

The logo for SYNACKTIV features a stylized icon on the left consisting of a 3x3 grid of squares. The top-left square is white, the top-middle square is white with a red dot, and the top-right square is white. The remaining squares are black. To the right of this icon, the word "SYNACKTIV" is written in a bold, sans-serif font. "SYNA" is in white, and "CKTIV" is in red.

SYNACKTIV



# American Conquest

(et ses dingeries)

# Motivations

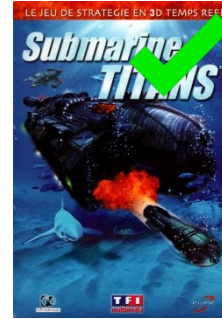
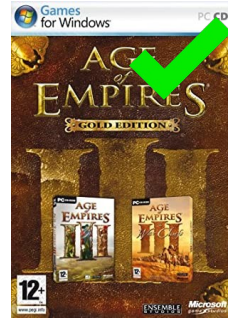
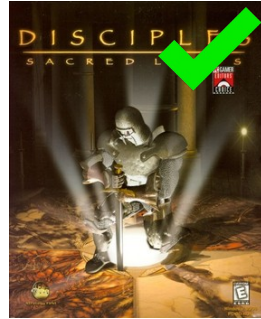
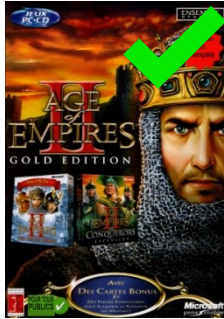
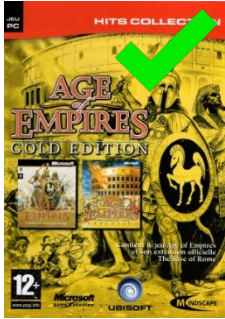


- Pourquoi chercher des vulnérabilités dans les vieux jeux vidéo ?
  - Pour le fun
  - Pour recycler sa collection de jeux vidéo
  - Intéressant lorsque les jeux sont réédités
  - De nombreux bugs, exploitation pas toujours évidente
- Concentration uniquement sur les RCE (pas de technique de cheat)

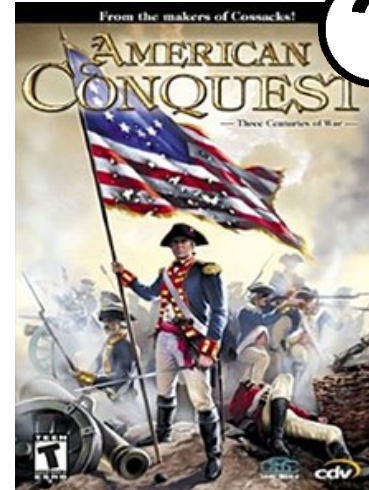
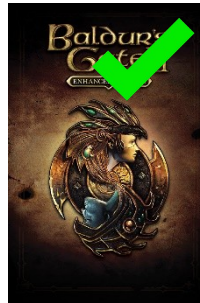
# Tableau de chasse



## Windows XP



## Windows 10 (re-released)





# Où trouver des vieux jeux réédités ?



# American Conquest



- Type - Stratégie
- Date de sortie - 7 Février 2003
- Editeur - GSC Game World



# Windows 10: Mitigations



Sécurité Windows

## Exploit Protection

Affichez les paramètres d'Exploit protection pour votre système et vos programmes. Vous pouvez personnaliser les paramètres de votre choix.

Paramètres système Paramètres du programme

**Prévention de l'exécution des données (PED)**  
Empêche l'exécution du code depuis des pages mémoire composées de données uniquement.

