SYNACKTIV

Anno 1404

Bière sécu Rennes

25/11/2025

OpenMIC

Introduction

Whoami

- Thomas Dubier **@Tomtombinary**
  - Security Researcher at Synacktiv
  - Reverse engineering team
- Synacktiv
  - Offensive security company
  - ~200 Ninja

Introduction

Sommaire

- Introduction
- Rétro-ingénierie
- Formats des assets
- Format RDA
- Format GR2
- Heap Windows
- Exploitation
- Démonstration
- Bonus
- Anno 2070

Introduction

Anno 1404

SYNACKTIV

- Type : Jeu de stratégie (RTS)
- Développeur : Related Designs
- Editeur : Ubisoft
- Date de sortie : 2010
- Moteur de jeu : Propriétaire
- Version : GOG v2.01.5010

Rétro-ingénierie

Mécanisme de sauvegarde

SYNACKTIV

Rétro-ingénierie

SYNACKTIV

Protocole réseau

- Propriétaire basé sur UDP
- Plusieurs types de message d'après les logs : JOIN,UPDATE, DELETE, **RMC_CALL** …
- RMC : **R**emote **M**ethod **C**all ?

```
methodName = ClassToMethodName(&v10, methodID);
targetName = TargetName(&v10);
v6 = TargetName(&v11);
WString::Format(
  a5,
  (wchar_t *)L"RMC_CALL message RMC_ID: %d, Flags: %d, Source: %x (%s), TargetObject: %x (%s), Method: %s",
  (unsigned __int16)input,
  Flags,
  source,
  v6,
  targetObject,
  targetName,
  methodName);
```

Rétro-ingénierie

Surface d'attaque RMC

SYNACKTIV

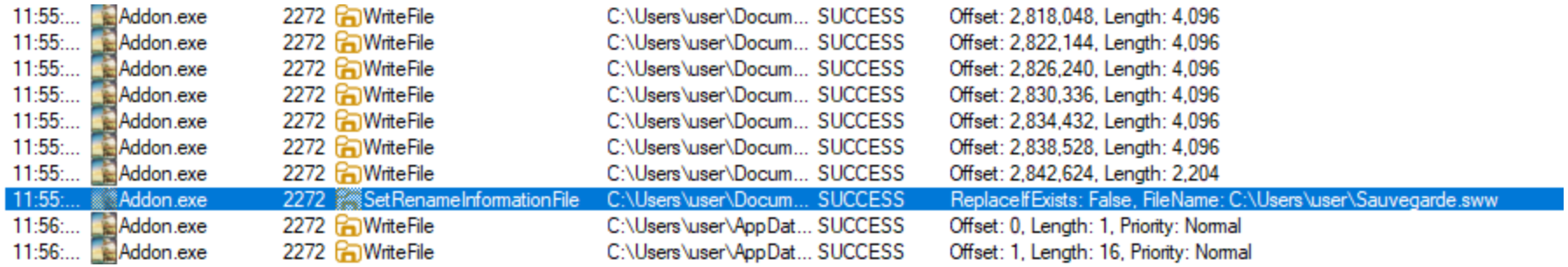
- Station
  - SignalAsFaulty
- Session
  - RetrieveURLs
  - SynchronizeTermination
- IDGenerator
  - RequestIDRangeFromMaster
- PromotionReferee
  - ConfirmElection
  - DeclinePromotion
  - ElectNewMaster
- SessionClock
  - AdjustTime
  - SyncRequest
  - SyncResponse
- Player
  - ForceKickPlayer
  - Kick
  - **OnCancelSendFile**
  - **OnReceivedFileData**
  - **OnSendFileData**
  - **OnSendFileInit**
- Chat
  - onNewChatLine
- GameSettings
  - ExecuteOnHost 🙈
- SyncProtocol
  - ClientToServerPing
  - ClientToServerSync
  - ConfirmHost
  - IdentifyHost
  - LeftGame
  - RequestMsgResend
  - ServerToClientPing
  - ServerToClientSync

Rétro-ingénierie

Path Traversal

SYNACKTIV

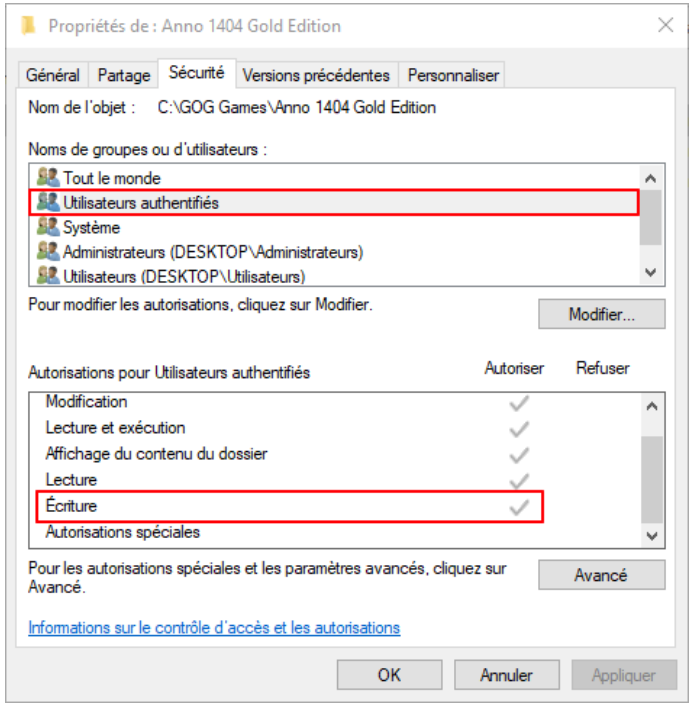
- **OnSendFileInit** récupère le nom du fichier de sauvegarde depuis le paquet
- Lors du transfert un fichier temporaire est créé dans `MPShare`
- A la fin du transfert ce fichier est renommé
- Pas de sanitization sur le nom de fichier
- Path traversal : `../../../Sauvegarde.sww`



