EXPLOITATION DE BINAIRES, REVERSE ENGINEERING

Gauvain Roussel-Tarbouriech (gauvain@govanify.com)

J'utiliserais les noms anglais, allez sur bitoduc.fr si vous êtes gênés!

J'utiliserais les noms anglais, allez sur bitoduc.fr si vous êtes gênés!

N'hésitez pas a poser des questions et à me dire si je vais trop rapidement sur un sujet!

J'utiliserais les noms anglais, allez sur bitoduc.fr si vous êtes gênés!

N'hésitez pas a poser des questions et à me dire si je vais trop rapidement sur un sujet!

Des ressources, les slides et les fichiers étudiés en cours sont disponibles sur code.govanify.com/govanify/esgi-re

IMPORTANCE DU REVERSE

La majorité de nos équipements sont "closed source"

IMPORTANCE DU REVERSE

La majorité de nos équipements sont "closed source"

Windows

IMPORTANCE DU REVERSE

La majorité de nos équipements sont "closed source"

Windows

Android (en partie)

IMPORTANCE DU REVERSE

La majorité de nos équipements sont "closed source"

Windows

Android (en partie)

MacOS/iOS

IMPORTANCE DU REVERSE

La majorité de nos équipements sont "closed source"

Windows

Android (en partie)

MacOS/iOS

Les consoles de jeu

IMPORTANCE DU REVERSE

La majorité de nos équipements sont "closed source"

Windows

Android (en partie)

MacOS/iOS

Les consoles de jeu

Même votre processeur!

Les vendeurs ne veulent pas partager leur code pour **protéger leur propriété intellectuelle.**

Les vendeurs ne veulent pas partager leur code pour **protéger leur propriété intellectuelle.**

Ca ne les protège pas vraiment mais ça nous embête

Les vendeurs ne veulent pas partager leur code pour **protéger leur propriété intellectuelle.**

Ca ne les protège pas vraiment mais ça nous embête

Du coup faire de la recherche sur leurs produits, en sécu ou autre, nécessite souvent du reverse.

LA SÉCURITÉ PAR L'OBSCURITÉ

LA SÉCURITÉ PAR L'OBSCURITÉ

Théorème: si on ne sait pas comment ça fonctionne on ne peux pas trouver de problèmes de sécurité!

LA SÉCURITÉ PAR L'OBSCURITÉ

Théorème: si on ne sait pas comment ça fonctionne on ne peux pas trouver de problèmes de sécurité!

Contraposée: Les CVEs et Internet.

IMPORTANCE DE L'ATTAQUE SUR BINAIRE

IMPORTANCE DE L'ATTAQUE SUR BINAIRE

Les attaquants travaillent très(trop) souvent sur des binaires selon les équipements

IMPORTANCE DE L'ATTAQUE SUR BINAIRE

Les attaquants travaillent très(trop) souvent sur des binaires selon les équipements

Il est bon de connaitre les méthodes des attaquants aussi bien dans l'attaque que la défense

Examples d'attaques connues

Examples d'attaques connues

- Les consoles de jeux vidéo

Examples d'attaques connues

- Les consoles de jeux vidéo
- EternalBlue(WannaCry), NSA

Examples d'attaques connues

- Les consoles de jeux vidéo
- EternalBlue(WannaCry), NSA
- Exploits iOS utilisés contre les Uighurs, MSS

DÉBOUCHÉES

DÉBOUCHÉES

Un marché de 151Mds de dollars d'ici 2023

DÉBOUCHÉES

Un marché de 151Mds de dollars d'ici 2023

En pénurie de personnes hautement qualifiées

DÉBOUCHÉES

Un marché de 151Mds de dollars d'ici 2023

En pénurie de personnes hautement qualifiées

Bon ok on vous a déjà raconté ça, mais je suis dispo si il y'a des questions sur le marché de la RE!

