Sécurité des systèmes d'information

Grenoble-INP Ensimag 3A 2021/2022

Travail réalisé par :

Hamza HASSOUNE
Mohamed SABI

1 Introduction

Bash est un logiciel libre publié sous licence publique générale GNU. Il est l'interprète par défaut sur de nombreux Unix libres, notamment sur les systèmes GNU/Linux.

Fondé sur le Bourne shell, *Bash* lui apporte de nombreuses améliorations, mais sur certaines versions de celui-ci, une fonction mal implémentée peut donner lieu à des failles plus ou moins graves, mais surtout exploitables afin de détourner le fonctionnement auquel il a été destiné.

C'est notamment le cas dans plusieurs Patch de la version *Bash 5.0*, où l'exploitation locale d'une faille du système permettait une élévation de privilège, donnant la possibilité à un utilisateur malintentionné de générer des actions indésirables.

Il s'agit de la faille : **CVE-2019-18276**, Cette dernière était initialement basée sur une vulnérabilité plus ancienne en 1999 (`suidperl`), 20 ans après en 2019, la même vulnérabilité dans *bash* a été découverte par Ian Pudney, ingénieur logiciel chez Google, Inc. qui travaille au sein de l'équipe Sécurité et confidentialité.

Dans le travail qui suit, il s'agira d'étudier cette faille en détail, de préciser le contexte de son exploitation, d'en déterminer les éventuelles répercussions, et de faire une démonstration de faisabilité de celle-ci sur un conteneur adapté. Certaines mesures seront également proposées afin de limiter l'impact d'une telle faille ou de l'éviter.

Les autorités de Gotham City n'auront donc plus de soucis à se faire !!

2 La faille CVE-2019-18276 :

- **Service compromis et type de compromission :**

Le service compromis est, comme évoqué ci-dessus, tout système d'exploitation utilisant l'une des versions compromises de *Bash*, selon le site de référence http://www.cvedetails.com, les versions touchées par cette faille sont celle de *Bash 5.0* (Mises à jour : *RC1, Beta1, Beta2 Patch1 jusqu'à Patch10*) ainsi que les versions antérieures, les mises à jours ont permis depuis de régler le problème.
Et en exploitant cette faille de manière frauduleuse, un attaquant malveillant pourrait effectuer des manipulations sur le système sans forcément en avoir les droits.
Il peut pour cela procéder en local sur le système.

- La compromission dont il s'agit dans ce cas là, est de type : exécution de code et élévation de privilège.

Une élévation des privilèges est un mécanisme permettant à un utilisateur d'obtenir des privilèges supérieurs à ceux dont il dispose, dans le but de lancer des processus sensibles ou d'effectuer des tâches qu'il n'a pas le droit de faire en temps normal.

- **Vulnérabilité et mécanisme :**

Cette vulnérabilité d'élévation de privilège présente un score dans le système d'évaluation CVSS *(Common Vulnerability Scoring System)* de : 7.2 , cela s'explique par le fait qu'elle puisse causer d'énormes soucis au niveau du système :
En effet, bien que l'utilisation de cette faille nécessite que l'attaquant aie un accès au système, très peu de connaissance et d'expertise sont nécessaires pour l'exploiter. Et plus important : la confidentialité des données n'est plus assurée, puisque l'exploitation de la faille peut engendrer une divulgation totale des informations et par conséquent la révélation de fichiers parfois protégés par une restriction des droits.

Son mécanisme de fonctionnement est expliqué dans ce qui suit :

Les systèmes d'exploitation de type Unix identifient un utilisateur par une valeur appelée identifiant utilisateur, souvent abrégée en UID, ce dernier, ainsi que l'identifiant de groupe (GID) et d'autres critères de contrôle d'accès, sont utilisés pour déterminer les ressources système auxquelles un utilisateur peut accéder.
(Le fichier de mot de passe `/etc/passwd` *relie les noms d'utilisateur aux UID)*

Avant de poursuivre, on se doit de définir quelques notions clés :

◇ Real UserID : il s'agit du propriétaire du processus.

◇ Effective UserID : Par défaut, c'est le même que le Real UserID, mais parfois il est modifié pour donner plus de privilèges à l'utilisateur.

◇ Saved UserID : parfois, lorsqu'un utilisateur privilégié (généralement root) souhaite effectuer des activités non privilégiées, cela peut être réalisé en passant temporairement à un compte non privilégié.

