SYNACKTIV

Tale of an RCE in a video game Neverwinter Nights

04/10/2024

HEXACON 2024

Agenda

SYNACKTIV

- Introduction
- Concepts
- Old games old bugs
- Neverwinter Nights
- Attack Surface
- Vulnerabilities
- Exploitation
- Conclusion

whoami

SYNACKTIV

- **Thomas DUBIER @Tomtombinary**
  - Security researcher @Synacktiv
  - In the Reverse Engineering team
- **Synacktiv**
  - Offensive security company
  - Based in France
  - ~170 Ninjas
  - We are hiring !

Research motivations

SYNACKTIV

- **Why look for vulnerabilities in old video games ?**
  - ~~Bug bounty~~
  - To have fun
  - To recycle my old video game collections
  - Interesting when old games are re-released
  - There are always bugs, but sometimes complicated to exploit
  - Training
- **Focus on RCE (no cheating technique)**

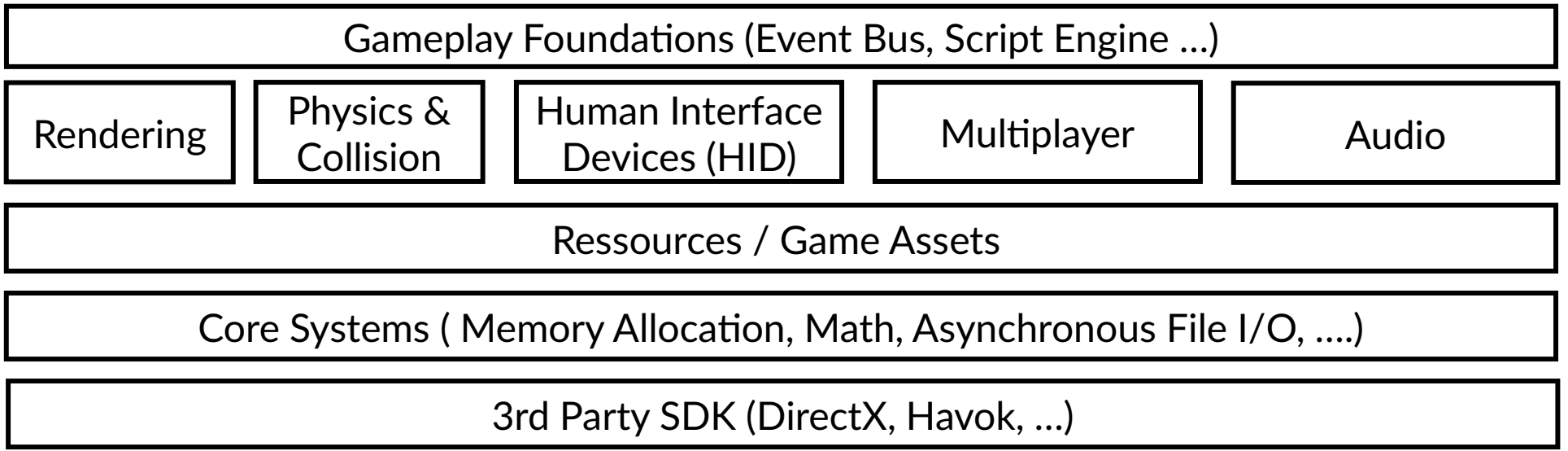
Concepts

What is video game ?

SYNACKTIV

What is game engine ?

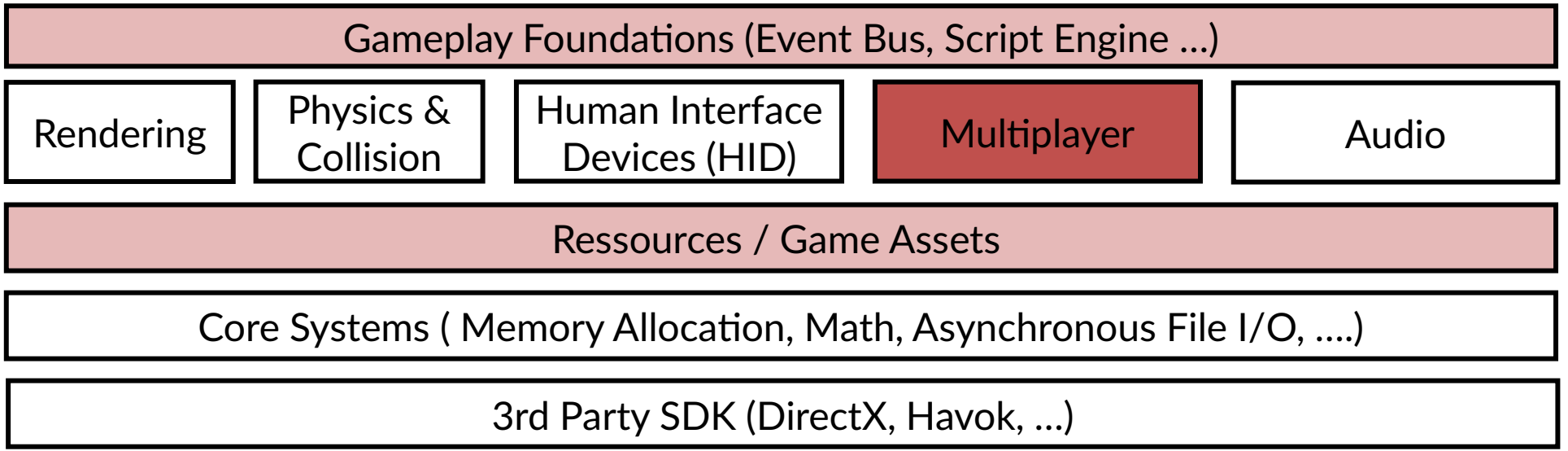
SYNACKTIV



Source: https://www.gameenginebook.com/
« Game Engine Architecture » by Jason Gregory

Where to focus ?

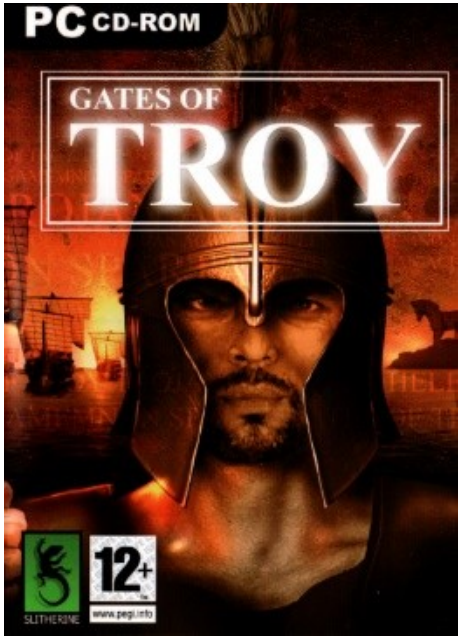
SYNACKTIV



Old games old bugs

Stack buffer overflow

SYNACKTIV

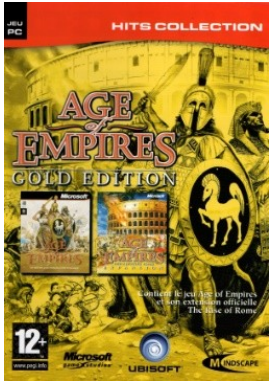


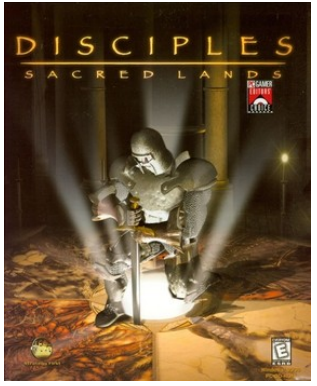

```
void __thiscall sub_404160(comm_t *this)
{
  [...]
  char buf[0x800]; // [esp+34h] [ebp-800h] BYREF

  [...]
  while ( 1 )
  {
    bytes_recv = recvfrom(this->sockfd, ::buf, 0x8000, 0, &from, &fromlen);
    if ( bytes_recv == -1 )
      break;
    if ( bytes_recv <= 0 )
      goto LABEL_6;
    qmemcpy(buf, ::buf, bytes_recv);
```

Stack buffer overflow

SYNACKTIV






```
int __thiscall sub_4DC120(_DWORD *this)
{
  const char *SessionName; // eax
  int NumberOfPlayer; // [esp-8h] [ebp-9Ch]
  int NumberMax; // [esp-4h] [ebp-98h]
  char GameName[122]; // [esp+8h] [ebp-8Ch] BYREF
  [...]
  NumberMax = Array_GetNumberMax(v9);
  NumberOfPlayer = Array_GetNumberOfPlayer(v9);
  SessionName = (const char *)Array_GetSessionName(v9++);
  sprintf(GameName, "%s ( %.1d / %.1d )", SessionName, NumberOfPlayer, NumberMax);
```

Stack buffer overflow

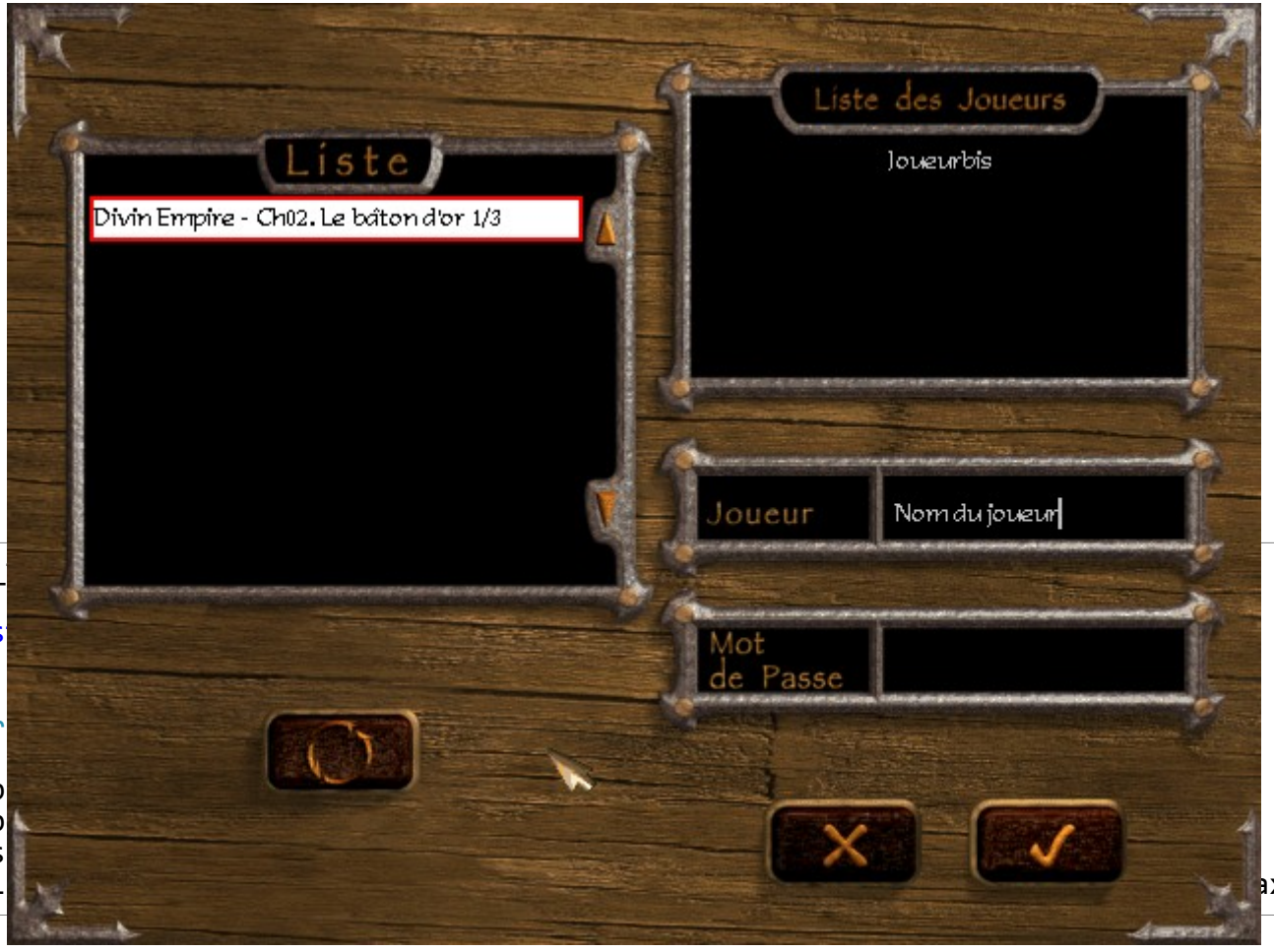
SYNACKTIV


```
int _
{
  cons
  int
  int
  char
  [..
  Numb
  Numb
  Sess
  spri
```

```
);
```

