



Tale of an RCE in a video game Neverwinter Nights

04/10/2024

Agenda

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

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

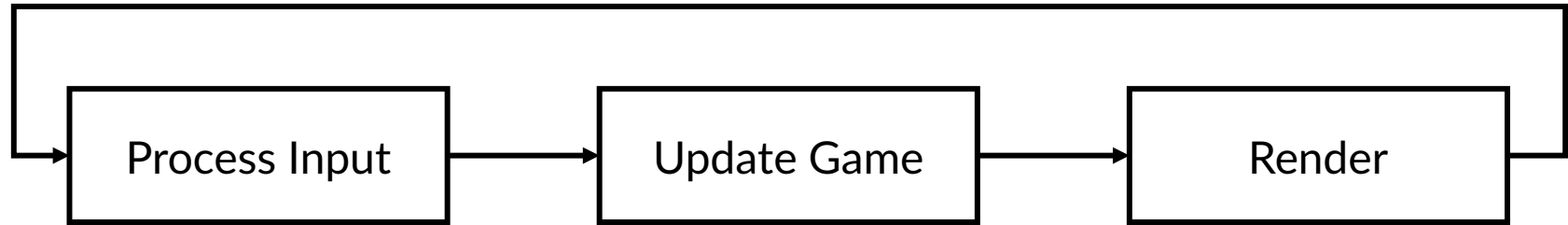




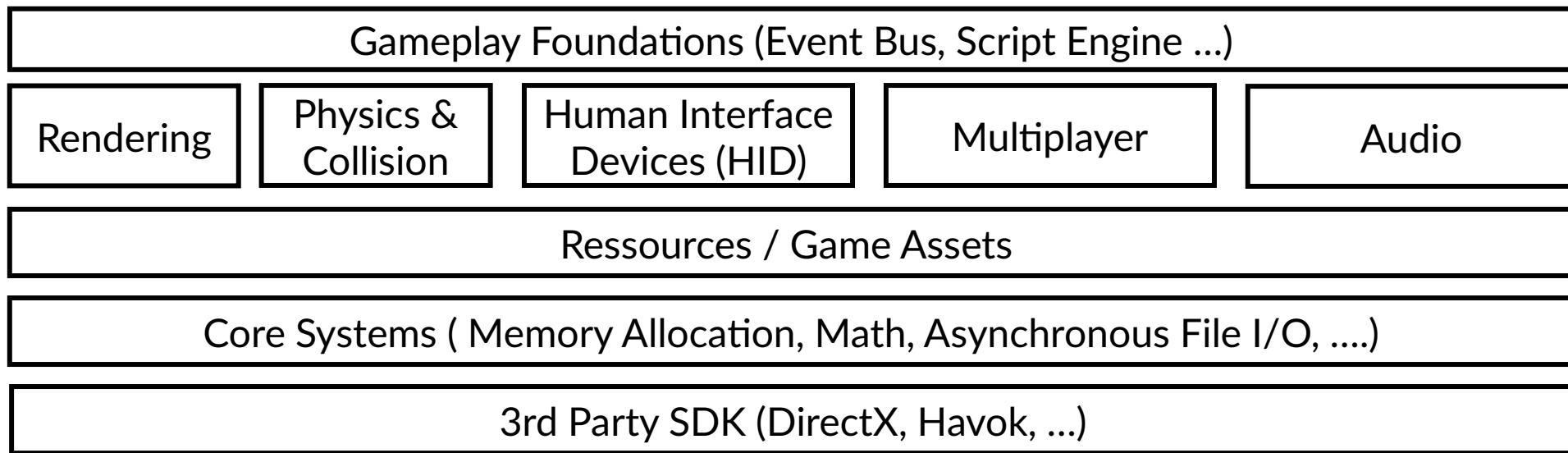
- Why look for vulnerabilities in old video games ?
 - Bug bounty
 - To have fun
 - To recycle my old video game collections
 - Interesting when old games are re-released
 - There are always bugs, but sometimes complicated to exploit
 - Training
- Focus on RCE (no cheating technique)

Concepts

What is video game ?

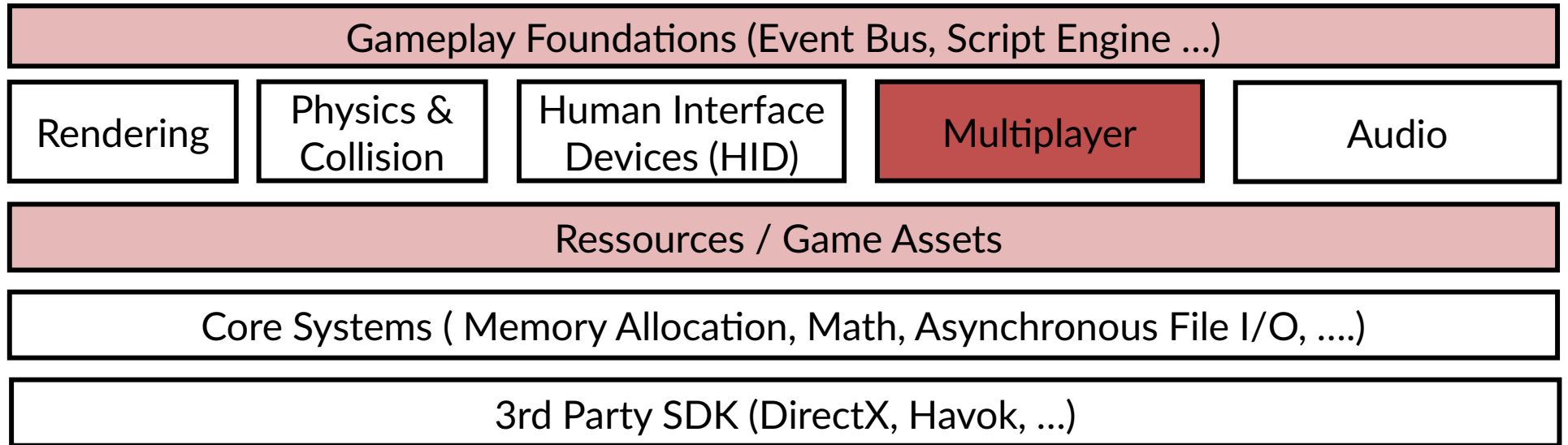


What is game engine ?



