SYNACKTIV

Le pwner du grenier

Rump SSTIC

Saison 2 - Episode 11

4/06/2026

Introduction

Sommaire

SYNACKTIV

- **Introduction**
- **Rétro-ingénierie**
- **Exploitation**
- **Demo**

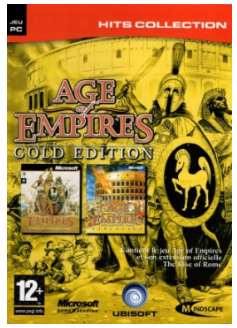
Introduction

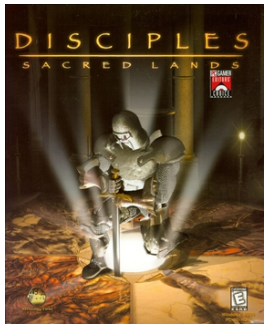
Whoami

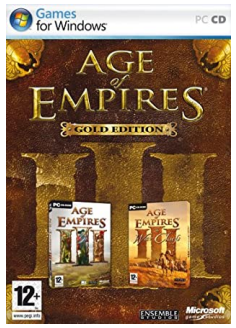
- Thomas Dubier **@Tomtombinary**
  - Security Researcher at Synacktiv
  - Reverse engineering team
- Synacktiv
  - Offensive security company
  - ~200 Ninja
  - We're hiring !