Stack buffer overflow

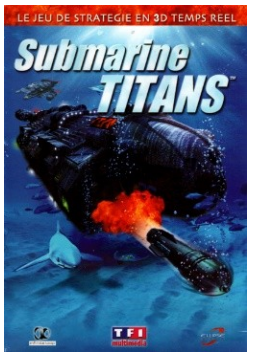
SYNACKTIV




```
int __
{
  cons
  int
  int
  char
  [...
  Numb
  Numb
  Sess
  spri                                    ax);
```

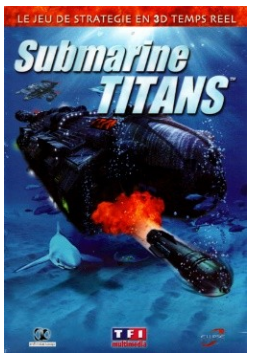
Game assets handling

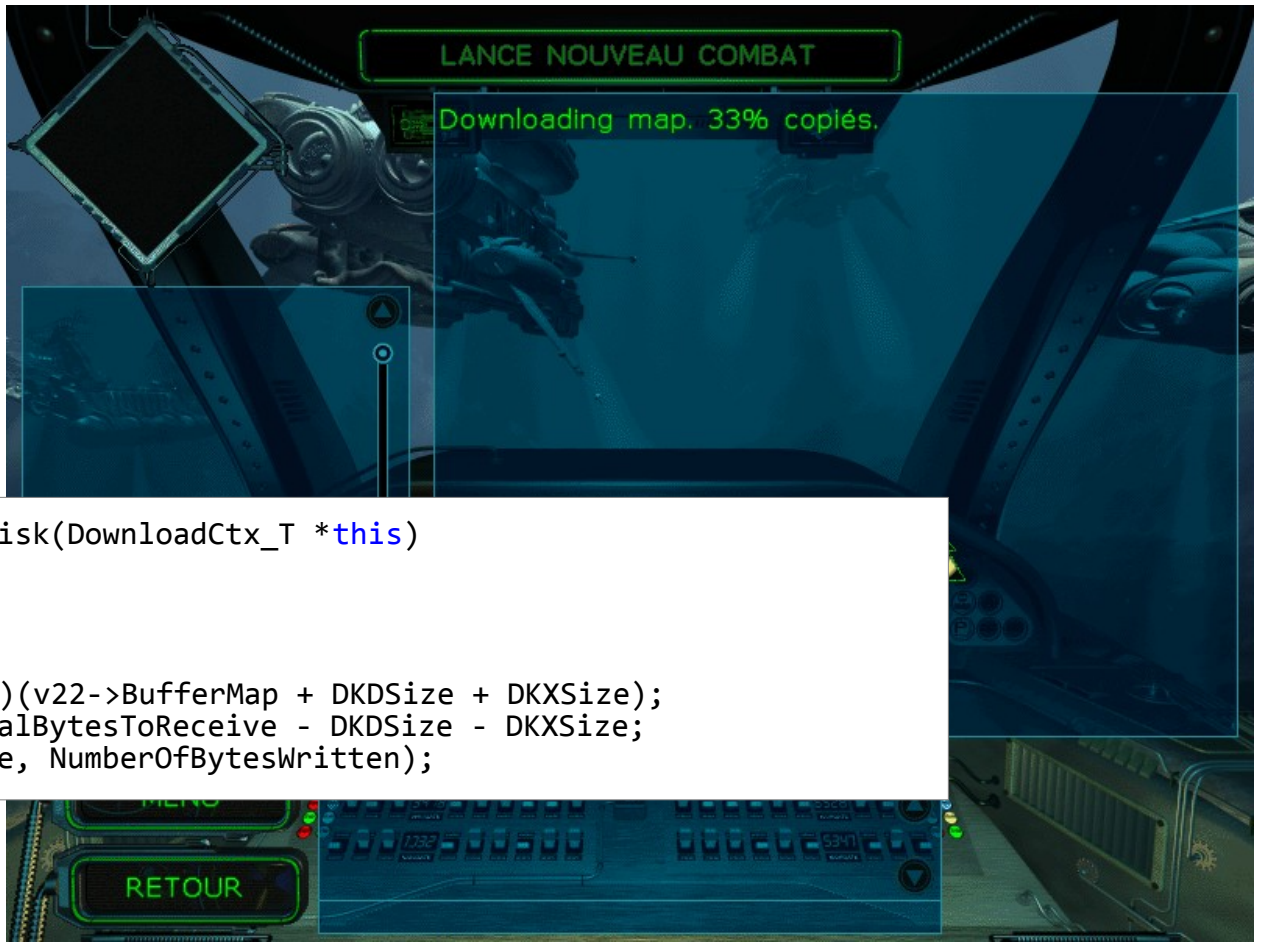
SYNACKTIV



Game assets handling

SYNACKTIV

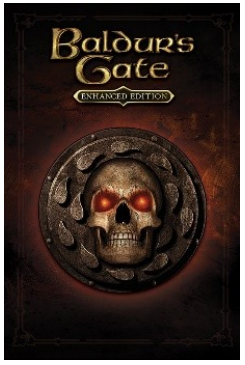


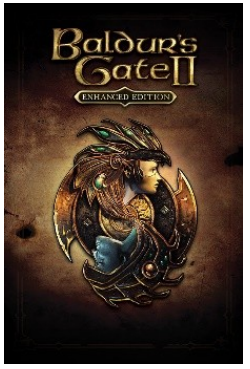


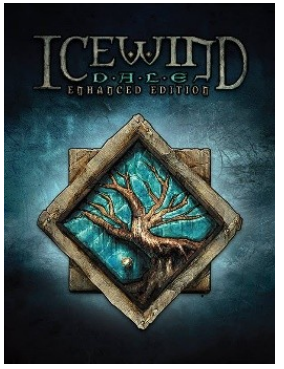

```
signed int __thiscall WriteMapToDisk(DownloadCtx_T *this)
{
[...]
  DKXSize = v22->DKXSize;
  DKDSize = v22->DKDSize;
  MapNameFromFile = (const char *)(v22->BufferMap + DKDSize + DKXSize);
  NumberOfBytesWritten = v22->TotalBytesToReceive - DKDSize - DKXSize;
  strncpy(MapName, MapNameFromFile, NumberOfBytesWritten);
```

Index out of bounds

SYNACKTIV








```
__int64 __fastcall CGamePermission::SetSinglePermission(
        CGamePermission *perms,
        int index,
        char value)
{
  __int64 result; // rax

  result = index;
  perms->m_permissions[index] = value;
  return result;
}
```

Index out of bounds

SYNACKTIV

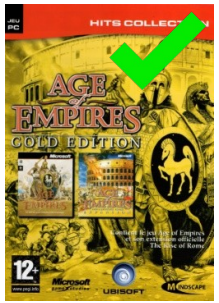
```
__int64 __fastcall CG
        CGamePermissio
        int index,
        char value)
{
  __int64 result; //

  result = index;
  perms->m_permissio
  return result;
}
```

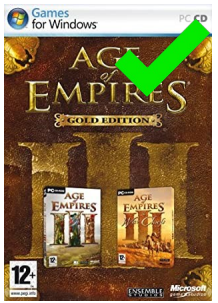
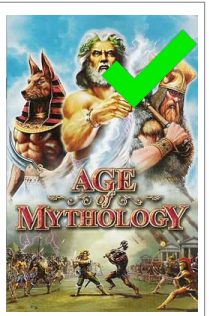
Hunting table

SYNACKTIV

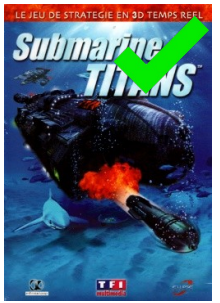
Windows XP

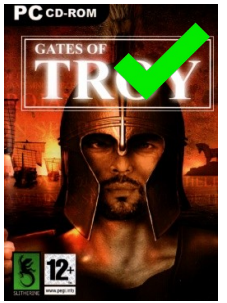


Same bugs

Same bugs



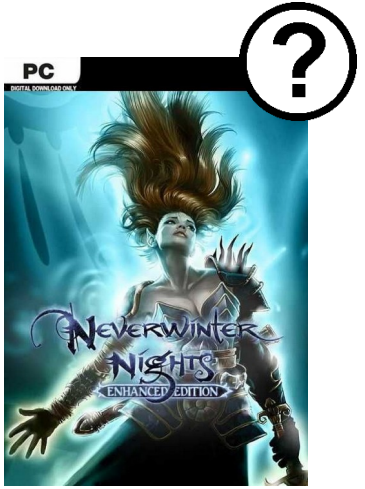


Windows 10 (re-released)




Same bugs


Same bugs



Hunting table

SYNACKTIV

Game	Stack Cookie	DEP	ASLR
Aliens versus Predator 2	No	No	No
Diablo I	No	Yes	Partial
Baldur’s Gates Enhanced Edition	Yes	Yes	No
Baldur’s Gates II Enhanced Edition	Yes	Yes	No
Icewind Dale Enhanced Editions	Yes	Yes	No
American Conquest	No	No	No
Cossacks II	Yes	No	No

Neverwinter Nights

Neverwinter Nights

SYNACKTIV

- RPG
- Developped by BioWare (2002)
- Reedited by Beamdog (2018)
- Available on Steam, GOG
- Aurora Engine
- Multiplayer LAN and Online

Neverwinter Nights: Enhanced Edition

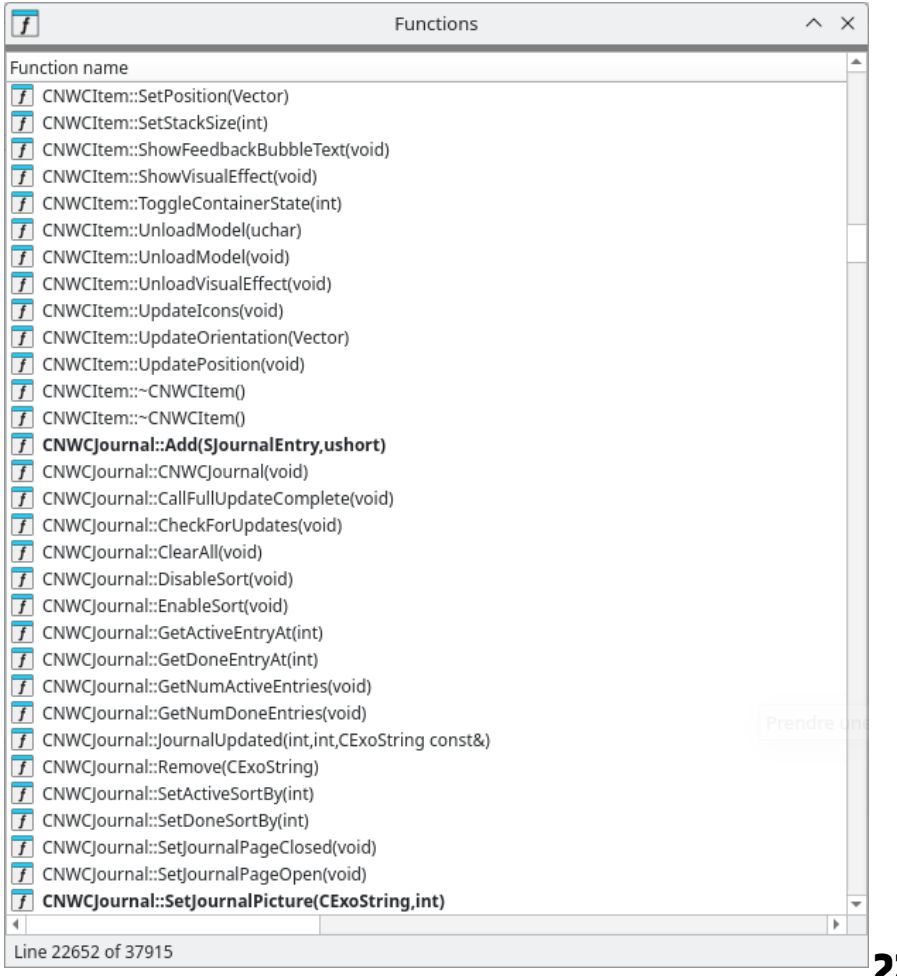
15,99 €

[BEAMDOG]