Utiliser la valeur par défaut (Activé) ▼

**Forcer la randomisation des images (randomisation du format d'espace d'adresse obligatoire)**  
Forcer le réadressage des images non compilées avec /DYNAMICBASE

Utiliser la valeur par défaut (Désactivé) ▼

[Exporter les paramètres](#)

Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-5PL66GD\Lab-01] (Administrateur)

File Options View Process Find Users DLL Help

Process	CPU	Private Byt...	Working Set	PID	Description	Company Name	DEP	ASLR	Control Flow Guard
csrss.exe	< 0.01	2 780 K	24 952 K	712			Enabled (permanent)	n/a	n/a
wininit.exe		1 356 K	7 280 K	720			Enabled (permanent)	n/a	n/a
winlogon.exe		2 888 K	13 332 K	776	Applicatio...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
SecurityHealthSystray.exe		1 716 K	9 768 K	4144	Windows ...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
vmtoolsd.exe		24 660 K	45 496 K	6276	VMware T...	VMware, Inc.	Enabled (permanent)	ASLR	CFG
OneDrive.exe		45 208 K	103 408 K	6772	Microsoft ...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
msedge.exe		45 316 K	109 692 K	6792	Microsoft ...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
explorer.exe	< 0.01	56 860 K	118 860 K	7792	Explorateur...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
DMCR.EXE	82.90	139 212 K	126 100 K	6036	dmcrc	-GSC-	Disabled (permanent)		
vopl.exe	< 0.01	3 160 K	11 404 K	3708			Disabled (permanent)		
dplaysvr.exe		2 268 K	8 072 K	8216	Applicatio...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG

Handles DLLs Threads

Name	Description	Company Name	Path
locale.nls			C:\Windows\System32\locale.nls
R000000000000006.clb			C:\Windows\Registration\R000000000000006.clb
wdmaud.drv.mui	Pilote du système audio Winmm	Microsoft Corporation	C:\Program Files\WindowsApps\Microsoft.LanguageExperiencePackfr-FR_1904...
MMDevAPI.dll.mui	API MMDevice	Microsoft Corporation	C:\Program Files\WindowsApps\Microsoft.LanguageExperiencePackfr-FR_1904...
Patch01.GS1			C:\GOG Games\American Conquest\Patch01.GS1

CPU Usage: 85.86% Commit Charge: 26.68% Processes: 146 Physical Usage: 32.12%

■ Non applicable pour les programmes compilé en 32 bits



# Quick Win



- Session multijoueur
- Ecrire un message dans le chat



# Quick Win



## Retrouver le message dans une capture réseau

The screenshot shows a Wireshark capture of network traffic on a VMware Network Adapter VMnet1. A filter is applied to show only UDP packets on port 2350. The packet list pane shows several packets, with packet 1176 selected. The packet details pane shows the structure of packet 1176: Ethernet II, Internet Protocol Version 4, and User Datagram Protocol. The data field shows a 55-byte payload. The packet bytes pane displays the raw hex and ASCII data, with a red box highlighting the ASCII sequence "TAHC" and a blue box highlighting the ASCII sequence "hubert t oujours le mot p our rire".

No.	Time	Source	Destination	Protocol	Length	Info
1172	121.5700...	192.168.136.133	192.168.136.134	UDP	74	2350 → 2350 Len=32
1173	121.6027...	192.168.136.133	192.168.136.134	UDP	74	2350 → 2350 Len=32
1174	121.6047...	192.168.136.134	192.168.136.133	UDP	74	2350 → 2350 Len=32
1175	121.7517...	192.168.136.134	192.168.136.133	UDP	62	2350 → 2350 Len=20
1176	121.8718...	192.168.136.134	192.168.136.133	UDP	97	2350 → 2350 Len=55
1177	122.0009...	192.168.136.134	192.168.136.133	UDP	74	2350 → 2350 Len=32
1178	122.0017...	192.168.136.133	192.168.136.134	UDP	74	2350 → 2350 Len=32