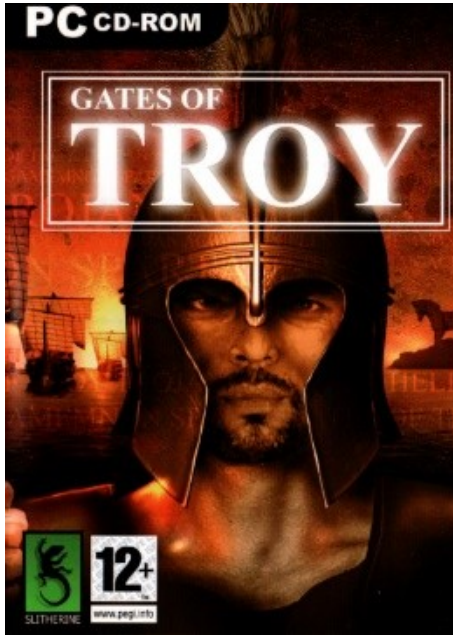
Source: <https://www.gameenginebook.com/>
« Game Engine Architecture » by Jason Gregory

Where to focus ?



Old games old bugs

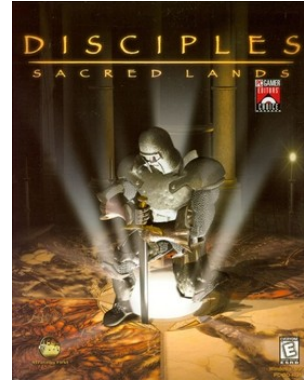
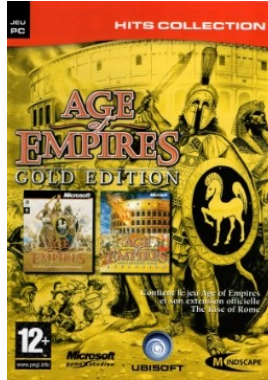
Stack buffer overflow



```
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;
        memcpy(buf, ::buf, bytes_recv);
    }
}
```

Stack buffer overflow



```
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



```
int  
{  
  cons  
  int  
  int  
  char  
  [...]  
  Num  
  Num  
  Sess  
  spr
```

);

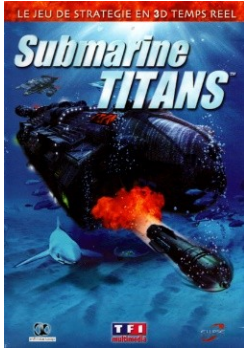
Stack buffer overflow



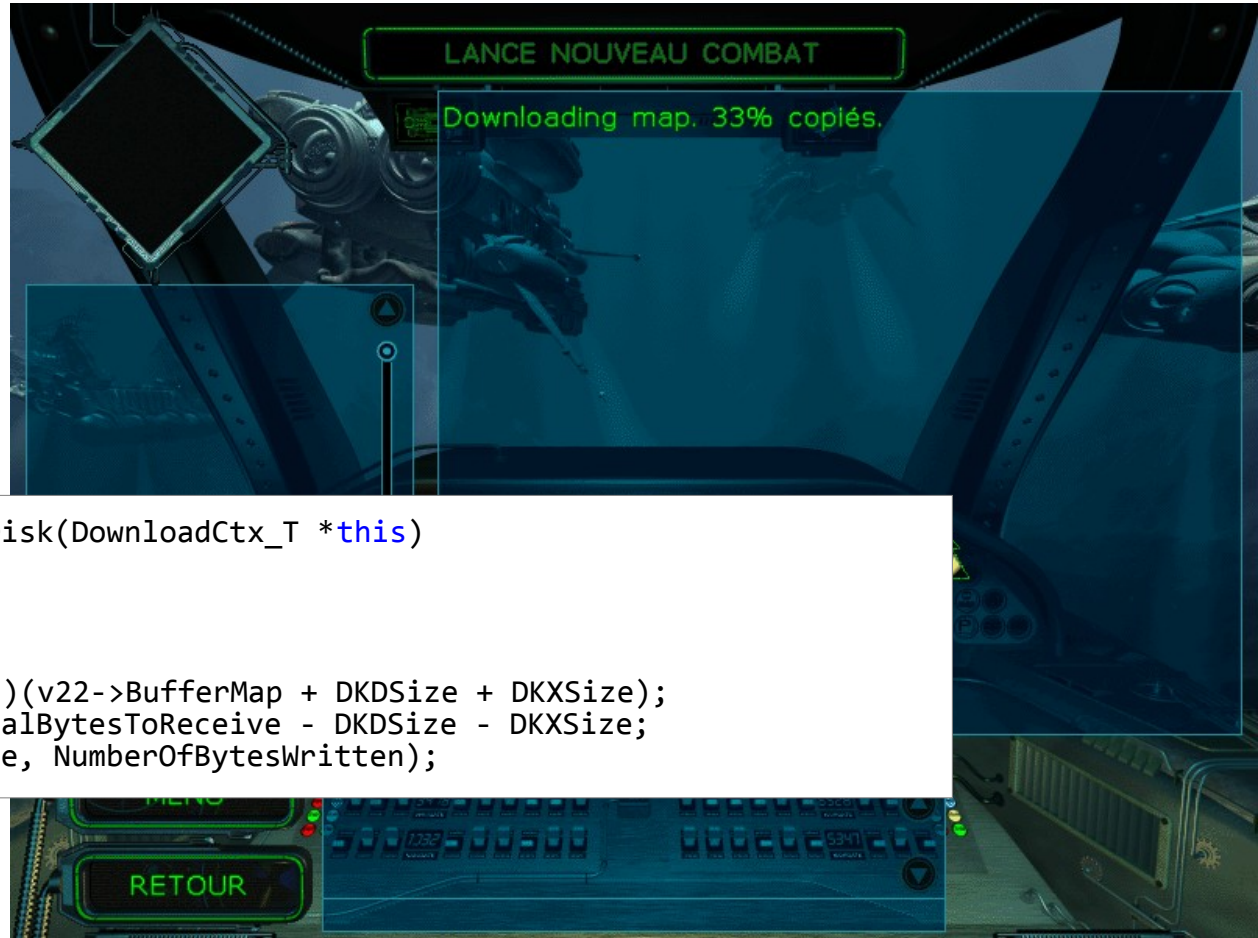
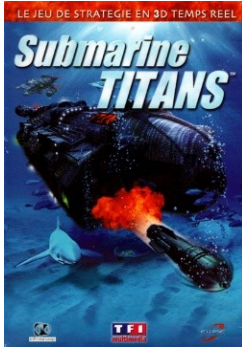
```
int  
{  
  cons  
  int  
  int  
  char  
  [...  
  Numb  
  Numb  
  Sess  
  spri
```

ax);

Game assets handling



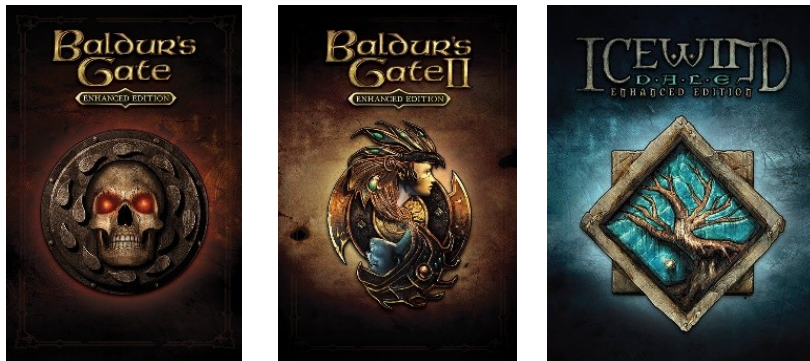
Game assets handling