Informations gathering

SYNACKTIV

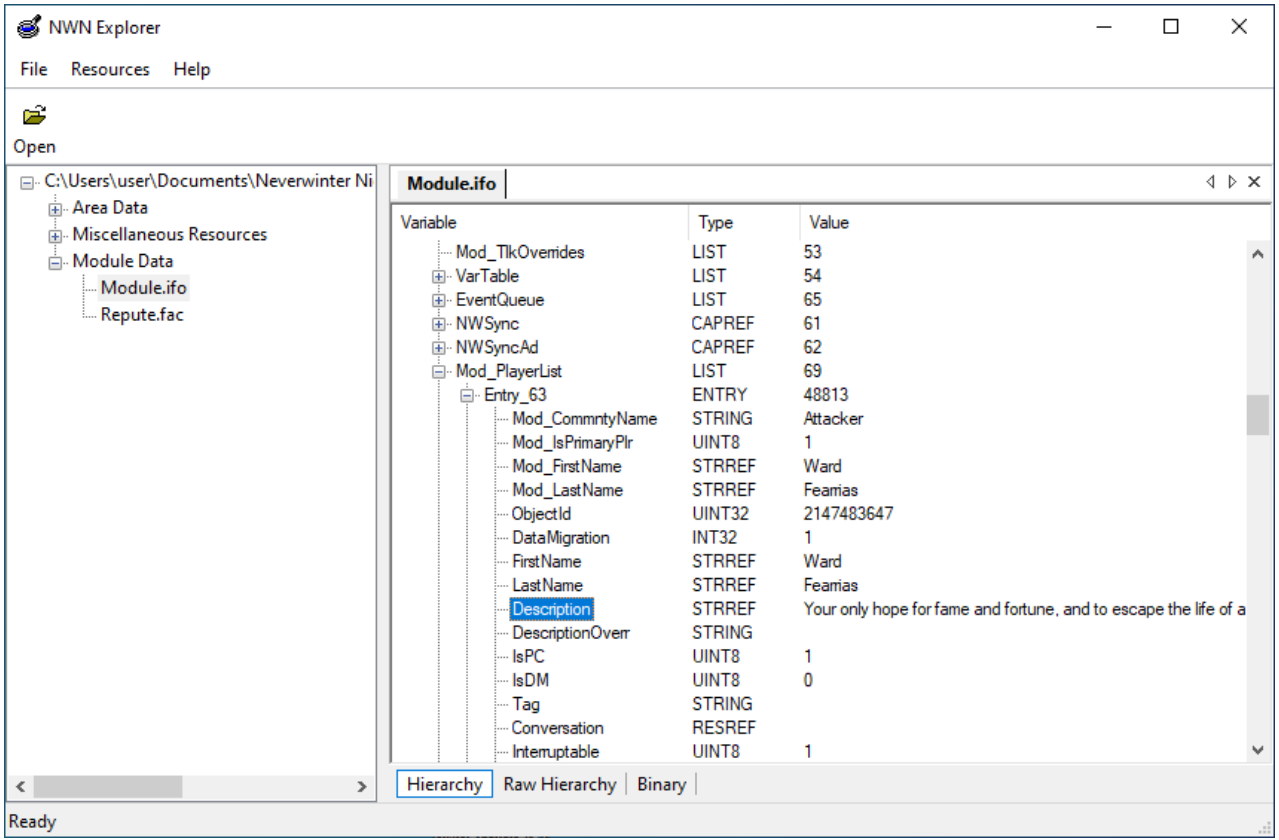
- **Linux version has debug symbols**
- **Modding community**
  - Xoreos (open-source clone of Aurora Engine)
- **https://github.com/Nostritius/nwn-wireshark/**
  - Few information about multiplayer



Tooling

SYNACKTIV

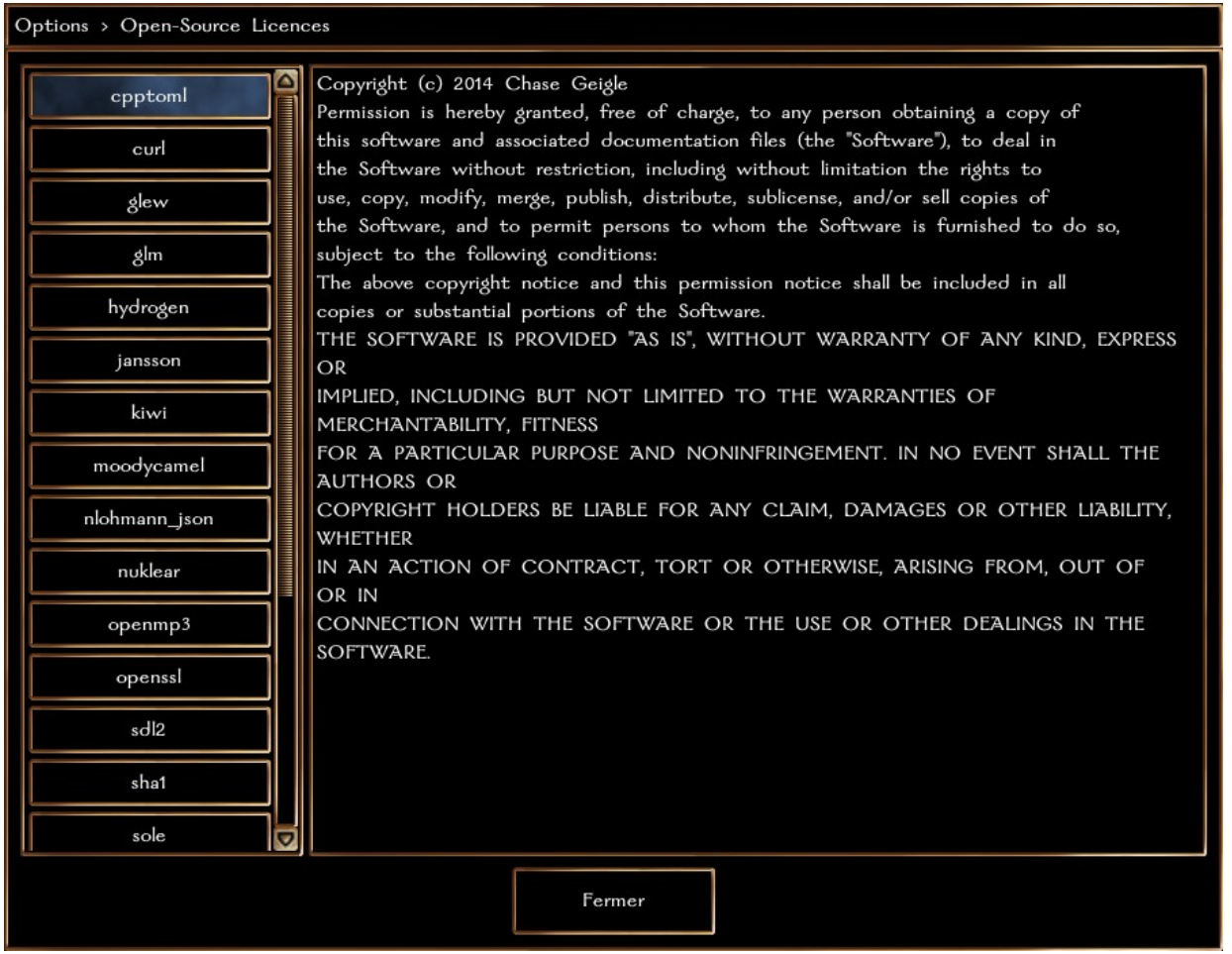
- **Existing tools to inspect**
  - Character file (.BIC)
  - Saved Games (.SAV)
  - NeverwinterNights Modules (.NWM)