Rétro-ingénierie

Perspective d'exploitation

- DLL Hijacking : Pas de chemin évident pour relancer le programme
- 💡 Remplacer les ressources du jeu à chaud



ACL sur le dossier d'installation

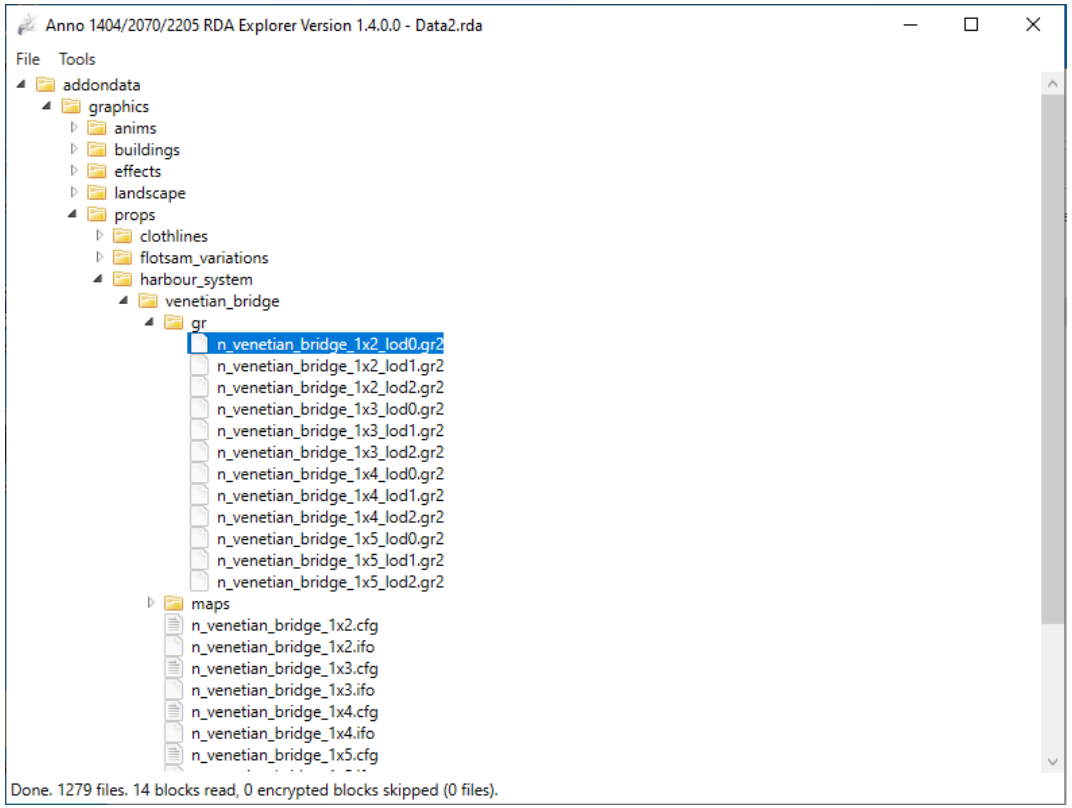
Formats des assets

Format RDA

SYNACKTIV

RDAExplorerGUI

- Format d'archive propriétaire
- Partiellement documenté
- Assets
  - Modèle 3D
  - XML
  - Son
  - Config
  - Shaders
  - ...