```
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



```
__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

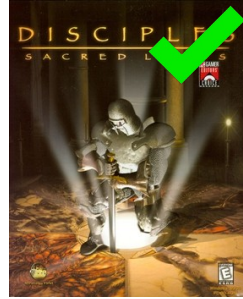
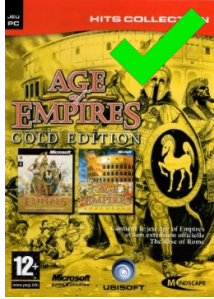
```
__int64 __fastcall CGamePermissions::SetPermission(
    int index,
    char value)
{
    __int64 result; // [Return Value]

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

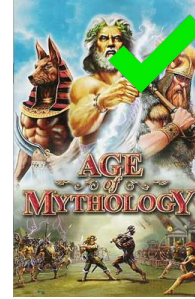
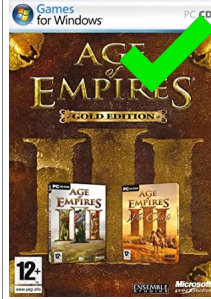


Hunting table

Windows XP



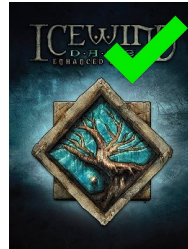
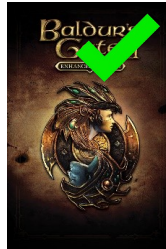
Same bugs



Same bugs



Windows 10 (re-released)



Same bugs



Same bugs



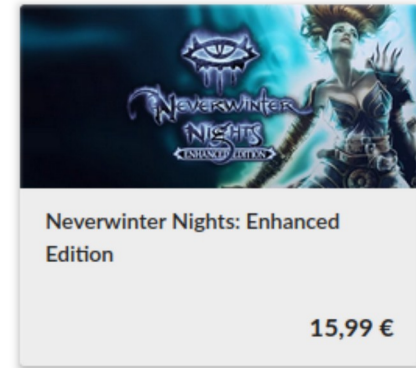
Hunting table

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

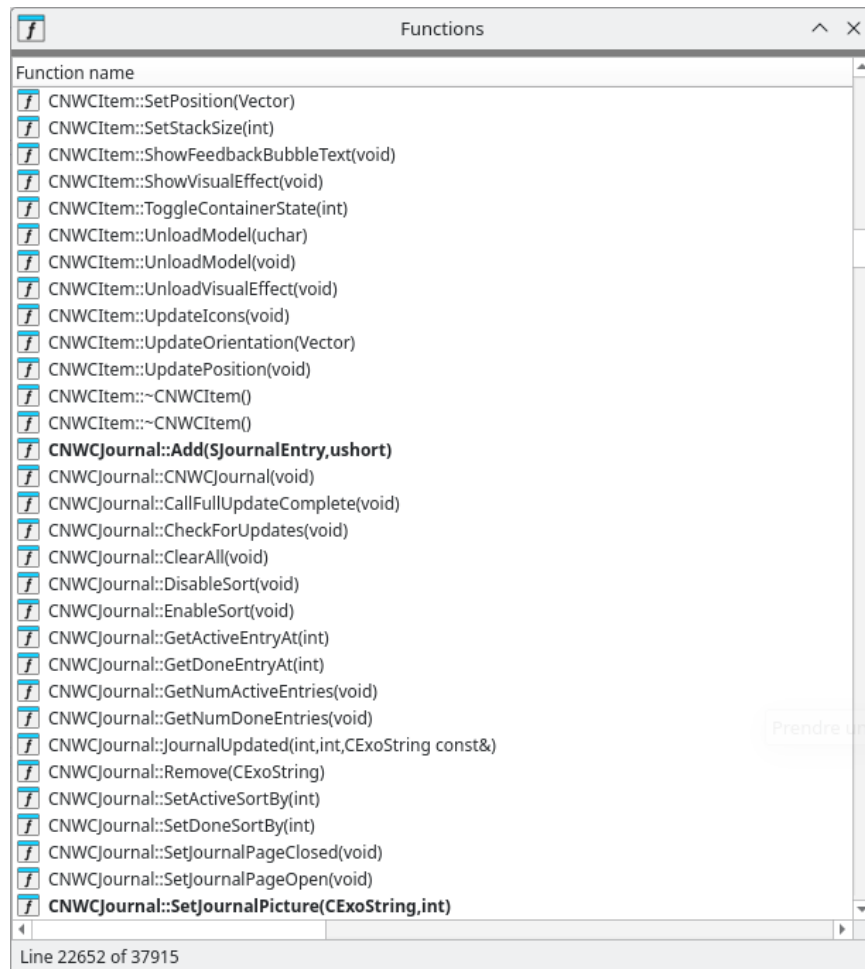
- **RPG**
- **Developped by BioWare (2002)**
- **Reedited by Beamdog (2018)**
- **Available on Steam, GOG**
- **Aurora Engine**
- **Multiplayer LAN and Online**



[Beamdog]

