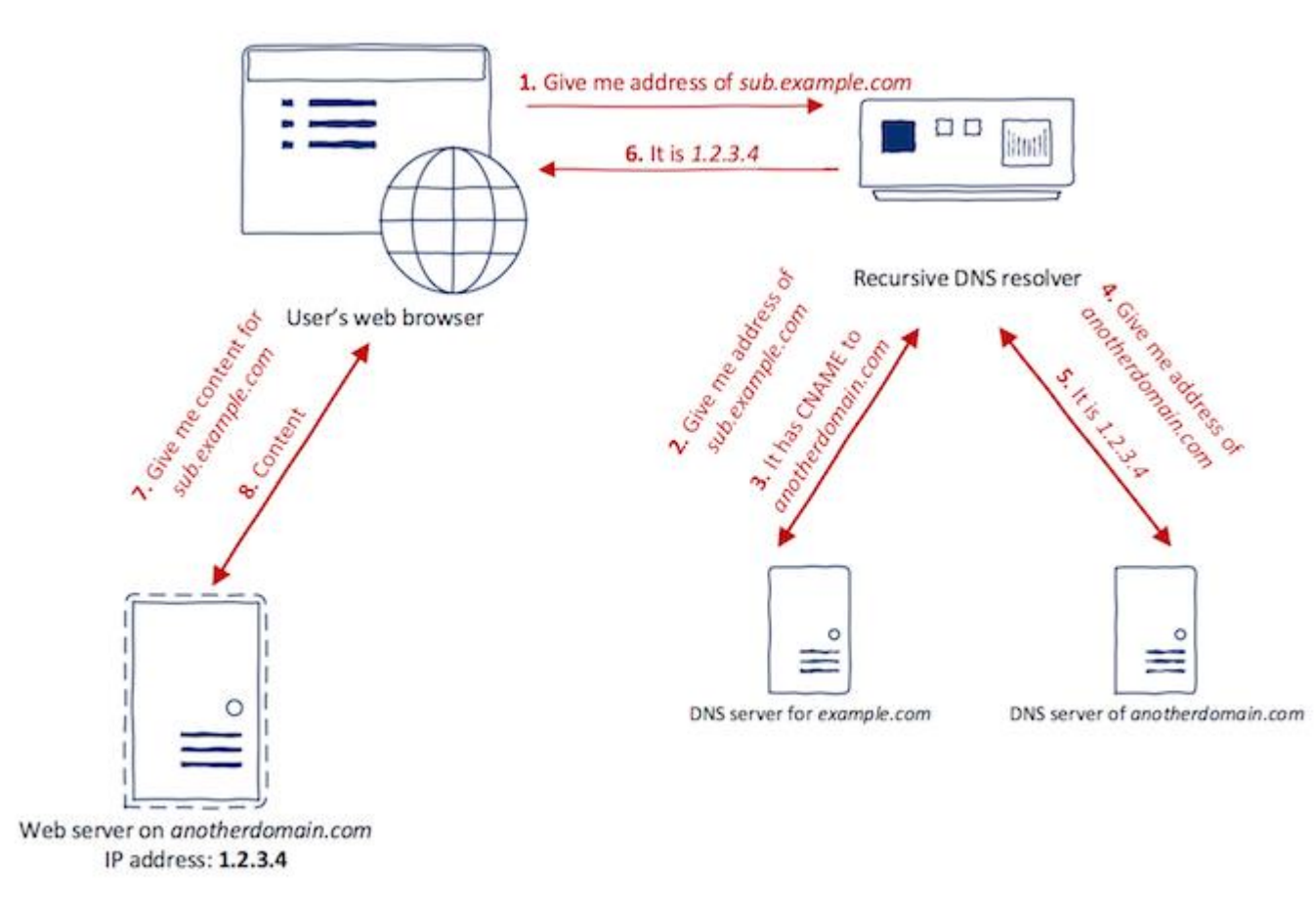
- https://www.youtube.com/watch?v=nZqidp5ZukI
- https://www.youtube.com/watch?v=dQ-_TO1zuvA
- https://www.youtube.com/watch?v=OlVezhUS4NQ
- https://medium.com/@trapp3rhat/command-injection-through-blh-3c32614bb395
- https://medium.com/bugbountywriteup/when-i-found-multiple-command-injection-ad891d3ad9e6
- https://www.hackerone.com/blog/how-to-command-injections
- https://portswigger.net/web-security/os-command-injection
- https://www.youtube.com/watch?v=nzrd_Dozufo
- https://portswigger.net/web-security/os-command-injection

SUBDOMAIN TAKEOVER

- Os ataques de controle de subdomínio são uma classe de problemas de segurança em que um invasor pode assumir o controle do subdomínio de uma organização por meio de serviços em nuvem como AWS ou Azure. Eles geralmente acontecem quando os projetos da Web são finalizados, mas as entradas DNS do subdomínio não são totalmente desativadas.
- Quando páginas da web são hospedadas em provedores de nuvem, a página da Web geralmente é criada em um subdomínio no provedor de nuvem primeiro. Por exemplo, no Azure, esse subdomínio teria o formato *webproject.azurewebsites.net* . Em última análise, o cliente deseja que o projeto pareça estar hospedado em um subdomínio do próprio domínio do cliente . Portanto, as consultas ao subdomínio do cliente - por exemplo, *webproject.example.org* - seriam encaminhadas para o subdomínio hospedado na nuvem - nesse caso, *webproject.azurewebsites.net* .
- Para efetuar essa alteração, um registro DNS (sistema de nome de domínio) CNAME - um registro para um nome canônico - está configurado para encaminhar todas as consultas ao subdomínio do cliente, por exemplo, *webproject.example.org* , ao subdomínio do provedor de nuvem, *webproject .azurewebsites.net* , onde o projeto da web está hospedado.
- O potencial para uma aquisição de subdomínio ocorre quando a página da web hospedada no provedor de nuvem é excluída, mas a entrada DNS é mantida. Há uma razão para essa ocorrência comum: enquanto a hospedagem no provedor de nuvem custa dinheiro, ter uma entrada DNS antiga geralmente é gratuita. Portanto, embora exista um incentivo para excluir páginas da Web obsoletas, as entradas DNS são frequentemente esquecidas.

https://searchsecurity.techtarget.com/answer/What-is-subdomain-takeover-and-why-does-it-matter

SUBDOMAIN TAKEOVER - EXEMPLOS



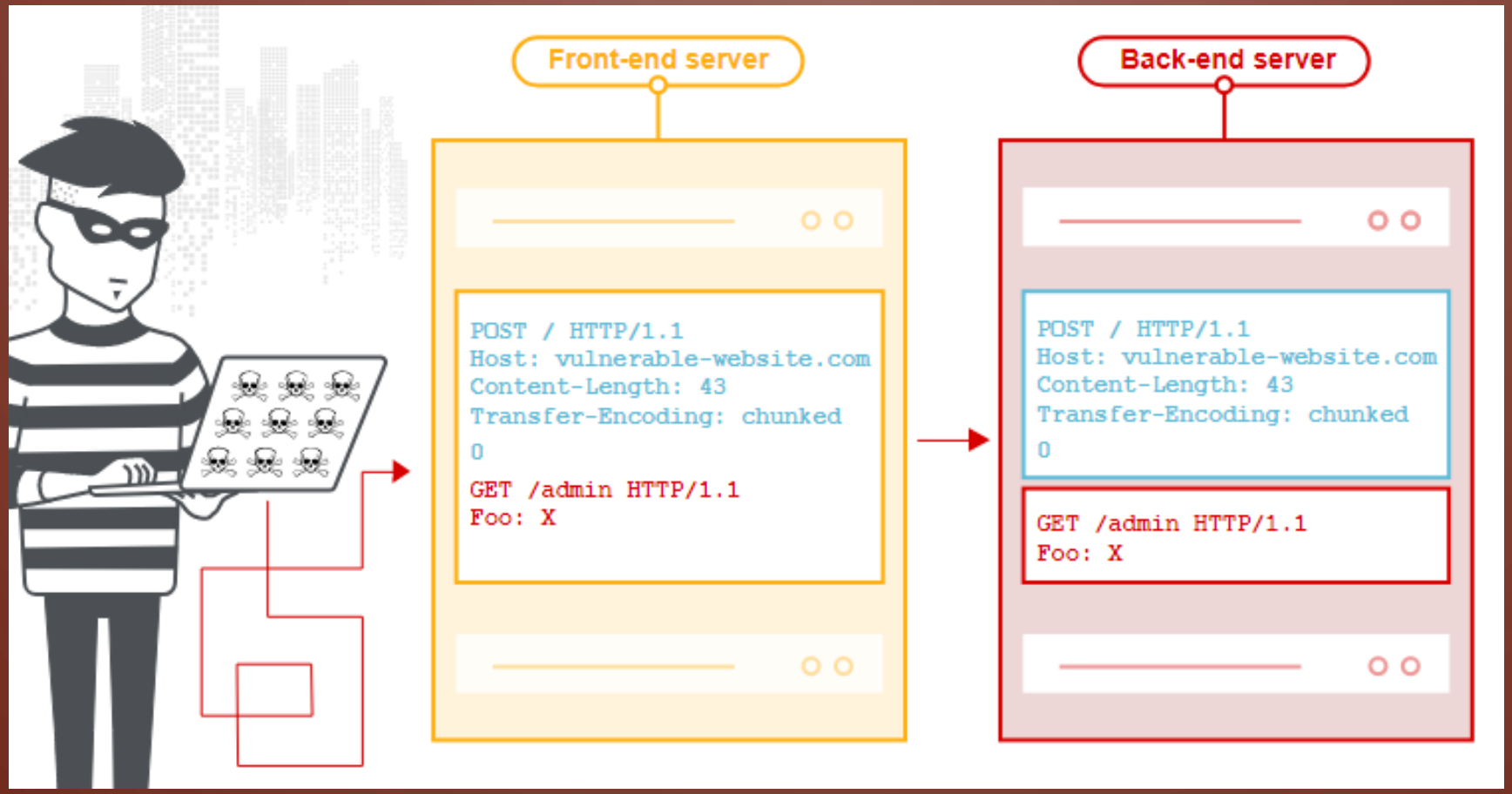
SUBDOMAIN TAKEOVER - EXEMPLOS

- https://www.youtube.com/watch?v=u_nXZ4YRcto
- https://www.youtube.com/watch?v=kMAtAWFBPLA
- https://www.youtube.com/watch?v=jqyOpSrf02U
- https://www.youtube.com/watch?v=9DYEg_j-_hw
- https://www.youtube.com/watch?v=ffQ38bMT4Kk
- https://www.youtube.com/watch?v=srKlqhj_ki8
- https://www.youtube.com/watch?v=vpCl8foxP00
- https://www.youtube.com/watch?v=9Nve8HwxkC8
- https://www.hackerone.com/blog/Guide-Subdomain-Takeovers
- https://hackerone.com/reports/325336
- https://0xpatrik.com/takeover-proofs/

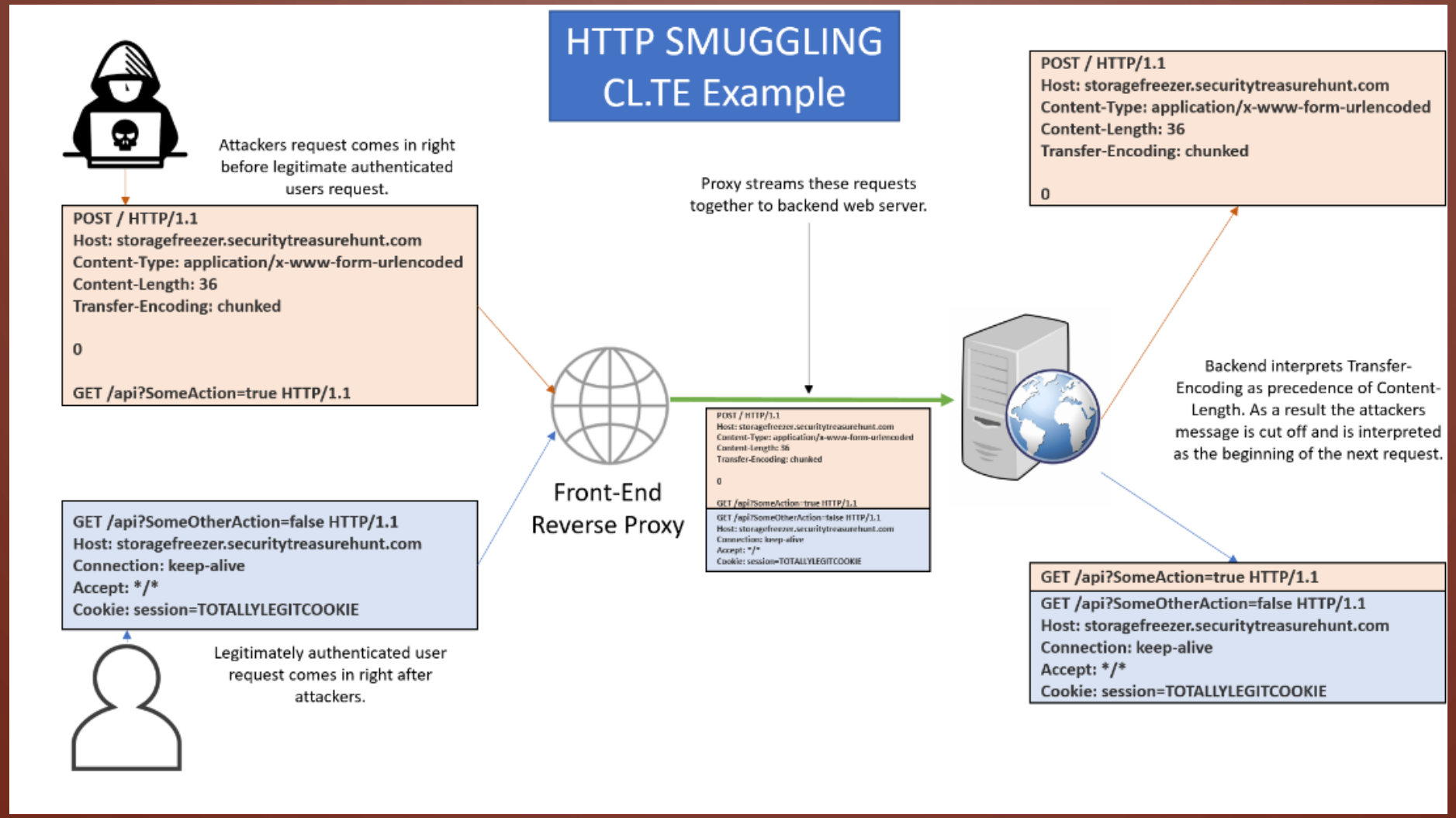
HTTP REQUEST SMUGGLING

- O contrabando de solicitação HTTP é uma técnica para interferir na maneira como um site processa seqüências de solicitações HTTP recebidas de um ou mais usuários. As vulnerabilidades de contrabando de solicitação geralmente são de natureza crítica, permitindo que um invasor ignore os controles de segurança, obtenha acesso não autorizado a dados confidenciais e comprometa diretamente outros usuários do aplicativo.
- https://portswigger.net/web-security/request-smuggling

HTTP REQUEST SMUGGLING - EXEMPLOS



HTTP REQUEST SMUGGLING - EXEMPLOS



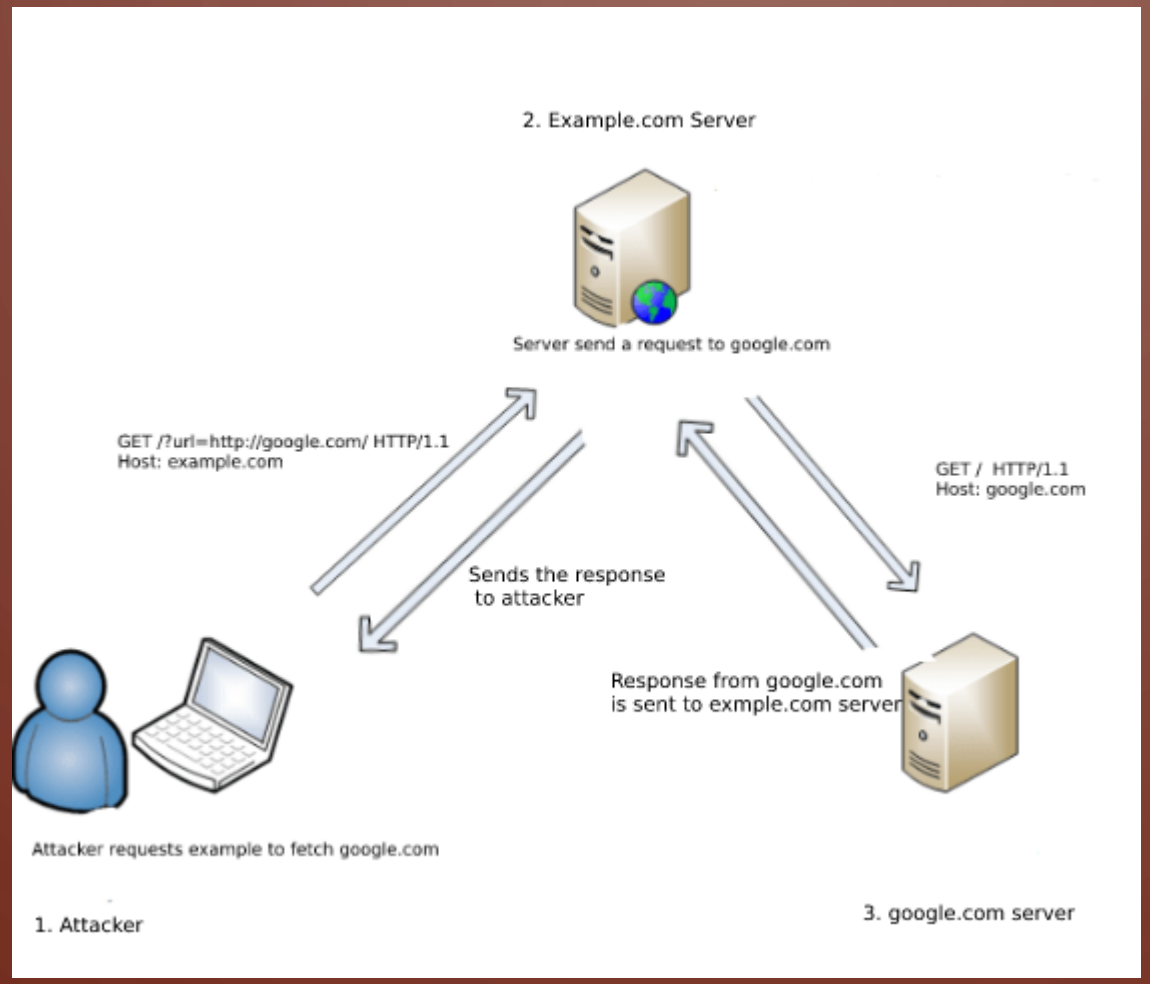
HTTP REQUEST SMUGGLING - EXEMPLOS

- https://portswigger.net/web-security/request-smuggling/finding
- https://www.youtube.com/watch?v=Ec8Cc6C0nS8
- https://www.youtube.com/watch?v=_A04msdpIXs
- https://www.youtube.com/watch?v=vkfBFuH54G4
- https://www.youtube.com/watch?v=eWQJtvWcsNw
- https://www.youtube.com/watch?v=IzpONjsQIXo
- https://www.youtube.com/watch?v=B7zdq-K2IpE
- https://hackerone.com/reports/737140
- https://hackerone.com/reports/726773
- https://hackerone.com/reports/866382
- https://blog.detectify.com/2020/05/28/hiding-in-plain-sight-http-request-smuggling/
- https://medium.com/cyberverse/earn-bounty-with-http-request-smuggling-attack-c68b4f2db363