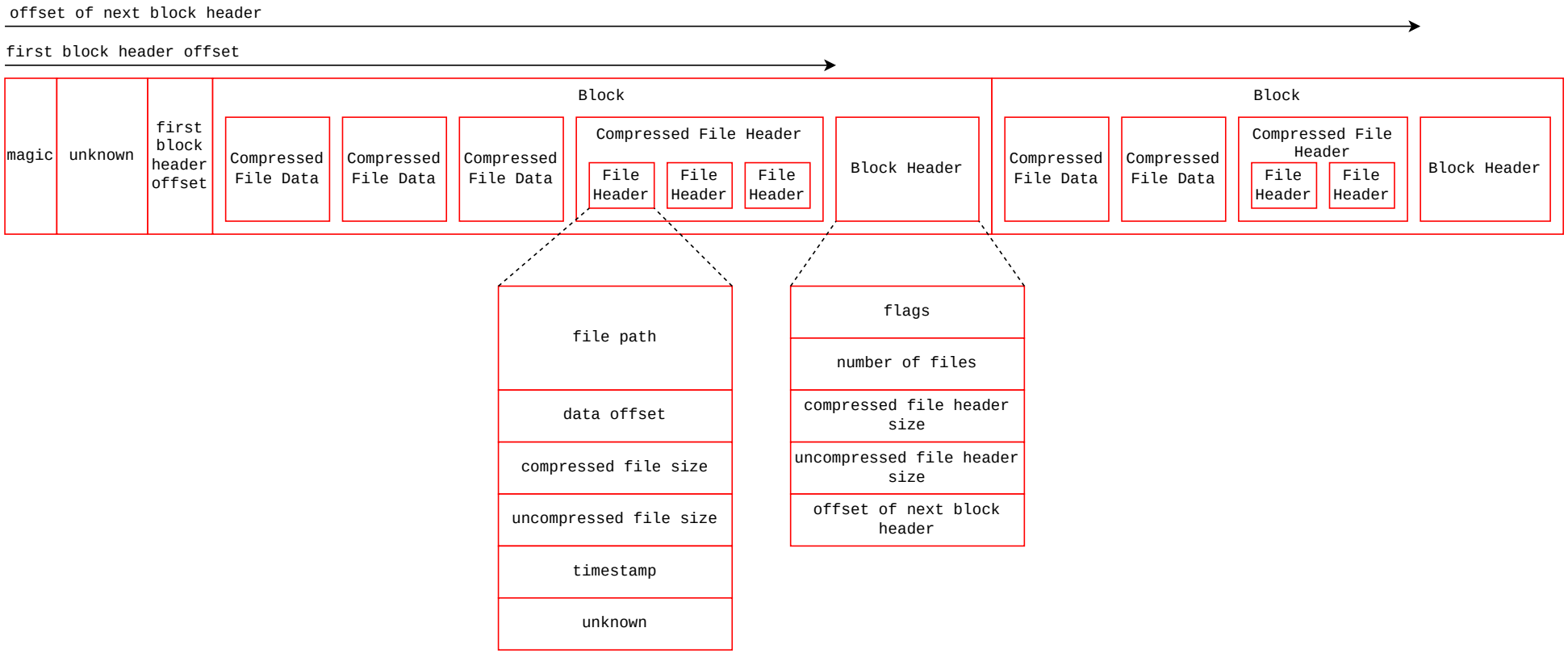


- https://github.com/lysanntranvouez/RDAExplorer/wiki/RDA-File-Format

Format RDA

Structure

- Compression Zlib
- Chiffrement propriétaire



Format RDA

Military Grade Encryption

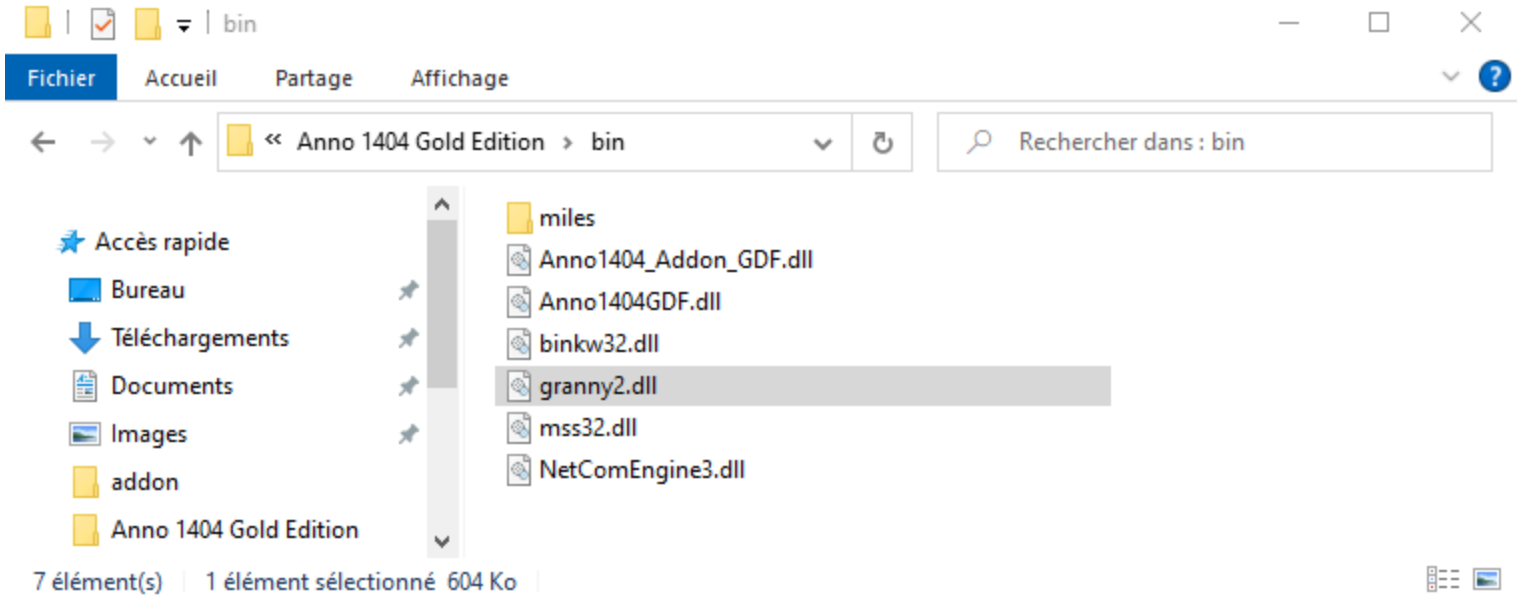
SYNACKTIV

```
1  char __cdecl xor_decrypt(wchar_t *buf, unsigned int size)
2  {
3    signed int index; // esi
4
5    srand(0xA2C2Au);
6    index = 0;
7    if ( size >> 1 )
8    {
9      do
10       buf[index++] ^= rand();
11     while ( index < (int)(size >> 1) );
12   }
13   return 1;
14 }
```