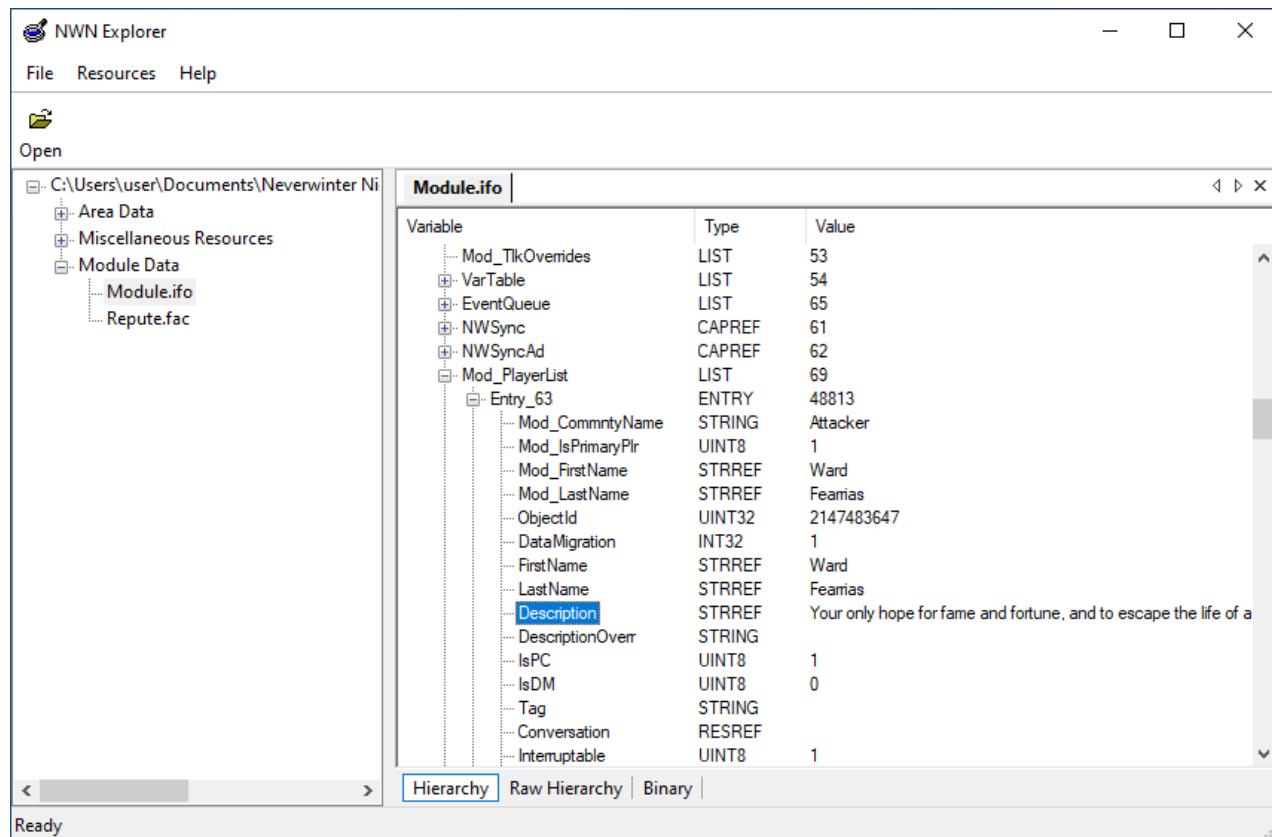
Informations gathering

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



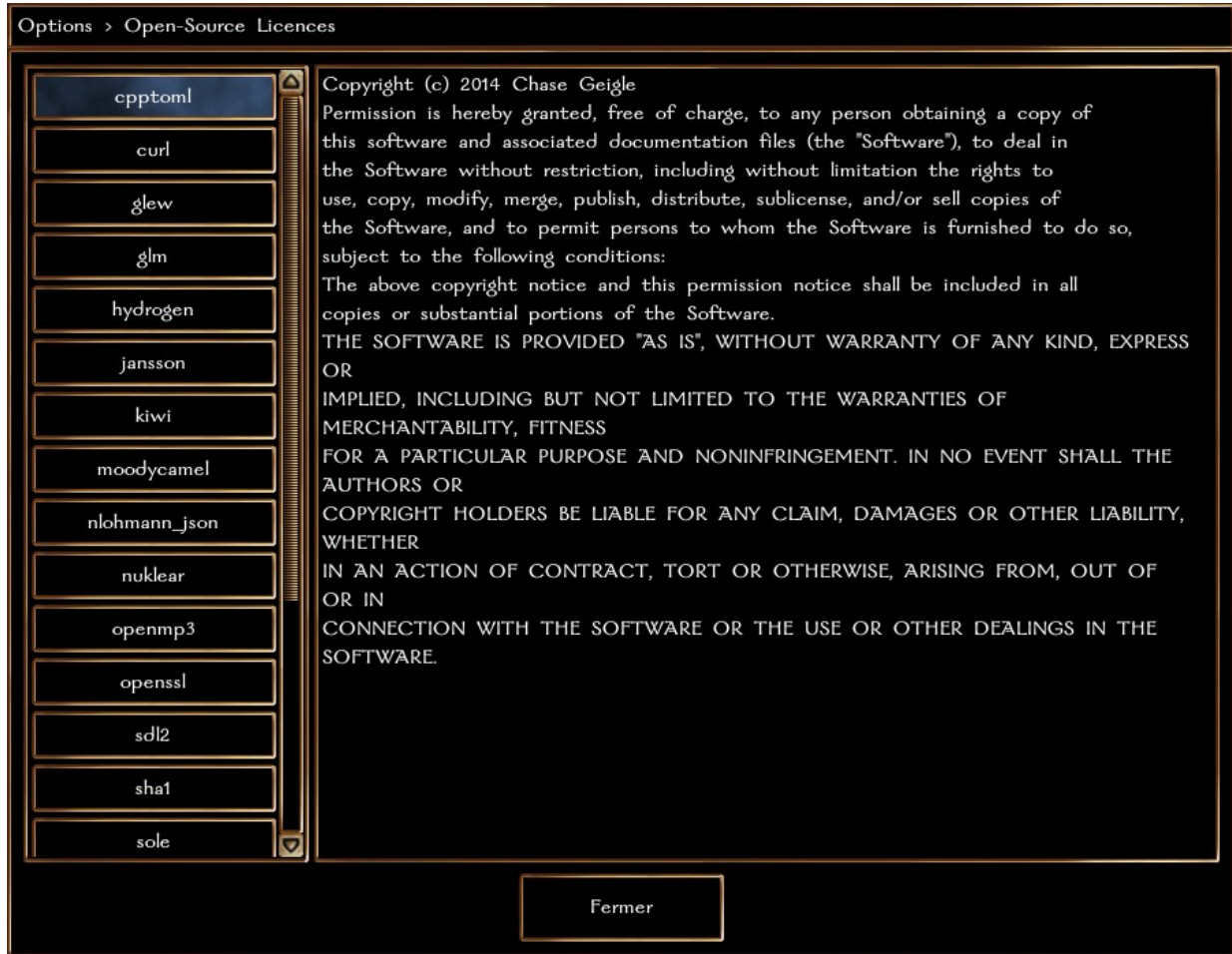
■ Existing tools to inspect

- Character file (.BIC)
- Saved Games (.SAV)
- NeverwinterNights Modules (.NWM)