PLAN DU COURS

PLAN DU COURS

Une introduction pratique a la sécurité offensive et au reverse engineering

PLAN DU COURS

Une introduction pratique a la sécurité offensive et au reverse engineering

- Rappels d'assembleur

PLAN DU COURS

Une introduction pratique a la sécurité offensive et au reverse engineering

- Rappels d'assembleur
- Introduction aux outils et au reverse engineering

PLAN DU COURS

Une introduction pratique a la sécurité offensive et au reverse engineering

- Rappels d'assembleur
- Introduction aux outils et au reverse engineering
- Corruption de mémoire / Surface d'attaque

PLAN DU COURS

Une introduction pratique a la sécurité offensive et au reverse engineering

- Rappels d'assembleur
- Introduction aux outils et au reverse engineering
- Corruption de mémoire / Surface d'attaque
- Stack et Heap overflow avec mitigations

PLAN DU COURS

Une introduction pratique a la sécurité offensive et au reverse engineering

- Rappels d'assembleur
- Introduction aux outils et au reverse engineering
- Corruption de mémoire / Surface d'attaque
- Stack et Heap overflow avec mitigations
- ROP avec ret2libc

PLAN DU COURS

Une introduction pratique a la sécurité offensive et au reverse engineering

- Rappels d'assembleur
- Introduction aux outils et au reverse engineering
- Corruption de mémoire / Surface d'attaque
- Stack et Heap overflow avec mitigations
- ROP avec ret2libc
- Format String vulnerabilities

PLAN DU COURS

Une introduction pratique a la sécurité offensive et au reverse engineering

- Rappels d'assembleur
- Introduction aux outils et au reverse engineering
- Corruption de mémoire / Surface d'attaque
- Stack et Heap overflow avec mitigations
- ROP avec ret2libc
- Format String vulnerabilities
- Race conditions

PLAN DU COURS

Une introduction pratique a la sécurité offensive et au reverse engineering

- Rappels d'assembleur
- Introduction aux outils et au reverse engineering
- Corruption de mémoire / Surface d'attaque
- Stack et Heap overflow avec mitigations
- ROP avec ret2libc
- Format String vulnerabilities
- Race conditions
- Injections de commandes **et + si possible!**

RAPPELS D'ASM

RAPPELS D'ASM

2 régions mémoire: le stack et la heap

RAPPELS D'ASM

2 régions mémoire: le stack et la heap

Le stack est utilisé pour stocker des variables, en appellant des fonctions etc

RAPPELS D'ASM

2 régions mémoire: le stack et la heap

Le stack est utilisé pour stocker des variables, en appellant des fonctions etc

La heap est allouée dynamiquement pour stocker des données variables

Example d'allocation sur la stack en C

```c
// 8 chars + un \00 a la fin
char array[9]="COVID-19";
```

Example d'allocation sur la heap en C

```
char* array = malloc(sizeof(char)*9);
strcpy(array,"COVID-19");
```

Les callings conventions(conventions d'appel)

Les callings conventions(conventions d'appel)

Elles diffèrent selon le compilateur et la plateforme

Les callings conventions(conventions d'appel)

Elles diffèrent selon le compilateur et la plateforme

Nous étudierons ici la plus commune pour le C et C++, __cdecl

Fonction exemple

```
_cdecl int exemple(int a, int b)
{
        return a + b;
}
```

Son équivalent en assembleur

```
push 3
push 2
call exemple
add esp, 8
```

```
exemple:
push ebp
mov ebp, esp
mov eax, [ebp + 8]
mov edx, [ebp + 12]
add eax, edx
pop ebp
ret
```

C'est pas un peu trop théorique?

C'est pas un peu trop théorique?

DEMO TIME!

Allez sur godbolt.org et testez par vous même!

Essayez d'appeller des fonctions, définir des struct, etc

INTRODUCTION AUX OUTILS DE RE

Cette partie sera une démonstration en pratique, n'hésitez pas a poser des questions sur mes actions!