Quand un utilisateur se connecte sur un système *GNU/Linux*, il détient donc 2 UID : le réel (RUID) et l'effectif (EUID). Au moment de l'exécution d'une commande, un UID effectif lui est attribué. Quand les droits d'endossement ne sont pas positionnés, alors l'UID effectif est identique à l'UID réel. Si les droits d'endossement sont positionnés alors l'UID effectif est celui de la commande. Ce qui signifie que ce dernier est celui qui contrôle les droits d'accès à une commande.

Un problème a été découvert dans la fonction : `disable_priv_mode` du `shell.c` dans *GNU Bash*, qui, comme son nom l'indique, se charge de désactiver un mode privilégié.
Par défaut, si *Bash* est exécuté avec son UID effectif différent de son UID réel, il supprimera les privilèges en définissant son UID effectif sur son UID réel.
Cependant, il le fait de manière incorrecte :
Sur Linux et d'autres systèmes qui prennent en charge la fonctionnalité SUID (Saved UID), ce dernier n'est pas supprimé.
Un attaquant, avec l'exécution de commandes dans le shell peut utiliser la commande `enable -f` pour le chargement à l'exécution d'une nouvelle fonction intégrée, qui peut être un objet partagé qui appelle la fonction `setuid()` et récupère donc les privilèges.

-Remarque : Il faut cependant noter que, les binaires exécutés avec un EUID de 0 (EUID de root) ne sont pas affectés.

- **Machines cibles :**

Elle concerne bien évidemment toutes les machines utilisant une version corrompue de *Bash* (*Cf. liste ci-dessus*), puisqu'un script malveillant peut y être exécuté et permettre une élévation de privilège non désirée, ce qui peut avoir diverses répercussions négatives si les intentions de l'attaquant sont malveillantes.

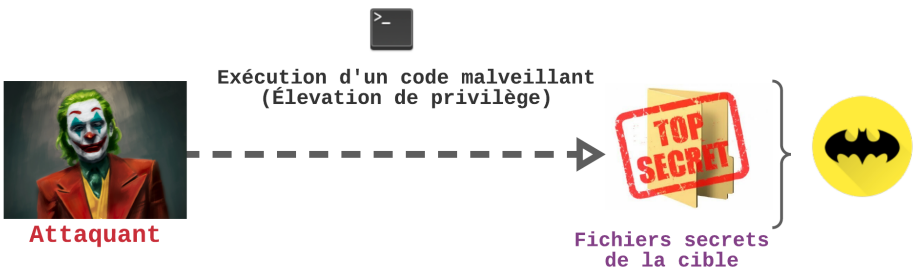

FIGURE 1 – Exploitation de la faille pour exécuter une tâche non permise
(Lecture d'un fichier protégé)

- **Architecture typique (Schéma) :**

Imaginons une entreprise dans laquelle un administrateur (root) se charge de créer des comptes utilisateurs pour chacun des employés pour qu'ils puissent s'authentifier et travailler sur le même poste (*Cf. Figure 2 ci-dessous*).
L'administrateur peut être amené à changer le propriétaire du `/bin/bash` dans certaines situations, supposons dans notre cas que le propriétaire de ce dernier est différent du root.
Un utilisateur malintentionné pourrait, moyennant une exploitation de la faille **CVE-2019-18276** afin d'effectuer une élévation de privilège, profiter de cette vulnérabilité du système afin d'accéder à des fichiers propres à un autre utilisateur, afin de les copier, les altérer ou les détruire. Il n'aura qu'à exécuter un fichier malveillant pour pouvoir "usurper" en quelque sorte l'identité d'un autre utilisateur.