Frame 1176: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface 0  
Ethernet II, Src: VMware\_10:d0:02 (00:0c:29:10:d0:02), Dst: VMware\_10:d0:02 (00:0c:29:10:d0:02)  
Internet Protocol Version 4, Src: 192.168.136.134, Dst: 192.168.136.133  
User Datagram Protocol, Src Port: 2350, Dst Port: 2350  
Data (55 bytes)  
Data: 84ca020000000000544148432700000053616372e9 [Length: 55]

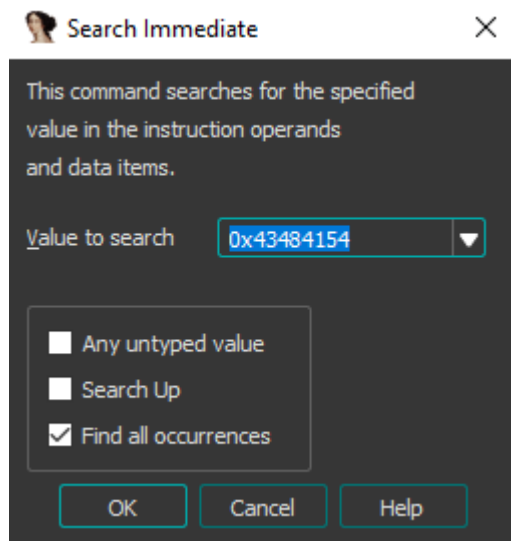
```
0000 00 0c 29 30 bc bb 00 0c 29 10 d0 02 08 00 45 00  ..)0.... ).....E.  
0010 00 53 f3 90 00 00 80 11 b4 ac c0 a8 88 86 c0 a8  .S.....  
0020 88 85 09 2e 09 2e 00 3f 6d 1c 84 ca 02 00 00 00  .....? m.....  
0030 00 00 54 41 48 43 27 00 00 00 53 61 63 72 e9 20  ..TAHC...Sacr.  
0040 68 75 62 65 72 74 20 74 6f 75 6a 6f 75 72 73 20  hubert t oujours  
0050 6c 65 20 6d 6f 74 20 70 6f 75 72 20 72 69 72 65  le mot p our rire  
0060 00
```



# Quick Win



- Retrouver l'identifiant de message
- Démarrer la rétro-ingénierie sur le gestionnaire de message



```
case 'CHAT':  
    memcpy(ChatTempBuffer, &packet_1->data.ping.field_4, packet_1->data.chat.length);  
    ChatTempBuffer[packet_1->data.chat.length] = 0;  
    dword_6B89D8 = idTo;  
    break;
```

*Buffer overflow in .bss, charset without constraint*

What is FIDN ?

```
else if ( packet_1->magic == 'FIDN' )  
{  
    packet_1->magic = 'FRPL';  
    len = GetRessourceLength(packet_1->data.chat.data);  
    packet_1->data.chat.length = len;  
    _send_message(8u, packet_1, idTo);  
}
```

# Vulnérabilité



- **GetResourceLength**
- RReset
  - UnixToWindowsPathAndOpen

```
struct TGSCfile *__cdecl UnixToWindowsPathAndOpen(char *path)
{
    bool v2; // [esp+0h] [ebp-114h]
    char _path[256]; // [esp+4h] [ebp-110h] BYREF
    int pathlen; // [esp+104h] [ebp-10h]
    int i; // [esp+108h] [ebp-Ch]
    bool v6; // [esp+10Fh] [ebp-5h]
    struct TGSCfile *v7; // [esp+110h] [ebp-4h]

    strcpy(_path, path);
    pathlen = strlen(_path);
    for ( i = 0; i < pathlen; ++i )
    {
        if ( _path[i] == '/' )
            _path[i] = '\\';
    }
    v2 = g_CGSCset.next && byte_67DC31;
    v6 = v2;
    _search_unrar_dll();
    v7 = CGSCset::gOpenFile(&g_CGSCset, _path, v6);
    if ( v7 )
        return v7;
    else
        return (struct TGSCfile *)-1;
}
```