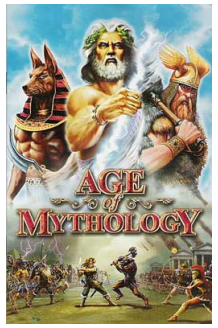
Introduction

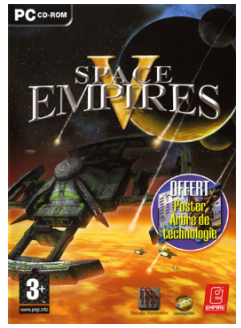
Tableau de chasse

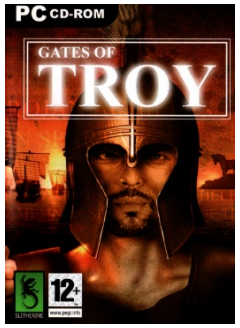


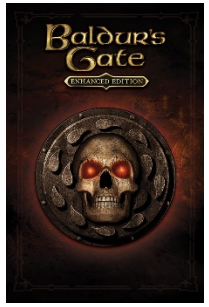


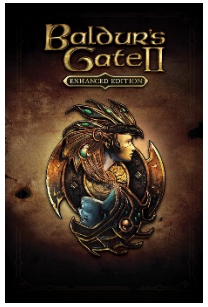


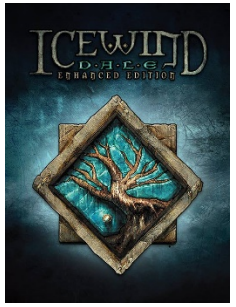


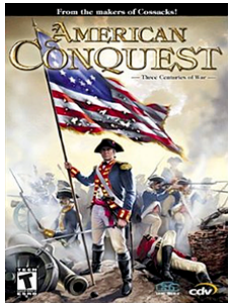


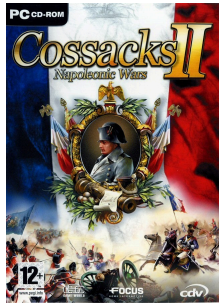


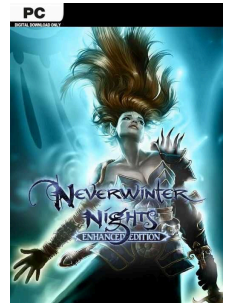


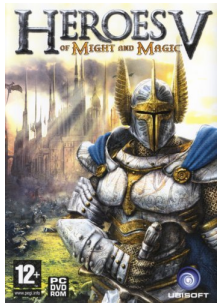












Introduction

Stronghold HD

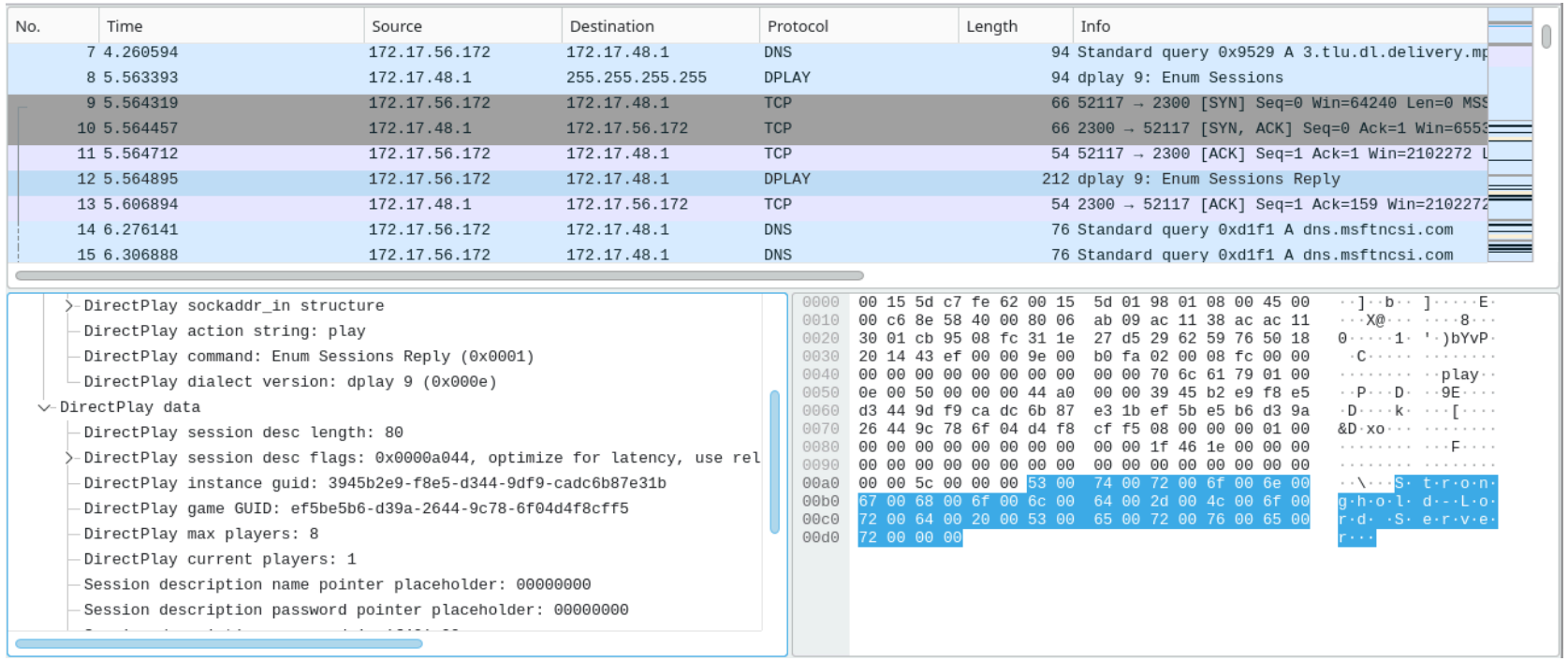
SYNACKTIV

- **Type : Jeu de stratégie (RTS)**
- **Développeur : Firefly Studios**
- **Editeur : Gathering of Developers**
- **Date de sortie : 2001**
- **Moteur de jeu : Propriétaire**
- **Version : GOG v1.41**
- **Technologie : DirectPlay**

Introduction

DirectPlay

- API disponible dans la suite DirectX (jusqu'à la version 8)
- Utilisable sous la forme d'un objet COM (Component Object Model)
- Utilisé pour gérer les échanges réseaux dans les jeux vidéo
- Implémente un protocole réseau nommé DirectPlay



Introduction

Interface COM

- Interface COM documenté dans le dossier `includes` de DirectX

```
DECLARE_INTERFACE_(IDirectPlay4, IDirectPlay3)
{
    /*** IUnknown methods ***/
    STDMETHOD(QueryInterface)       (THIS_ REFIID riid, LPVOID * ppvObj) PURE;
    STDMETHOD_(ULONG, AddRef)        (THIS)  PURE;
    STDMETHOD_(ULONG, Release)       (THIS) PURE;
    /*** IDirectPlay2 methods ***/
    STDMETHOD(AddPlayerToGroup)     (THIS_ DPID, DPID) PURE;
    STDMETHOD(Close)                (THIS) PURE;
    STDMETHOD(CreateGroup)          (THIS_ LPDPID, LPDPNAME, LPVOID, DWORD, DWORD) PURE;
    STDMETHOD(CreatePlayer)         (THIS_ LPDPID, LPDPNAME, HANDLE, LPVOID, DWORD, DWORD) PURE;
    STDMETHOD(DeletePlayerFromGroup)(THIS_ DPID, DPID) PURE;
    STDMETHOD(DestroyGroup)         (THIS_ DPID) PURE;
    STDMETHOD(DestroyPlayer)        (THIS_ DPID) PURE;
    STDMETHOD(EnumGroupPlayers)     (THIS_ DPID, LPGUID, LPDPENUMPLAYERSCALLBACK2, LPVOID, DWORD) PURE;
    STDMETHOD(EnumGroups)           (THIS_ LPGUID, LPDPENUMPLAYERSCALLBACK2, LPVOID, DWORD) PURE;
    STDMETHOD(EnumPlayers)          (THIS_ LPGUID, LPDPENUMPLAYERSCALLBACK2, LPVOID, DWORD) PURE;
    STDMETHOD(EnumSessions)         (THIS_ LPDPSESSIONDESC2, DWORD, LPDPENUMSESSIONSCALLBACK2, LPVOID, DWORD) PURE;
    [...]
};
```