Open-source components

- License file
- Check outdated components



Mitigations

- Stack Cookie
- DEP
- ASLR
- No CFG

Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP\user] (Administrator)

File Options View Process Find Users DLL Help

<Filter by name>

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	DEP	ASLR	Control Flow G
svchost.exe		2,380 K	10,228 K	4756			Enabled (permanent)	n/a	n/a
SgmBroker.exe		3,004 K	7,116 K	5576			Enabled (permanent)	n/a	n/a
svchost.exe		2,696 K	11,636 K	1676	Processus hôte pour les serv...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
svchost.exe		2,200 K	8,860 K	6788	Processus hôte pour les serv...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
svchost.exe		3,288 K	11,192 K	6820			Enabled (permanent)	n/a	n/a
svchost.exe		3,036 K	12,920 K	6272	Processus hôte pour les serv...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
lsass.exe	< 0.01	5,472 K	17,364 K	656	Local Security Authority Proc...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
fontdrvhost.exe	< 0.01	1,276 K	3,420 K	780	Usemode Font Driver Host	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
csrss.exe	< 0.01	2,036 K	5,332 K	516			Enabled (permanent)	n/a	n/a
winlogon.exe		2,600 K	12,236 K	604	Application d'ouverture de s...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
fontdrvhost.exe		2,892 K	6,508 K	776	Usemode Font Driver Host	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
dwm.exe	< 0.01	60,492 K	65,540 K	1020	Gestionnaire de fenêtres du ...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
explorer.exe	< 0.01	69,692 K	143,148 K	4580	Explorateur Windows	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
nwmain.exe	23.87	264,456 K	193,732 K	5276	Neverwinter Nights	Beamdog	Enabled (permanent)	ASLR	
SecurityHealthSystray.exe	< 0.01	1,748 K	9,508 K	6768	Windows Security notificatio...	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
VBoxTray.exe	< 0.01	2,540 K	11,496 K	6984	VirtualBox Guest Additions Tr...	Oracle and/or its affiliates	Enabled (permanent)	ASLR	CFG
OneDrive.exe		19,816 K	74,424 K	7072	Microsoft OneDrive	Microsoft Corporation	Enabled (permanent)	ASLR	CFG
procexp64.exe	1.49	25,576 K	52,232 K	4648	Sysinternals Process Explorer	Sysinternals - www.sysinter...	Enabled (permanent)	ASLR	

Handles DLLs Threads

Name	Description	Company Name	Path	Base	ASLR	Control Flow Gu...
Galaxy64.dll	GOG Galaxy Library		C:\GOG Games\Neverwinter Nights Enhanced Edition\...	0x7FEEED690000	ASLR	
libcrypto-1_1-x64.dll	OpenSSL library	The OpenSSL Project, h...	C:\GOG Games\Neverwinter Nights Enhanced Edition\...	0x7FEEED3D0000	ASLR	
libssl-1_1-x64.dll	OpenSSL library	The OpenSSL Project, h...	C:\GOG Games\Neverwinter Nights Enhanced Edition\...	0x7FEEFCBE0000	ASLR	
nwmain.exe	Neverwinter Nights	Beamdog	C:\GOG Games\Neverwinter Nights Enhanced Edition\...	0x7FF795990000	ASLR	
opengl32.dll	Main implementation library		C:\GOG Games\Neverwinter Nights Enhanced Edition\...	0x5E820000	ASLR	
StaticCache.dat			C:\Windows\Fonts\StaticCache.dat	0x1FCC9640000	n/a	n/a
SortDefault.nls			C:\Windows\Globalization\Sorting\SortDefault.nls	0x1FCB7920000	n/a	n/a
advapi32.dll	API avancées Windows 32	Microsoft Corporation	C:\Windows\System32\advapi32.dll	0x7FFF0FD0000	ASLR	CFG
apphelp.dll	Fichier DLL du client de com...	Microsoft Corporation	C:\Windows\System32\apphelp.dll	0x7FFF0BD10000	ASLR	CFG
AudioSes.dll	Session audio	Microsoft Corporation	C:\Windows\System32\AudioSes.dll	0x7FFF06E50000	ASLR	CFG
bcrypt.dll	Bibliothèque de primitives de ...	Microsoft Corporation	C:\Windows\System32\bcrypt.dll	0x7FFF0E940000	ASLR	CFG
bcryptprimitives.dll	Windows Cryptographic Primit...	Microsoft Corporation	C:\Windows\System32\bcryptprimitives.dll	0x7FFF0E8B0000	ASLR	CFG
cfgmgr32.dll	Configuration Manager DLL	Microsoft Corporation	C:\Windows\System32\cfgmgr32.dll	0x7FFF0EDD0000	ASLR	CFG
clbcatq.dll	COM+ Configuration Catalog	Microsoft Corporation	C:\Windows\System32\clbcatq.dll	0x7FFF0F020000	ASLR	CFG
coloradapclnt.dll	Microsoft Color Adapter Client	Microsoft Corporation	C:\Windows\System32\coloradapclnt.dll	0x7FFF09600000	ASLR	CFG
combase.dll	Microsoft COM pour Windows	Microsoft Corporation	C:\Windows\System32\combase.dll	0x7FFF0F650000	ASLR	CFG
CoreMessaging.dll	Microsoft CoreMessaging DLL	Microsoft Corporation	C:\Windows\System32\CoreMessaging.dll	0x7FFF0B830000	ASLR	CFG
CoreUIComponents.dll	Microsoft Core UI Component...	Microsoft Corporation	C:\Windows\System32\CoreUIComponents.dll	0x7FFF0B3D0000	ASLR	CFG
crypt32.dll	Crypto API32	Microsoft Corporation	C:\Windows\System32\crypt32.dll	0x7FFF0E970000	ASLR	CFG
cryptbase.dll	Base cryptographic API DLL	Microsoft Corporation	C:\Windows\System32\cryptbase.dll	0x7FFF0E0E0000	ASLR	CFG

CPU Usage: 34.31% Commit Charge: 37.17% Processes: 112 Physical Usage: 43.23%

Attack surface

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)