Format GR2

Implémentation

SYNACKTIV



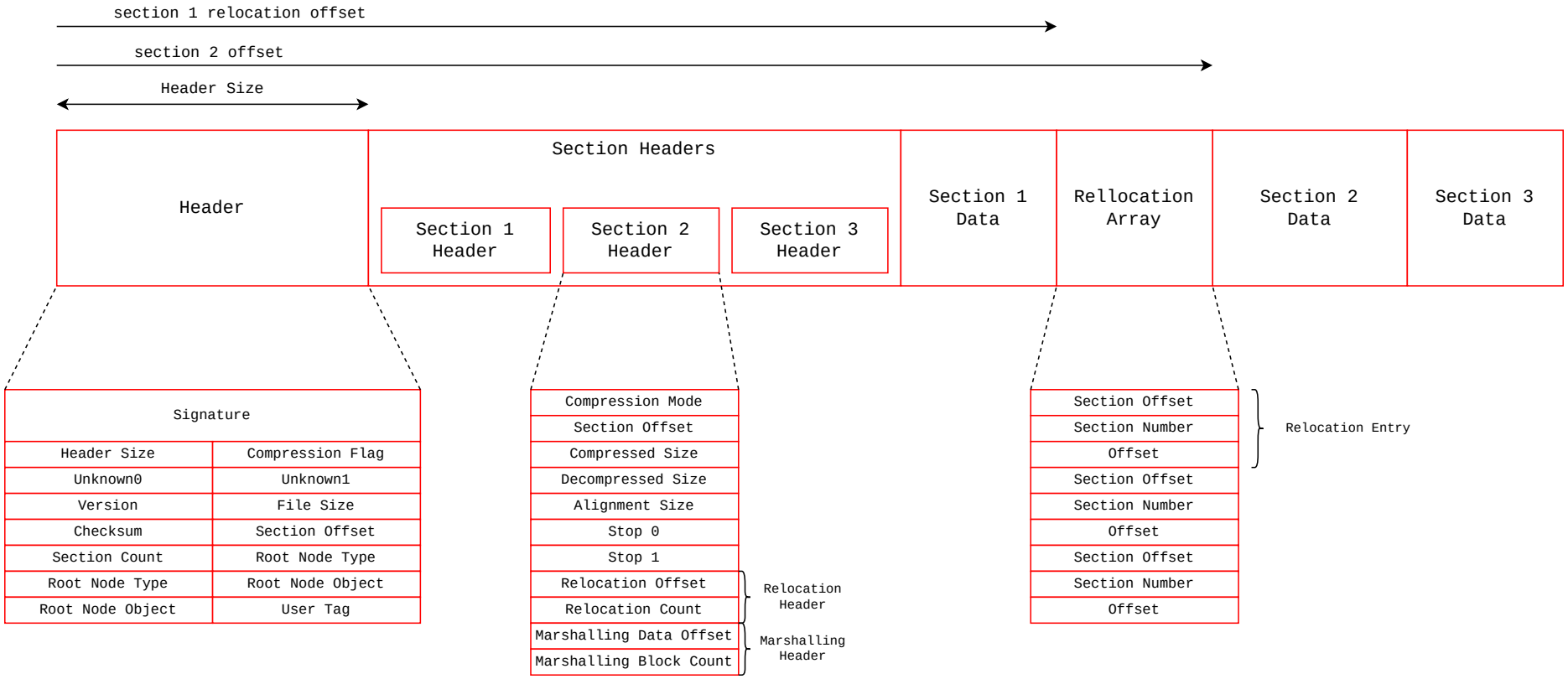
Format GR2

Implémentation

Format GR2

Structure

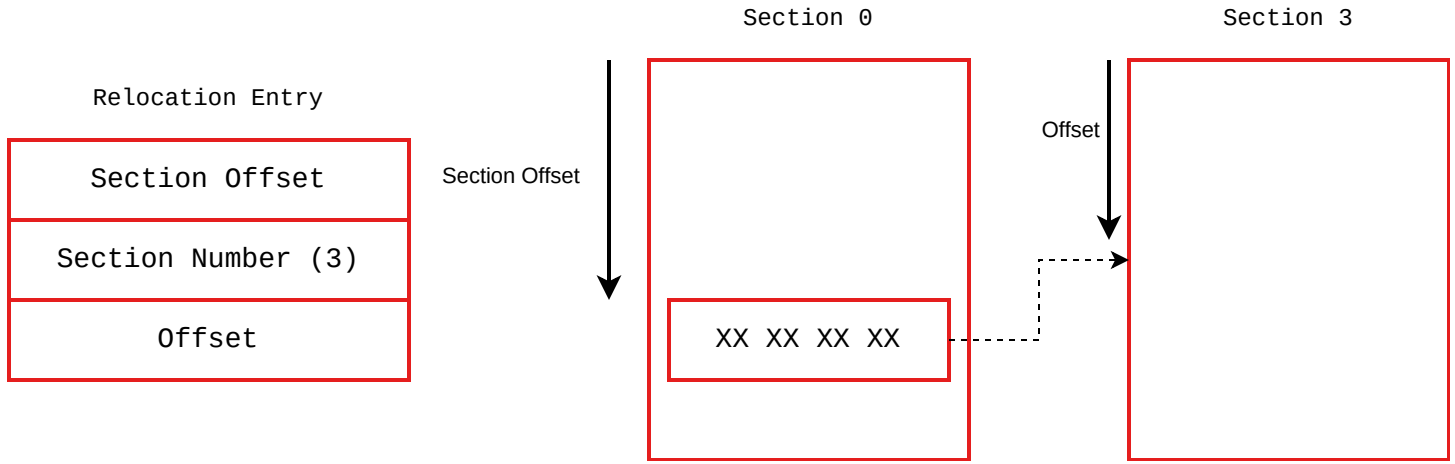
- Conteneur pour les Mesh, Textures, Bones, Materials
- Partiellement documenté : https://github.com/rdw-archive/RagnarokFileFormats/blob/master/GR2.MD



Format GR2

Relocations

- Les sections contiennent des pointeurs vers des éléments des autres sections
- Chaque section en mémoire débute à une certaine adresse
- Les adresses des sections varient selon l'environement
- Les pointeurs doivent être mis à jour
- Ces informations sont contenues dans une table de ***relocations***