Rétro-ingénierie

Retrouver l'initialisation de l'interface

SYNACKTIV

- Les interfaces COM sont identifiées par un GUID

```
DEFINE_GUID(IID_IDirectPlay4A, 0xab1c531, 0x4745, 0x11d1, 0xa7, 0xa1, 0x0, 0x0, 0xf8, 0x3, 0xab, 0xfc);
```

- Trouver les appels à CoCreateInstance

```
1 hr = CoCreateInstance(&rclsid, 0, 1u, &IID_IDirectPlay4, (LPVOID *)&ppv);
2 if ( hr >= 0 )
3 {
4   v11 = ((int (__stdcall *)(IDirectPlay4 *, LPCVOID, _DWORD))ppv->vftable->InitializeConnection)(ppv, *v7, 0);
5   if ( v11 < 0 )
6   {
7 LABEL_17:
8     ((void (__stdcall *)(IDirectPlay4 *))ppv->vftable->Release)(ppv);
```

Rétro-ingénierie

Comment trouver le dispatcher ?

```
1  IDirectPlay4A = this->pidirectplay427C;
2  if ( !IDirectPlay4A )
3      break;
4  lpData = this->lpData;
5  this->lpDataSize = 45000;
6  p_dpIdFrom = &this->dpIdFrom;
7  rc = IDirectPlay4A->vftable->Receive(
8          IDirectPlay4A,
9          &this->dpIdFrom,
10         &this->dpIdTo,
11         1,
12         this->lpData,
13         &this->lpDataSize);
14 this->lastDpError = rc;
15 if ( rc == 0x8000000A || rc == 0x887700BE ) // DPERR_NOMESSAGES
16     break;
17 [...]
18     HandleApplicationMessage(this, packet_type, *p_dpIdFrom, __extracted_size, &this->lpData[4]);
```

Rétro-ingénierie

Comment trouver le dispatcher ?

```
1  void __thiscall HandleApplicationMessage(
2          GameEngine *this,
3          char packet_type,
4          DPID dpIdFrom,
5          int size,
6          unsigned __int8 *data)
7  {
8   [...]
9      this->RxBuffer[this->index_copy].field_9 = 1;
10     this->RxBuffer[this->index_copy].dpIdFrom = dpIdFrom;
11     this->RxBuffer[this->index_copy].type = packet_type;
12     this->RxBuffer[this->index_copy].size = size;
13     index_copy = this->index_copy;
14     this->offset = 0;
15     this->bIsClient = 2;
16     StateMachineHandlers[this->RxBuffer[index_copy].type]();
```

