Cours de Antoine Allard

mercredi 12 juin 2019 14:06

Injection par buffer overflow et injection SQL

Qu'est-ce que une vulnérabilité

Toujours des bugs dans un code

Ce qui est exploitable c'est une série de bugs (exploit)
Si on patch un des bugs de la série, l'attaque ne peut plus avoir lieu

C.V.E: Common vulnerabilities & exposure. C'est un dictionnaire fait par les gars de MITRE et supporté par les USA

Pratique très courante: jamais release le vendredi car derrière les gens parte en week-end

Vulnérabilité zero-day: Bugs malveillants qui ont jamais été trouvés

55% des bugs viennent du C

Example
IE5 url overflow: Ils ont décidés de codés les caractères sur 16 bits et ça peut faire des overflow de mort

Buffer: tableau qui s'attend à recevoir des données
Buffer overflow: quand on tape là où on est pas censé tapé

Eviter un buffer overflow: pas si simple, on laisse souvent les devs avoir du undefined behaviour

Stack canaries: ils contrent les méchant buffer corruption exploit.
-> Principe: on place un canari dans la stack et après une série d'instruction on vérifie que le canari n'as pas changé

str(n)cpy, str(n)cat: ces fonctions sont pas terribles; elle permettent de faire un buffer overflow
-> on est pas toujours assuré que les chaînes finissent avec un "\0"

Avec strncat et strncpy ca rempli de "\0" jusqu'à la fin

strlcpy, strlcat: conseillé à la place
Raisons:

- Bonne troncature
- Ça finira toujours par NULL
- La taille est toujours en bytes
- 0 fill

Sprintf est aussi déconseillé: snprintf conseillé à la place

Le pire cas c'est quand on laisse l'utilisateur écrire directement dans le buffer sans aucune vérifications (taille…)

Drepper: Ancien RedHat (société d'audit de sécurité)

"If you can fix everything then don't fix anything"

Sturgeon: 90% de tout ce qui existe c'est de la merde

Le prof a tendance a beaucoup apprécié l'utilisation de emalloc au lieu de malloc

Qu'est ce qui se passe si on donne une taille négatif à un malloc ?
-> Ça prend beaucoup de mémoire, malloc prend un size_t il va donc convertir le truc en positif
-> Undefined behaviour
-> Possibilité de buffer overflow

emalloc: gentil wrapper autour de malloc

"Il fut un temps où toute les librairies GFX était vulnérables"

Injection SQL

Bonne exemple: WebSense, pro en sécurité à l'époque, ils avaient comme clients iTunes

Attaque par injection en 2011: vulnérabilités chez tout leur clients

SELECT * FROM student WHERE login='x'

Si on met x = hugo';DROP TABLE student; --

Bah la table student disparait

Instruction DO de PHP: il prenait les variables directement sans vérifications, et on pouvait en faire ce que l'on voulait

Solution:
- Rajouter des or à la fin comme fix potentiel
- Prepared statement
- Regex matching

Essaye de matcher x pour voir si il comporte des single quotes, ça c'est du negative matching
-> Mauvaise habitude

Positive matching: un truc spécifique, bien encadré et précis
Essayer de minimiser les inputs utilisateurs qui mènent à la BDD

Autre form d'exploit:
Certains sites voient "Fuizziy@gmail.com" identique à "F..ui…z.z.y@gmail.com"
Avec ça on peut avoir un compte Netflix gratuit à vie par example

Format String Attack

Autre forme d'injection

Lorsqu'on écrit printF(msg) avec msg qui est une variable
Si msg contient {'%n', '%p', '%x'}, on peut accéder à des choses par la suite

char *c = …;

printf("coucou %nouais", c);
c -> "coucou ouais7";

Si on place aux bons endroits des '%n', on peut changer totalement les variables

System et popen en C

C'est des wrappers sur quelque chose qui va exécuter directement du script
-> C'est dangereux car ça peut mener à des Hidden Shells

popen(x)
sh -x x

Si jamais on veut s'en protéger il faudrait tout quote

Pour éviter que le mec passe des arguments à un program, on peut mettre des "--"
Permet d'arrêter le parsing des options

Script kiddies

Ces des mecs sans connaissances qui arrive à s'introduire dans le system mais qui ont aucune connaissance
Considérer comme plus dangereux que des gamins normaux

Anecdote:
- Un gamin qui est arrivé à s'introduire dans le serveur d'Antoine, et qui a tapé tout et n'importe quoi
- Antoine qui donne les perms admin de son serveur mc à ses admins pour qu'ils arrêtent la pluie. Le mec sait pas ce qu'il fait et tape "/stop rain". "/stop" étant la commande pour arrêter le serveur, le serveur s'éteint

mktemp: crée un fichier temporaire

Utiliser mkstemp à la place

stat, chmod et chown (c'est des commandes unix et C)

Il vaut mieux utiliser: fstat, fchmod, et fchown

Différence à fstat on donne un filedescriptor, moins de vulnérabilités (stat prend directement une chaîne)

En audit ou review, il faut faire attention à:
- Memory handling
- Le flow d'exécution (control flow en anglais). Data flow aussi (askip faut pas faire attention)
- Faire attention au hidden state (trucs genre errno, local blocking status pour les file descriptor, les bloquing status)
- Le error hadling (liés un peu avec le hidden state) = Bonne gestion des erreurs !
- User roles (setuid, setguid)
- Process handling