[illegible]

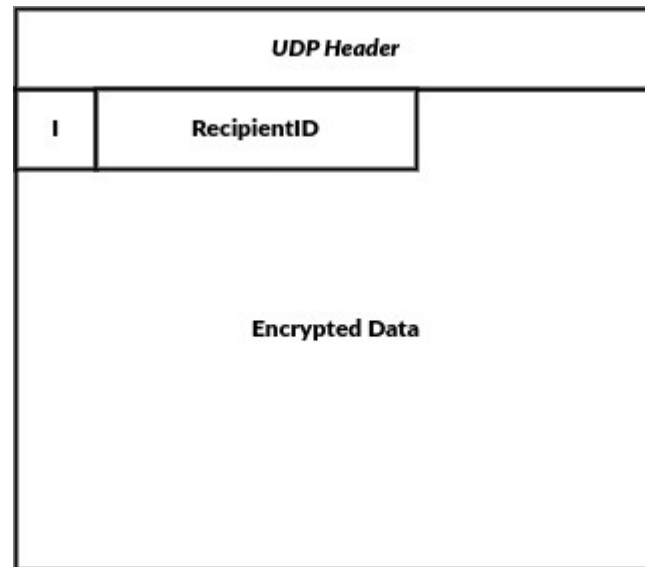
- **Encrypted Communication**
- **Use open-source LibHydrogen**
 - Based on Curve25519 elliptic curve and Gimli permutation
 - Noise Protocol Framework
- **Static keypair derived from CD-Key**

■ Divided in 3 layer

- Layer 1 : Handle deciphering
- Layer 2 : Handle compressed and fragmented data
- Layer 3 : Handle game message

Layer 1

- **UDP Based**
- **1 magic byte**
- **4 random bytes**



Layer 2

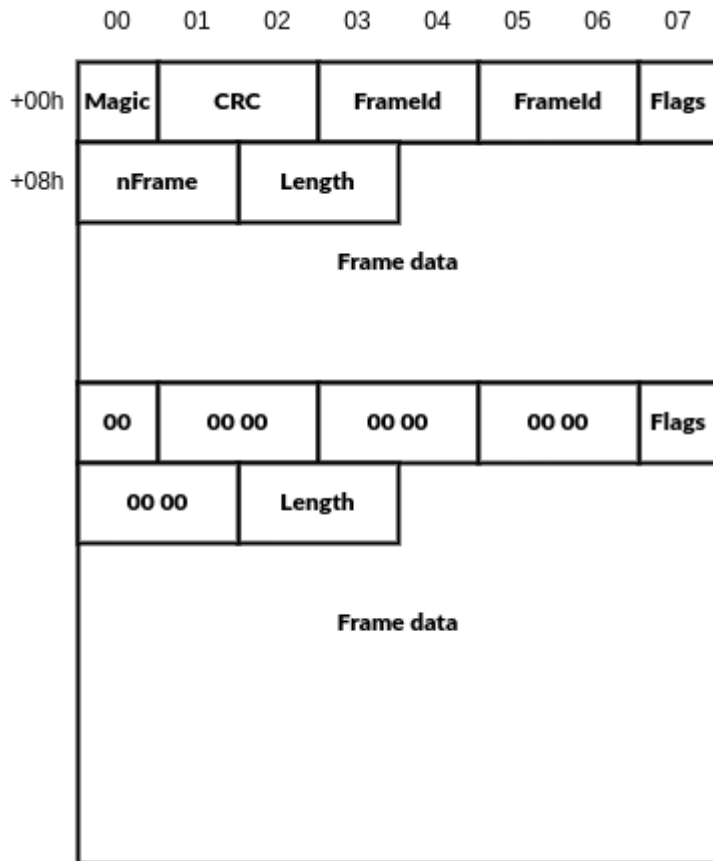
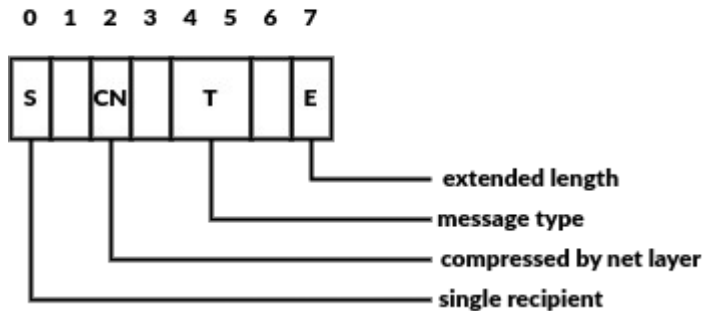
3 Frame Types

- DATA
- ACK
- NAK

Fragmentation

Compression

flags details



■ 4 types of messages

- “P” Server to Player
- “p” Player to Server
- “S” Server to SysAdmin
- “s” SysAdmin to Server

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

The screenshot shows a window titled "NWServer - Server - 1/4 0:31:48". It contains various configuration fields and checkboxes. The "Server Name" is "Server", "Difficulty" is "Easy", "Server AutoSave Interval (minutes)" is "0", and "Levels" is "1 -> 40". The "IP Address" is "192.168.56.104:5121". The "Module Name" is empty, "Game Type" is "Action", and "Player vs. Player" is "Party". There are checkboxes for "Local Characters Allowed" (unchecked), "Enforce Legal Characters" (checked), "Item Level Restrictions" (unchecked), "Only One Party" (checked), "Player Pause Enabled" (unchecked), "Reload When Empty" (checked), and "Post Game To Internet" (checked). The "Players / Max Players:" is "1/4", and "Player 1" is selected. Below this is a table with columns "Ban Name", "Ban CD", "Ban IP", and "Boot". The "Save Game" button is next to a text field, and the "Slot Number" is "1". The "Server Message:" field is empty. The "Server status:" is "Build 8193.36-13 [d77dd024] Idle, login disabled.".