Rétro-ingénierie

SYNACKTIV

Evaluer la surface d'attaque

- 108 handlers

```
.data:007E8FB0 StateMachineHandlers dd offset nullsub_1, offset nullsub_1, offset HandleMessage02h
.data:007E8FB0                                         ; DATA XREF: sub_46DD20+90↑r
.data:007E8FB0                                         ; HandleApplicationMessage+125↑r ...
.data:007E8FBC                 dd offset HandleMessage03h, offset HandleMessage04h, offset HandleMessage05h
.data:007E8FC8                 dd offset HandleMessage06h, offset HandleMessage07h, offset HandleMessage08h
.data:007E8FD4                 dd offset HandleMessage09h, offset HandleMessage0Ah, offset HandleMessage0Bh
.data:007E8FE0                 dd offset HandleMessage0Ch, offset HandleMessage0Dh, offset HandleMessage0Eh
.data:007E8FEC                 dd offset HandleMessage0Fh, offset HandleMessage10h, offset HandleMessage11h
.data:007E8FF8                 dd offset HandleMessage12h, offset HandleMessage13h, offset HandleMessage14h
.data:007E9004                 dd offset HandleMessage15h, offset HandleMessage16h, offset HandleMessage17h
.data:007E9010                 dd offset HandleMessage18h, offset HandleMessage19h, offset HandleMessage1Ah
.data:007E901C                 dd offset HandleMessage1Bh, offset HandleMessage1Ch, offset HandleMessage1Dh
.data:007E9028                 dd offset HandleMessage1Eh, offset HandleMessage1Fh, offset HandleMessage08h
.data:007E9034                 dd offset HandleMessage21h, offset HandleMessage22h, offset HandleMessage23h
.data:007E9040                 dd offset sub_470040, offset sub_470170, offset sub_470240
.data:007E904C                 dd offset sub_475FD0, offset sub_476210, offset sub_4702E0
.data:007E9058                 dd offset sub_4703F0, offset sub_470590, offset sub_4705F0
.data:007E9064                 dd offset sub_470700, offset sub_4707D0, offset sub_470CB0
.data:007E9070                 dd offset sub_4763B0, offset sub_470E00, offset sub_4765D0
.data:007E907C                 dd offset sub_470EB0, offset sub_470F60, offset sub_4711C0
.data:007E9088                 dd offset sub_479E90, offset sub_479F20, offset sub_4712C0
.data:007E9094                 dd offset sub_4713D0, offset sub_471490, offset sub_471550
.data:007E90A0                 dd offset sub_471660, offset sub_471720, offset sub_4717B0
.data:007E90AC                 dd offset sub_471870, offset sub_471CB0, offset sub_471FE0
.data:007E90B8                 dd offset sub_4720C0, offset sub_472120, offset sub_472170
.data:007E90C4                 dd offset sub_472280, offset sub_472480, offset sub_472520
.data:007E90D0                 dd offset sub_472630, offset sub_4726D0, offset sub_472740
.data:007E90DC                 dd offset sub_4727E0, offset sub_472910, offset sub_476870
.data:007E90E8                 dd offset sub_472A70, offset sub_472AE0, offset sub_472B40
.data:007E90F4                 dd offset sub_472C00, offset sub_472CC0, offset sub_476A50
.data:007E9100                 dd offset sub_472D80, offset sub_472E40, offset sub_472F00
.data:007E910C                 dd offset sub_476E30, offset sub_472F60, offset sub_472FC0
.data:007E9118                 dd offset sub_473080, offset sub_476F30, offset sub_476FE0
.data:007E9124                 dd offset sub_473140, offset sub_4731A0, offset sub_477190
.data:007E9130                 dd offset sub_4731E0, offset sub_473370, offset sub_473410
.data:007E913C                 dd offset sub_4734A0, offset sub_473520, offset sub_4735D0
.data:007E9148                 dd offset nullsub_1, offset nullsub_1, offset nullsub_1
.data:007E9154                 dd offset nullsub_1, offset nullsub_1, offset nullsub_1
```

Rétro-ingénierie

Observons les paquets

```
ecx=0xf1c858,packet_type=10,dpIdFrom=0x32ee7,size=0x0,data=0xf1d4d4
           0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
00f1d4d4  01 00 00 00 4c 00 6f 00 72 00 64 00 20 00 53 00  ....L.o.r.d. .S.
00f1d4e4  65 00 72 00 76 00 65 00 72 00 00 00 00 00 00 00  e.r.v.e.r.......
00f1d4f4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d504  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d514  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d524  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d534  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d544  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d554  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d564  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d574  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d584  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d594  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d5a4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d5b4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00f1d5c4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Rétro-ingénierie

SYNACKTIV

Traitement du paquet 10 (0Ah)

- Multiple out of bound write