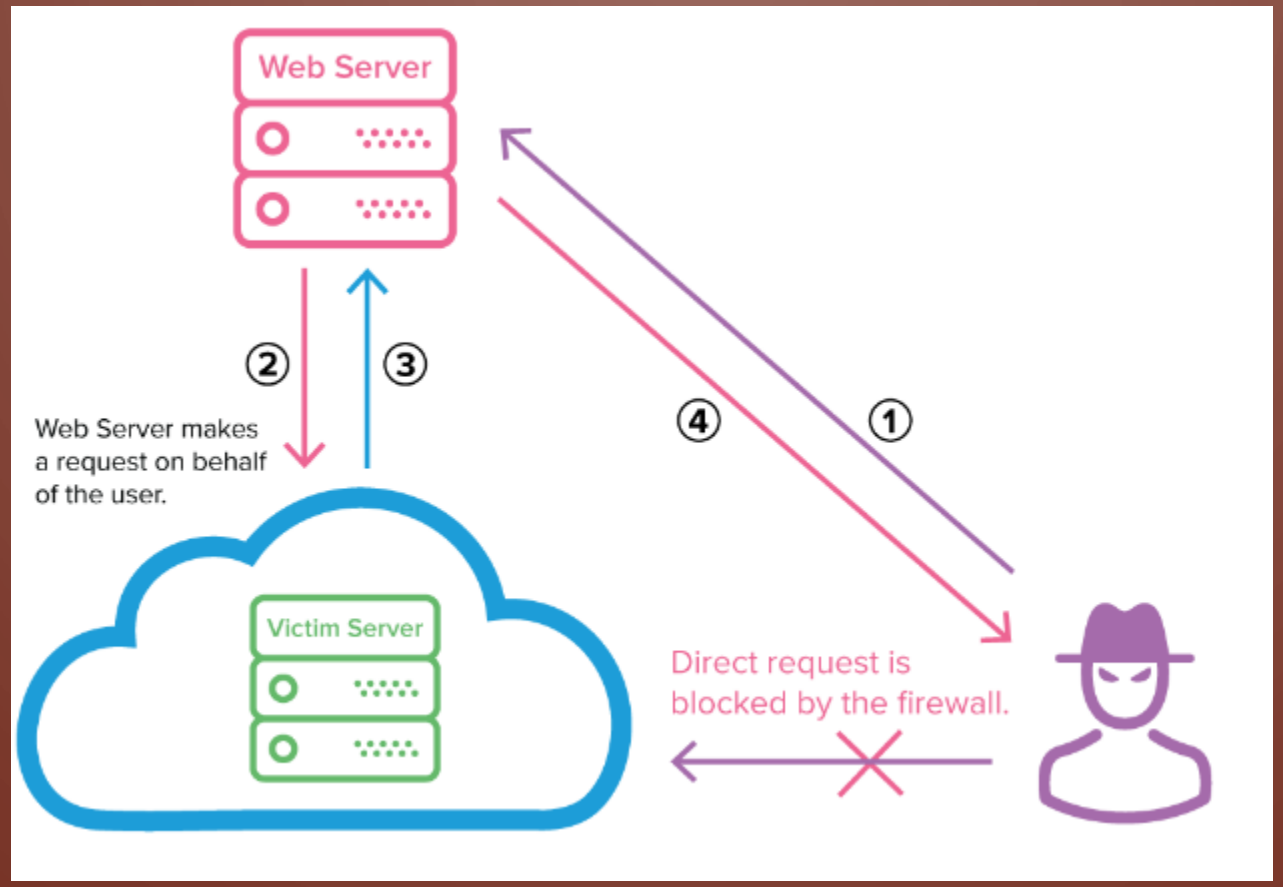
SERVER SIDE REQUEST FORGERY

- A falsificação de solicitação do lado do servidor (também conhecida como SSRF) é uma vulnerabilidade de segurança da Web que permite que um invasor induza o aplicativo do servidor a fazer solicitações HTTP para um domínio arbitrário de sua escolha.
- Em exemplos típicos de SSRF, o invasor pode fazer com que o servidor faça uma conexão de volta para si mesmo ou para outros serviços baseados na Web na infraestrutura da organização ou para sistemas externos de terceiros.
- Um ataque bem-sucedido do SSRF geralmente pode resultar em ações não autorizadas ou no acesso a dados dentro da organização, no próprio aplicativo vulnerável ou em outros sistemas de back-end com os quais o aplicativo pode se comunicar. Em algumas situações, a vulnerabilidade do SSRF pode permitir que um invasor execute a execução arbitrária de comandos .
- Uma exploração de SSRF que causa conexões com sistemas externos de terceiros pode resultar em ataques maliciosos que parecem se originar da organização que hospeda o aplicativo vulnerável, levando a possíveis responsabilidades legais e danos à reputação.
- https://portswigger.net/web-security/ssrf

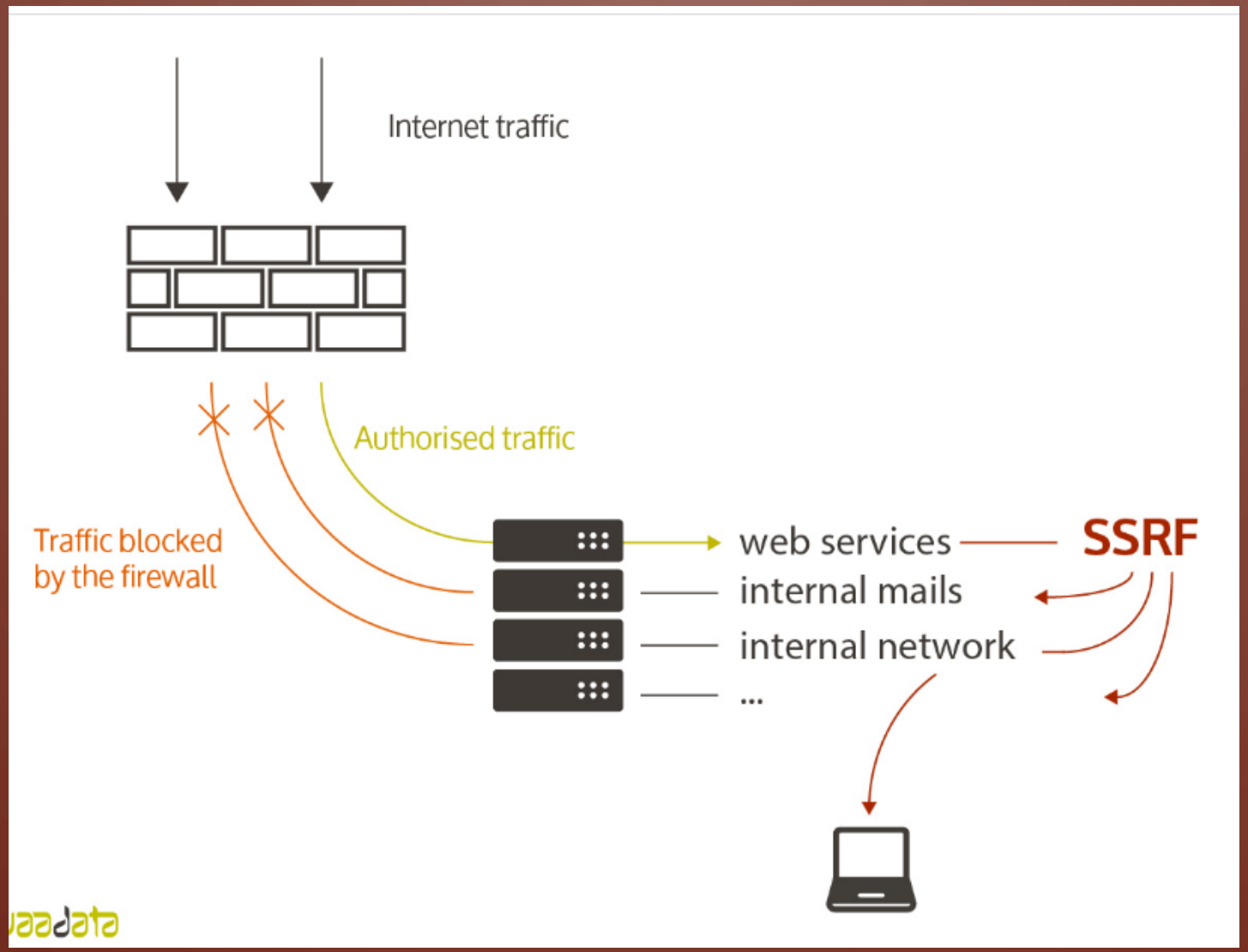
SERVER SIDE REQUEST FORGERY - EXEMPLOS



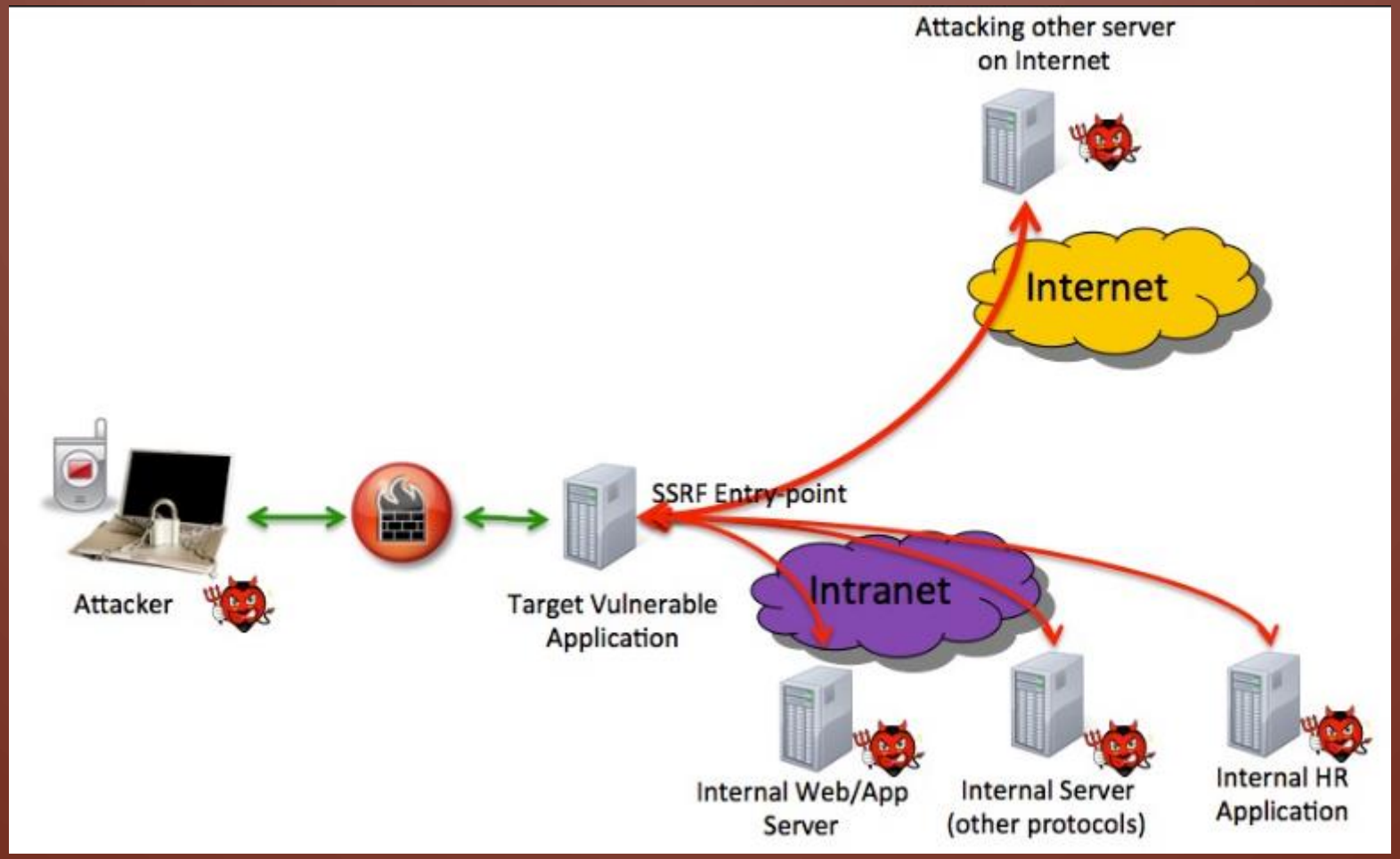
SERVER SIDE REQUEST FORGERY - EXEMPLOS



SERVER SIDE REQUEST FORGERY - EXEMPLOS



SERVER SIDE REQUEST FORGERY - EXEMPLOS



SERVER SIDE REQUEST FORGERY - EXEMPLOS

- https://www.youtube.com/watch?v=66ni2BTljS8
- https://www.youtube.com/watch?v=sZ9SbXDBR8k
- https://www.youtube.com/watch?v=yfH3ChlaT4g
- https://www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/
- https://www.youtube.com/watch?v=voTHFdL9S2k
- https://www.youtube.com/watch?v=t9ttt5bZaTE
- https://www.youtube.com/watch?v=_IVjvNelzMw
- https://www.youtube.com/watch?v=Fi32ZkFofpU
- https://medium.com/@madrobot/ssrf-server-side-request-forgery-types-and-ways-to-exploit-it-part-1-29d034c27978
- https://www.youtube.com/watch?v=x9UFlxMz-po
- https://hackerone.com/reports/514224
- https://hackerone.com/reports/341876
- https://portswigger.net/daily-swig/facebook-security-researcher-scoops-31k-bug-bounty-for-flagging-ssrf-vulnerabilities

2AF BYPASS

- A autenticação multifator é um método de autenticação no qual um usuário de computador recebe acesso somente após apresentar com sucesso duas ou mais evidências a um mecanismo de autenticação: conhecimento, posse e herança. A autenticação de dois fatores é um tipo, ou subconjunto, de autenticação de múltiplos fatores.
- https://en.wikipedia.org/wiki/Multi-factor_authentication
- **Vulnerabilidades**:

https://medium.com/@surendirans7777/2fa-bypass-techniques-32ec135fb7fe

https://shahmeeramir.com/4-methods-to-bypass-two-factor-authentication-2b0075d9eb5f

https://hackerone.com/reports/121696

2AF BYPASS - EXEMPLOS

- https://www.youtube.com/watch?v=QJL63_LO6c8
- https://www.youtube.com/watch?v=4Y_NQbNQLg8
- https://www.youtube.com/watch?v=kljojaKfKuE
- https://www.youtube.com/watch?v=kHI90LbBwaQ
- https://www.youtube.com/watch?v=ftpOrSuM39M
- https://www.youtube.com/watch?v=xaOX8DS-Cto
- https://www.youtube.com/watch?v=KN6e1mqcB9s
- https://www.youtube.com/watch?v=T7xG4ODwMzM
- https://www.youtube.com/watch?v=Yp6OFjeXdIw
- https://www.sans.org/webcasts/multi-factor-authentication-bypass-techniques-about-115255
- https://www.datto.com/library/how-attackers-bypass-multi-factor-authentication-mfa

LOCAL FILE INCLUSION

- A falha de **local file inclusion** permite que o atacante inclua um arquivo para explorar o mecanismo de dynamic file inclusion(inclusão dinâmica de arquivo) implementado na aplicação web. A falha ocorre devido ao fato de que o atacante pode passar qualquer valor para o parâmetro da aplicação alvo e a mesma não faz a validação correta do valor informado antes de executar a operação. Esse tipo de falha faz com que a aplicação web mostre o conteúdo de alguns arquivos, mas dependendo da severidade, essa falha também permite:

- – Execução de código no servidor
  – Execução de código no client-side. Por exemplo, JavaScript, o que pode levar a ocorrência de outros tipos de ataques como XSS por exemplo
  – Negação de Serviço(DoS)
  – Vazamento de informações sensíveis

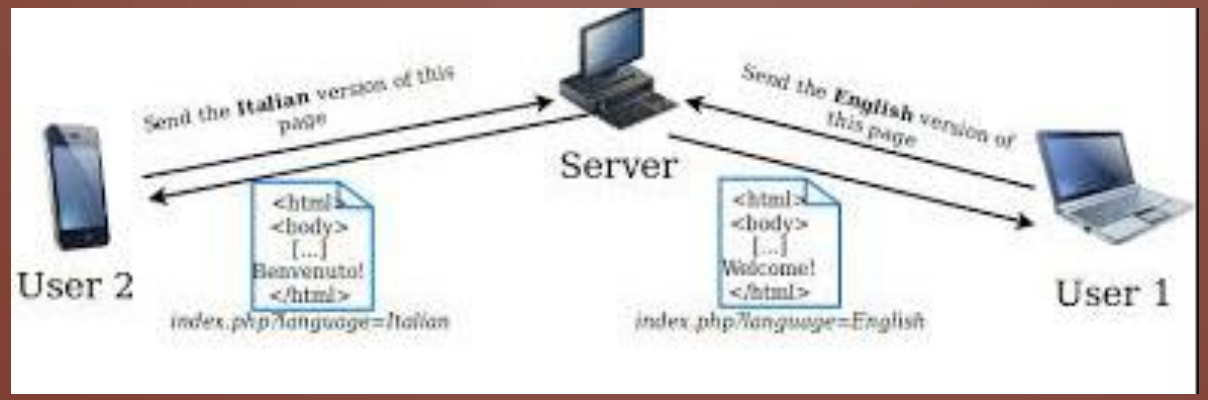
- **Local File Inclusion** (LFI) é o processo de inclusão de arquivos, que já estão presentes localmente no servidor em questão, através da exploração de processos de inclusão vulneráveis, implementados na aplicação web.
  Esta falha ocorre, por exemplo, quando uma página recebe como entrada, o caminho para o arquivo que será incluído, e esta entrada não é validada de forma correta pela aplicação web, possibilitando assim que caracteres de directory traversal(../../) sejam injetados.