Open-source components

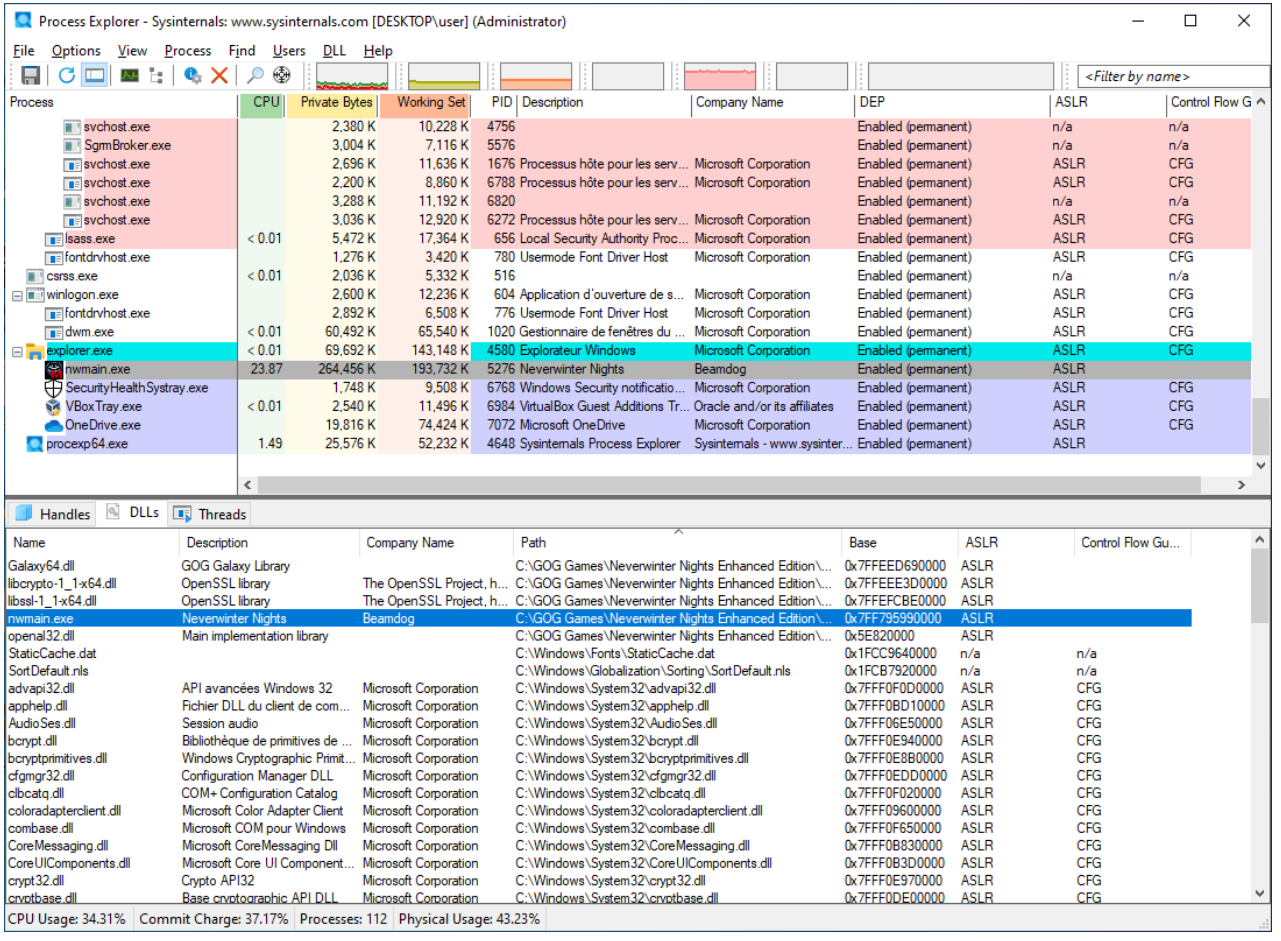
SYNACKTIV

- License file
- Check outaded components



Mitigations

- Stack Cookie
- DEP
- ASLR
- No CFG



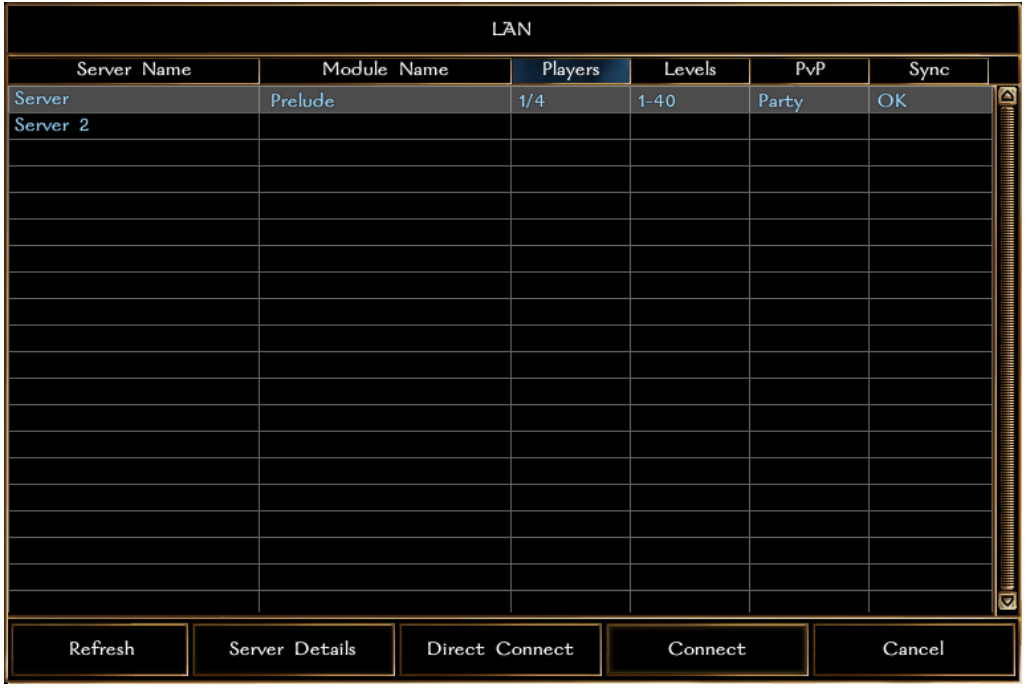
Attack surface

Lobby

SYNACKTIV

- ~19 “Non Window Message”

Type	Description
BNES	Game Search Broadcast
BNER	Game Search (Response)
BNCS	Game CD Key
BNCR	Game CD Key (Response)
BNVS	Password Related
BNVR	Password Related (Response)



Key Exchange

SYNACKTIV

- **Encrypted Communication**
- **Use open-source LibHydrogen**
  - Based on Curve25519 elliptic curve and Gimli permutation
  - Noise Protocol Framework
- **Static keypair derived from CD-Key**

Network stack

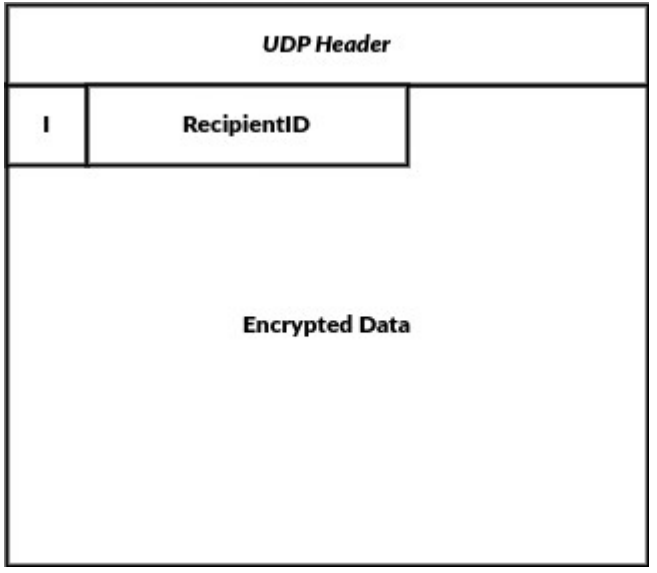
SYNACKTIV

- **Divided in 3 layer**
  - Layer 1 : Handle deciphering
  - Layer 2 : Handle compressed and fragmented data
  - Layer 3 : Handle game message

Layer 1

SYNACKTIV

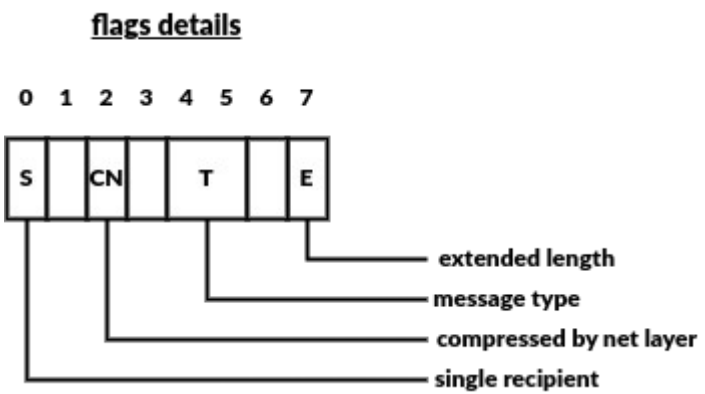
- UDP Based
- 1 magic byte
- 4 random bytes

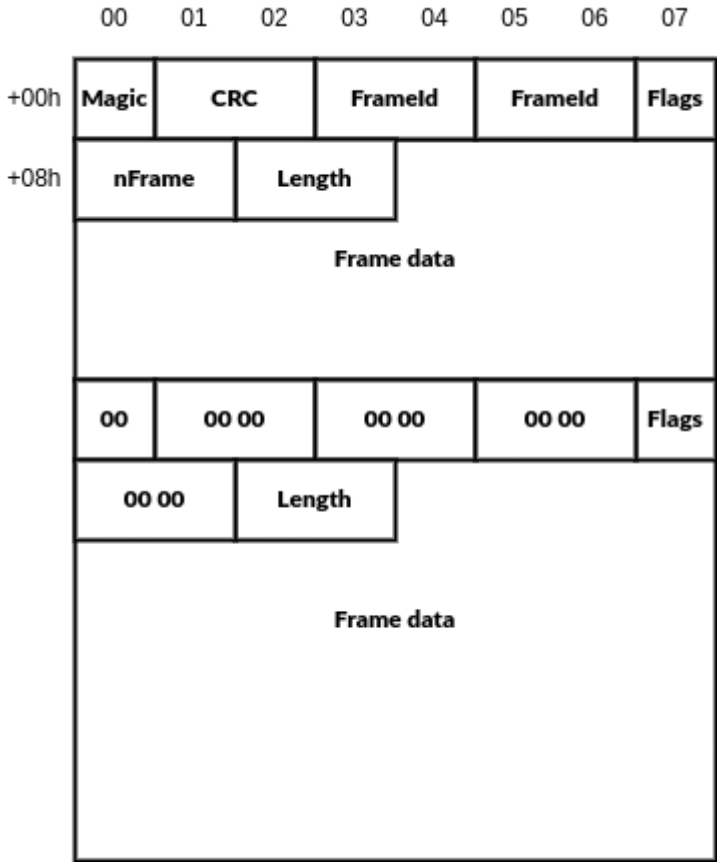


Layer 2

SYNACKTIV

- **3 Frame Types**
  - DATA
  - ACK
  - NAK
- **Fragmentation**
- **Compression**





Layer 3

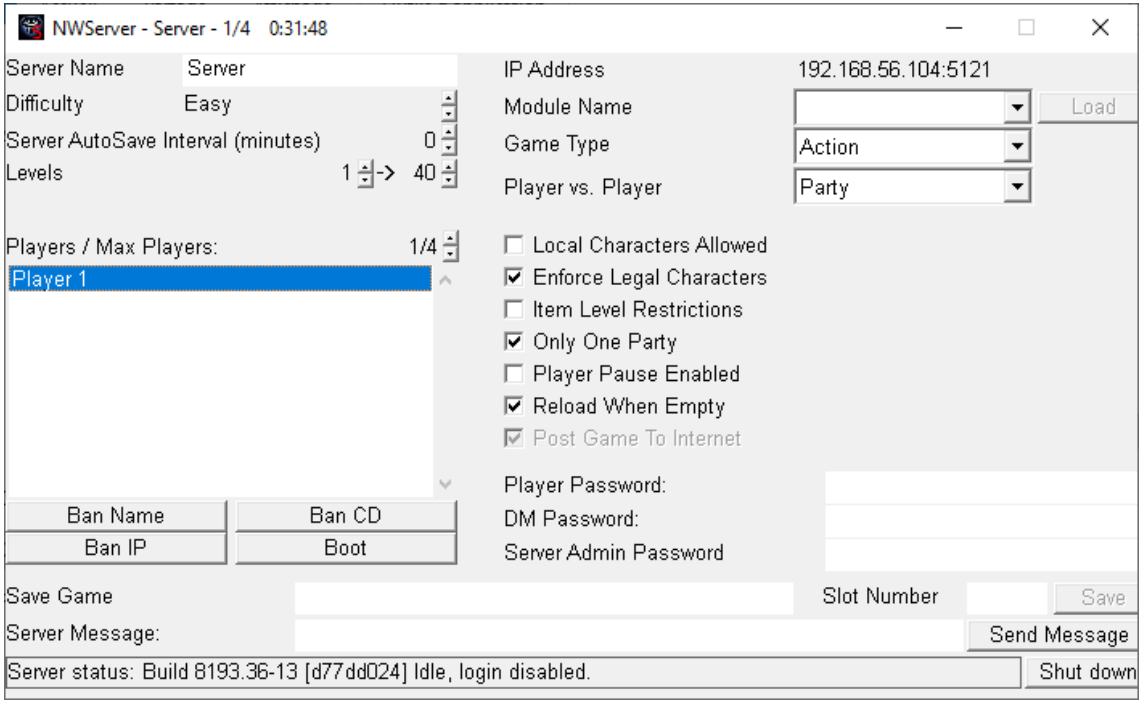
SYNACKTIV

- **4 types of messages**
  - “P” Server to Player
  - “p” Player to Server
  - “S” Server to SysAdmin
  - “s” SysAdmin to Server

SysAdmin Message

SYNACKTIV

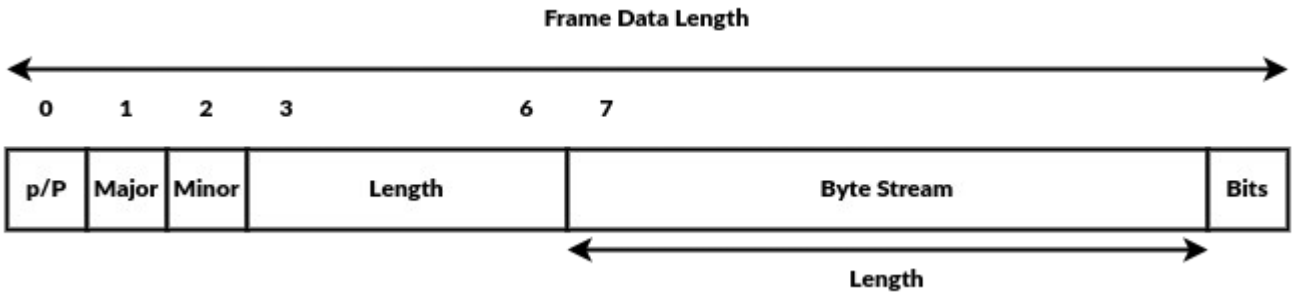
- Only between Server and the “Administrator”
- Textual command “Control.Boot 1”