```
1   void HandleMessage0Ah()
2   {
3     Engine.__packet_size = 504;
4     Engine.RxBuffer[Engine.index_copy].size = 0;
5     if ( Engine.__isHost == 1 )
6     {
7       [...]
8     }
9     else if ( !Engine.__isHost )
10    {
11      ReadOrWriteInRxOrTxBuffer(&Engine, &index, 4, 0, 1);
12      ReadOrWriteInRxOrTxBuffer(&Engine, name, 500, 0, 1);
13      WideCharToCP1252((LPSTR)(250 * index + 0xFB4DD4), name, 250);
14      _index = index;
15      v2 = Engine.field_9857C[index];
16      v3 = (char *)&Engine.field_9b078 + 250 * index - v2;
17      do
18      {
19        v4 = *v2;
20        v2[v3] = *v2;
21        ++v2;
22      }
23      while ( v4 );
24      Engine.field_39694[_index] = 1;
```

Exploitation

Mitigations

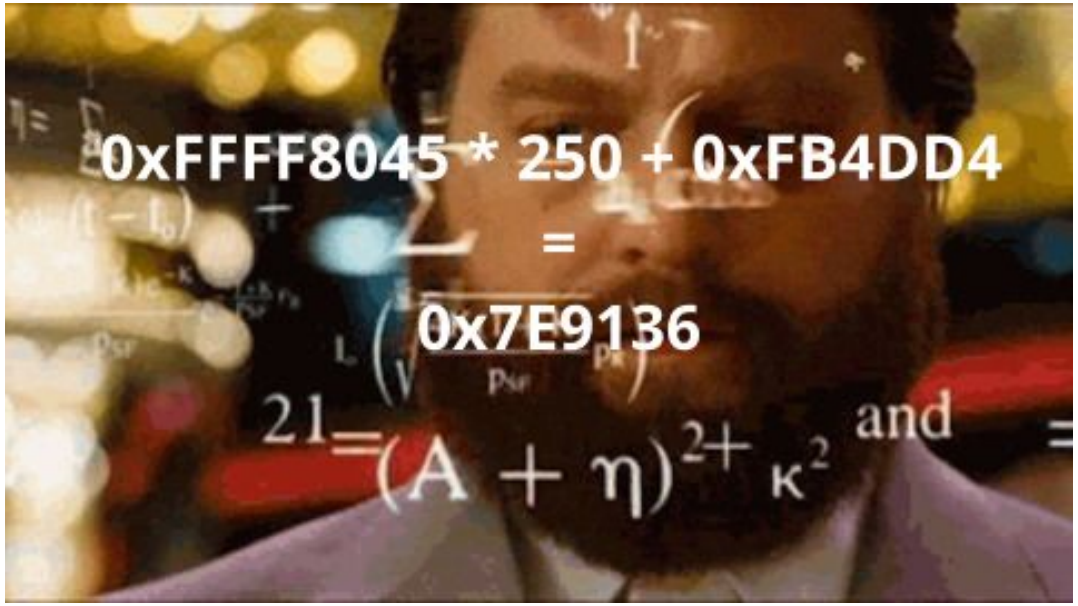
- 32 bits
- No DEP
- No ASLR
- No Read Only Data

Exploitation

Faire des maths

SYNACKTIV

```
WideCharToCP1252((LPSTR)(250 * index + 0xFB4DD4), name, 250);
```



Exploitation

Fumer un pointeur

SYNACKTIV

- Remplacer le pointeur du handler n°98

```
.data:007E90D0                 dd offset sub_472630, offset sub_4726D0, offset sub_472740
.data:007E90DC                 dd offset sub_4727E0, offset sub_472910, offset sub_476870
.data:007E90E8                 dd offset sub_472A70, offset sub_472AE0, offset sub_472B40
.data:007E90F4                 dd offset sub_472C00, offset sub_472CC0, offset sub_476A50
.data:007E9100                 dd offset sub_472D80, offset sub_472E40, offset sub_472F00
.data:007E910C                 dd offset sub_476E30, offset sub_472F60, offset sub_472FC0
.data:007E9118                 dd offset sub_473080, offset sub_476F30, offset sub_476FE0
.data:007E9124                 dd offset sub_473140, offset sub_4731A0, offset sub_477190
.data:007E9130                 dd offset sub_4731E0, offset sub_473370, offset sub_473410
.data:007E913C                 dd offset sub_4734A0, offset sub_473520, offset sub_4735D0
.data:007E9148                 dd offset nullsub_1, offset nullsub_1, offset nullsub_1
.data:007E9154                 dd offset nullsub_1, offset nullsub_1, offset nullsub_1
```

0x7E9136

Exploitation

Contrainte CP-1252

Windows-1252 (CP1252)																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8x	€		‚	*ƒ*	„	…	†	‡	ˆ	‰	Š	‹	Œ		Ž	
9x		‘	’	“	”	•	–	—	˜	™	š	›	œ		ž	Ÿ
Ax	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Exploitation

Sauter sur le buffer de reception

Demo

C'est crasseux mais ça marche

SYNACKTIV

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

https://twitter.com/synacktiv

https://synacktiv.com

Un incident ? Contactez csirt@synacktiv.com.