REMOTE FILE INCLUSION

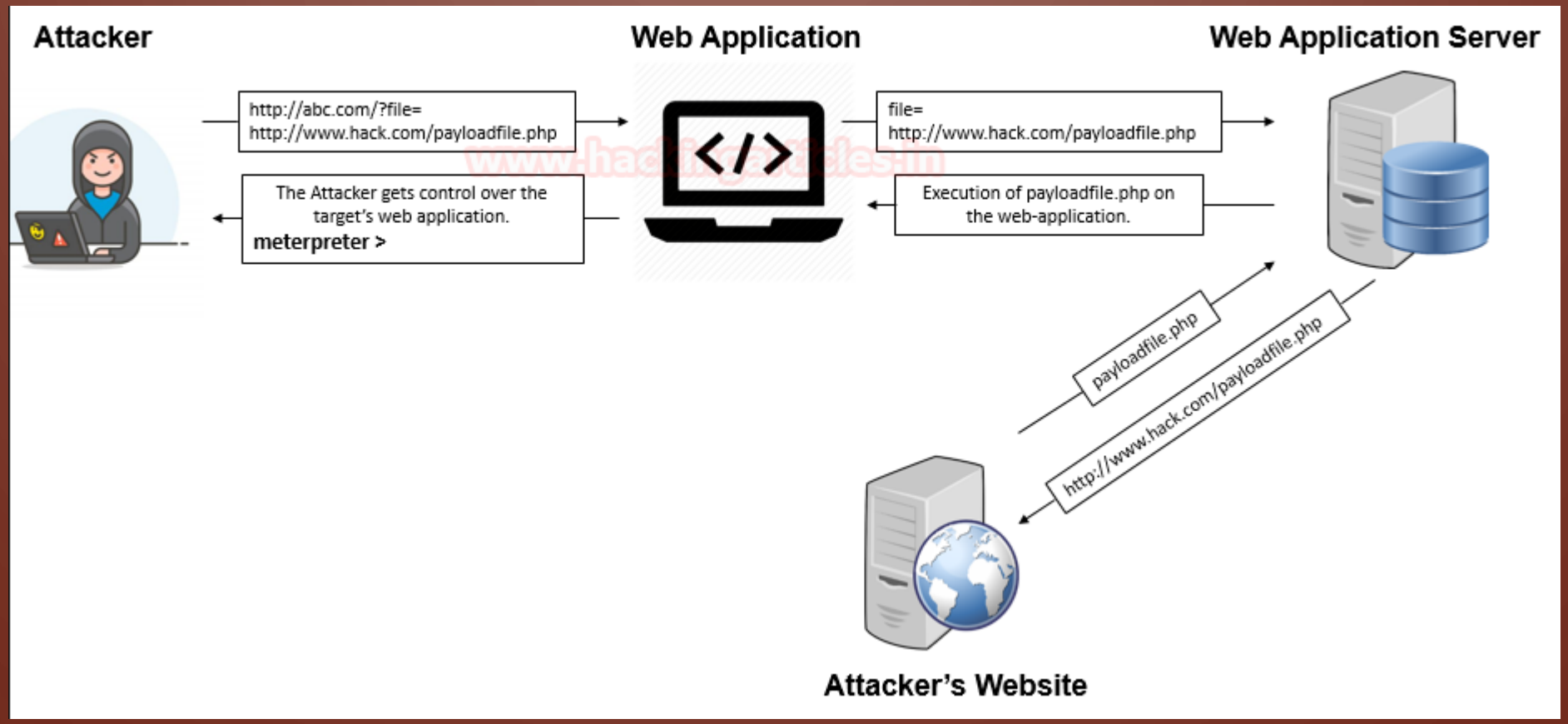
- Apesar de que a maior parte desse tipo de falha se manifeste em aplicações PHP, é muito importante lembrar que ela também pode ocorrer em JSP, ASPX e outras tecnologias.
- **Remote File Inclusion** (RFI) é o processo de inclusão de arquivos remotos, através da exploração dos processos de inclusão vulneráveis, implementados na aplicação web.
  Esta falha ocorre, por exemplo, quando uma página recebe como entrada, o caminho para o arquivo que será incluído, e esta entrada não é validada de forma correta pela aplicação web, permitindo assim que uma URL externa seja injetada na aplicação.
- Apesar de que a maior parte desse tipo de falha se manifeste em aplicações PHP, é muito importante lembrar que ela também pode ocorrer em JSP, ASPX e outras tecnologias.

https://www.infosec.com.br/local-file-inclusion-remore-file-inclusion/

LOCAL FILE INCLUSION - EXEMPLOS



LOCAL FILE INCLUSION - EXEMPLOS



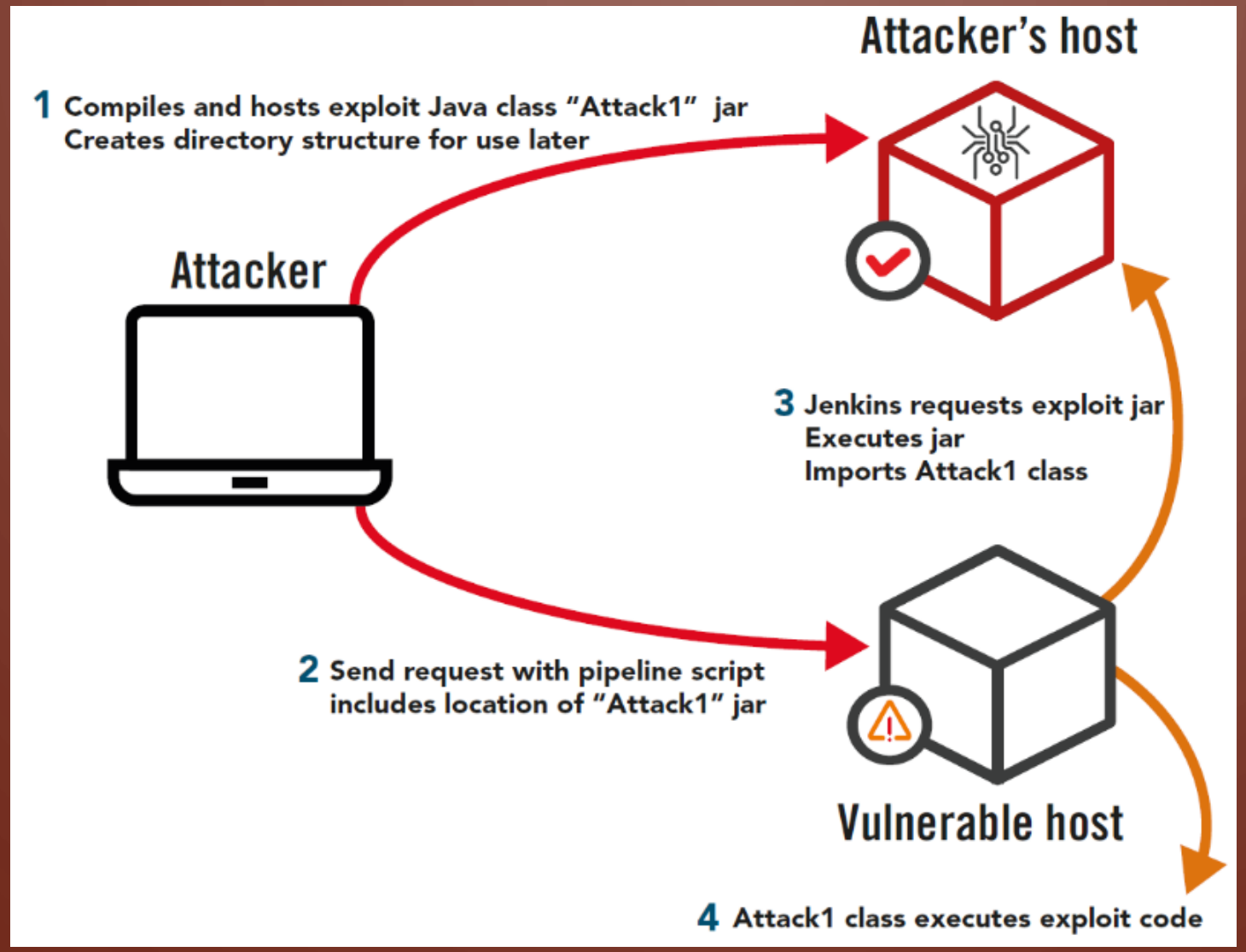
LOCAL FILE INCLUSION - EXEMPLOS

- https://www.acunetix.com/blog/articles/local-file-inclusion-lfi/
- https://www.offensive-security.com/metasploit-unleashed/file-inclusion-vulnerabilities/
- http://labs.siteblindado.com/2017/08/o-que-e-vulnerabilidade-file-inclusion.html
- https://www.youtube.com/watch?v=kcojXEwolls
- https://www.youtube.com/watch?v=V1gdO3QT-XY
- https://www.youtube.com/watch?v=s3XQ1n5kdeQ
- https://www.hacker101.com/sessions/file_inclusion.html
- https://www.youtube.com/watch?v=vg9BEEsnQ6A
- https://medium.com/bugbountywriteup/finding-path-traversal-vulnerability-e2506d390569
- https://www.youtube.com/watch?v=MHBoCVvzXzc
- https://www.youtube.com/watch?v=khvwTKJqcsg

REMOTE CODE EXECUTION

- A Execução Remota de Código é uma vulnerabilidade que pode ser explorada se a entrada do usuário for injetada em um Arquivo ou String e executada (avaliada) pelo analisador da linguagem de programação. Normalmente, esse comportamento não é pretendido pelo desenvolvedor do aplicativo Web. Uma avaliação remota de código pode levar a um comprometimento total do aplicativo da web vulnerável e também do servidor da web. É importante observar que quase todas as linguagens de programação possuem funções de avaliação de código.
- Uma Execução de código pode ocorrer se você permitir a entrada do usuário dentro de funções que estão avaliando o código na respectiva linguagem de programação. Isso pode ser implementado de propósito, por exemplo, para acessar funções matemáticas da linguagem de programação para criar uma calculadora ou acidentalmente, porque não é de esperar que o desenvolvedor controle a entrada do desenvolvedor dentro dessas funções. Geralmente não é aconselhável fazê-lo. De fato, é considerado uma prática ruim usar a avaliação de código.
- https://www.netsparker.com/blog/web-security/remote-code-evaluation-execution/

REMOTE CODE EXECUTION - EXEMPLO



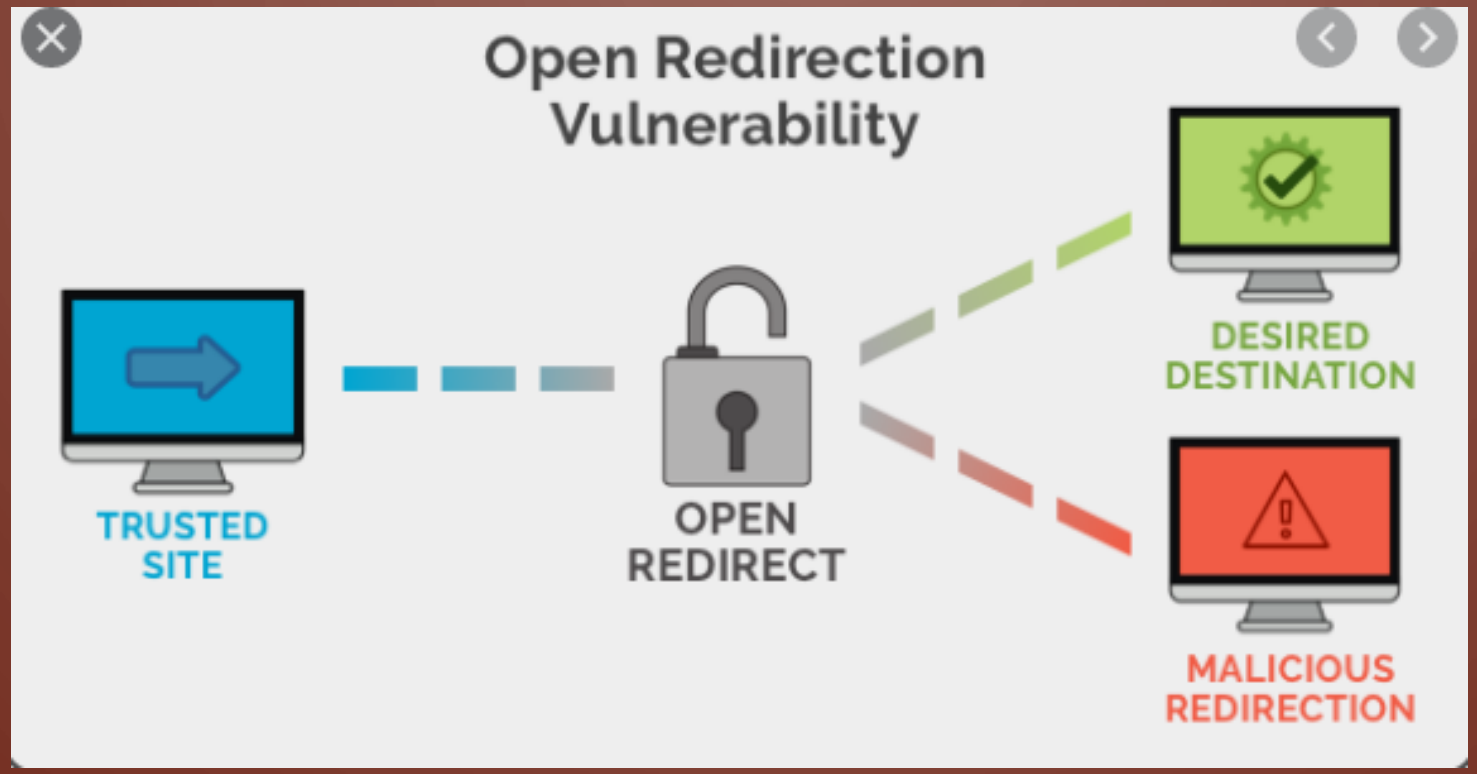
REMOTE CODE EXECUTION - EXEMPLOS

- https://www.youtube.com/watch?v=AuNwk--lfxU
- https://www.ophtek.com/remote-code-execution-used/
- https://www.thezdi.com/blog/2020/2/24/cve-2020-0688-remote-code-execution-on-microsoft-exchange-server-through-fixed-cryptographic-keys
- https://www.youtube.com/watch?v=EqlNWsORnRg
- https://www.contextis.com/en/blog/frag-grenade-a-remote-code-execution-vulnerability-in-the-steam-client
- https://www.youtube.com/watch?v=kcnJMKXnW1k
- https://www.youtube.com/watch?v=mjCAcQ5HFII
- https://www.youtube.com/watch?v=xIUFQc6M0Hk
- https://medium.com/@corneacristian/top-25-rce-bug-bounty-reports-bc9555cca7bc
- https://www.bugcrowd.com/resources/glossary/remote-code-execution-rce/
- https://hackerone.com/reports/546753
- https://github.com/ngalongc/bug-bounty-reference

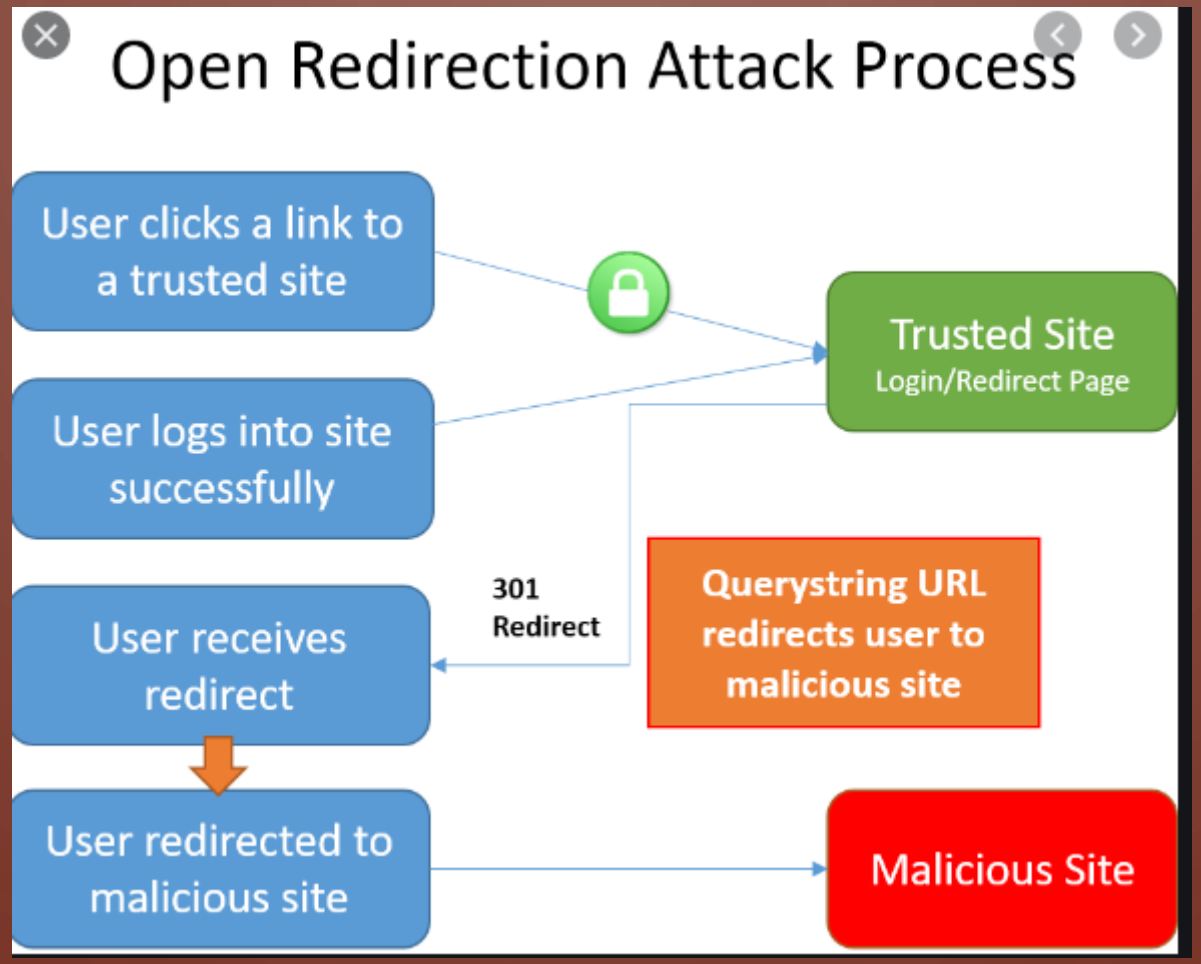
OPEN REDIRECT

- As vulnerabilidades de redirecionamento aberto surgem quando um aplicativo incorpora dados controláveis pelo usuário no destino de um redirecionamento de maneira insegura. Um invasor pode construir uma URL no aplicativo que causa um redirecionamento para um domínio externo arbitrário. Esse comportamento pode ser aproveitado para facilitar ataques de phishing contra usuários do aplicativo. A capacidade de usar um URL de aplicativo autêntico, direcionado ao domínio correto e com um certificado SSL válido (se o SSL for usado), confere credibilidade ao ataque de phishing, porque muitos usuários, mesmo que verifiquem esses recursos, não notarão o redirecionamento subsequente para um domínio diferente.
- https://portswigger.net/kb/issues/00500100_open-redirection-reflected

OPEN REDIRECT - EXEMPLO



OPEN REDIRECT - EXEMPLO



OPEN REDIRECT ATTACK - EXEMPLOS

- https://www.youtube.com/watch?v=4Jk_l-cw4WE
- https://www.youtube.com/watch?v=ReVORQcecGU
- https://www.youtube.com/watch?v=5-xzOCIMJts
- https://medium.com/@corneacristian/top-25-open-redirect-bug-bounty-reports-5ffe11788794
- https://hackerone.com/reports/504751
- https://hackerone.com/reports/373916
- https://www.youtube.com/watch?v=xBQY_b5MUkk
- https://www.youtube.com/watch?v=3P10Kd8m9bY
- https://www.youtube.com/watch?v=ul1a7EgHNNU
- https://medium.com/@nnez/1st-bug-bounty-write-up-open-redirect-vulnerability-on-login-page-5e0dd9a6eb69
- https://hackerone.com/reports/665651
- https://www.youtube.com/watch?v=Ozbygzfdzv4
- https://www.youtube.com/watch?v=X0mV9HXbKHY
- https://www.youtube.com/watch?v=X0mV9HXbKHY

CRLF INJECTION

- O termo refere-se a CRLF Carriage Return (ASCII 13, \r) Line Feed (ASCII 10, \n). Eles são usados para observar o término de uma linha, no entanto, tratado de maneira diferente nos populares sistemas operacionais de hoje. Por exemplo: no Windows, é necessário um CR e LF para observar o final de uma linha, enquanto no Linux / UNIX, um LF é necessário apenas. No protocolo HTTP, a sequência CR-LF é sempre usada para terminar uma linha.
- Um ataque de injeção de CRLF ocorre quando um usuário consegue enviar um CRLF para um aplicativo. Isso geralmente é feito modificando um parâmetro HTTP ou URL.