Player/Server messages

SYNACKTIV

- **Game related message**
  - Update creature appearance
  - Start a progress bar
  - Add a quest in journal
  - Level up
  - …
- **~250 messages types**



Message Parsing

SYNACKTIV

- **CNWMessage helps to decode byte stream**
- **21 simple types**
  - CNWMessage::ReadBOOL
  - CNWMessage::ReadBYTE
  - CNWMessage::ReadINT
  - CNWMessage::ReadDOUBLE
  - CNWMessage::ReadVOIDPtr
  - CNWMessage::ReadCExoString
  - …

Vulnerabilities

First Bug

SYNACKTIV

```
__int64 __fastcall CNWCMessage::HandleServerToPlayerLogin(CNWMessage *this, char Minor) {
[...]
int Class[8];       // [rsp+F0h] [rbp-18h] BYREF
char ClassLevel[8]; // [rsp+110h] [rbp+8h] BYREF
[...]
switch(Minor)
{
      […]
      case 10:
            ClassListSize = CNWMessage::ReadBYTE(this, 8);
            _ClasListSize = ClassListSize;
            if ( ClassListSize )
            {
                  _ClassLevel = ClassLevel;
                  _Class = Class;
                  n = ClassListSize;
                  do
                  {
                        // stack buffer overflow
                        *_Class = CNWMessage::ReadINT(this, 32);
                        *_ClassLevel = CNWMessage::ReadBYTE(this, 8);
                        ++_Class;
                        ++_ClassLevel;
                        --n;
                  }
                  while ( n );
            }
            Experience = CNWMessage::ReadDWORD(this, 32);
            [...]
            CPanelCharVersionPopup::SetSaveCharacterInfo(v13, _ClassListSize, Class, ClassLevel, Experience);
```

First Bug

SYNACKTIV

- Classic stack buffer overflow
- count 0..255
- Inexploitable due to stack cookie …

Second bug

SYNACKTIV

```
__int64 __fastcall CNWCMessage::HandleServerToPlayerCreatureUpdate_Appearance(CNWMessage *this)
{
[…]
 unsigned __int16 Buf[18]; // [rsp+216h] [rbp-11Ah] BYREF
[…]
    Count = CNWMessage::ReadBYTE(this, 8);
    _Count = Count;
    if ( Count )
    {
      v53 = (int *)Buf;
      if ( Count <= 9u )
      {
        CNWCCreatureAppearance::GetPartVariations(
          *((CNWCCreatureAppearance **)CreatureByGameObjectID + 102),
          (unsigned __int8 *)Buf,
;
        n = 0;
        while ( 1 )
        {
          index = CNWMessage::ReadBYTE(this, 8);
          if ( CNWMessage::MessageReadOverflow(this) )
            goto LABEL_82;
          if ( _bVersionSup_8193_35 )
            value = CNWMessage::ReadWORD(this, 16);
          else
            value = (unsigned __int8)CNWMessage::ReadBYTE(this, 8);
          ++n;
          Buf[index] = value;
          if ( _Count == n )
            goto LABEL_130;
        }
      }
```

Second bug

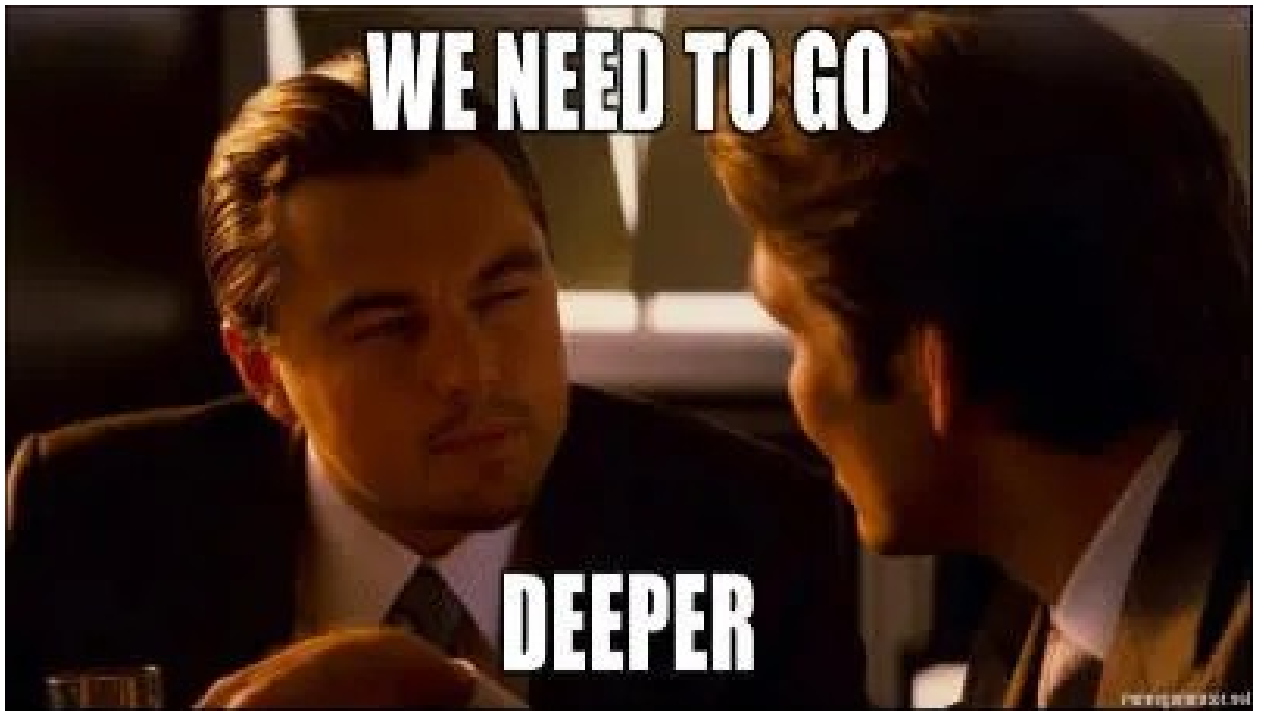
SYNACKTIV

- Out of bound write in stack
- 10 word write
- Enough to rewrite return address
- Need a leak …

Find a leak

SYNACKTIV

- By design server doesn't need to query information about client …



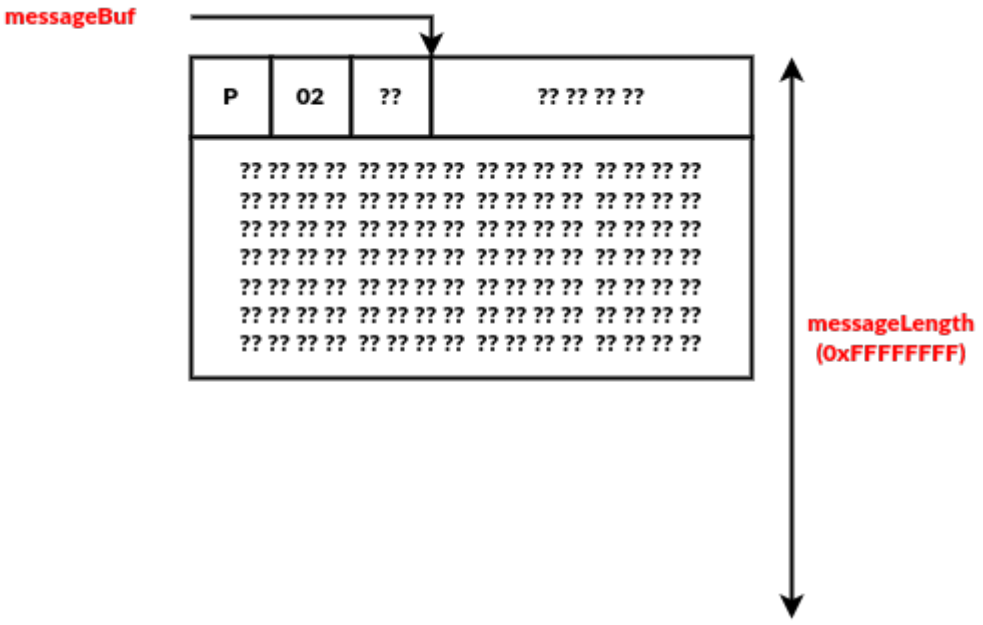
Integer underflow bug

SYNACKTIV

```
__int64 __fastcall CNWCMessage::HandleServerToPlayerMessage(CNWMessage *this, char *Buf, int Len)
{
  […]
  Magic = *Buf;
  Major = Buf[1];
  Minor = Buf[2];
  if ( CNWMessage::SetReadMessage(this, Buf + 3, Len - 3, -1, 1)
    && (v8 = g_pAppManager->CClientExoApp->vtable->CClientExoApp::GetNetLayer)(g_pAppManager->CClientExoApp),
        CNetLayer::GetClientConnected(v8))
    && !CNWMessage::MessageReadOverflow(this)
    && Magic == 'P' )
  {
    CNetworkProfiler::AddMessageToProfile((const void **)g_cNetworkProfiler, 82, Major, Buf[2], Len);
    CExoString::Format(&a1, "unknown Major (0x%.2X)", Major);
    switch ( Major )
    {
      case 1u:
        CExoString::operator=(&a1, "ServerStatus");
        active = CNWCMessage::HandleServerToPlayerServerStatus(this, Minor);
        goto LABEL_9;
      case 2u:
        CExoString::operator=(&a1, "Login");
        active = CNWCMessage::HandleServerToPlayerLogin(this, Minor);
        goto LABEL_9;
```

Integer underflow bug

SYNACKTIV

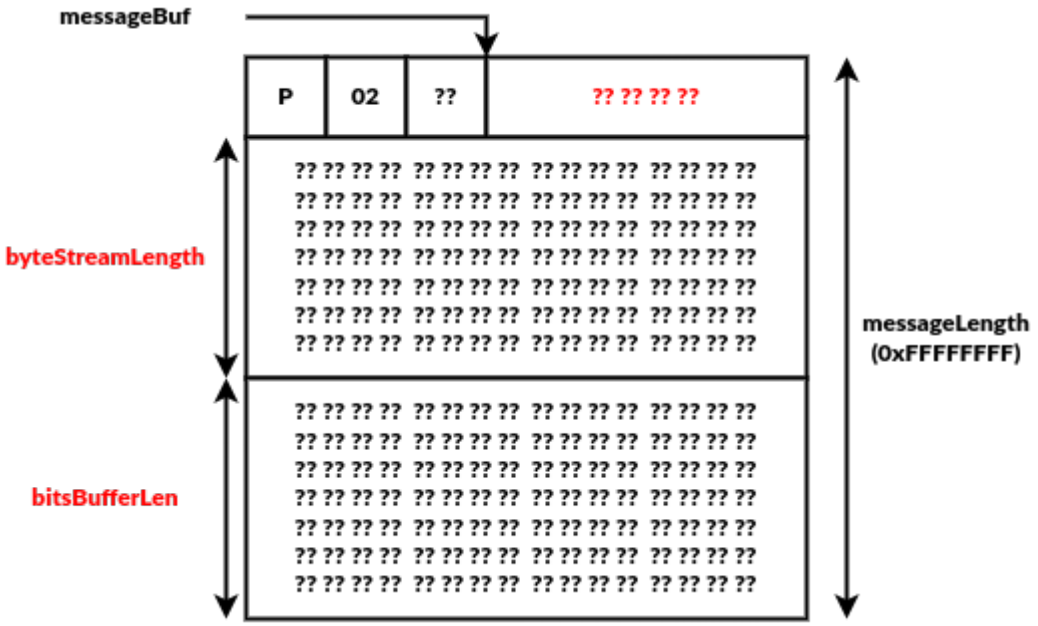



```
unsigned int __fastcall CNWMessage::SetReadMessage(
        CNWMessage *this,
        unsigned __int8 *messageBuf,
        unsigned int Length,
        int a4,
        int a5)
{
  unsigned int bytesStreamLength; // ecx
  unsigned int res; // eax

  this->messageBuf = messageBuf;
  this->messageLength = Length;
  this->curPos = 0;
  [...]
  if ( Length )
  {
    this->curPos = 4;
    bytesStreamLength = *(_DWORD *)messageBuf - 3;
    res = 0;
    this->bitsBufferOffset = bytesStreamLength;
    if ( bytesStreamLength < Length )
    {
      this->bitsBuffer = &messageBuf[bytesStreamLength];
      this->bitsBufferLen = Length - bytesStreamLength;
      this->bitsPos = 0;
      this->messageLength = bytesStreamLength;
      this->nEncodedBits = CNWMessage::ReadBYTE(this, 3);
      return 1;
    }
  }
  else
  {
  […]
```