Format GR2

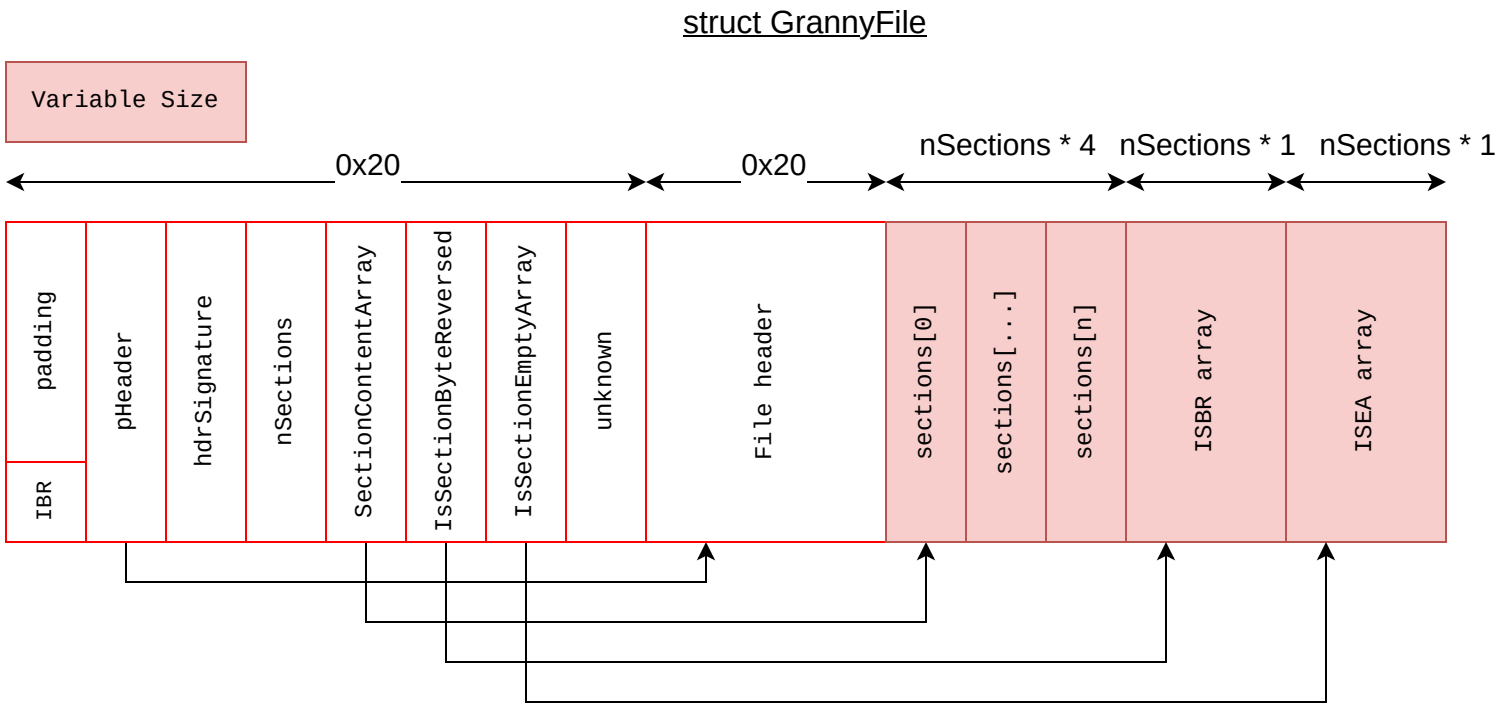
OOBR/OOBW

```
1  int *__cdecl GrannyGRNFixUp_0(DWORD RelocationCount, Relocation *PointerFixupArray, int *array, char *destination)
2  {
3    int *result; // eax
4    DWORD v6; // ebp
5    Relocation *v7; // ecx
6    int v8; // edx
7
8    result = (int *)RelocationCount;
9    if ( RelocationCount )
10   {
11     v6 = RelocationCount;
12     do
13     {
14       v7 = PointerFixupArray;
15       v8 = array[PointerFixupArray->SectionNumber];
16       result = (int *)&destination[PointerFixupArray->SectionOffset];
17       ++PointerFixupArray;
18       *result = v8;
19       if ( v8 )
20         *result = v8 + v7->Offset;
21       --v6;
22     }
23     while ( v6 );
24   }
25   return result;
26 }
```

Format GR2

GrannyFile en mémoire

- Structure de taille variable (dépend du nombre de section)
- Allocation possible hors du LFH