CRLF INJECTION

- Dependendo de como o aplicativo é desenvolvido, isso pode ser um problema menor ou uma falha de segurança bastante séria. Vejamos o último porque, afinal, esta é uma postagem relacionada à segurança.
- Vamos supor que um arquivo seja usado em algum momento para ler / gravar dados em algum tipo de log. Se um invasor conseguiu colocar uma CRLF, pode injetar algum tipo de método programático de leitura no arquivo. Isso pode resultar na gravação do conteúdo na tela na próxima tentativa de usar esse arquivo.
- Outro exemplo são os ataques de "divisão de respostas", em que os CRLFs são injetados em um aplicativo e incluídos na resposta. Os CRLFs extras são interpretados por proxies, caches e talvez navegadores como o final de um pacote, causando problemas.

https://owasp.org/www-community/vulnerabilities/CRLF_Injection

CRLF INJECTION - EXEMPLO

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Options | Alerts

1 × 2 × ...

Go | Cancel | < | > | Follow redirection

Target: http://[redacted]edu

Request

Raw | Params | Headers | Hex

```
GET
/redir.php?C=7dd4d1be87c147beabcb295451a65d41&URL=//xssposed.org+CRLF+INJECTION+XSS+OPEN+REDIR
ECT%0D%0A%20Content-Type:text/html%0D%0A%20Content-Length%0D%0A%20222%0A%20%3Cscript%3Ealert%2
8%27XSSPOSED+e3xpl0it%27%29%3C%2fscript HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: PHPSESSID=ng0rr06bvsmlt9d5stllloeb370
Connection: keep-alive
```

Response

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 301 Moved Permanently
Server: Apache/2.2.15 (Red Hat)
X-Powered-By: PHP/5.3.3
Location: //xssposed.org CRLF INJECTION XSS OPEN REDIRECT   Content-Type:text/html
Content-Length   222   <script>alert('XSSPOSED e3xpl0it')</script
Vary: Accept-Encoding,User-Agent
Content-Type: text/html; charset=UTF-8
cache-control: max-age=1800
Content-Length: 306
Date: Wed, 03 Jun 2015 16:31:41 GMT
X-Varnish: 716562032
Age: 0
Via: 1.1 varnish
Connection: keep-alive