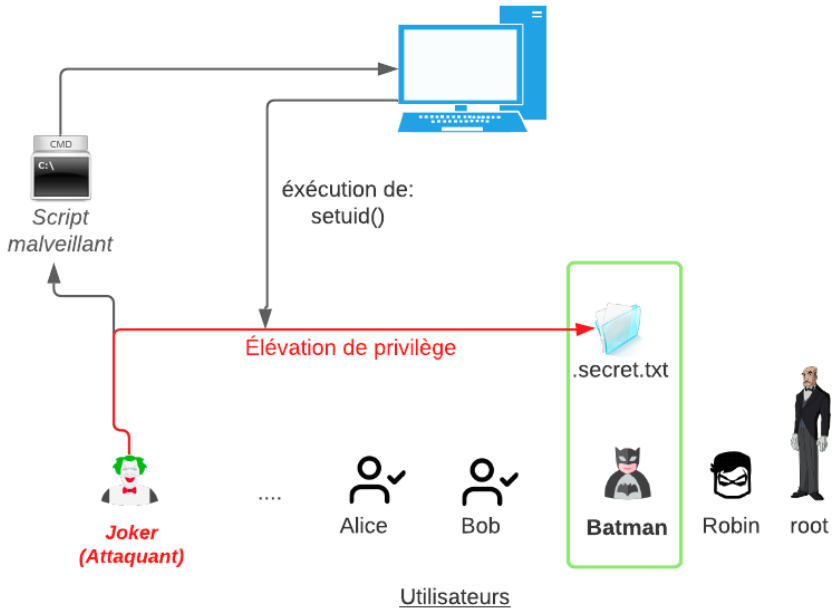

FIGURE 2 – Exemple d'architecture cible de l'attaque *CVE-2019-18276*

- **Limitation de l'impact/ Empêcher l'exploitation :**

Dans une logique de prévention, il semble judicieux et intéressant que l'administrateur du système fasse ce qui suit afin de limiter l'impact de l'exploitation de cette faille, et au mieux empêcher celle-ci :

-La mise à jour régulière des appareils avec les derniers correctifs peut aider à prévenir les attaques qui exploitent cette vulnérabilité, ceci est particulièrement applicable dans le cas du *Bash*, car cette vulnérabilité a bien été neutralisée dans les dernières versions.

-Empêcher l'exécution ou l'installation d'exécutables suspects, pour un utilisateur ne disposant pas des droits d'administrateur système.

-Proscrire l'utilisation de clés usb externes dans les machines de l'entreprise, pour éviter d'introduire et d'exécuter des scripts malveillants (comme celui utilisé dans la partie Expérimentation).

-Contrôler les contenus téléchargés, via des mails frauduleux ou des liens de téléchargement pour éviter l'éxécution de scripts malveillants, et sensibiliser l'ensemble des utilisateurs à ce type de faille.

-Crypter le contenu des fichiers sensibles de l'entreprise susceptibles d'être enregistrés sur le système, pour empêcher leur exploitation ou toute utilisation frauduleuse de ceux-ci dans le cas d'un vol ou d'une divulgation d'informations suite à l'exploitation de cette faille.

- **Bonnes pratiques :**

Voici quelques "bonnes pratiques" qu'il est conseillé d'adopter afin de limiter la menace que peut représenter cette vulnérabilité :

-Vérifier régulièrement la présence de fichiers inconnus ou suspects dans le système, ainsi que les droits de lecture, écriture et exécution de ceux-ci, et signaler toute présence de tels fichiers à l'administrateur système ou le DSSI.

-Ne pas utiliser de dispositif de stockage externe personnel (clés usb, disque dur etc.) au sein de la structure.

-Mise à jour régulière de tous les logiciels utilisés par la structure.

-Utilisation d'un pare-feu afin de surveiller et contrôler les flux de données, pour faciliter la détection de toute action suspecte sur le système (notamment le téléchargement de fichiers éxécutables).

- **Faille de développement :**

Oui, il s'agit bien d'une faille de développement qui correspond au code CWE (*Common Weakness Enumeration*) suivant :

CWE-273 : Vérification incorrecte des privilèges abandonnés,

Et qui est référencée dans la liste des vulnérabilités que l'on peut rencontrer dans les logiciels.

Pour être plus explicite : Le logiciel tente de supprimer les privilèges mais ne vérifie pas du tout ou alors vérifie de manière incorrecte si la suppression a bien été effectuée.
Une conséquence directe de cette faille de développement serait,en cas d'échec de la suppression, la poursuite de l'exécution du logiciel en question avec les privilèges élevés, ce qui peut fournir un accès supplémentaire à des utilisateurs non privilégiés.
Cela pourrait permettre à une personne malintentionnée d'exécuter des commandes ou des tâches dont il n'a pas les accès ou le privilège et qui peuvent représenter un danger direct sur le système d'exploitation.