Heap Windows

Heap Windows

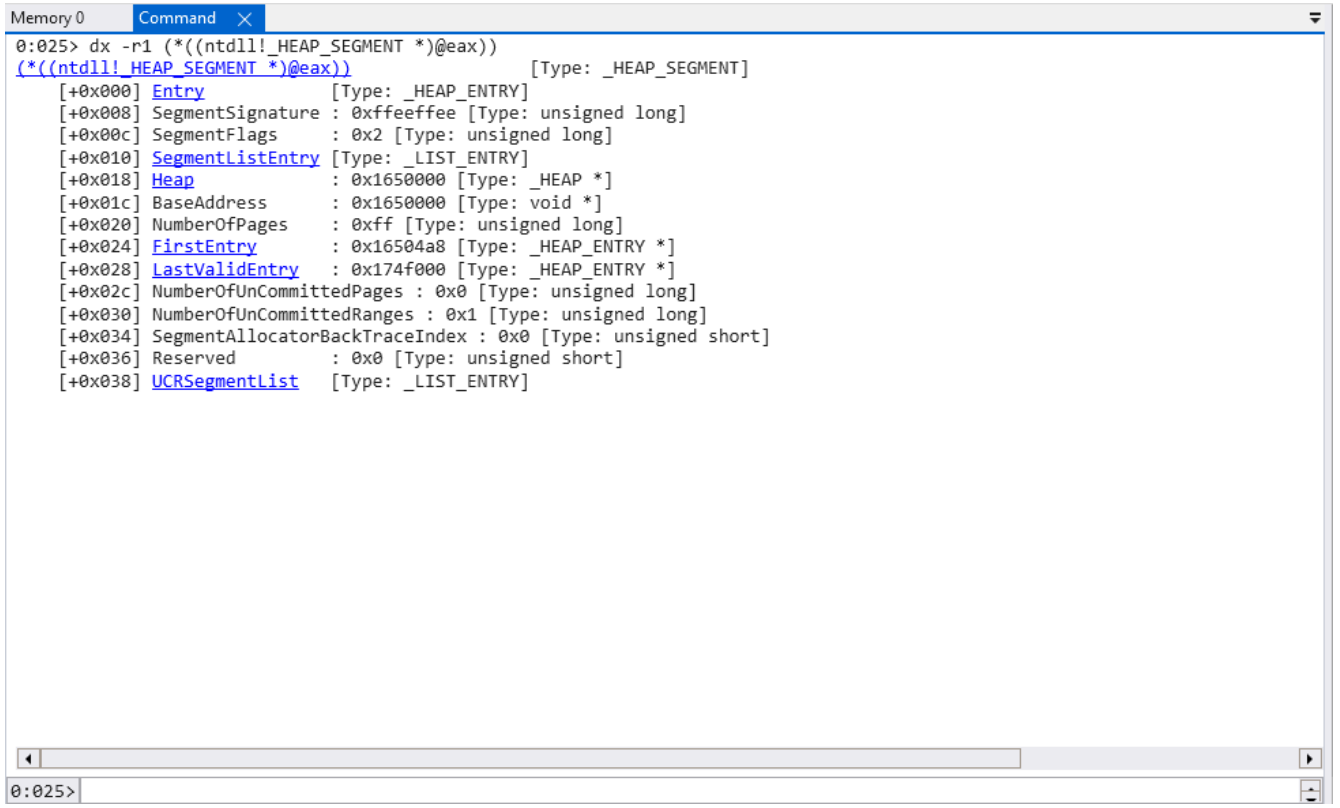
Introduction

- NT Heap
  - Fontend : Low Fragmentation Heap (LFH)
  - Backend
- Segment Heap
  - Low Fragmentation Heap (LFH)
  - Variable Size Allocation (VS)
  - Backend
  - Large Blocks Allocation
- https://www.blackhat.com/docs/us-16/materials/us-16-Yason-Windows-10-Segment-Heap-Internals.pdf

Heap Windows

Windbg

- SegmentSignature = 0xFFEEFFEE = **NT Heap**



Heap Windows

SYNACKTIV

LFH Randomization

- Randomly select a bit position in a ***BitmapBits***
- RtlpAllocateHeapInternal
- RtlpLargestLfhBlock (0x4000)

- https://www.blackhat.com/docs/us-16/materials/us-16-Yason-Windows-10-Segment-Heap-Internals.pdf

Exploitation

Exploitation

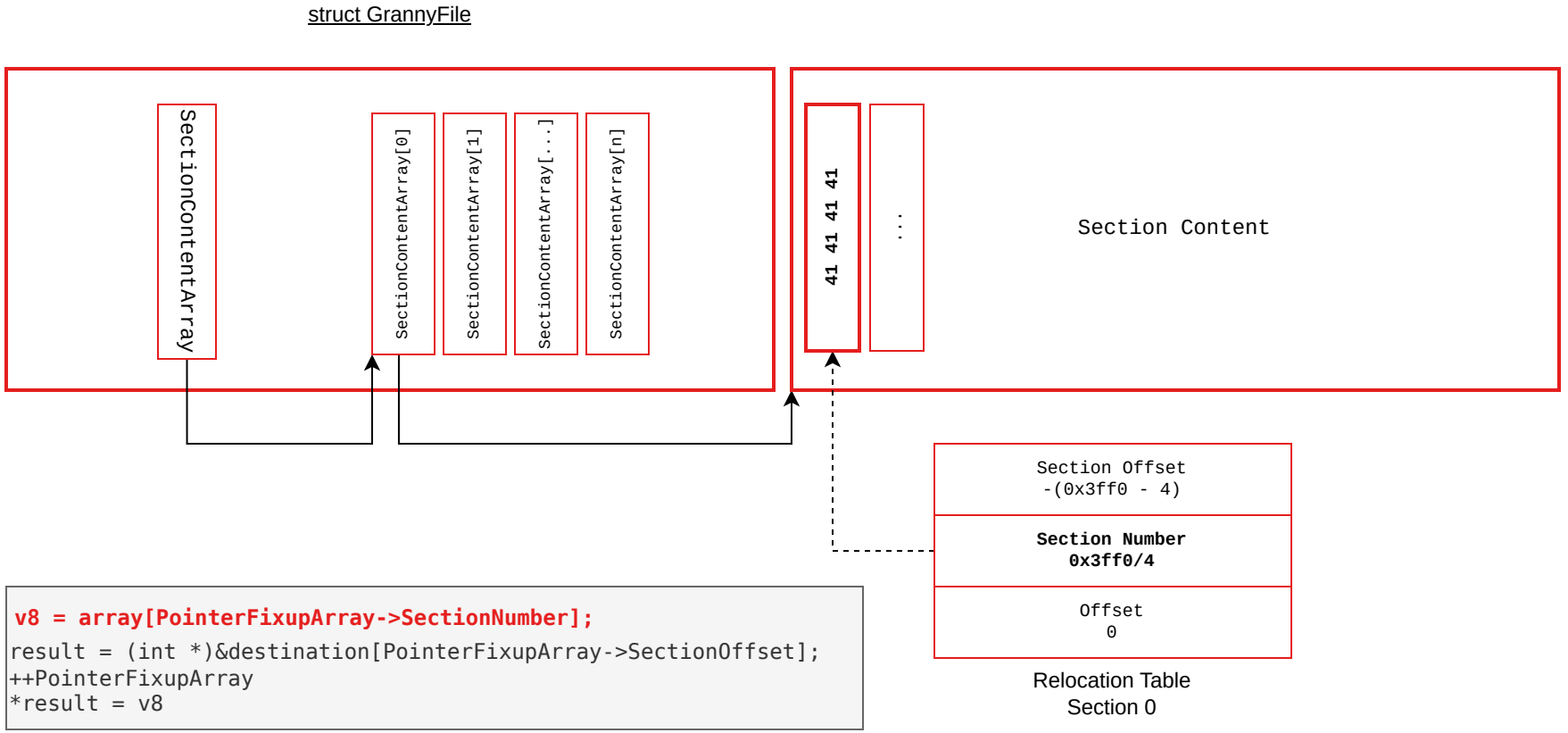
Résumé

SYNACKTIV

- 1 vulnérabilité type Path Traversal
  - **Déclenchement** : dans le salon de pré-partie via le mécanisme de sauvegarde multijoueur
  - **Capacité** : Permet de remplacer l'archive contenant les assets du jeu
- 1 vulnérabilité type OOBR/OOBW
  - **Déclenchement** : au lancement de la partie, lors du chargement d'un fichier GR2 corrompu
  - **Capacité** : corruption mémoire