Redirecting to <a href="//xssposed.org CRLF INJECTION XSS OPEN REDIRECT
Content-Type:text/html
Content-Length
222
<script>alert('XSSPOSED e3xpl0it')</script">//xssposed.org CRLF INJECTION XSS OPEN REDIRECT
Content-Type:text/html
Content-Length
222
<script>alert('XSSPOSED e3xpl0it')</script>
```

CRLF - EXEMPLOS

- https://medium.com/@briskinfosec/crlf-injection-attack-f0cb50554aab
- https://medium.com/bugbountywriteup/bugbounty-exploiting-crlf-injection-can-lands-into-a-nice-bounty-159525a9cb62
- https://www.netsparker.com/blog/web-security/crlf-http-header/
- https://hackerone.com/reports/446271
- https://medium.com/cyberverse/crlf-injection-playbook-472c67f1cb46
- https://hackerone.com/reports/52042
- https://www.youtube.com/watch?v=ODFfwW2kxCE
- https://www.youtube.com/watch?v=RUlb2LKEia8
- https://www.youtube.com/watch?v=rAzgbAvqfhI
- https://github.com/EdOverflow/bugbounty-cheatsheet/blob/master/cheatsheets/crlf.md

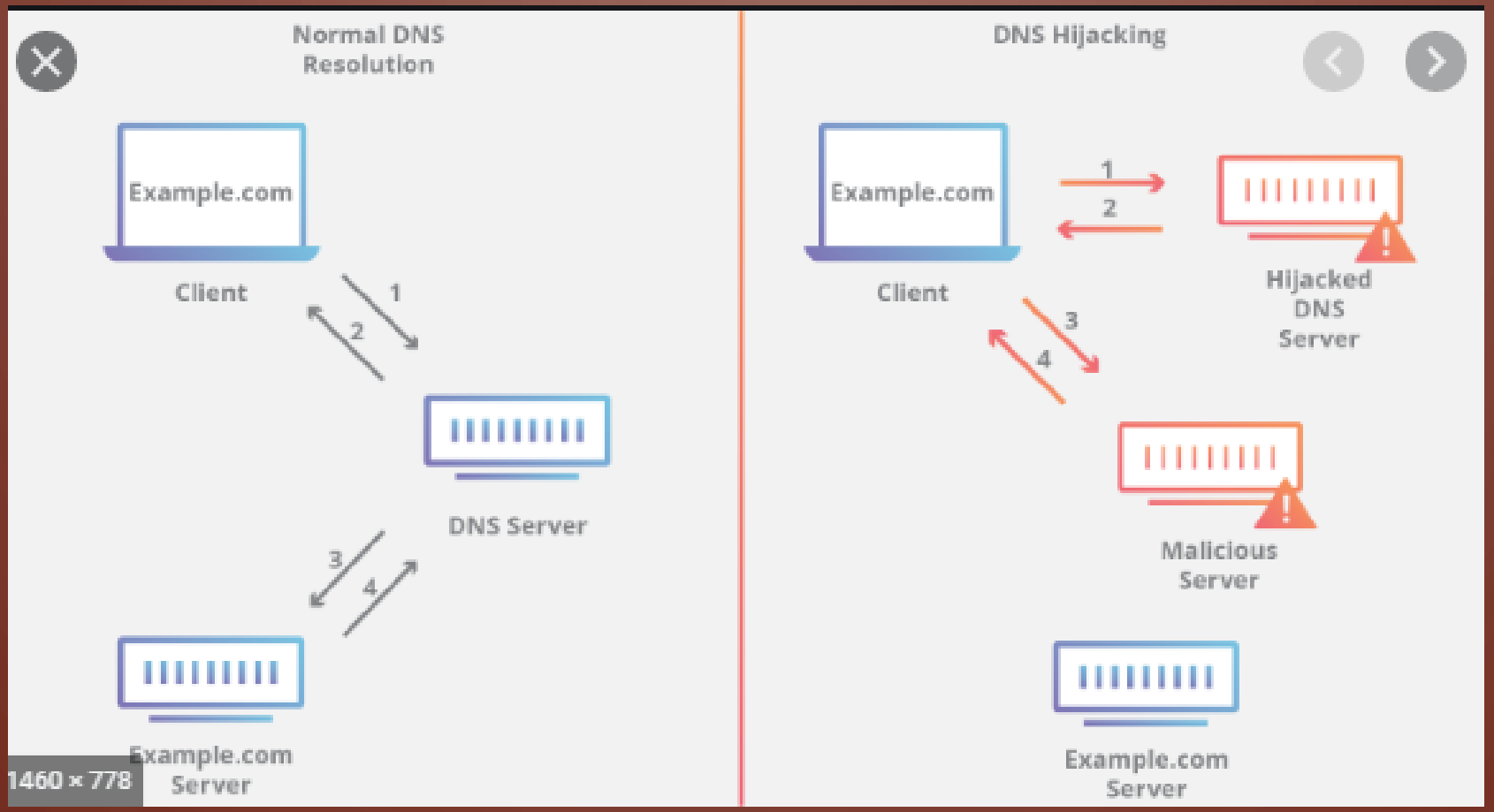
DNS HIJACKING

- O sequestro de servidor de nome de domínio (DNS), também chamado de redirecionamento de DNS, é um tipo de ataque DNS no qual as consultas DNS são resolvidas incorretamente para redirecionar inesperadamente usuários para sites maliciosos. Para realizar o ataque, os autores instalam malware nos computadores dos usuários, assumem o controle de roteadores ou interceptam ou cortam a comunicação DNS.
- O sequestro de DNS pode ser usado para pharming (nesse contexto, os atacantes geralmente exibem anúncios indesejados para gerar receita) ou para phishing (exibindo versões falsas de sites acessados por usuários e roubando dados ou credenciais).
- Muitos provedores de serviços de Internet (ISPs) também usam um tipo de sequestro de DNS para controlar as solicitações de DNS de um usuário, coletar estatísticas e retornar anúncios quando os usuários acessam um domínio desconhecido. Alguns governos usam o sequestro de DNS para censura, redirecionando usuários para sites autorizados pelo governo.

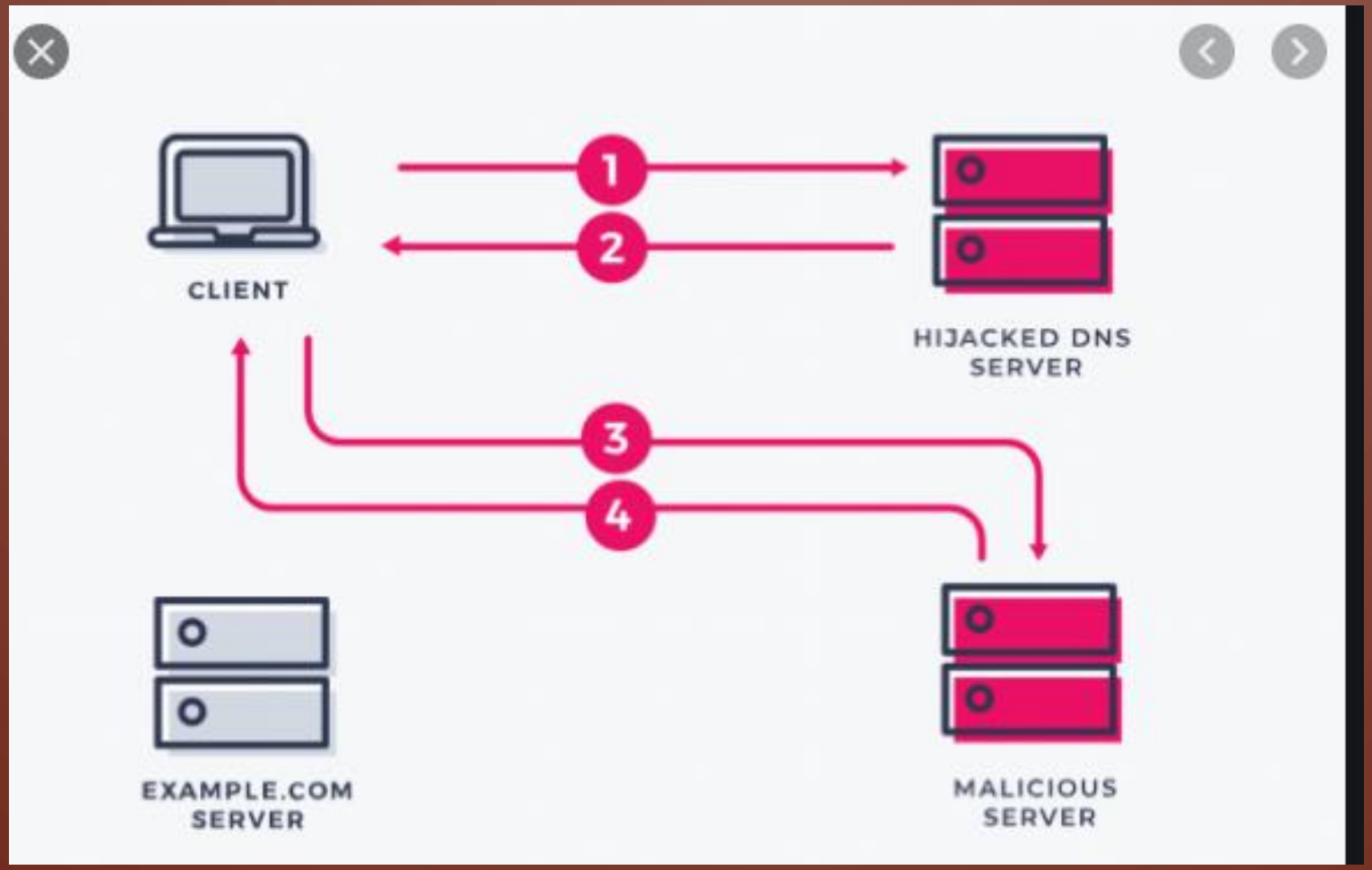
DNS HIJACKING

- Existem quatro tipos básicos de redirecionamento de DNS:
- Sequestro de DNS local - os invasores instalam o malware Trojan no computador do usuário e alteram as configurações de DNS local para redirecionar o usuário para sites maliciosos.
- Sequestro de DNS do roteador - muitos roteadores têm senhas padrão ou vulnerabilidades de firmware. Os invasores podem assumir o controle de um roteador e substituir as configurações de DNS, afetando todos os usuários conectados a esse roteador.
- MITM ataques DNS - os invasores interceptam a comunicação entre um usuário e um servidor DNS e fornecem endereços IP de destino diferentes, apontando para sites maliciosos.
- Servidor DNS não autorizado - os invasores podem invadir um servidor DNS e alterar os registros DNS para redirecionar solicitações de DNS para sites maliciosos.
- https://www.imperva.com/learn/application-security/dns-hijacking-redirection/

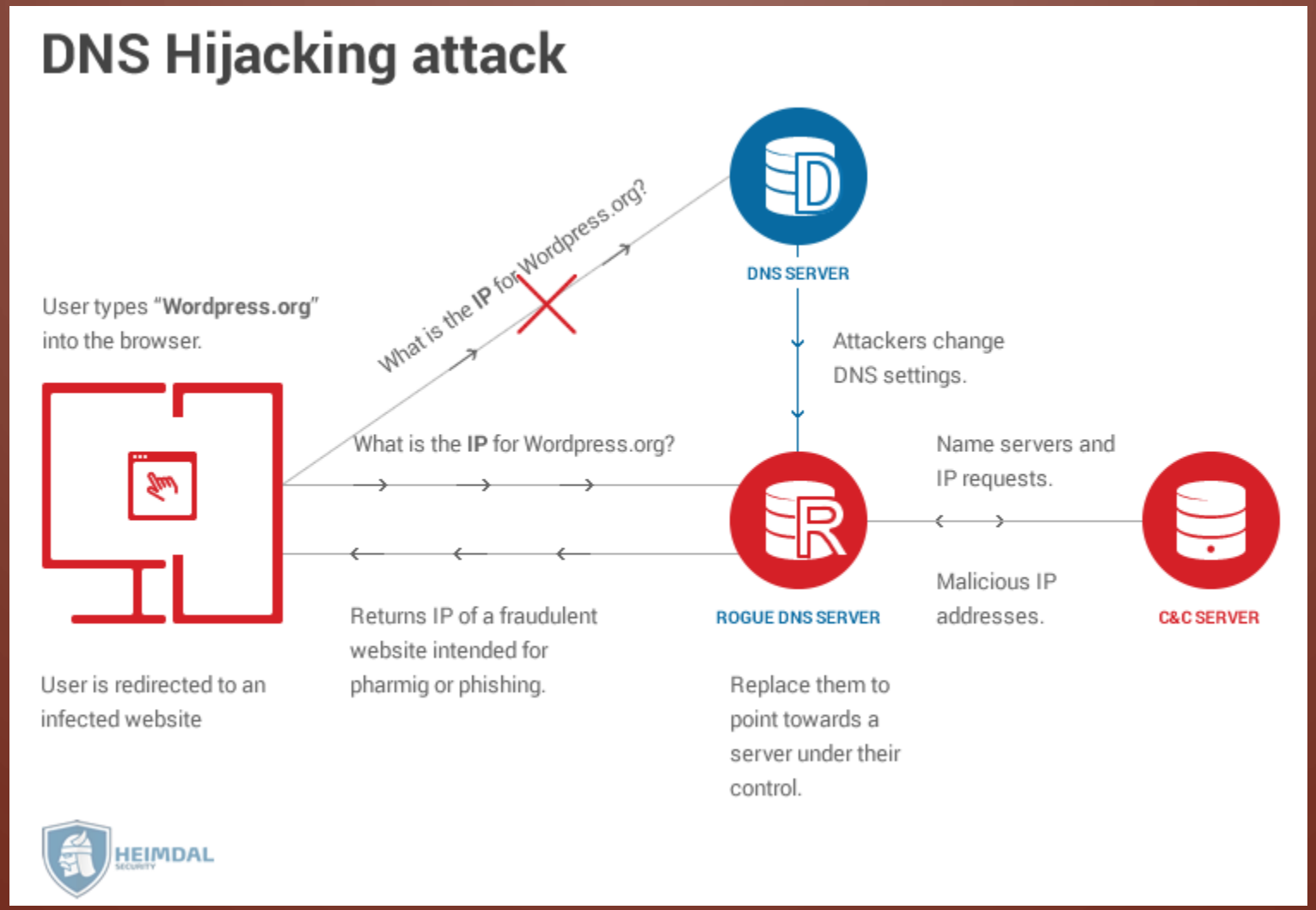
DNS HIJACKING - EXEMPLO



DNS HIJACKING - EXEMPLO



DNS HIJACKING - EXEMPLO



DNS HIJACKING - EXEMPLOS

- https://blog.eccouncil.org/what-is-dns-hijacking-and-how-to-combat-it/
- https://www.youtube.com/watch?v=4HkRpCBcXoE
- https://www.youtube.com/watch?v=HhJv8CU-RIk
- https://en.wikipedia.org/wiki/DNS_hijacking
- https://securitytrails.com/blog/dns-hijacking
- https://www.paloaltonetworks.com/cyberpedia/what-is-dns-hijacking

SESSION HIJACKING

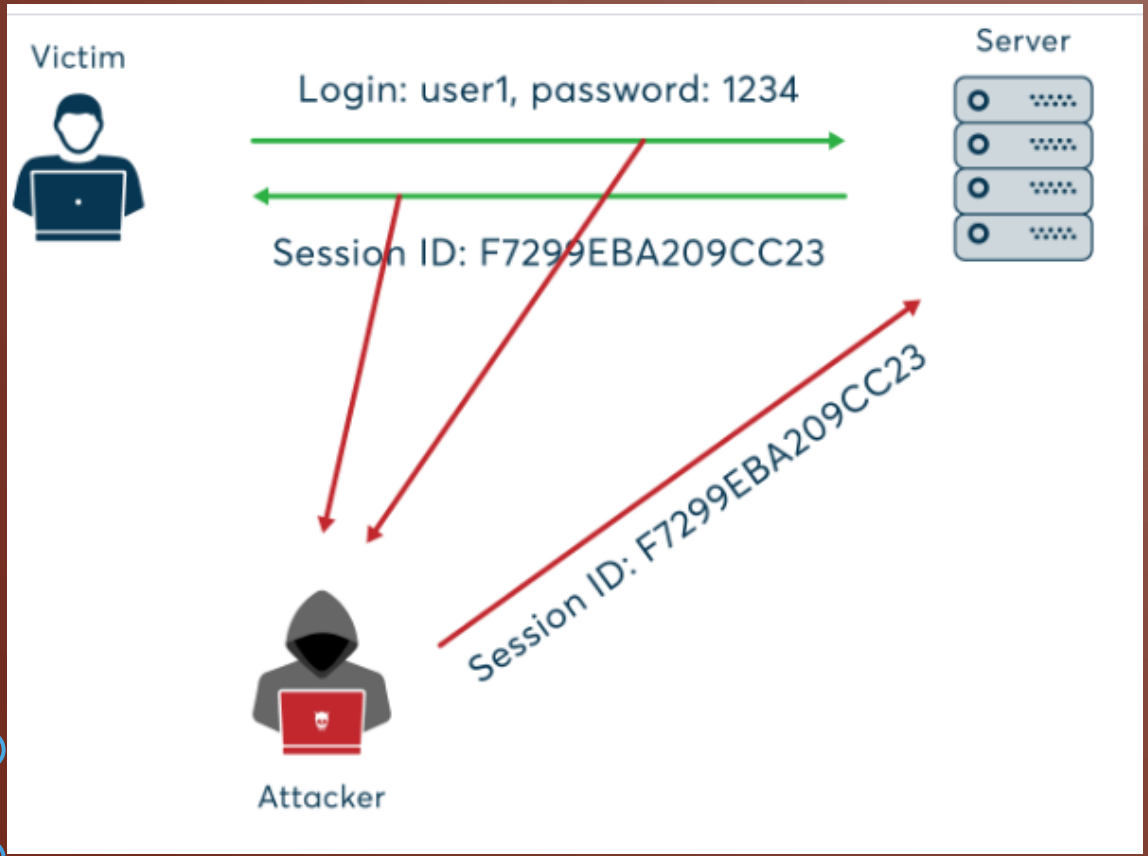
- O ataque de seqüestro de sessão consiste na exploração do mecanismo de controle de sessão da web, que normalmente é gerenciado para um token de sessão.
- Como a comunicação http usa muitas conexões TCP diferentes, o servidor da web precisa de um método para reconhecer as conexões de todos os usuários. O método mais útil depende de um token que o servidor da Web envia ao navegador do cliente após uma autenticação bem-sucedida do cliente. Um token de sessão é normalmente composto por uma sequência de largura variável e pode ser usado de diferentes maneiras, como na URL, no cabeçalho da requisição http como um cookie, em outras partes do cabeçalho da solicitação http ou ainda no corpo da requisição http.
- O ataque de seqüestro de sessão compromete o token da sessão roubando ou prevendo um token de sessão válido para obter acesso não autorizado ao servidor Web.

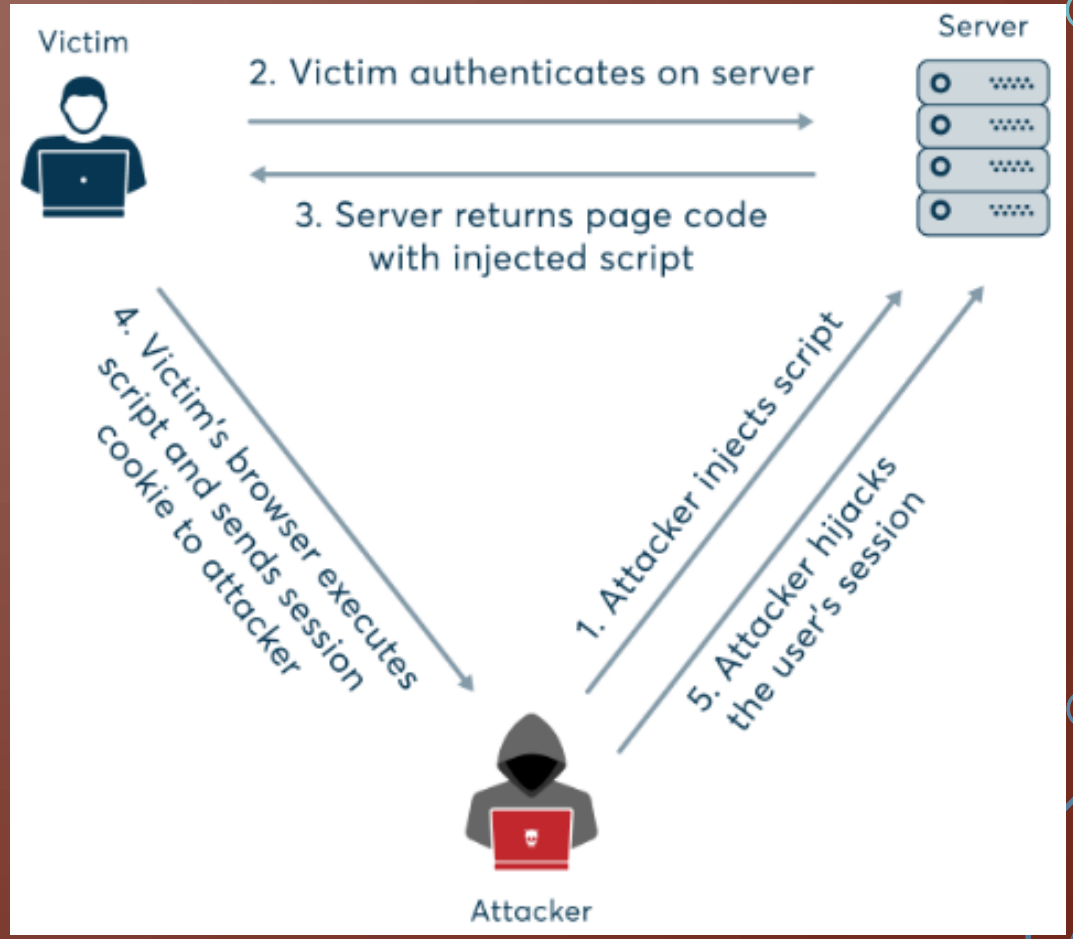
O token da sessão pode ser comprometido de diferentes maneiras; os mais comuns são:

- Token de sessão previsível;
- Sniffing de sessão;
- Ataques do lado do cliente (XSS, códigos JavaScript maliciosos, cavalos de Troia, etc);
- MITM
- MITB

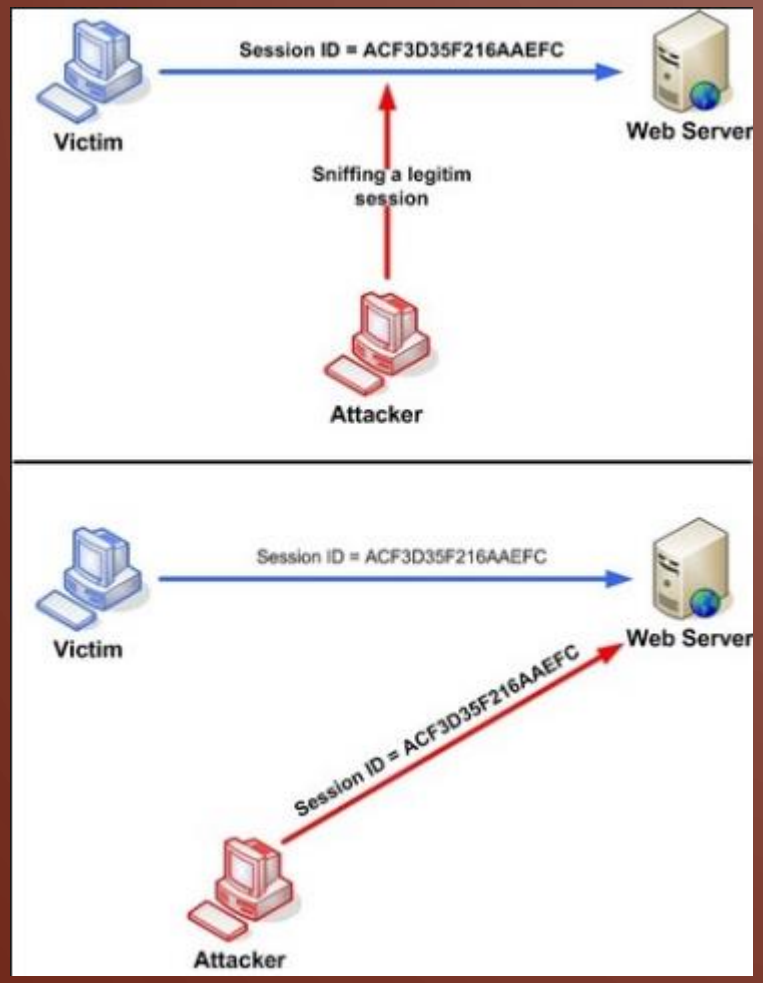
https://owasp.org/www-community/attacks/Session_hijacking_attack

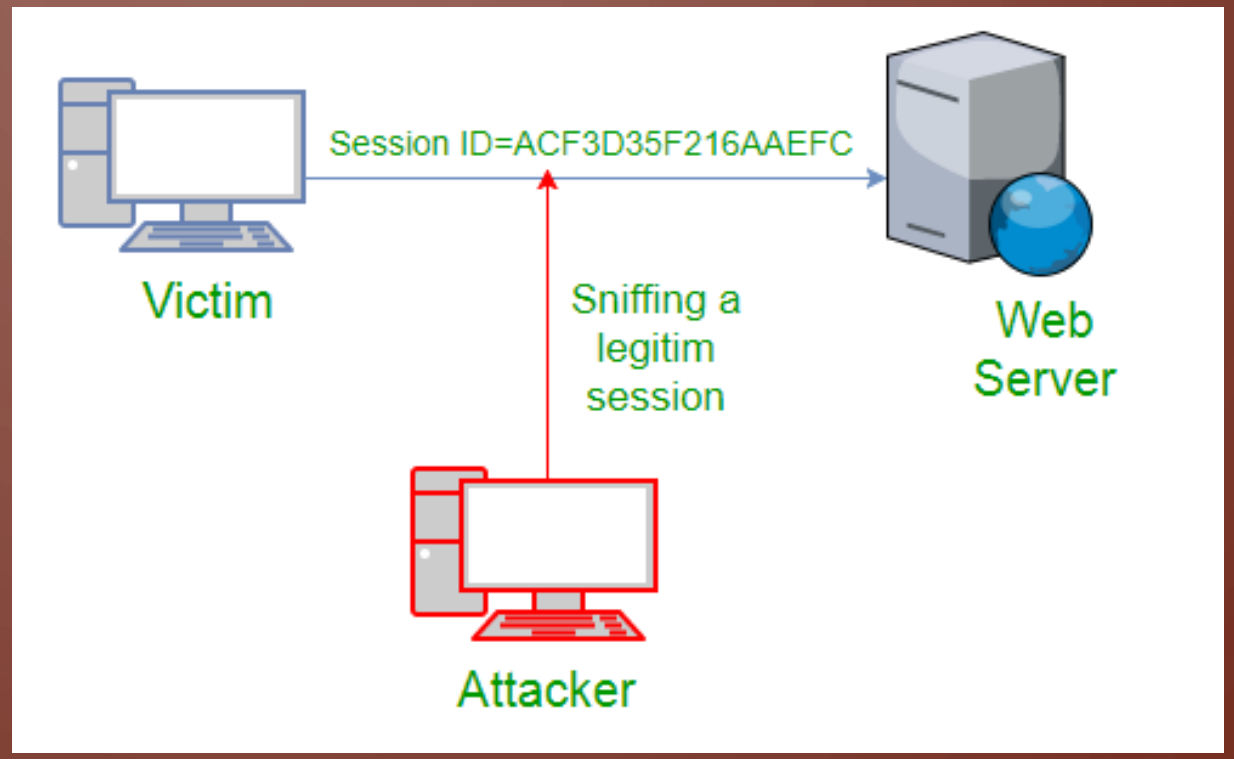
SESSION HIJACKING - EXEMPLO





SESSION HIJACKING - EXEMPLO





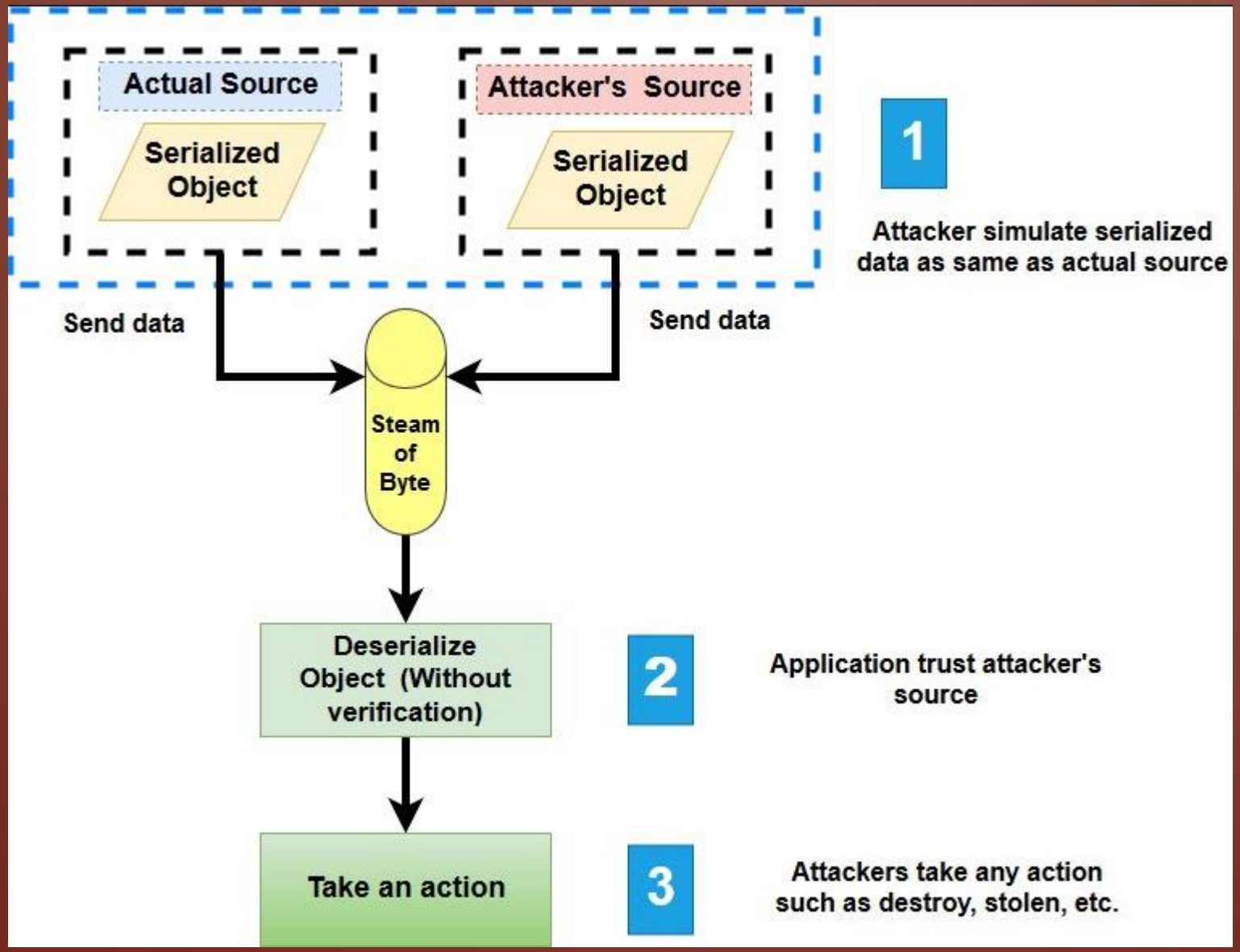
SESSION HIJACKING - EXEMPLOS

- https://www.youtube.com/watch?v=fxrCJNQ96Kg
- https://www.youtube.com/watch?v=OtlaTf9065w
- https://www.netsparker.com/blog/web-security/session-hijacking/
- https://www.diegomacedo.com.br/entendendo-o-sequestro-de-sessao-session-hijacking/
- https://www.youtube.com/watch?v=J5gAK1X8Jlk
- https://www.youtube.com/watch?v=0ep-vqrwhyM
- https://hackerone.com/reports/167460
- https://www.youtube.com/watch?v=tkSmaMlSQ9E
- https://www.youtube.com/watch?v=Ae6Dl9fYqWI
- https://threatpost.com/session-hijacking-bug-exposed-gitlab-users-private-tokens/127747/

INSECURE DESEREALIZATION

- A desserialização insegura é uma vulnerabilidade que ocorre quando dados não confiáveis são usados para abusar da lógica de um aplicativo, infligir um ataque de negação de serviço (DoS) ou mesmo executar código arbitrário após a **desserialização** . Também ocupa o 8º lugar na lista dos 10 melhores da OWASP 2017 .
- Para entender o que é desserialização insegura, primeiro precisamos entender o que são serialização e desserialização. Em seguida, abordaremos alguns exemplos de desserialização insegura e como ela pode ser usada para executar código, além de discutir algumas possíveis mitigações para essa classe de vulnerabilidade.
- **Serialização** refere-se a um processo de conversão de um objeto em um formato que pode ser mantido em disco (por exemplo, salvo em um arquivo ou armazenamento de dados), enviado por fluxos (por exemplo, stdout) ou enviado por uma rede. O formato no qual um objeto é serializado pode ser um texto binário ou estruturado (por exemplo, XML, JSON YAML...). JSON e XML são dois dos formatos de serialização mais usados em aplicativos da web.
- **A desserialização,** por outro lado, é o oposto da serialização, ou seja, transformar dados serializados provenientes de um arquivo, fluxo ou soquete de rede em um objeto.
- Os aplicativos da Web usam serialização e desserialização regularmente e a maioria das linguagens de programação fornece recursos nativos para serializar dados (especialmente em formatos comuns como JSON e XML). É importante entender que a desserialização **segura** de objetos é uma prática normal no desenvolvimento de software. O problema, no entanto, começa ao desserializar **a entrada não confiável do usuário** .
- https://www.acunetix.com/blog/articles/what-is-insecure-deserialization/

INSECURE DESEREALIZATION - EXEMPLO



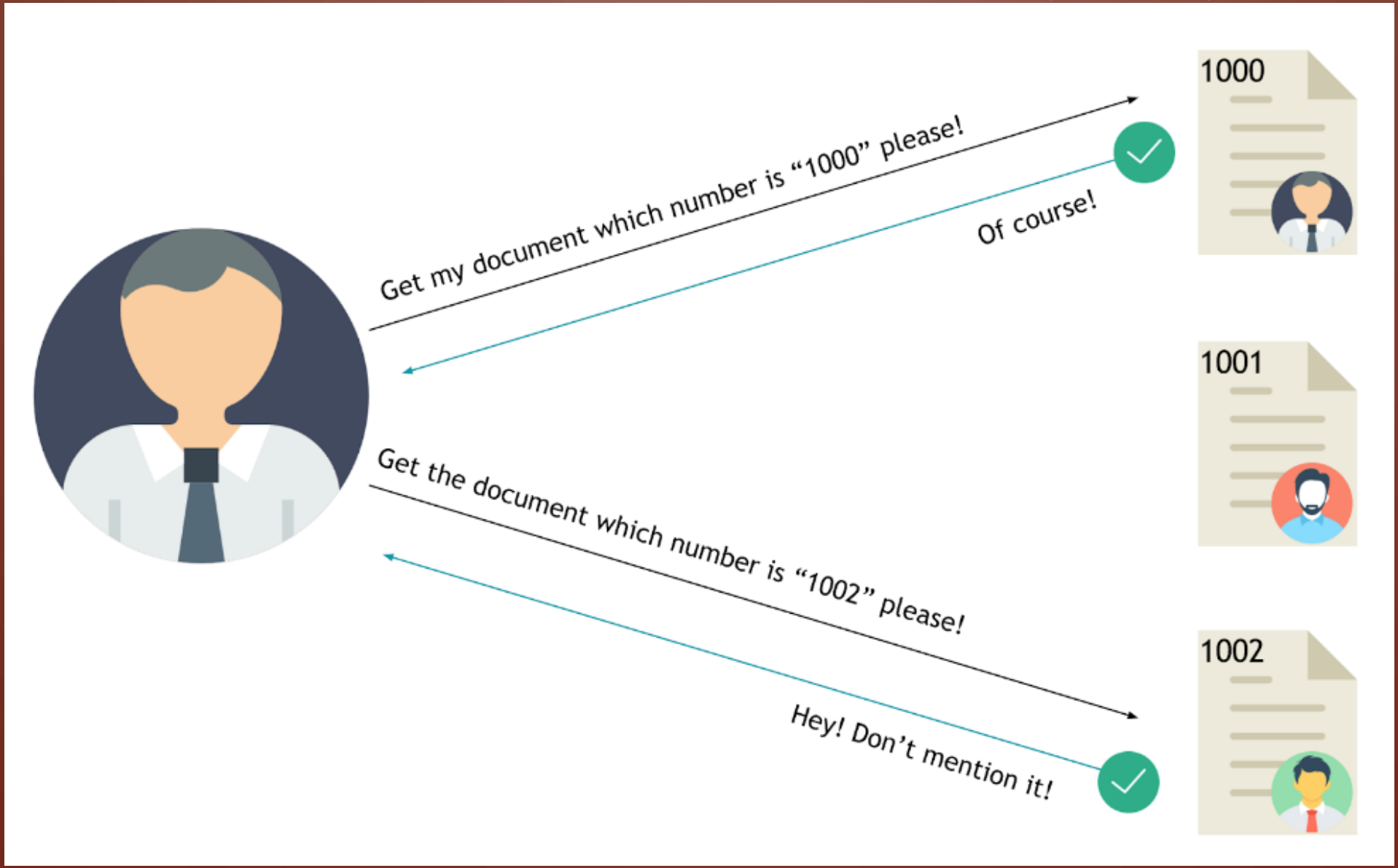
INSECURE DESERALIZATION - EXEMPLOS

- https://hdivsecurity.com/bornsecure/insecure-deserialization-attack-examples-mitigation/
- https://portswigger.net/web-security/deserialization
- https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A8-Insecure_Deserialization
- https://www.youtube.com/watch?v=nkTBwbnfesQ
- https://www.youtube.com/watch?v=EzOquQNQAUs
- https://www.youtube.com/watch?v=5grJYo9lqY0
- https://thehackerish.com/tag/insecure-deserialization-bug-bounty/
- https://hackerone.com/reports/350401
- https://hackerone.com/reports/838196
- https://www.exploit-db.com/docs/english/44756-deserialization-vulnerability.pdf

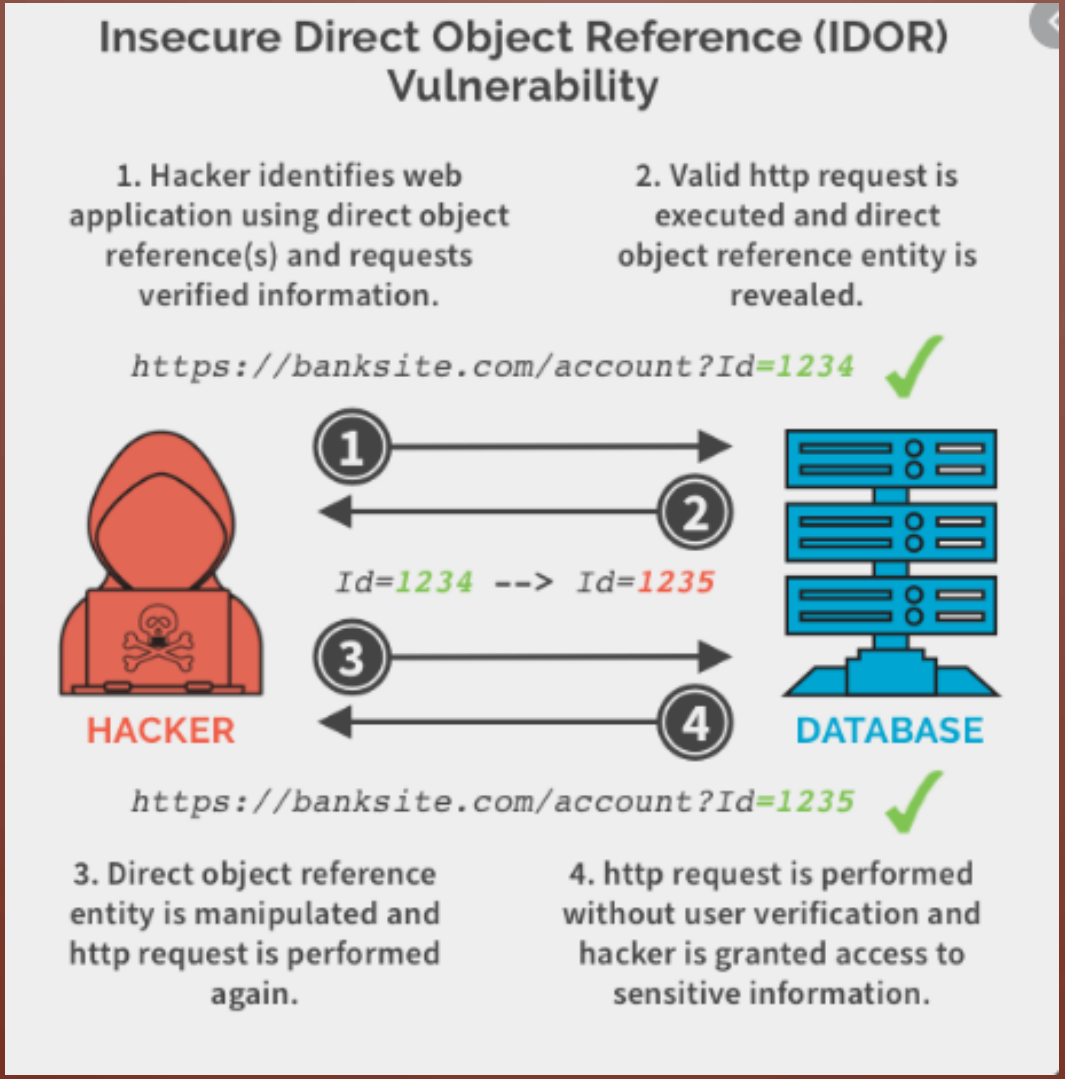
INSECURE DIRECT OBJECT REFERENCE (IDOR)

- As referências diretas a objetos inseguras (IDOR) são um tipo de vulnerabilidade de controle de acesso que surge quando um aplicativo usa a entrada fornecida pelo usuário para acessar objetos diretamente. O termo IDOR foi popularizado por sua aparição no Top Ten da OWASP 2007. No entanto, é apenas um exemplo de muitos erros de implementação do controle de acesso que podem levar a que os controles de acesso sejam contornados. As vulnerabilidades do IDOR são mais comumente associadas à escalação de privilégios horizontais, mas também podem surgir em relação à escalação de privilégios verticais.
- Existem muitos exemplos de vulnerabilidades de controle de acesso em que valores de parâmetros controlados pelo usuário são usados para acessar recursos ou funções diretamente.

INSECURE DIRECT OBJECT REFERENCE (IDOR)



INSECURE DIRECT OBJECT REFERENCE (IDOR)



INSECURE DIRECT OBJECT REFERENCE - EXEMPLOS