Ban Name	Ban CD	Ban IP	Boot
----------	--------	--------	------

Save Game: [] Slot Number: [1] Save

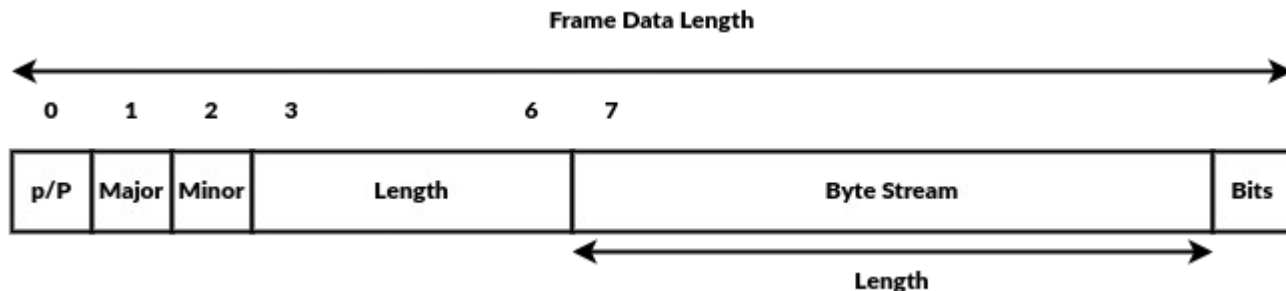
Server Message: [] Send Message

Server status: Build 8193.36-13 [d77dd024] Idle, login disabled. Shut down

■ Game related message

- Update creature appearance
- Start a progress bar
- Add a quest in journal
- Level up
- ...

■ ~250 messages types



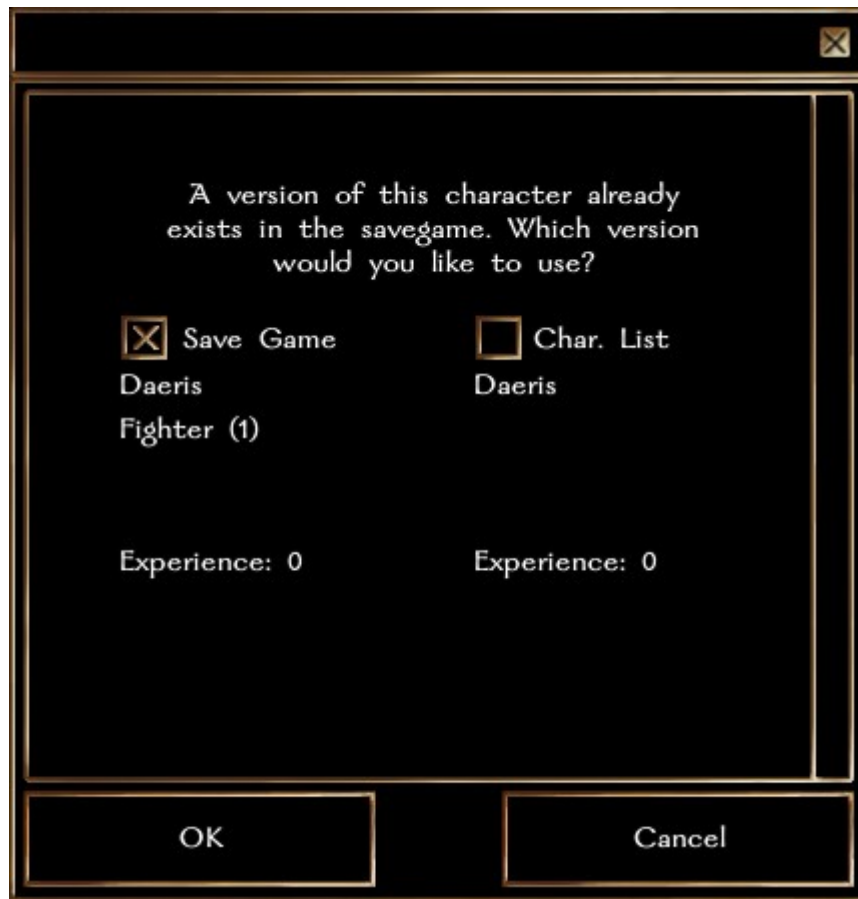
- **CNWMessage helps to decode byte stream**
- **21 simple types**
 - `CNWMessage::ReadBOOL`
 - `CNWMessage::ReadBYTE`
 - `CNWMessage::ReadINT`
 - `CNWMessage::ReadDOUBLE`
 - `CNWMessage::ReadVOIDPtr`
 - `CNWMessage::ReadCExoString`
 - ...

Vulnerabilities

```
__int64 __fastcall CNWMessage::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);  
        _ClassListSize = 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

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



Second bug

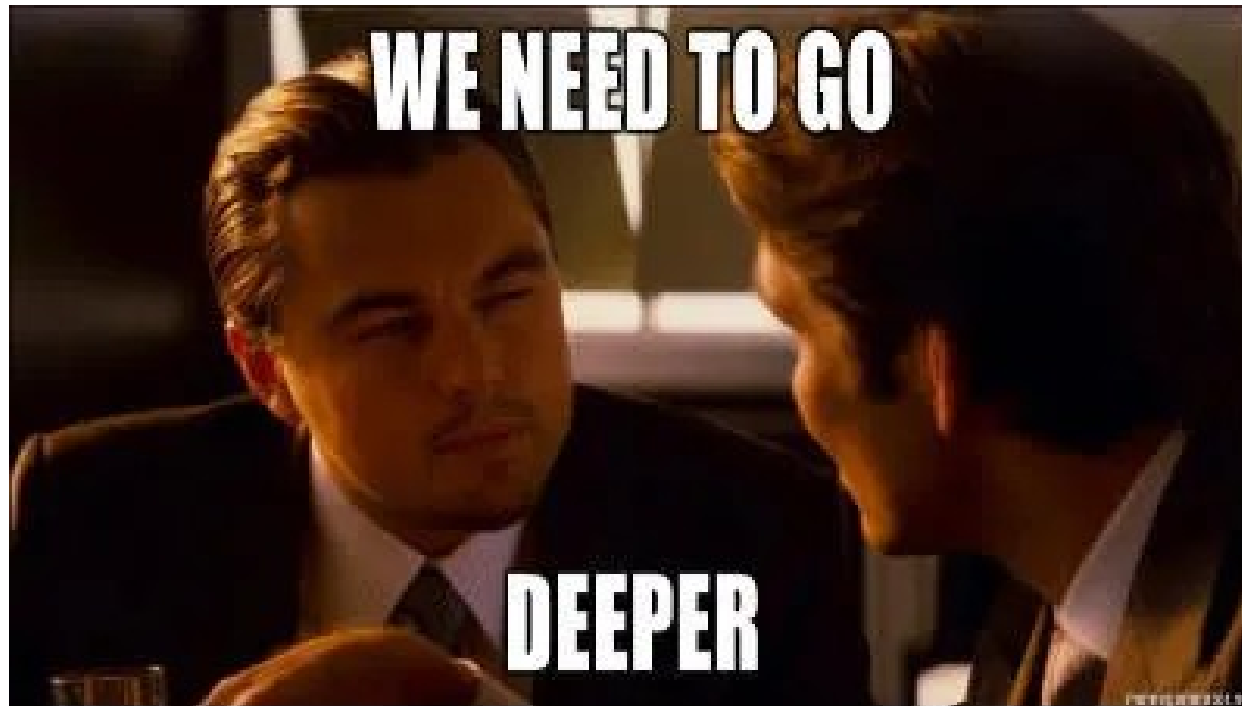
```
__int64 __fastcall CNWMessage::HandleServerToPlayerCreatureUpdate_Appearance(CNWMessage *this)
{
[...]  
    unsigned __int16 Buf[18]; // [rsp+216h] [rbp-11Ah] BYREF  
[...]  
    Count = CNWMessage::ReadBYTE(this, 8);  
    _Count = Count;  
    if ( Count )  
    {  
        v53 = (int *)Buf;  
        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