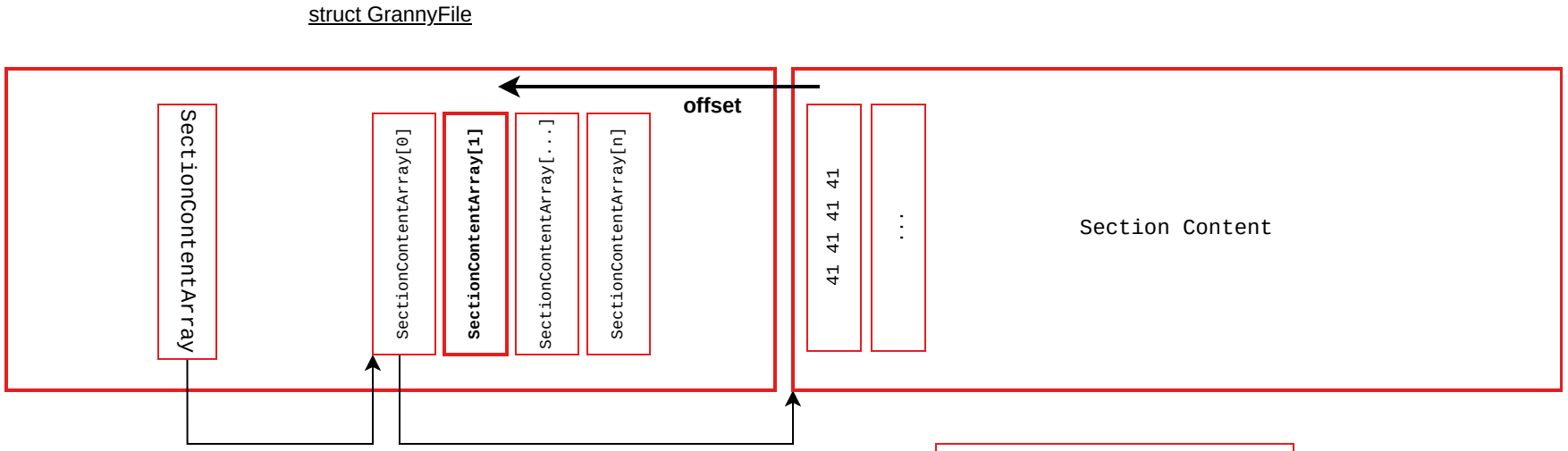
Exploitation

Primitive Write



Exploitation

Primitive Write



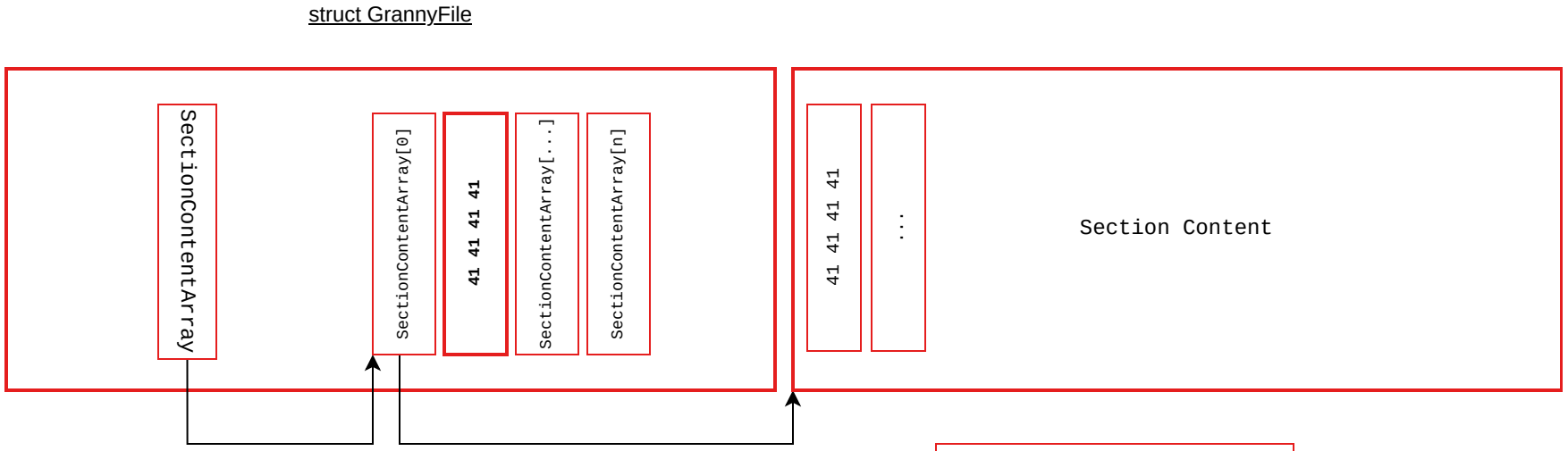

```
v8 = array[PointerFixupArray->SectionNumber];
result = (int *)&destination[PointerFixupArray->SectionOffset];
++PointerFixupArray
*result = v8
```