- https://owasp.org/www-chapter-ghana/assets/slides/IDOR.pdf
- https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html
- https://www.youtube.com/watch?v=bMYpGj2xzpM
- https://www.youtube.com/watch?v=rIoqMGcPMkI
- https://www.youtube.com/watch?v=TRDyvgkBcUs
- https://www.youtube.com/watch?v=ZodA76-CB10
- https://www.youtube.com/watch?v=zBQqJfLNm2I
- https://www.youtube.com/watch?v=jNuougtx1M0
- https://www.youtube.com/watch?v=3K1-a7dnA60
- https://www.youtube.com/watch?v=rcfEq6NUF7E
- https://www.youtube.com/watch?v=DwdiBZuix1k
- https://www.youtube.com/watch?v=-y-OcymRcZs
- https://www.youtube.com/watch?v=HQUxXE1oaIE
- https://www.youtube.com/watch?v=HQZ0ZawqhDY

RACE CONDITION

- Um ataque de condição de corrida acontece quando um sistema de computação projetado para lidar com tarefas em uma sequência específica é forçado a executar duas ou mais operações simultaneamente. Essa técnica aproveita um intervalo de tempo entre o momento em que um serviço é iniciado e o momento em que um controle de segurança entra em vigor. Esse ataque, que depende de aplicativos multithread, pode ser realizado de duas maneiras: interferência causada por processos não confiáveis (essencialmente um trecho de código que desliza em uma sequência entre as etapas de programas seguros) e interferência causada por um processo confiável, que podem ter os privilégios "mesmos ". Sem controles adequados, processos diferentes podem interferir entre si. Outros nomes usados para se referir a esta vulnerabilidade incluem ataques Tempo de verificação / tempo de uso ou TOC / TOU.
- Aplicativos da Web, sistemas de arquivos e ambientes de rede são todos vulneráveis a um ataque de condição de corrida. Os invasores podem ter como alvo uma lista de controle de acesso (ACL), uma folha de pagamento ou banco de dados de recursos humanos, um sistema transacional, um razão financeiro ou algum outro repositório de dados. Embora os ataques das condições de corrida não ocorram com freqüência - porque são relativamente difíceis de projetar e os atacantes devem explorar uma breve janela de oportunidade - quando acontecem, eles podem levar a sérias repercussões, incluindo um sistema que concede privilégios não autorizados. Além disso, ataques de condição de corrida são inerentemente difíceis de detectar.
- https://www.veracode.com/security/race-condition

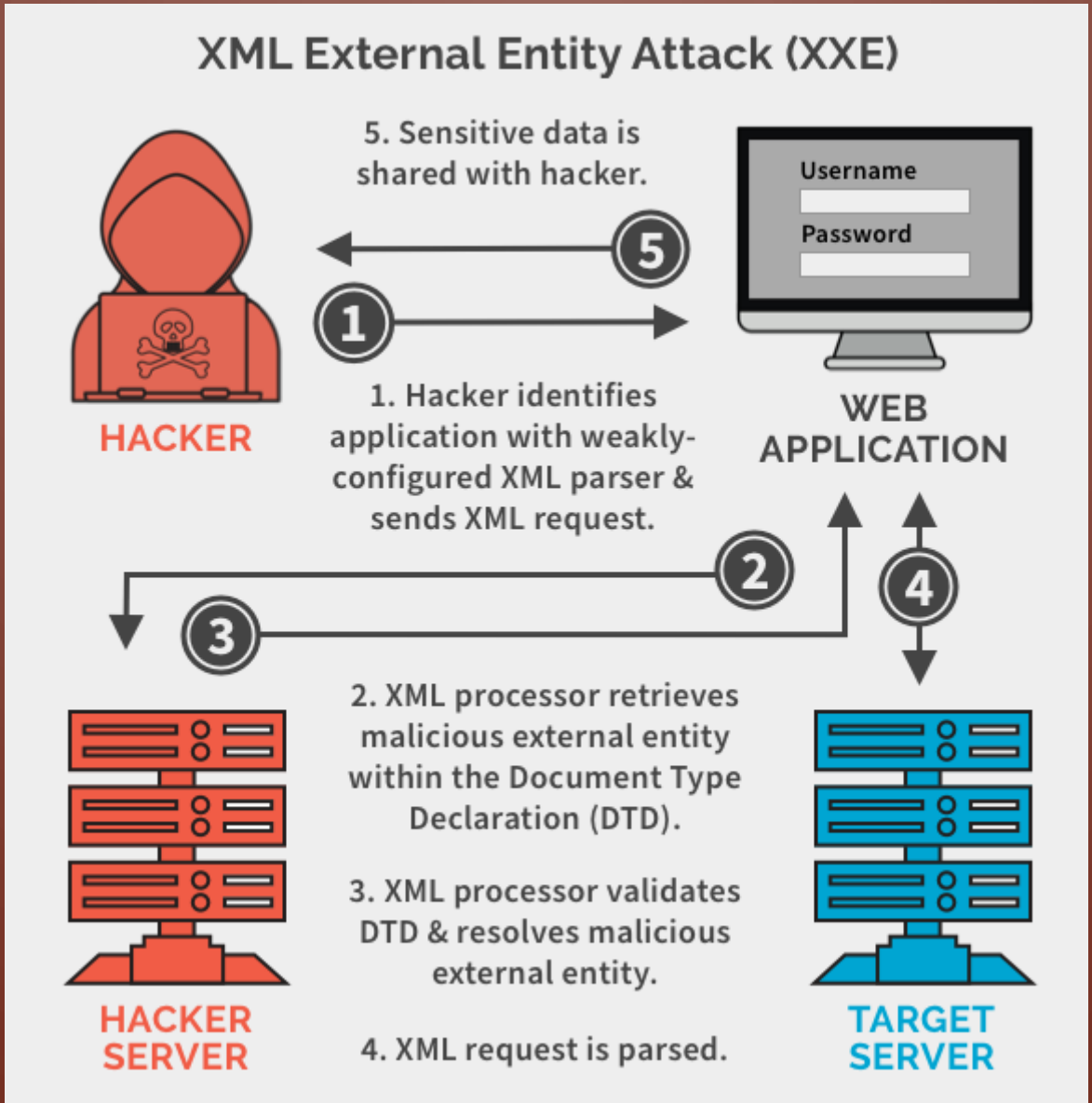
RACE CONDITION - EXEMPLOS

- https://hackaday.com/2015/04/29/race-conditions-exploit-granted-free-money-on-web-services/
- https://resources.infosecinstitute.com/category/certifications-training/secure_coding/race-condition-vulnerabilities/#gref
- http://www.cis.syr.edu/~wedu/Teaching/IntrCompSec/LectureNotes_New/Race_Condition.pdf
- https://en.wikipedia.org/wiki/Race_condition
- https://resources.securitycompass.com/blog/moving-beyond-the-owasp-top-10-part-1-race-conditions-2
- https://medium.com/@corneacristian/top-25-race-condition-bug-bounty-reports-84f9073bf9e5
- https://medium.com/@ciph3r7r0ll/chaining-improper-authorization-to-race-condition-to-harvest-credit-card-details-a-bug-bounty-effe6e0f5076
- https://hackerone.com/reports/454949
- https://blog.intigriti.com/2019/09/24/bug-bytes-37-how-to-find-more-idors-race-condition-to-rce-tracy/
- https://www.youtube.com/watch?v=R3B3JaaYpbI
- https://www.youtube.com/watch?v=9qPusaW6_CM
- https://medium.com/@stokochtrubbel/how-to-get-started-in-bug-bounty-9-pro-tips-69c13f3c74c6
- https://www.bugcrowd.com/resources/webinars/turbo-intruder-abusing-http-misfeatures-to-accelerate-attacks-by-james-kettle/

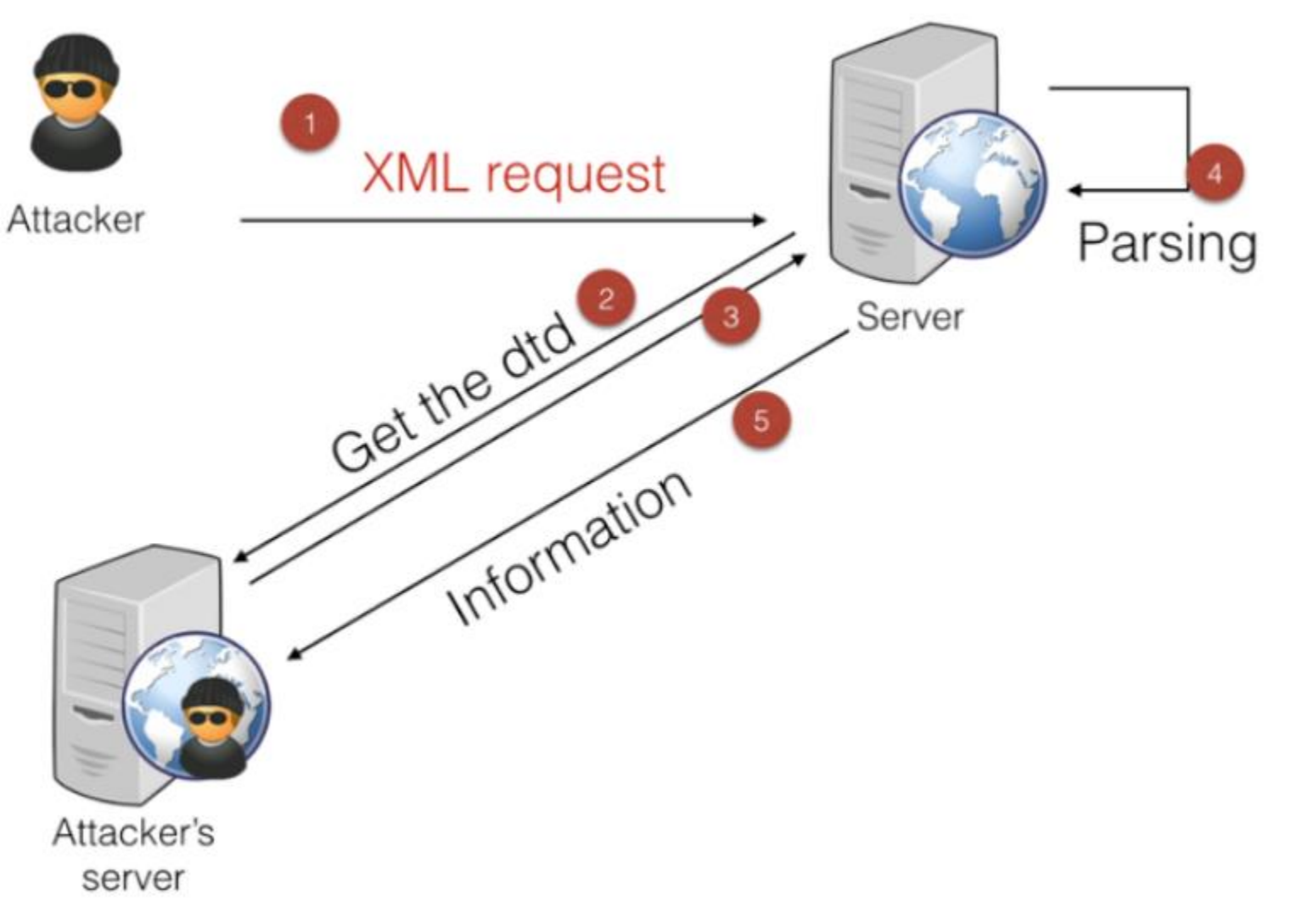
XML EXTERNAL ENTITY INJECTION (XXE)

- Um ataque de *entidade externa XML* é um tipo de ataque contra um aplicativo que analisa a entrada XML. Esse ataque ocorre quando a **entrada XML que contém uma referência a uma entidade externa é processada por um analisador XML mal configurado** . Esse ataque pode levar à divulgação de dados confidenciais, negação de serviço, falsificação de solicitação do servidor, varredura de portas na perspectiva da máquina em que o analisador está localizado e outros impactos no sistema.
- Os ataques podem incluir a divulgação de arquivos locais, que podem conter dados confidenciais, como senhas ou dados de usuários particulares, usando esquemas file: ou caminhos relativos no identificador do sistema. Como o ataque ocorre em relação ao aplicativo que processa o documento XML, um invasor pode usar esse aplicativo confiável para dinamizar outros sistemas internos, possivelmente divulgando outro conteúdo interno por meio de solicitações http (s) ou iniciando um CSRFataque a quaisquer serviços internos desprotegidos. Em algumas situações, uma biblioteca de processador XML vulnerável a problemas de corrupção de memória no cliente pode ser explorada desreferenciando um URI mal-intencionado, possivelmente permitindo a execução arbitrária de código na conta do aplicativo. Outros ataques podem acessar recursos locais que podem não parar de retornar dados, possivelmente afetando a disponibilidade do aplicativo se muitos processos ou threads não forem liberados.
- https://owasp.org/www-community/vulnerabilities/XML_External_Entity_(XXE)_Processing

XML EXTERNAL ENTITY INJECTION (XXE) - EXEMPLOS



XML EXTERNAL ENTITY INJECTION (XXE) - EXEMPLOS



XML EXTERNAL ENTITY INJECTION (XXE) - EXEMPLOS

- http://mamaquieroserpentester.blogspot.com/2018/02/xml-external-entity-you-can-call-me-xxe.html
- https://www.infosec.com.br/xml-external-entity/
- https://www.youtube.com/watch?v=UhgCFQb_dw8
- https://www.youtube.com/watch?v=gjm6VHZa_8s
- https://www.youtube.com/watch?v=IMw2C6EJaDo
- https://www.youtube.com/watch?v=FR1uq-hDpHg
- https://www.youtube.com/watch?v=EZxGa6dqero
- https://www.youtube.com/watch?v=DREgLWZqMWg
- https://portswigger.net/web-security/xxe
- https://www.youtube.com/watch?v=mF7SSXYMS2o
- https://www.youtube.com/watch?v=IGz5MOUz7Ws
- https://www.bugcrowd.com/resources/glossary/xml-external-entity-injection-xxe/
- https://hackerone.com/reports/500515

USER ACCOUNT TAKEOVER

- As aquisições de contas acontecem regularmente em praticamente qualquer site com uma função de login.
- O preenchimento de credenciais e a quebra de cartões estão entre as técnicas mais comuns de controle de contas e cada uma usa bots automatizados para obter entrada de força bruta em uma conta.
  - Os ataques de preenchimento de credenciais vasculham listas de nomes de usuário e senhas vazados, usando bots para testar continuamente combinações em vários sites até que sejam bem-sucedidas.
  - Os ataques de quebra de cartão usam bots automatizados para combinar nomes de usuário e dicionários de senhas vazados, até que o código seja quebrado.
- Nomes de usuário e senhas são adquiridos a partir de despejos de dados em massa que são facilmente acessíveis na dark web. Cada despejo de dados pode consistir em milhões de combinações de nome de usuário e senha após anos de violações de dados realizadas em vários sites.
- O desafio para as empresas reside não apenas na disponibilidade e preço baixo dos despejos de dados, mas também no comportamento do consumidor. Com mais senhas para controlar, os consumidores frequentemente reutilizam detalhes de login em vários sites e negligenciam atualizações de senha por anos seguidos.
- https://www.netacea.com/what-is-account-takeover

USER ACCOUNT TAKEOVER - EXEMPLOS

- https://www.shieldsquare.com/what-is-account-takeover/
- https://precognitive.com/2019/09/26/account-takeover-scenarios/
- https://www.youtube.com/watch?v=pmqzztDVolw
- https://www.usenix.org/conference/enigma2018/presentation/milka
- https://www.youtube.com/watch?v=U3Of-jF1nWo
- https://www.securityweek.com/instagram-account-takeover-vulnerability-earns-hacker-30000
- https://www.youtube.com/watch?v=2326m6ddthg
- https://www.youtube.com/watch?v=R0mWptLglhk
- https://medium.com/@0xankush/readme-com-account-takeover-bugbounty-fulldisclosure-a36ddbe915be
- https://blog.securitybreached.org/2020/01/22/user-account-takeover-via-signup-feature-bug-bounty-poc/
- https://hackerone.com/reports/745324

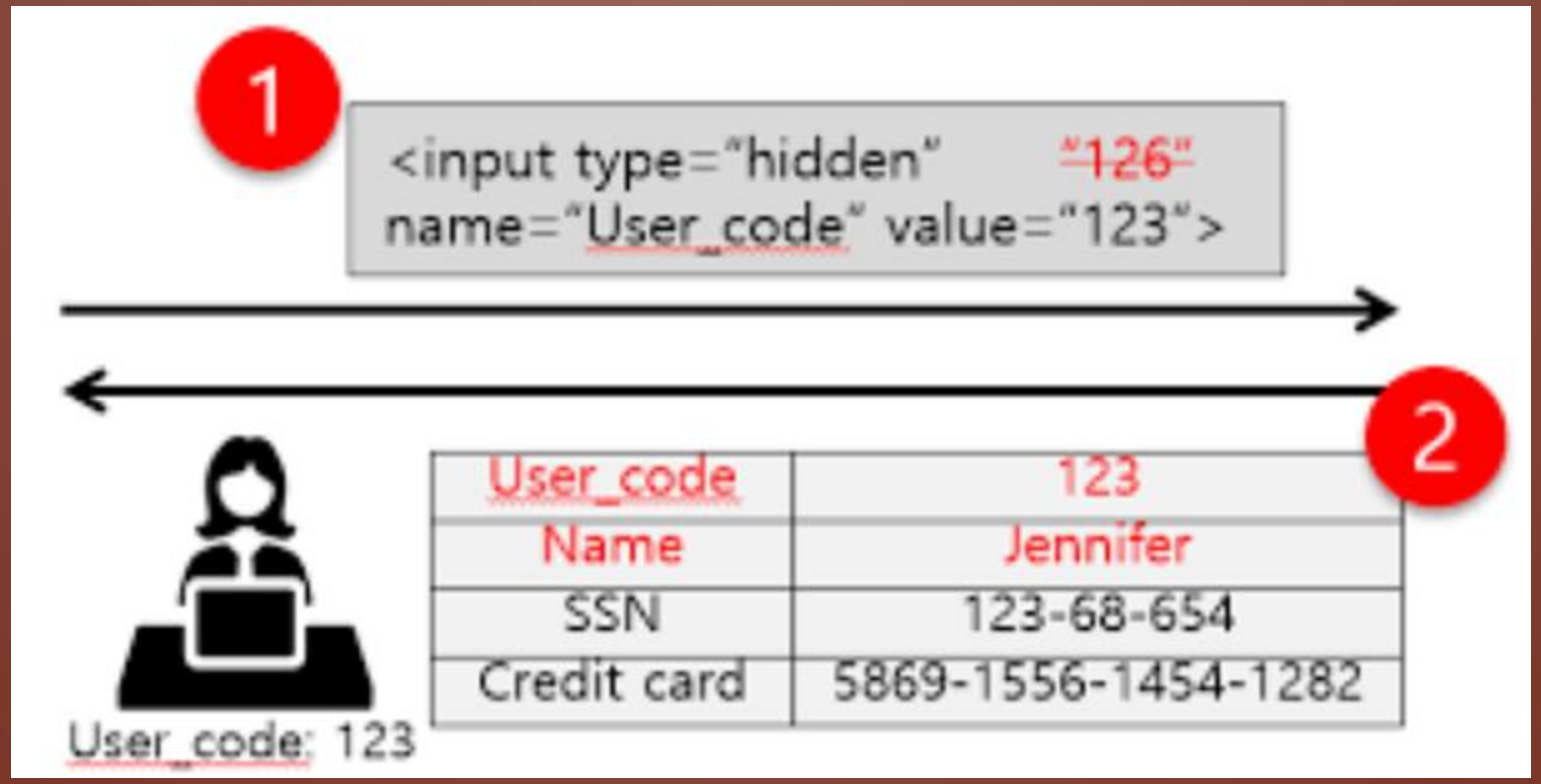
IMPROPER ACCESS CONTROL

- O controle de acesso, às vezes chamado de autorização, é como um aplicativo da Web concede acesso a conteúdo e funções para alguns usuários e não para outros. Essas verificações são realizadas após a autenticação e controlam o que os usuários 'autorizados' podem fazer. O controle de acesso parece um problema simples, mas é insidiosamente difícil de implementar corretamente. O modelo de controle de acesso de um aplicativo da Web está intimamente ligado ao conteúdo e às funções que o site fornece. Além disso, os usuários podem se enquadrar em vários grupos ou funções com diferentes habilidades ou privilégios.