Integer underflow bug

SYNACKTIV




```
unsigned int __fastcall CNWMessage::SetReadMessage(
        CNWMessage *this,
        unsigned __int8 *messageBuf,
        unsigned int Length,
        int a4,
        int a5)
{
  unsigned int bytesStreamLength; // ecx
  unsigned int res; // eax

  this->messageBuf = messageBuf;
  this->messageLength = Length;
  this->curPos = 0;
  [...]
  if ( Length )
  {
    this->curPos = 4;
    bytesStreamLength = *(_DWORD *)messageBuf - 3;
    res = 0;
    this->bitsBufferOffset = bytesStreamLength;
    if ( bytesStreamLength < Length )
    {
      this->bitsBuffer = &messageBuf[bytesStreamLength];
      this->bitsBufferLen = Length - bytesStreamLength;
      this->bitsPos = 0;
      this->messageLength = bytesStreamLength;
      this->nEncodedBits = CNWMessage::ReadBYTE(this, 3);
      return 1;
    }
  }
  else
  {
  […]
```

Transform into OOBR

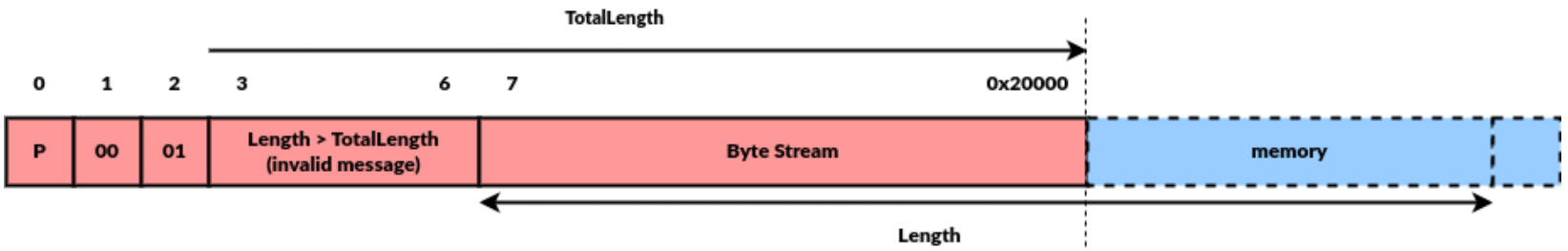
SYNACKTIV

- **Find a way to have the same memory buffer used between two message**
  - Frame can contains compressed data
  - CNetLayerInternal::UncompressMessage use a temporary buffer
    - Dynamically allocated when uncompressed size is >= 0x20000
    - else *rx_buffer* member is used
    - *rx_buffer* is not erased after use

Transform into OOBR

SYNACKTIV

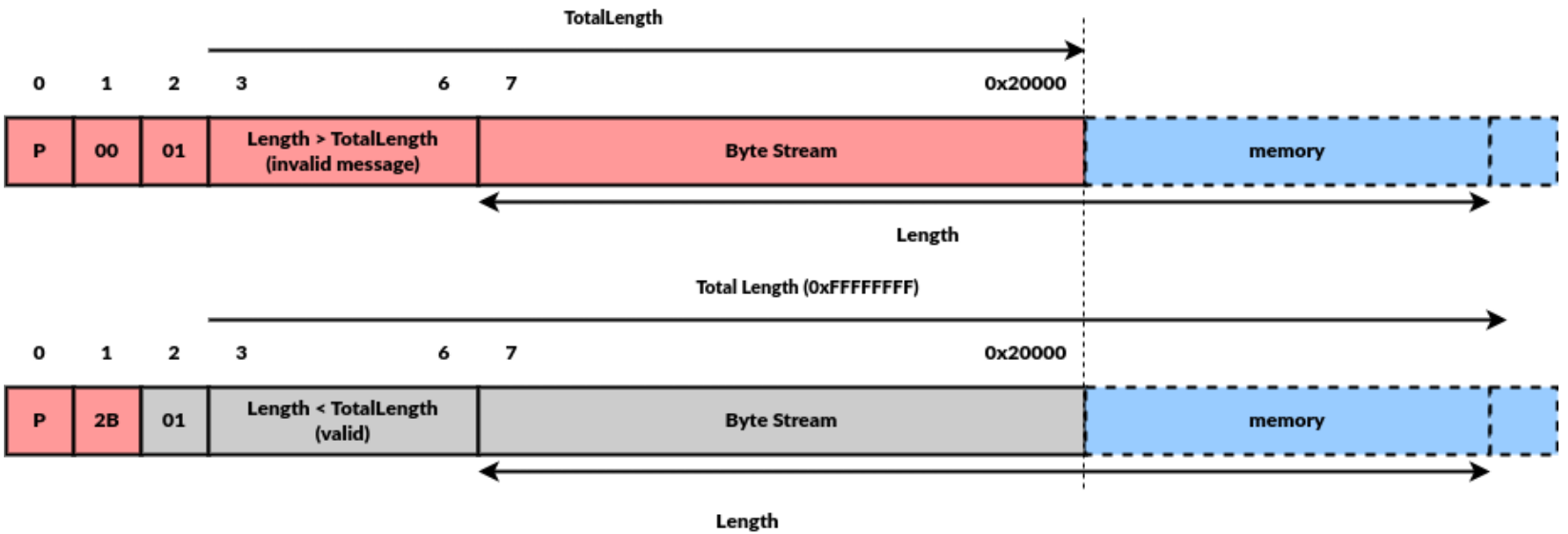
- Send first invalid message to initialise buffer



Transform into OOBR

SYNACKTIV

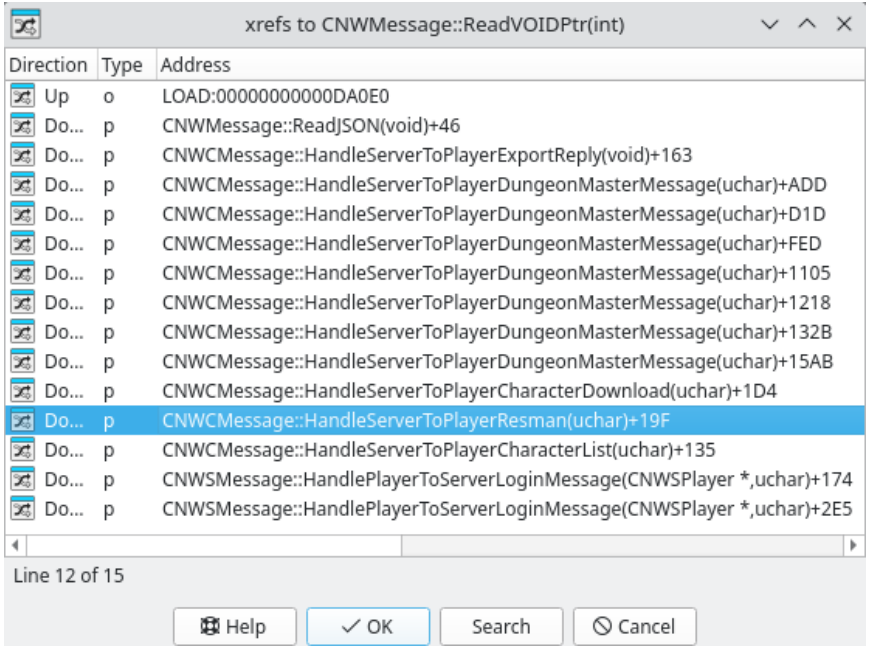
- Send first invalid message to initialise buffer
- Send second message of 2 bytes



Transform into OOBR

SYNACKTIV

- **CNWMessage::ReadVOIDPtr**
  - return a pointer to buffer of arbitrary size
- **Usage of CNWMessage::ReadVOIDPtr**



Transform into OOBR

SYNACKTIV

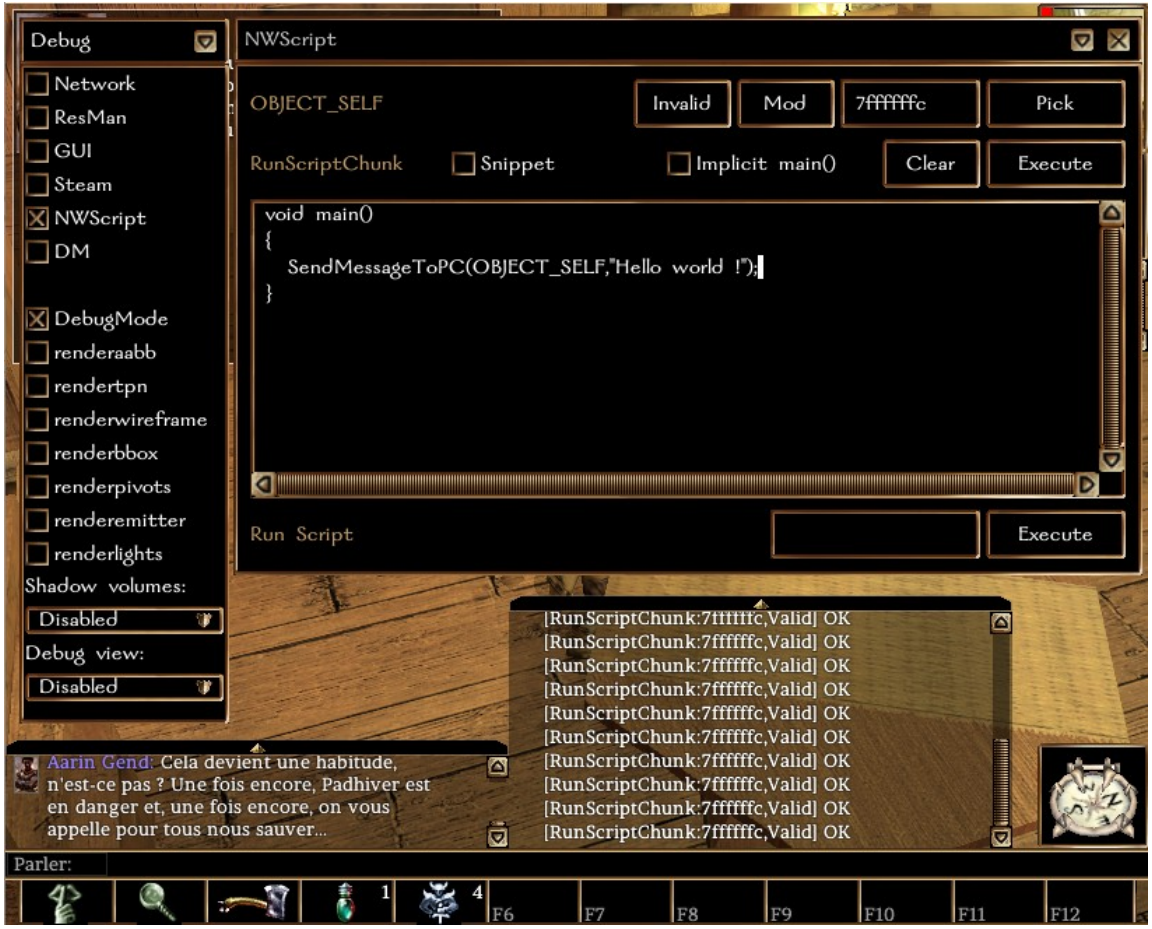
- Upload assets to client
- Write some heap data into a client file
- No way to request file :(
- Give up …

```
__int64 __fastcall CNWCMessage::HandleServerToPlayerResman(
      CNWMessage *this,
      char minor)
{
[…]
    switch ( minor )
    {
[…]
case 5:
        CNWMessage::ReadCResRef((CResRef *)v34, this, 16);
        CExoString::CExoString(&v29, (const CResRef *)v34);
        v27 = CNWMessage::ReadSHORT(this, 16);
        Length = CNWMessage::ReadDWORD(this, 32);
        VOIDPtr = CNWMessage::ReadVOIDPtr(this, Length);
        if ( CNWMessage::MessageReadOverflow(this) )
          goto LABEL_32;
        CExoString::StripNonAlphaNumeric(&v29, 1, 0, 0);
        v7 = CExoString::CStr(&v29);
        CExoString::CExoString(&v30);
        CExoString::Format(&v30, "TEMPCLIENT:%s", v7);
        if ( Length )
        {
          CExoString::CExoString(&v32, "wb");
          CExoFile::CExoFile((CExoFile *)&v31, &v30, v27, &v32);
          [...]
          CExoFile::Write((CExoFile *)&v31, VOIDPtr, 1u, Length);
          CExoFile::~CExoFile((#204 *)&v31);
        }
```