Afin de limiter les effets de cette faille, les quelques points abordés dans ce qui suit, aurait dû être pris en considération lors du développement du `shell.c` :

Il fallait s'assurer qu'une compartimentation appropriée a été intégrée dans la conception du système, et que cette dernière permet et renforce la fonctionnalité de séparation des privilèges. Un architecte ou un concepteur devrait s'appuyer sur le principe du moindre privilège afin de décider du moment adéquat pour utiliser les privilèges et du moment pour abandonner les privilèges.

Le principe de moindre privilège (ou PoLP) est un principe qui stipule, selon l'ANSSI, qu'une tâche ne doit bénéficier que de privilèges strictement nécessaires à l'exécution du code menant à bien ses fonctionnalités. Il est largement considéré comme une bonne pratique de cybersécurité.

Au niveau de la mise en oeuvre du logiciel, il est nécessaire de vérifier les résultats de toutes les fonctions qui renvoient une valeur, et de s'assurer que c'est bel et bien la valeur attendue.
Dans notre cas, il s'agit de vérifier que l'euid renvoie la bonne valeur...

Enfin, même si une telle technique est en général suffisante et d'une efficacité élevée, un attention particulière devrait être portée sur les conditions de concurrence dans un environnement concurrent *(Cf. la faille CWE-362 pour plus d'informations).*

- **Extrait PSSI :**

Reprenons l'exemple de l'entreprise décrite ci-dessus, en supposant qu'elle utilise une des versions vulnérables du logiciel *bash* sur ses postes, la faille CVE-2019-18276 est tout à fait exploitable. En suivant les recommandations du Guide d'élaboration de politiques de sécurité des systèmes d'information, disponible sur internet et proposé par l'ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information), et dans le but de mettre à jour la Politique de Sécurité du Système d'Information (PSSI), un extrait de cette dernière est proposé dans ce qui suit :

Une vulnérabilité ayant été détecté sur notre système, avec une probabilité d'exploitation et un impact moyens sur nos machines, sa criticité a été jugée : Moyenne.

Dans l'attente d'effectuer une mise à jour de celui-ci avec les correctifs nécessaires, et dans une logique de prévention et de vérification, vous êtes priés de respecter les consignes qui suivent afin de limiter la surface d'exploitation de celle-ci :

-Les administrateurs et les utilisateurs du système sont tenus d'utiliser des mots de passe robustes et de ne jamais les communiquer sous aucun prétexte.

-Il est fortement recommandé de changer son mot de passe tout les 3 mois.

-Ne téléchargez des documents ou fichiers sur votre machine, qu'en cas de nécessité, ou d'impossibilité de lecture et/ou modification en ligne.

-Vous êtes prié de consulter la liste noire des extensions de fichiers qui a été établie, avant d'entamer une quelconque action (téléchargement, copie, exécution etc.) sur le système.

-Signalez à l'administrateur la présence de tout fichier suspect, inconnu, et/ou anormal sur nos systèmes.

-Analyser régulièrement tout le système afin de détecter d'éventuels fichiers suspects, voire des processus inconnus.

-Effectuer des sauvegardes de tout fichier ou document jugé important pour l'entreprise, afin de pouvoir le restaurer en cas de suppression/compromission de celui-ci.

Merci de bien vouloir respecter ces quelques règles afin de limiter l'impact de la faille, et de signaler toute anomalie remarquée à l'administrateur du système ou à la DSSI (Direction de Sécurité des Systèmes d'Information) le plus tôt possible.

---

3 Expérimentation :

Dans cette partie, un PoC (*Proof of Concept*) sera détaillé afin de montrer la faisabilité de tout ce qui a été décrit précédemment.

Pour ce faire, nous utiliserons une image Docker de la distribution Debian :10, on pourra vérifier par la suite que la version du *bash* utilisée est : *5.0.3(1)-release*, et ce en utilisant la commande :
`$ echo $BASH_VERSION`
une fois le conteneur lancé, cette version est bien entendu vulnérable.

Sur le Dockerfile utilisé, afin de préparer l'environnement de travail, il a été question de créer deux utilisateurs : d'une part, **batman** (notre victime) qui disposera sur son propre répertoire d'un fichier caché nommé `.secret.txt` qu'il est le seul à pouvoir lire ou modifier, et d'autre part, **joker** (l'attaquant) qui tentera de percer le secret de batman, pour cela il a préparé sur son propre répertoire `/home/joker` un fichier malveillant nommé `exploit.sh` contenant une succession de commande à exécuter afin de se donner le droit le lire le fichier secret.