- Os desenvolvedores frequentemente subestimam a dificuldade de implementar um mecanismo de controle de acesso confiável. Muitos desses esquemas não foram deliberadamente projetados, mas simplesmente evoluíram junto com o site. Nesses casos, as regras de controle de acesso são inseridas em vários locais em todo o código. À medida que o site se aproxima da implantação, a coleção ad hoc de regras se torna tão difícil que é quase impossível entender.

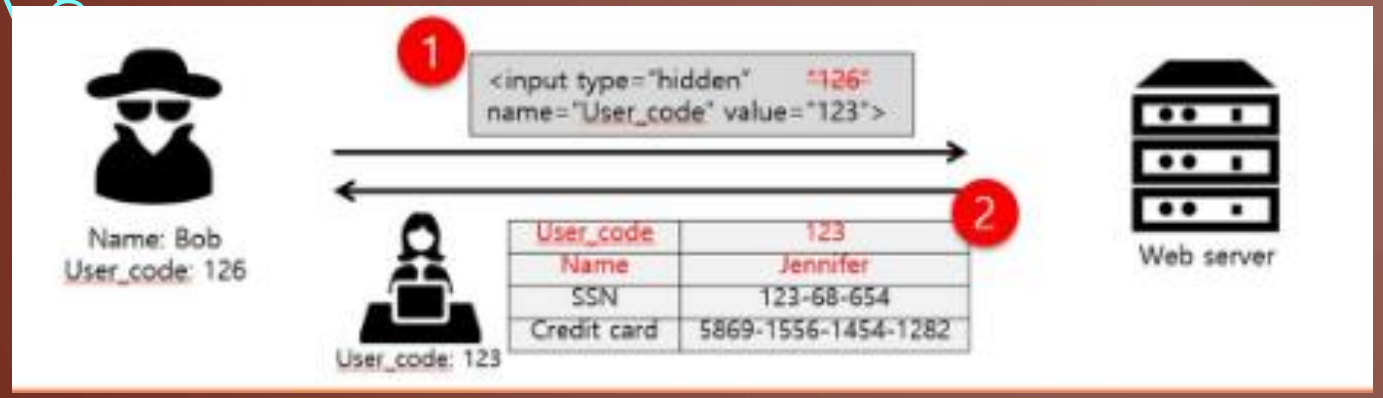
IMPROPER ACCESS CONTROL

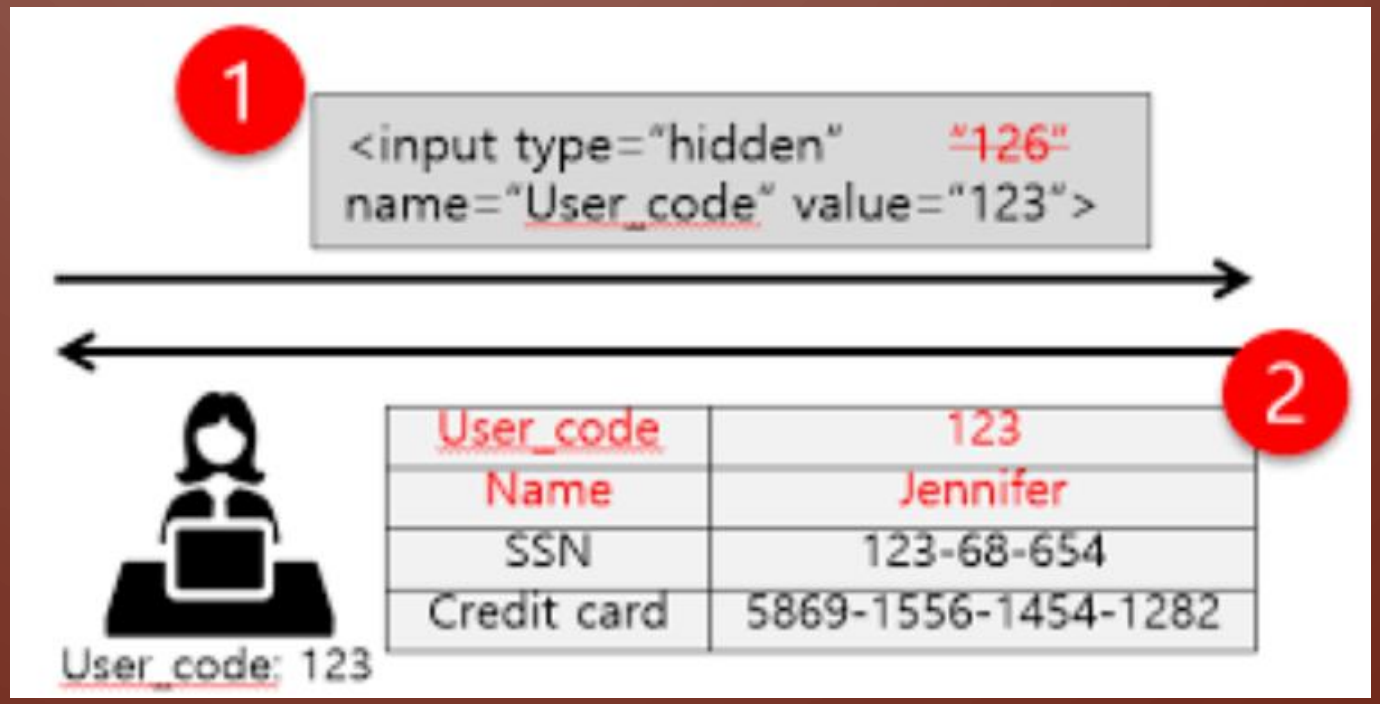
- Muitos desses esquemas de controle de acesso defeituosos não são difíceis de descobrir e explorar. Frequentemente, tudo o que é necessário é elaborar uma solicitação de funções ou conteúdo que não deve ser concedido. Depois que uma falha é descoberta, as consequências de um esquema de controle de acesso defeituoso podem ser devastadoras. Além de exibir conteúdo não autorizado, um invasor pode alterar ou excluir conteúdo, executar funções não autorizadas ou até mesmo assumir a administração do site.
- Um tipo específico de problema de controle de acesso são as interfaces administrativas que permitem que os administradores do site gerenciem um site pela Internet. Esses recursos são frequentemente usados para permitir que os administradores do site gerenciem com eficiência usuários, dados e conteúdo em seus sites. Em muitos casos, os sites oferecem suporte a várias funções administrativas para permitir granularidade mais fina da administração do site. Devido ao seu poder, essas interfaces são frequentemente os principais alvos de ataques de pessoas de fora e de dentro.
- https://owasp.org/www-community/Broken_Access_Control

IMPROPER ACCESS CONTROL - EXEMPLOS



IMPROPER ACCESS CONTROL - EXEMPLOS





IMPROPER ACCESS CONTROL - EXEMPLOS

- https://hdivsecurity.com/owasp-broken-access-control
- https://www.youtube.com/watch?v=94-tlOCApOc
- https://www.youtube.com/watch?v=ZG7gUwyxZBY
- https://www.youtube.com/watch?v=P38at6Tp8Ms
- https://www.packetlabs.net/broken-access-control/
- https://www.youtube.com/watch?v=EE2N2H3_RnE
- https://www.youtube.com/watch?v=TJQpOrtet8E
- https://www.youtube.com/watch?v=Z2Eo_5jEOwA
- https://www.youtube.com/watch?v=UNxSVYVyalo
- https://medium.com/@deepakdramz/bug-bounty-for-beginners-part-2-broken-access-control-7755f5c61937
- https://medium.com/bugbountywriteup/dank-writeup-on-broken-access-control-on-an-indian-startup-d29132a1ecd
- https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A5-Broken_Access_Control
- https://www.youtube.com/watch?v=A6YX6lvQfQY
- https://www.youtube.com/watch?v=zeb82DDH5HM
- https://hackerone.com/reports/502593

PHP OBJECT INJECTION

- O PHP Object Injection é uma vulnerabilidade no nível do aplicativo que pode permitir que um invasor execute diferentes tipos de ataques maliciosos, como Injeção de Código , Injeção de SQL , Traversal de Caminho e Negação de Serviço de Aplicativo , dependendo do contexto. A vulnerabilidade ocorre quando a entrada fornecida pelo usuário não é higienizada adequadamente antes de ser passada para a função PHP unserialize (). Como o PHP permite a serialização de objetos, os invasores podem transmitir seqüências serializadas ad-hoc a uma chamada unserialize () vulnerável, resultando em uma injeção arbitrária de objetos PHP no escopo do aplicativo.
- Para explorar com êxito uma vulnerabilidade de injeção de objetos PHP, duas condições devem ser atendidas:
  - O aplicativo deve ter uma classe que implemente um método mágico do PHP (como __wakeup ou __destruct) que possa ser usado para realizar ataques maliciosos ou para iniciar uma "cadeia POP".
  - Todas as classes usadas durante o ataque devem ser declaradas quando o unserialize () vulnerável está sendo chamado, caso contrário, o carregamento automático de objetos deve ser suportado para essas classes.

https://owasp.org/www-community/vulnerabilities/PHP_Object_Injection

PHP OBJECT INJECTION - EXEMPLOS

- https://www.youtube.com/watch?v=gTXMFrctYLE
- https://www.youtube.com/watch?v=HaW15aMzBUM
- https://www.youtube.com/watch?v=wp8LeTgNdyU
- https://www.youtube.com/watch?v=A-_Ow-qVD34
- https://www.youtube.com/watch?v=m13W6NqsgQY
- https://www.youtube.com/watch?v=LywLzazH1JA
- https://www.youtube.com/watch?v=pCbBfxJJn4E
- https://www.youtube.com/watch?v=GE2HyC7Gwrs
- https://www.youtube.com/watch?v=uW4yd9i9Ltw
- https://blog.ripstech.com/2018/php-object-injection/
- https://www.tarlogic.com/en/blog/how-php-object-injection-works-php-object-injection/

SSI INJECTION

- SSIs são diretrizes presentes em aplicativos da Web usados para alimentar uma página HTML com conteúdo dinâmico. Eles são semelhantes aos CGIs, exceto que os SSIs são usados para executar algumas ações antes que a página atual seja carregada ou enquanto a página está sendo visualizada. Para fazer isso, o servidor da web analisa o SSI antes de fornecer a página ao usuário.
- O ataque Inclui do lado do servidor permite a exploração de um aplicativo da Web injetando scripts em páginas HTML ou executando códigos arbitrários remotamente. Ele pode ser explorado através da manipulação do SSI em uso no aplicativo ou forçar seu uso através dos campos de entrada do usuário.

SSI INJECTION

- É possível verificar se o aplicativo está validando corretamente os dados dos campos de entrada inserindo caracteres usados nas diretivas SSI, como:
- < ! # = / . " - > and [a-zA-Z0-9]
- Outra maneira de descobrir se o aplicativo está vulnerável é verificar a presença de páginas com extensão .stm, .shtm e .shtml. No entanto, a falta desse tipo de página não significa que o aplicativo esteja protegido contra ataques SSI.
- De qualquer forma, o ataque será bem-sucedido apenas se o servidor da Web permitir a execução do SSI sem a validação adequada. Isso pode levar ao acesso e manipulação do sistema e processo de arquivos sob a permissão do proprietário do processo do servidor da web.
- O invasor pode acessar informações confidenciais, como arquivos de senha, e executar comandos do shell. As diretivas SSI são injetadas nos campos de entrada e enviadas ao servidor da web. O servidor da Web analisa e executa as diretivas antes de fornecer a página. Em seguida, o resultado do ataque estará visível na próxima vez que a página for carregada no navegador do usuário.

https://owasp.org/www-community/attacks/Server-Side_Includes_(SSI)_Injection

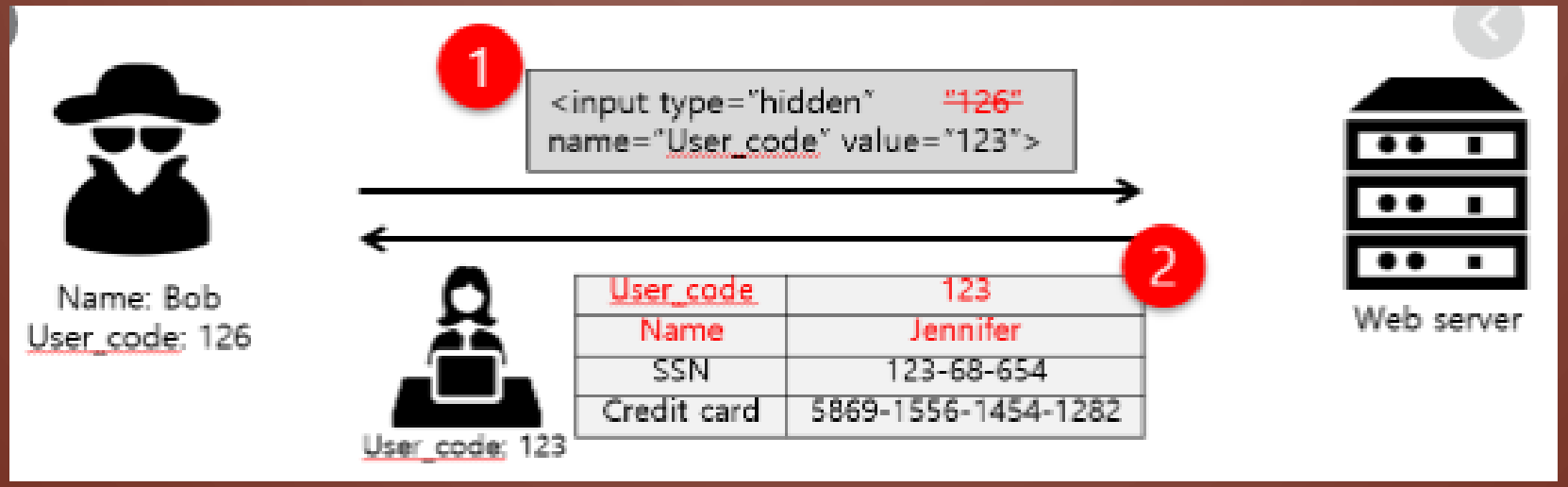
SSI INJECTION - EXEMPLOS

- https://www.youtube.com/watch?v=2Jo_-vDE4io
- https://www.youtube.com/watch?v=iH4TzbqHkkc
- https://www.youtube.com/watch?v=B5rkWcOAfmA
- https://owasp.org/www-community/attacks/Server-Side_Includes_(SSI)_Injection
- https://portswigger.net/kb/issues/00101100_ssi-injection
- https://medium.com/@shatabda/security-ssi-injection-what-how-fbce1dc232b9
- https://hackerone.com/reports/159985

WEB PARAMETER TAMPERING

- O ataque de violação de parâmetros da Web baseia-se na manipulação de parâmetros trocados entre cliente e servidor, a fim de modificar dados de aplicativos, como credenciais e permissões de usuário, preço e quantidade de produtos, etc. Geralmente, essas informações são armazenadas em cookies, de forma oculta campos ou cadeias de consulta de URL e é usado para aumentar a funcionalidade e o controle do aplicativo.
- Esse ataque pode ser realizado por um usuário mal-intencionado que deseja explorar o aplicativo para seu próprio benefício ou por um invasor que deseja atacar uma terceira pessoa usando um ataque Man-in-the-middle . Nos dois casos, ferramentas como Webscarab e Paros proxy são usadas principalmente.
- O êxito do ataque depende dos erros do mecanismo de validação da integridade e da lógica e sua exploração pode resultar em outras consequências, incluindo ataques XSS , Injeção SQL , inclusão de arquivo e divulgação de caminhos.
- https://owasp.org/www-community/attacks/Web_Parameter_Tampering

WEB PARAMETER TAMPERING - EXEMPLOS



WEB PARAMETER TAMPERING - EXEMPLOS

WEB PARAMETER TAMPERING - EXEMPLOS

- https://www.imperva.com/learn/application-security/parameter-tampering/
- https://www.youtube.com/watch?v=6zjAv_BBDUQ
- https://www.youtube.com/watch?v=YOLubxfJUnU
- https://www.youtube.com/watch?v=MNDkzkR8TBI
- https://www.youtube.com/watch?v=38lEboR-BI4
- https://www.youtube.com/watch?v=Gune0TXAjYg
- https://medium.com/@chawdamrunal/what-is-parameter-tampering-5b1beb12c5ba
- https://www.facebook.com/watch/?v=2179943992316316
- https://www.facebook.com/watch/?v=2200932383550741
- https://www.youtube.com/watch?v=qCU3vZrO3vA
- https://medium.com/bugbountywriteup/shopping-products-for-free-parameter-tampering-vulnerability-8e09e1471596

CLICKJACKING

- O clickjacking, também conhecido como “ataque de reparação da interface do usuário”, ocorre quando um invasor usa várias camadas transparentes ou opacas para induzir um usuário a clicar em um botão ou link em outra página quando pretendia clicar na página de nível superior. Assim, o invasor está "sequestrando" cliques destinados à sua página e os encaminha para outra página, provavelmente pertencente a outro aplicativo, domínio ou ambos.