Neverwinter Night Script

SYNACKTIV

- ***Weird symbol name RunScriptChunk***
- **Virtual Machine stack-based**
- **Server can execute arbitrary script send from player**
  - Only in DebugMode …



Portal feature

SYNACKTIV

- Found when looking NWScript surface
- Well-documented : https://nwnlexicon.com/index.php/ActivatePortal

Send a player's client to a new server, where the player's character will log in.

```
void ActivatePortal(
    object oTarget,
    string sIPaddress = "",
    string sPassword = "",
    string sWaypointTag = "",
    int bSeamless = FALSE
);
```

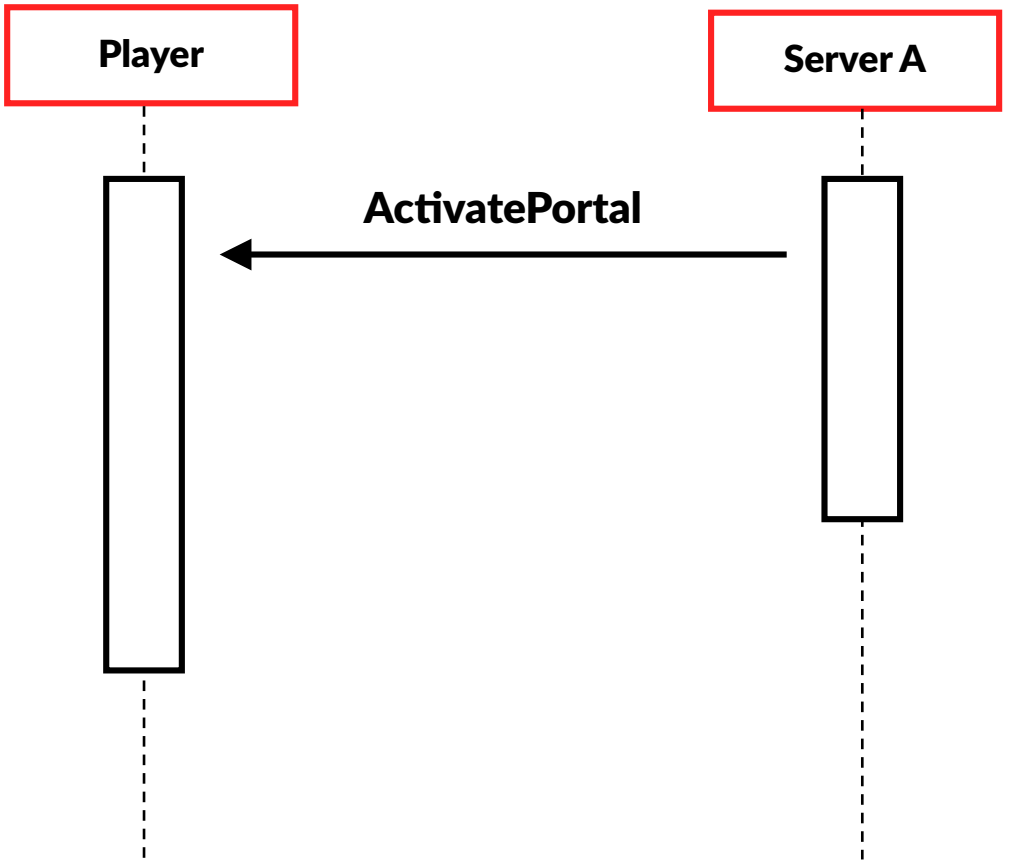
Portal feature

SYNACKTIV

- **CNWCMessage::HandleServerToPlayerPortalActivatePortal**
  - Major/Minor 2A/01
  - Work with DebugMode disabled
  - bSeamless => Automatic transfert without dialog box