- Dans ce qui suit, les commandes à exécuter afin d'exploiter la vulnérabilité :

Dans un premier temps, on commence par lancer le makefile sur notre Terminal ouvert dans le répertoire *gotham* :

```
$ make
```

(cela permettra de construire l'image et de lancer notre conteneur)

Au démarrage, c'est la session de l'utilisateur joker qui est ouverte, on pourra essayer d'aller ouvrir le fichier `.secret.txt` situé sur le répertoire `/home/batman`, en utilisant :

`$ cat /home/batman/.secret.txt` ,mais l'accès nous sera refusé.

À l'aide des commandes qui suivent, on va donc se placer sur le répertoire de l'utilisateur joker, ce dernier exécutera son fichier malveillant, ce fichier contient des commandes qui permettent de récupérer l'UID de batman, puis de lancer un code C qui obligera implicitement le bash d'éxécuter la fonction `setuid()` :

```
$ cd /home/joker
$ ./exploit.sh
```

(Il sera nécessaire de saisir l'UID qui s'affiche afin de poursuivre l'exécution)

Enfin, la dernière étape sera de lancer la commande suivante :

```
$ enable -f ./libpwn.so asd
```

Bien qu'un message d'erreur apparaisse à l'issue de cette étape, l'exploit est un succès car la fonction setuid() a déjà été lancé par le *bash*, ce qui a permis à joker de modifier son EUID et de le remplacer par l'UID de batman, d'où l'élevation de privilège que l'on pourra vérifier dans ce qui suit :

```
$ id
```

(permet de vérifier que l'EUID a bien été modifié)

```
$ cat /home/batman/.secret.txt
```

(La lecture du fichier caché et protégé par restriction des droits devient possible !)

Ainsi, joker découvrit l'identité de celui qui se cache derrière le masque de Batman, qui n'est autre que son demi-frère, le richissime Bruce Wayne.

On imagine bien les répercussions que pourrait avoir une telle élévation de privilège sur le système vulnérable et ses utilisateurs...

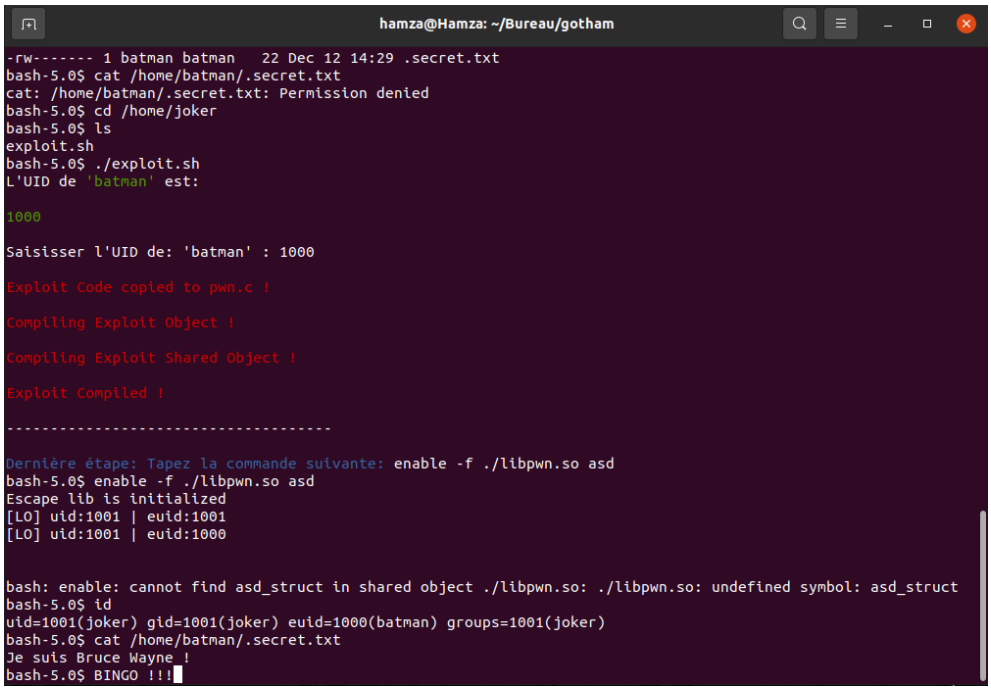

FIGURE 3 – Commandes de l'exploit et résultats sur la sortie standard

⇒ *Une démonstration vidéo qui illustre toutes les étapes décrites ci-dessus est jointe à ce rapport, les rendus fournis (Makefile, Dockerfile, exploit.sh) ont bien entendu été pensés et construits dans le but de faciliter la reproduction de ces étapes.*