# Vulnérabilités



## ■ GetRessourceLength

### ■ RReset

- UnixToWindowsPathAndOpen (stack bof)
  - CGSCset::gOpenFile
    - CGSCarch::GetFileHandle (stack bof)
      - *hash\_set\_func* (stack bof)

```
struct TGSCfile *__thiscall CGSCarch::GetFileHandle(CGSCarch *this, const char *path)
{
    int v2; // eax
    char _path[64]; // [esp+Ch] [ebp-50h] BYREF
    int v6; // [esp+4Ch] [ebp-10h]
    unsigned int i; // [esp+50h] [ebp-Ch]
    char *v8; // [esp+54h] [ebp-8h]
    unsigned int v9; // [esp+58h] [ebp-4h]

    i = 0;
    v8 = 0;
    v9 = 0;
    memset(_path, 0, sizeof(_path));
    strcpy(_path, path);
    _strupr(_path);
    v6 = hash_set_func(_path);
}
```

```
int __cdecl hash_set_func(char *String)
{
    char *v1; // eax
    int checksum; // edx
    int v3; // ecx
    char *v4; // ebx
    int v5; // eax
    int v6; // eax
    char v7; // t1
    char Destination[64]; // [esp+4h] [ebp-40h] BYREF

    memset(Destination, 0, sizeof(Destination));
    v1 = _strupr(String);
    strcpy(Destination, v1);
    checksum = 0;
    v3 = 0x10;
    v4 = Destination;
    do
    {
        v5 = *(_DWORD *)v4;
        BYTE1(v5) = *(_DWORD *)v4;
        LOBYTE(v5) = BYTE1(*(_DWORD *)v4);
        v6 = __ROL4__(v5, 0x10);
        v7 = BYTE1(v6);
        BYTE1(v6) = v7;
        LOBYTE(v6) = v7;
        checksum += v6;
        v4 += 4;
        --v3;
    }
    while ( v3 );
    return checksum;
}
```



# Contrainte



- **Jeu de caractères**
  - Octets null en fin de chaîne
  - Conversion minuscule en majuscule
- **DMCR.exe base address 0x00400000**
  - Un seul gadget
  - Pas de jmp esp
- **DLLs ne sont pas chargée selon l'ImageBase**



shellcode  
in stack

shellcode  
in bss

# Fonction de hash

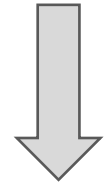


## ■ Contrôle EAX en sortie de fonction via le « Path »

```
mov ecx, 10h
lea ebx, [ebp + source]
HashLoop:
mov eax, [ebx]
xchg ah, al
rol eax, 10h
xchg ah, al
add edx, eax
add ebx, 4
loop HashLoop
mov eax, edx
```

0x00462a36 (6\*F ): jmp eax

AA  
AA| |!- ]{)



hash\_set\_func

0x006B88D8  
(ChatTempBuffer)