- Usando uma técnica semelhante, as teclas também podem ser seqüestradas. Com uma combinação cuidadosamente elaborada de folhas de estilo, iframes e caixas de texto, um usuário pode ser levado a acreditar que está digitando a senha do email ou da conta bancária, mas digitando em um quadro invisível controlado pelo invasor.

CLICKJACKING

- Por exemplo, imagine um invasor que constrói um site com um botão que diz "clique aqui para obter um iPod grátis". No entanto, no topo dessa página da web, o invasor carregou um iframe com sua conta de e-mail e alinhou exatamente o botão "excluir todas as mensagens" diretamente na parte superior do botão "iPod grátis". A vítima tenta clicar no botão "iPod grátis", mas na verdade clica no botão invisível "excluir todas as mensagens". Em essência, o invasor "sequestrou" o clique do usuário, daí o nome "Clickjacking".
- Um dos exemplos mais notórios de Clickjacking foi um ataque contra a página de configurações do plugin Adobe Flash . Ao carregar esta página em um iframe invisível, um invasor pode induzir um usuário a alterar as configurações de segurança do Flash, permitindo que qualquer animação em Flash utilize o microfone e a câmera do computador.
- https://owasp.org/www-community/attacks/Clickjacking

CLICKJACKING - EXEMPLOS

- https://pt.wikipedia.org/wiki/Clickjacking
- https://www.imperva.com/learn/application-security/clickjacking/
- https://portswigger.net/web-security/clickjacking
- https://hackerone.com/reports/405342
- https://medium.com/@raushanraj_65039/google-clickjacking-6a04132b918a
- https://www.youtube.com/watch?v=tJLWmr8ypZg
- https://www.paulosyibelo.com/2015/03/facebook-bug-bounty-clickjacking.html
- https://www.youtube.com/watch?v=Mmp0s1GBrNo
- https://www.youtube.com/watch?v=A7JVzglupxc
- https://www.youtube.com/watch?v=JuYULZdmd9U
- https://www.youtube.com/watch?v=Zm1IQAQOqJ0
- https://www.youtube.com/watch?v=keAtUgCbuq8

CROSS SITE PORT ATTACK

- O Ataque de porta entre sites (XSPA) é uma vulnerabilidade que permite que os invasores busquem o status das portas TCP (e obtenham faixas de serviço) pela Internet ou sistemas internos, abusando de um recurso em aplicativos Web que faz solicitações HTTP usando URLs fornecidos pelo invasor.
- https://www.briskinfosec.com/blogs/blogsdetail/Cross-Site-Port-Attack-XSPA
- https://ibreak.software/2012/11/cross-site-port-attacks-xspa-part-1/

CROSS SITE PORT ATTACK - EXEMPLOS

- https://www.youtube.com/watch?v=oVwHlESZTil
- https://www.youtube.com/watch?v=W0H3Ebmbvxl
- https://briskinfosec.blogspot.com/2018/03/cross-site-port-attack-xspa.html
- https://owasp.org/www-pdf-archive//2018-02-05-AhmadAshraff.pdf
- https://trouge.net/papers/SSR_raid2016.pdf
- https://media.blackhat.com/ad-12/Walikar/bh-ad-12-pokingserverswithFacebook-Walikar-WP.pdf

CACHE POISONING

- O impacto de uma resposta criada com códigos maliciosos pode ser aumentado se for armazenado em cache por um cache da Web usado por vários usuários ou mesmo pelo cache do navegador de um único usuário. Se uma resposta for armazenada em cache em um cache da Web compartilhado, como os comumente encontrados em servidores proxy, todos os usuários desse cache continuarão recebendo o conteúdo malicioso até que a entrada do cache seja removida. Da mesma forma, se a resposta for armazenada em cache no navegador de um usuário individual, ele continuará recebendo o conteúdo malicioso até que a entrada do cache seja removida, embora apenas o usuário da instância do navegador local seja afetado.
- Para realizar com sucesso esse ataque, um invasor:
  - Localiza o código de serviço vulnerável, o que lhes permite preencher o campo de cabeçalho HTTP com muitos cabeçalhos.
  - Força o servidor de cache a liberar seu conteúdo de cache real, que queremos que seja armazenado em cache pelos servidores.
  - Envia uma solicitação especialmente criada, que será armazenada no cache.
  - Envia a próxima solicitação. O conteúdo injetado anteriormente armazenado no cache será a resposta a essa solicitação.
- Esse ataque é bastante difícil de ser realizado em um ambiente real. A lista de condições é longa e difícil de ser realizada pelo invasor. No entanto, é mais fácil usar essa técnica do que a desfiguração entre usuários .
- Um ataque de envenenamento por cache é possível devido a divisão de resposta HTTP e falhas no aplicativo da web. É crucial, do ponto de vista do invasor, que o aplicativo permita preencher o campo de cabeçalho com mais de um cabeçalho usando caracteres CR (Retorno de carro) e LF (avanço de linha).

https://owasp.org/www-community/attacks/Cache_Poisoning

CACHE POISONING - EXEMPLOS

- https://portswigger.net/research/practical-web-cache-poisoning
- https://www.youtube.com/watch?v=oAiG-EVemUI
- https://www.youtube.com/watch?v=ZsrCoheszzo
- https://www.youtube.com/watch?v=w_nxsG-JXHA
- https://www.youtube.com/watch?v=cbp7M1Mj5ts
- https://www.youtube.com/watch?v=9FHBhTFcHyc
- https://www.iana.org/about/presentations/davies-viareggio-entropyvuln-081002.pdf
- https://portswigger.net/web-security/web-cache-poisoning
- https://hackerone.com/reports/409370
- https://hackerone.com/reports/492841

OUTRAS VULNERABILIDADES WEB

Arbitrary file access
Binary planting
Blind SQL Injection
Blind XPath Injection
Brute force attack
Buffer overflow attack
Cache Poisoning
Cash Overflow
Clickjacking
Command injection attacks
Comment Injection Attack
Content Security Policy
Content Spoofing
Credential stuffing
Cross Frame Scripting
Cross Site History Manipulation (XSHM)
Cross Site Tracing
Cross-Site Request Forgery (CSRF)
Cross Site Port Attack (XSPA)
Cross-Site Scripting (XSS)
Cross-User Defacement

Custom Special Character Injection
Denial of Service
Direct Dynamic Code Evaluation ('Eval Injection')
Execution After Redirect (EAR)
Exploitation of CORS
Forced browsing
Form action hijacking
Format string attack
Full Path Disclosure
Function Injection
Host Header injection
HTTP Response Splitting
HTTP verb tampering
HTML injection

LDAP injection
Log Injection
Man-in-the-browser attack
Man-in-the-middle attack
Mobile code: invoking untrusted mobile code
Mobile code: non-final public field
Mobile code: object hijack
One-Click Attack
Parameter Delimiter
Page takeover
Path Traversal
Reflected DOM Injection
Regular expression Denial of Service – ReDoS
Repudiation Attack
Resource Injection

OUTRAS VULNERABILIDADES WEB

Server-Side Includes (SSI) Injection
Session fixation
Session hijacking attack
Session Prediction
Setting Manipulation
Special Element Injection
SMTP injection
SQL Injection
SSI injection
Traffic flood
Web Parameter Tampering
XPATH Injection
XSRF or SSRF

CONCLUSÃO

Esses foram alguns dos conceitos envolvendo aplicações web e suas vulnerabilidades e obviamente que isso apenas é uma base de algo tão vasto;

Por isso, eu recomendo que você se aprofunde não apenas na prática, mas na teoria também, pois alguns ataques mesclam outros também, seja um XSS to SQL Injection ou um SQL Injection to RCE ou um Local File Inclusion to Command Injection e etc;

Muitos desses conteúdos apresentados você encontra em algumas certificações famosas como (CASE.JAVA e .NET, CEH, ECSA, OSWE, eWPT e eWPTX), sendo essas as mais conhecidas quando se fala em segurança de aplicação web;

CONCLUSÃO

E se você desejar aprender ataques web na prática e conhecer outros métodos, recomendo:

https://github.com/infoslack/awesome-web-hacking

https://libraries.io/github/infoslack/awesome-web-hacking

https://libraries.io/github/infoslack/awesome-web-hacking

https://www.elearnsecurity.com/certification/ewpt/

https://www.elearnsecurity.com/certification/ewptx/

https://www.offensive-security.com/awae-oswe/

https://acaditi.com.br/certificacoes-eccouncil/

https://www.linkedin.com/in/joas-antonio-dos-santos/ (Meus Artigos)

https://www.hackthebox.eu/

http://www.dvwa.co.uk/

https://www.vulnhub.com/

https://pentesterlab.com/

AGRADECIMENTO

Por fim, agradeço à todos que curtiram os meus e-books, isso me motiva à continuar desenvolvendo e compartilhando conteúdos para à comunidade e no meu próprio desenvolvimento pessoal com as dicas que recebo;

Para todos aqueles que gostou, esse local é aonde reservo meus agradecimento e pode ter certeza que isso significa muito para mim!

REFERENCE

https://www.hostinger.com.br/tutoriais/o-que-e-ssl-tls-https

https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status

https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods

https://pt.stackoverflow.com/questions/608/qual-a-diferen%C3%A7a-entre-c%C3%B3digo-client-side-e-server-side-em-desenvolvimento-web

https://pt.wikipedia.org/wiki/Servidor_de_aplica%C3%A7%C3%A3o

https://pt.wikipedia.org/wiki/Aplica%C3%A7%C3%A3o_web

https://sensedia.com/api/owasp-2017-top-10-riscos-seguranca-apis/

https://owasp.org/www-project-top-ten/

https://www.w3schools.in/ethical-hacking/information-gathering-techniques

https://pt.wikipedia.org/wiki/Base64

https://pt.wikipedia.org/wiki/Cookie_(inform%C3%A1tica)

https://en.wikipedia.org/wiki/Session_ID

https://pt.wikipedia.org/wiki/Nslookup

https://pt.wikipedia.org/wiki/WHOIS