4 Conclusion :

Compte tenu de l'analyse faite sur ce rapport de la faille CVE-2019-18276, il s'avère qu'elle constitue une vulnérabilité compromettante, surtout que son exploitation n'est pas très compliquée. D'ailleurs, elle a été rectifié dans les versions postérieures de *bash*.
Par ailleurs, il est intéressant de savoir qu'une faille très similaire à celle-ci (référencée sous le code : CVE-1999-0138 pour les plus curieux) a déjà été rencontrée des années auparavant dans le système d'exploitation *Linux Kernel 1.2.0*, force est de constater que certaines failles sont communes et risquent de revenir souvent si les développeurs et concepteurs logiciel baisse leur vigilance et ne retiennent pas les leçons des erreurs commises par leurs prédécesseurs.

Finalement, au vu des explications et de la démonstrations ci-dessus, et dans le cas d'utilisation du logiciel Bash, il est fortement conseillé à Bruce Wayne de garder sa vigilance, de vérifier régulièrement si la version de son système est bien à jour afin d'éviter les mauvaise surprises, et pourquoi pas d'investir une partie de sa fortune en cybersécurité : en effet, le soutien d'un Hacker White Hat peut s'avérer bien plus utile que celui de Robin dans certaines situations, surtout dans une ère de prédominance numérique.

5 Glossaire :

Shell : Il s'agit d'un interpréteur de commandes destiné aux systèmes d'exploitation Unix (ou de type Unix) qui permet d'accéder aux fonctionnalités internes du système d'exploitation.

Bash : Bash *(acronyme de Bourne-Again shell)* est un logiciel libre, interpréteur en ligne de commande de type script, il s'agit du shell Unix du projet GNU.

CVE : Common Vulnerabilities and Exposures, il s'agit d'un dictionnaire contenant des informations publiques relatives aux vulnérabilités de sécurité.

CVSS : Common Vulnerability Scoring System, c'est un système d'évaluation standardisé de la criticité des vulnérabilités selon des critères objectifs et mesurables.

CWE : Common Weakness Enumeration, il représente une liste des vulnérabilités que l'on peut rencontrer dans les logiciels.

UID : User ID, il s'agit d'un identifiant utilisé par les systèmes d'exploitation de type Unix pour déterminer les ressources système auxquelles un utilisateur peut accéder.

EUID : Effective User ID, il est propre à un processus et est utilisé pour la plupart des contrôles d'accès ; il est également utilisé comme propriétaire pour les fichiers créés par ce processus.

SUID : Saved User ID, il est utilisé lorsqu'un programme exécuté avec des privilèges élevés doit effectuer temporairement un travail non privilégié.

Criticité : Évalue la gravité et les conséquences potentielles d'une faille.
(Criticité = Probabilité d'exploitation × Impact potentiel)

ANSSI : Agence nationale de la sécurité des systèmes d'information, il s'agit de l'autorité nationale en matière de sécurité et de défense des systèmes d'information.

DSSI : Direction de la Sécurité des Systèmes d'Information.

PSSI : Politique de Sécurité du Système d'Information.

PoC : Proof of Concept, il s'agit d'une démonstration de faisabilité, ou une réalisation ayant pour vocation de montrer la faisabilité d'un procédé.

Docker : Docker est un logiciel libre permettant de lancer des applications dans des conteneurs logiciels.

6 Bibliographie :

https ://cve.mitre.org/cgi-bin/cvename.cgi ?name=CVE-2019-18276

https ://www.cvedetails.com/cve/CVE-2019-18276/

https ://cwe.mitre.org/

http ://cwe.mitre.org/data/definitions/273.html

https ://nvd.nist.gov/vuln/detail/CVE-2019-18276

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

https ://www.ssi.gouv.fr/guide/guide-des-bonnes-pratiques-de-linformatique/

https ://www.ssi.gouv.fr/guide/pssi-guide-delaboration-de-politiques-de-securite-des-systemes-dinformation/

https ://www.exploit-db.com/exploits/47726

https ://www.youtube.com/watch ?v=Dbwvzbb38W0

https ://vuldb.com/fr/ ?id.13739