# Demo





# Timeline



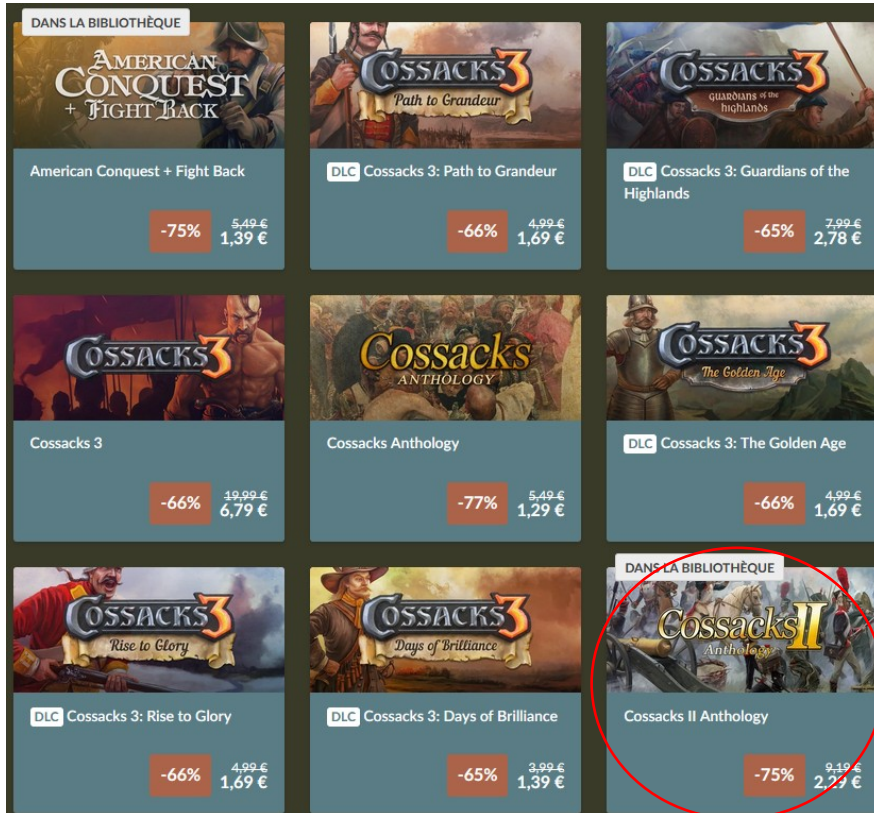
- 17/12/2023 PoC
- 24/01/2024 Envoi d'un mail à l'éditeur
- 05/04/2024 Relance éditeur
- 05/04/2024 L'éditeur répond, le bug ne sera pas corrigé

# Dommage collatéral



## Cossacks II

- Type – Stratégie
- Date de sortie - 26 Avril 2005
- Editeur - GSC Game World



# Dommage collatéral



## ■ Vulnérabilités toujours présentes mais

- pas d'archive .GSC
- stack cookie

## ■ Cas d'école

```
buffer      db 260 dup(?)
cookie      dd ?
s           db 4 dup(?)
r           db 4 dup(?)
pPath      dd ?
```

*Stack View*

```
push  ebp
mov   ebp, esp
sub   esp, 11Ch
mov   eax, cookie
mov   [ebp+cookie], eax
mov   eax, [ebp+pPath]
push  eax           ; Source
lea   ecx, [ebp+buffer]
push  ecx           ; Destination
call  _strcpy
add   esp, 8
```

...

```
mov   ecx, [ebp+cookie]
call  checkcookie
mov   esp, ebp
pop   ebp
retn
```



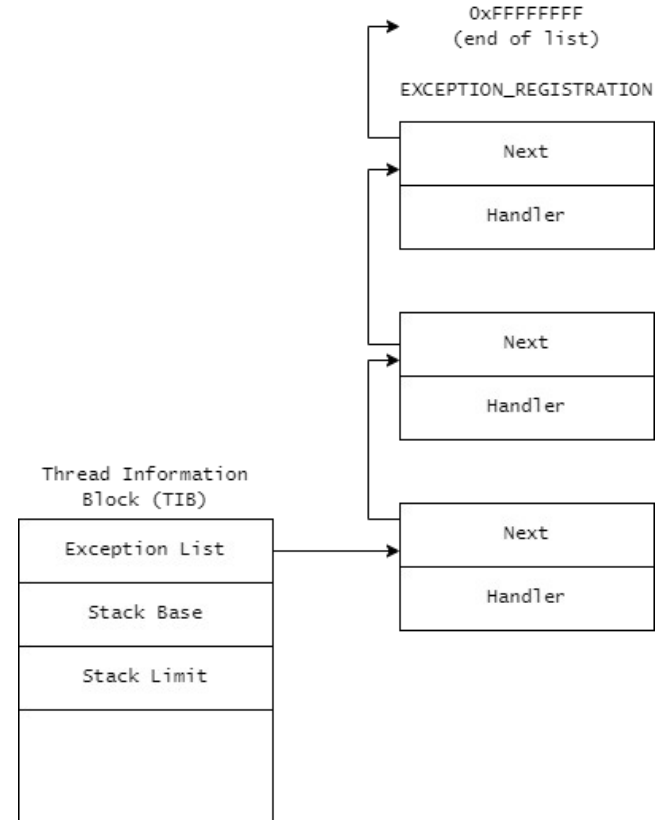
# Did you know SEH ?