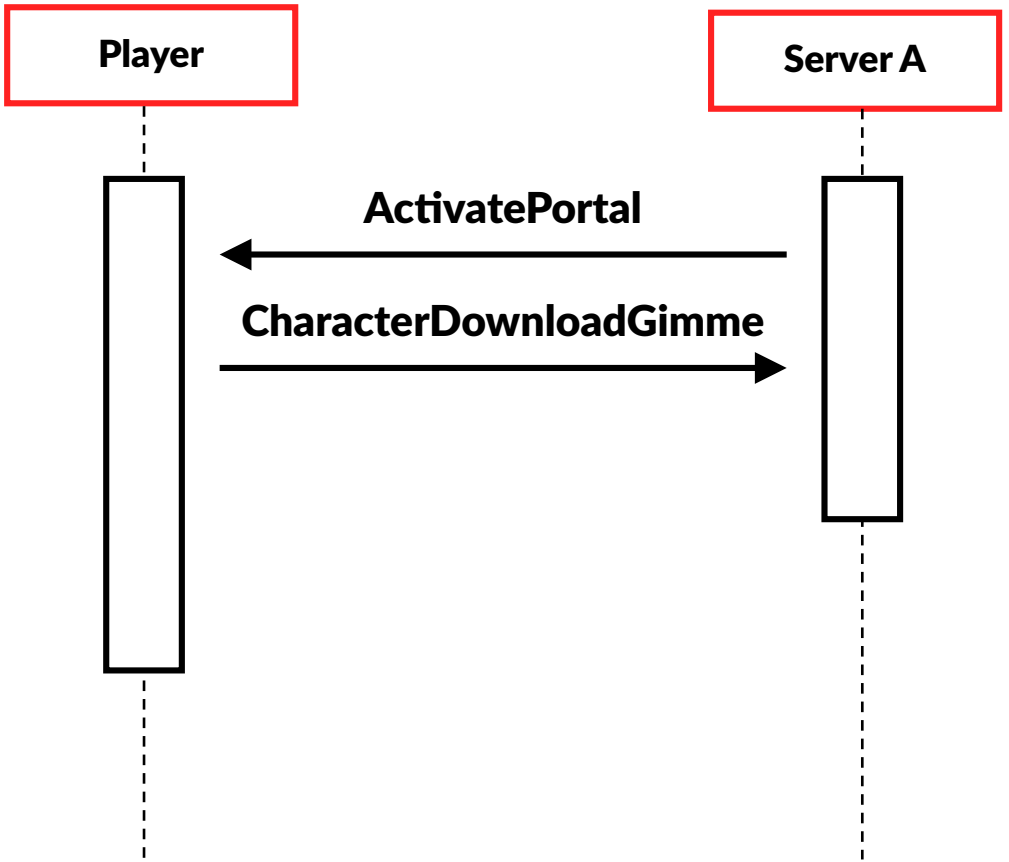
Portal feature

SYNACKTIV



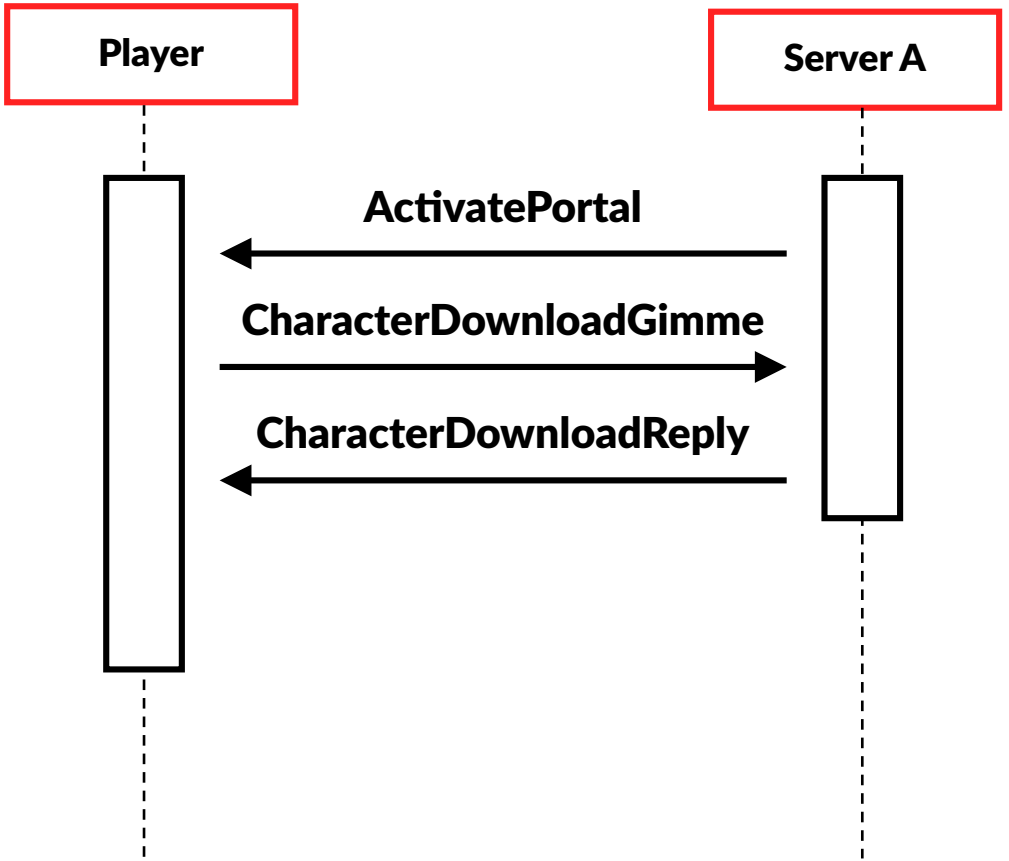
Portal feature

SYNACKTIV



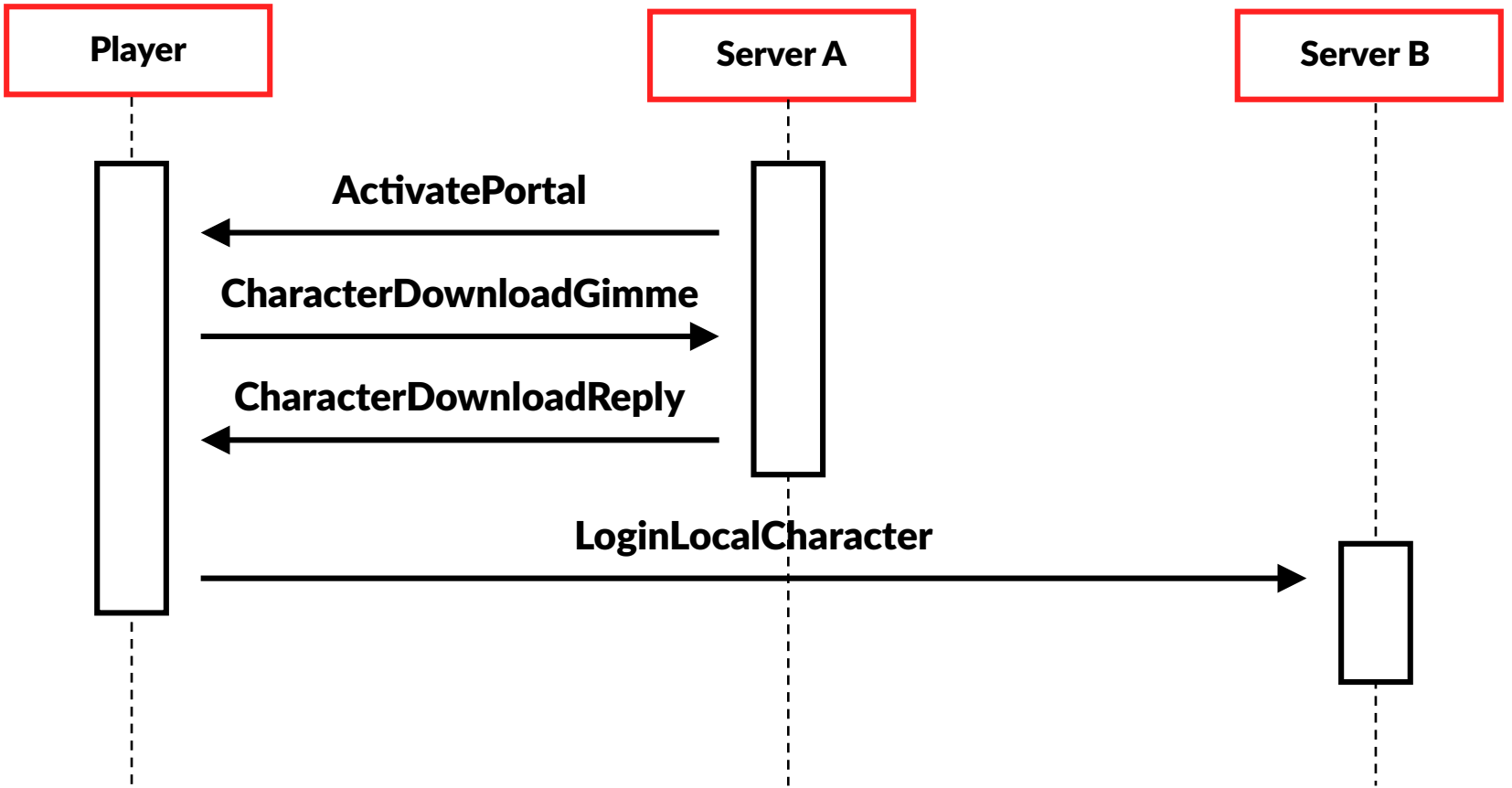
Portal feature

SYNACKTIV



Portal feature

SYNACKTIV



Exploitation

Leak

SYNACKTIV

- Activate Portal
- Use integer underflow bug to trigger OOBR during *SetCharacterFile*
- Second server received leaked heap memory
- Search pattern in memory to find program base

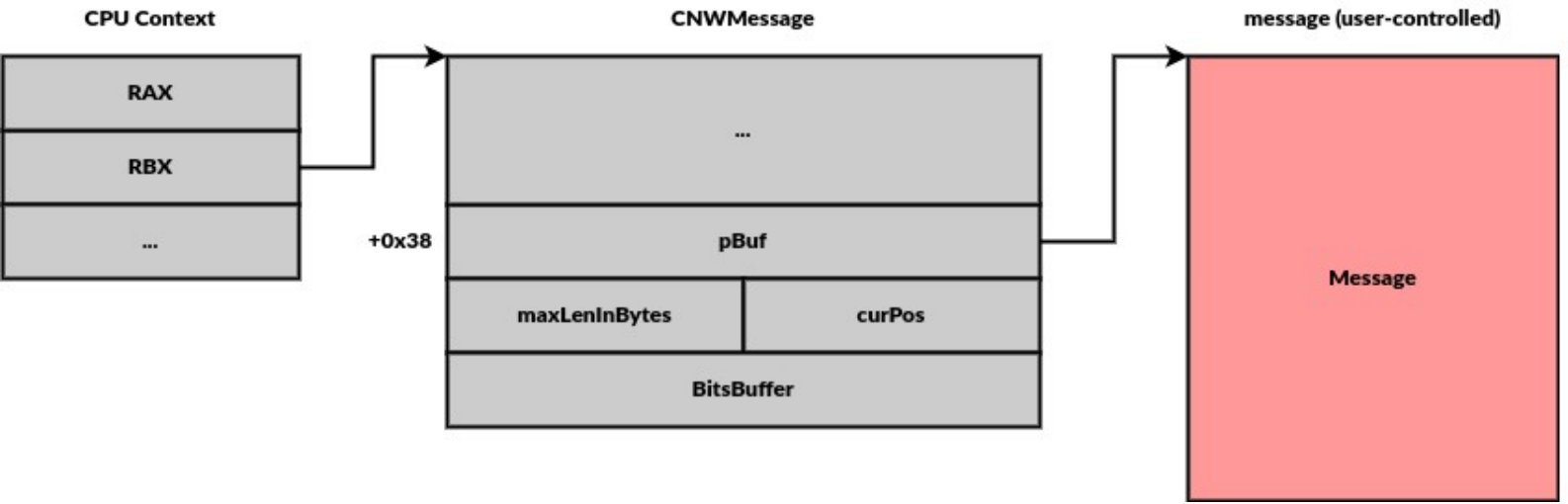
```
__int64 __fastcall CNWCMessage::HandleServerToPlayerCharacterDownload(CNWMessage *this, char Minor)
{
  [...]
  if ( Minor == 2 )
  {
    length = CNWMessage::ReadDWORD(this, 32);
    pointer = (unsigned __int8 *)CNWMessage::ReadVOIDPtr(this, length);
    CClientExoApp::SetCharacterFile(g_pAppManager->CClientExoApp, length, pointer, 0);
```

Pivot

SYNACKTIV

```
mov     rcx, [rbx+38h]
mov     rdi, [rcx+28h]
call    qword ptr [rdi+18h]
```

- OOBW to rewrite return address
- RBX points to CWNMessage
- Limited call to existing vtable functions

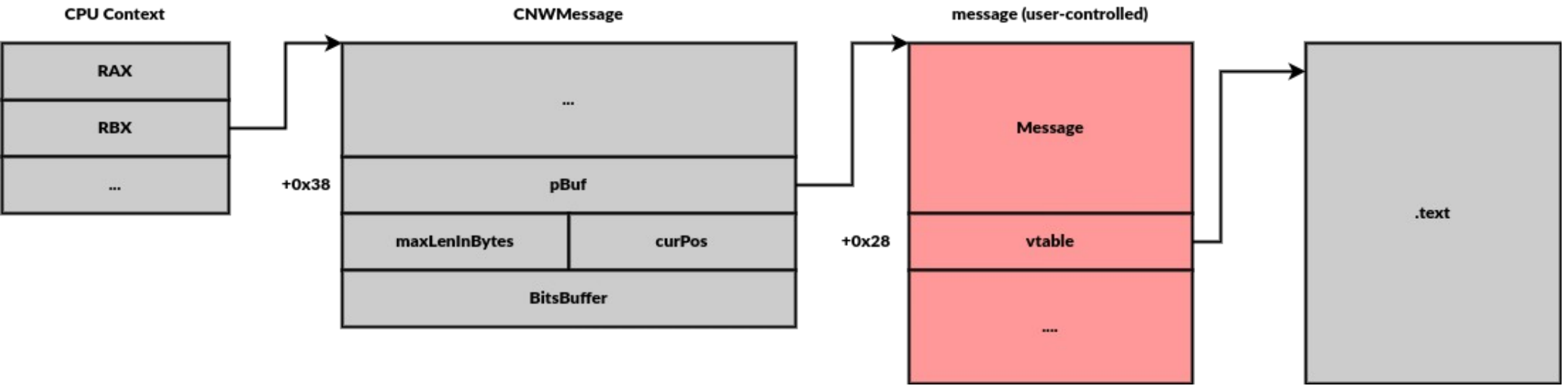


Pivot

SYNACKTIV

```
mov     rcx, [rbx+38h]
mov     rdi, [rcx+28h]
call    qword ptr [rdi+18h]
```

- OOBW to rewrite return address
- RBX points to CWNMessage
- Limited call to existing vtable functions

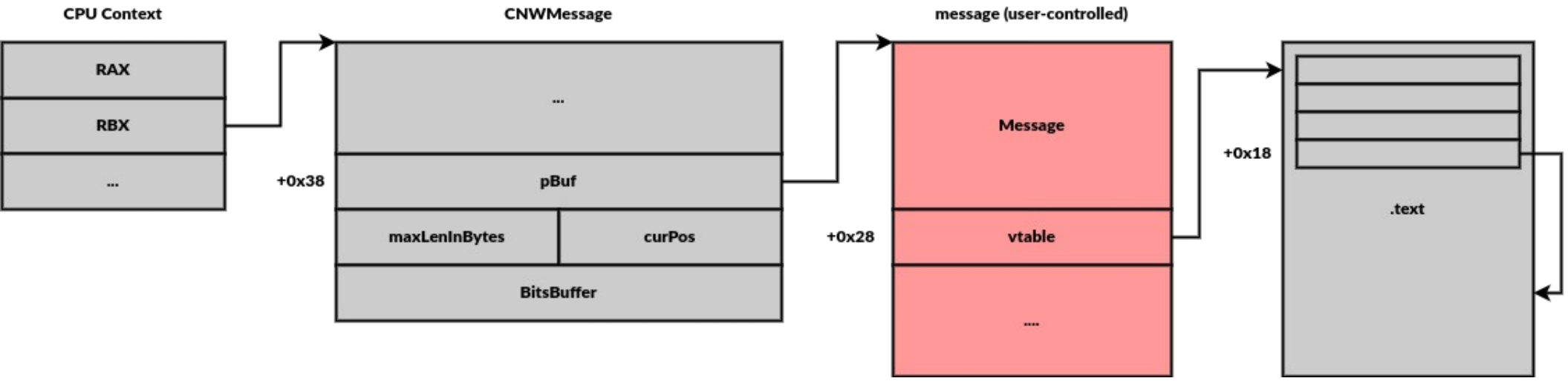


Pivot

SYNACKTIV

```
mov     rcx, [rbx+38h]
mov     rdi, [rcx+28h]
call    qword ptr [rdi+18h]
```

- OOBW to rewrite return address
- RBX points to CWNMessage
- Limited call to existing vtable functions



Pivot

SYNACKTIV

- **Goal : Limited call to arbitrary call**
- **Reuse “Steam gadget” (found in previous research)**
  - CCallResult<CSteamInternal,CreateItemResult_t>::Run
- **RCX points to buffer (due to previous gadget)**
- **JMP anywhere**

```
.text:00000001401D8850          mov     rax, rcx
.text:00000001401D8853          mov     qword ptr [rcx+10h], 0
.text:00000001401D885B          mov     rcx, [rcx+18h]
.text:00000001401D885F          xor     r8d, r8d
.text:00000001401D8862          jmp     qword ptr [rax+20h]
```

Pivot

SYNACKTIV

- **Goal : Start ROPChain placed in message**
- **ROPChain will execute calc.exe**
- **COP/JOP Chain**
  - RAX points to CWNMessage buffer
  - Move RAX into RSP with 3 gadgets

```
push rax ; mov rcx, rbx ; call qword ptr [rax + 0x48]
pop rdi ; jmp qword ptr [rax + 0x40]
pop rsp ; and al, 0x50 ; add rsp, 0x58 ; ret
```

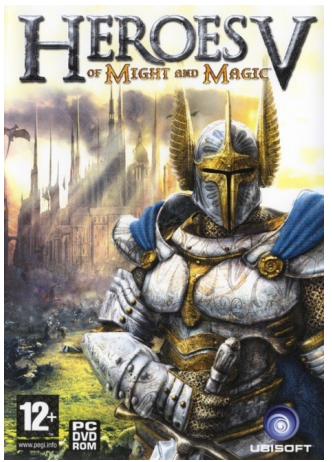
Demo

SYNACKTIV

Conclusion

SYNACKTIV

- Modding community provides great ressources for security researchers
- Devil hides in the details
- Intel CET will kill exploitation
- Next …



SYNACKTIV

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

https://twitter.com/synacktiv

https://synacktiv.com