Section Offset -(0x3ff0 - 4)
Section Number 0x3ff0/4
Offset 0

Relocation Table
Section 0

Exploitation

Primitive Write



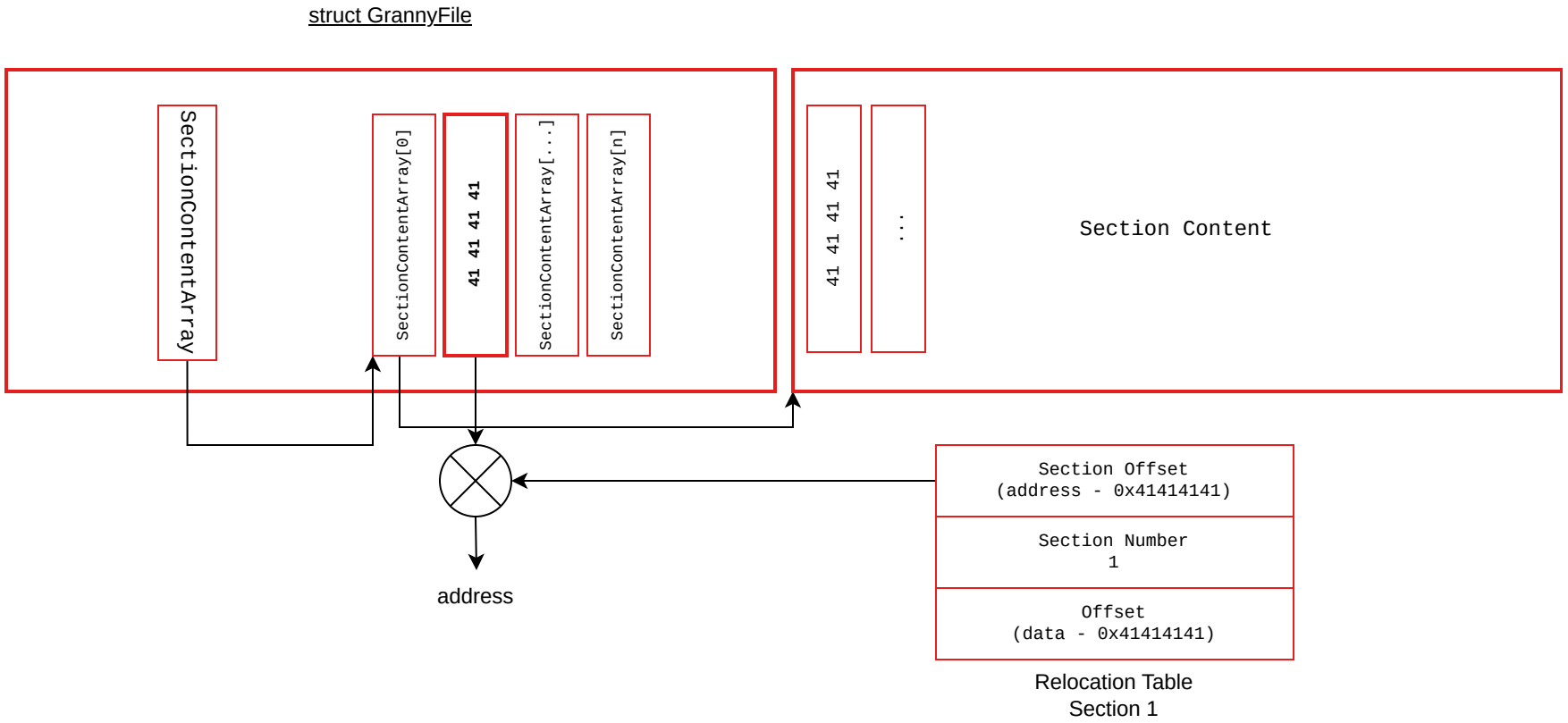

```
v8 = array[PointerFixupArray->SectionNumber];
result = (int *)&destination[PointerFixupArray->SectionOffset];
++PointerFixupArray
*result = v8
```

Section Offset -(0x3ff0 - 4)
Section Number 0x3ff0/4
Offset 0

Relocation Table
Section 0

Exploitation

Primitive Write



Exploitation

Stratégie

SYNACKTIV

- Ré-implémenter un patcher de RDA
- Tracer les modèles 3D chargé lors d'une partie multijoueur
- Envoyer un deuxième fichier
- Spray des sections de 0x4000 dans la heap
- Remplacer une des callbacks de gestion de la mémoire dans granny2.dll
- Ecriture du shellcode dans la section .data
- Win 🥳

Démonstration

SYNACKTIV

Bonus

Anno 2070

- Path traversal
- Repertoire d'installation accessible
- Game.cdf.lua
- Binding lua

```
1   _QWORD *__fastcall sub_1401AADC0(_QWORD *a1, __int64 a2, _QWORD *a3)
2   {
3     _QWORD *v5; // [rsp+30h] [rbp-18h]
4     _QWORD *v6; // [rsp+60h] [rbp+18h] BYREF
5     __int64 v7; // [rsp+68h] [rbp+20h]
6
7     v6 = a3;
8     v5 = operator new(0x28u);
9     luabind::detail::registration::registration(v5);
10    *v5 = &luabind::detail::function_registration<void (*)(CRDStringW const &,bool),luabind::detail::null_type>::`vftable';
11    v5[2] = "OpenUrl";
12    v5[3] = OpenURL;
13    v7 = 0;
14    v6 = v5;
15    sub_140E53FD0(a1, &v6);
16    return a1;
17  }
18
```

Anno 2070

Game over

SYNACKTIV

```
1  HINSTANCE __fastcall OpenURL(const WCHAR *lpFile, char a2)
2  {
3    HINSTANCE result; // rax
4
5    if ( qword_141BA1510 )
6    {
7      if ( a2 )
8        ShowWindow(*(HWND *)(qword_141BA1420 + 16), 6);
9      if ( *((_QWORD *)lpFile + 3) >= 8u )
10       lpFile = *(const WCHAR **)lpFile;
11     return ShellExecuteW(0, L"open", lpFile, 0, 0, 1);
12   }
13   return result;
14 }
```

Anno 2070

Demo

SYNACKTIV

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://twitter.com/synacktiv

https://synacktiv.com

Un incident ? Contactez csirt@synacktiv.com.