- Structured Exception Handler (SEH)
- Gestion des exceptions

Address	Name	Stack
01434469	.text:engine_?DecompressData@@YA_NPAPAIPAIP...	0019EF98
0148106D	.text:engine_?DecompressData@@YA_NPAPAIPAIP...	0019EFE4
01436E6D	.text:engine_?DecompressData@@YA_NPAPAIPAIP...	0019F238
0143EA9D	.text:engine_?DecompressData@@YA_NPAPAIPAIP...	0019F6EC
0143EA9D	.text:engine_?DecompressData@@YA_NPAPAIPAIP...	0019FB8C
01445A20	sub_898F00+BACB20	0019FD50
012ED35C	__except_handler3	0019FF60
7751AF30	ntdll.dll:ntdll_wcstombs+70	0019FFCC
77528C28	ntdll.dll:ntdll_RtlCaptureContext+F8	0019FFE4

Debugger > Debugger Windows > SEH List



# SEH Exploitation



- Remplacer le *Handler* d'une structure SEH
- Générer une exception
  - Ecrire sur une page Read-Only

Name	Start	End	R	W	X	D	L	Align	Base
debug001	00010000	00020000	R	W	.	D	.	byte	0000
debug002	00020000	00022000	R	W	.	D	.	byte	0000
debug003	00022000	00028000	?	?	?	D	.	byte	0000
debug004	00030000	00031000	R	.	.	D	.	byte	0000
debug005	00040000	0005D000	R	.	.	D	.	byte	0000
debug006	00060000	00095000	?	?	?	D	.	byte	0000
debug007	00095000	00098000	R	W	.	D	.	byte	0000
debug008	00098000	000A0000	R	W	.	D	.	byte	0000
debug009	000A0000	0017E000	?	?	?	D	.	byte	0000
Stack_PAGE_GUARD[00...	0017E000	00180000	R	W	.	D	.	byte	0000
Stack[000018C0]	00180000	001A0000	R	W	.	D	.	byte	0000
debug010	001A0000	001A4000	R	.	.	D	.	byte	0000
debug011	001B0000	001B2000	R	W	.	D	.	byte	0000
debug012	001C0000	001C1000	R	.	.	D	.	byte	0000
debug013	001D0000	001D1000	R	.	.	D	.	byte	0000
debug014	001E0000	001EF000	R	W	.	D	.	byte	0000

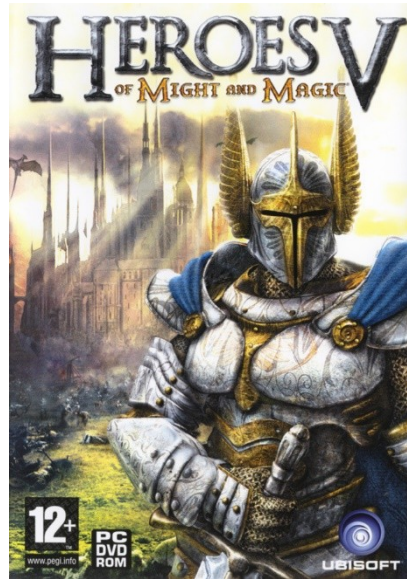
# Demo



# Conclusion



■ Next ...





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

<https://twitter.com/synacktiv>

Nos publications sur : <https://synacktiv.com>