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

Find a leak

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

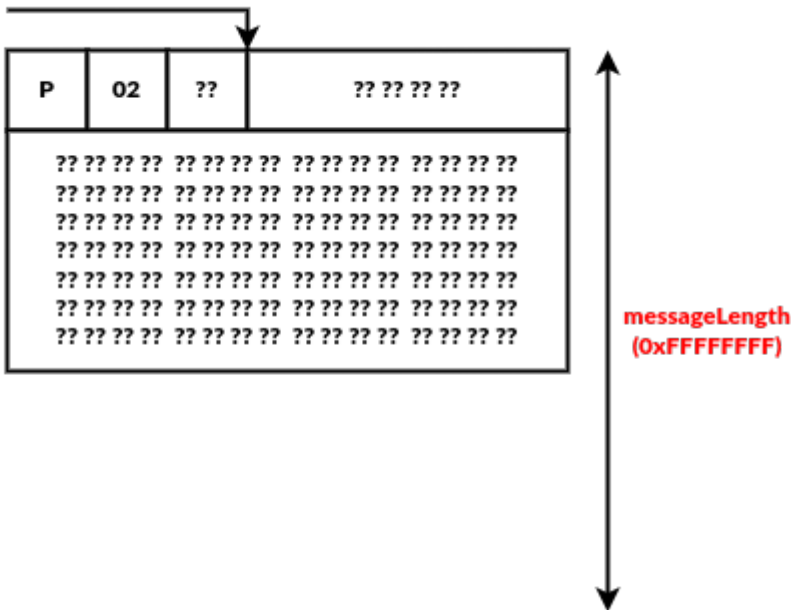


Integer underflow bug

```
__int64 __fastcall CNWMessage::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 = CNWMessage::HandleServerToPlayerServerStatus(this, Minor);
                goto LABEL_9;
            case 2u:
                CExoString::operator=(&a1, "Login");
                active = CNWMessage::HandleServerToPlayerLogin(this, Minor);
                goto LABEL_9;
        }
    }
}
```

Integer underflow bug

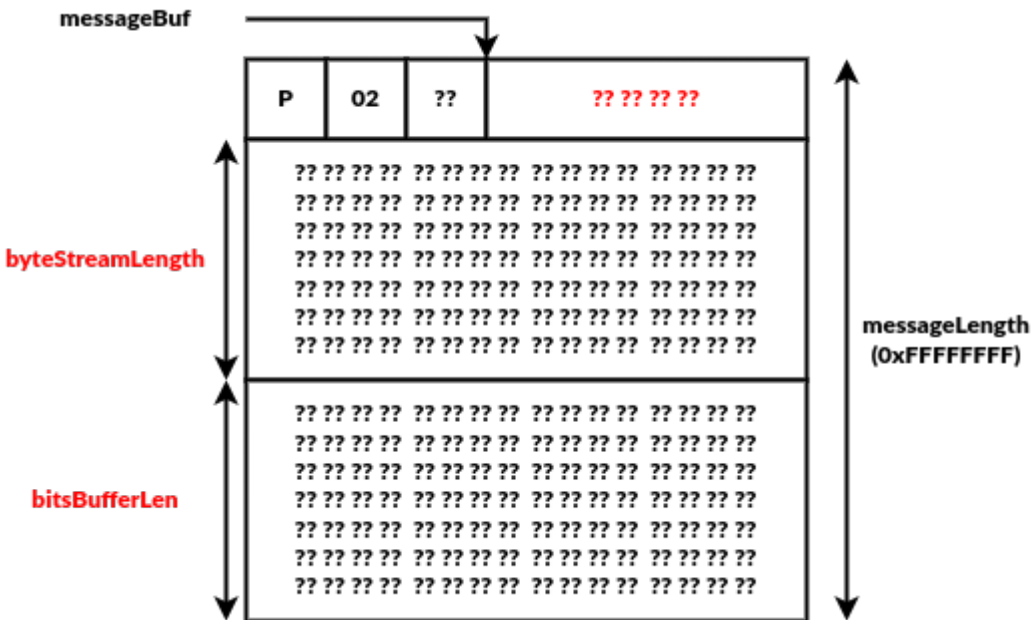
messageBuf



```
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



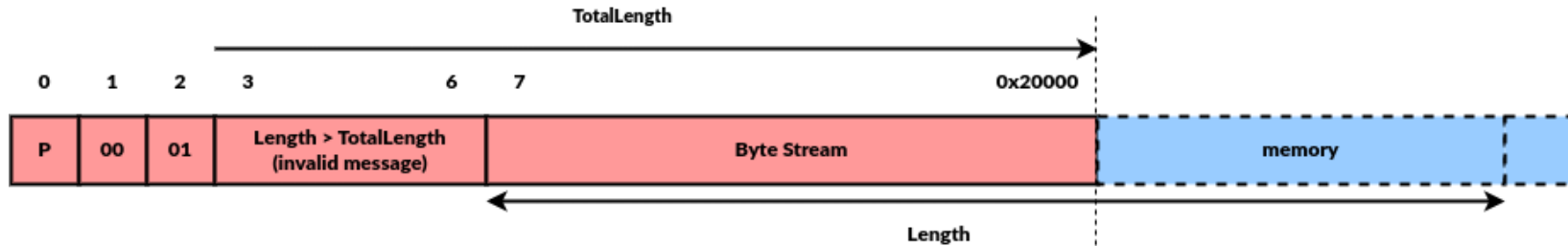
```
unsigned int __fastcall CNWMessage::SetReadMessage(
    CNWMessage *this,
    unsigned __int8 *messageBuf,
    unsigned int Length,
    int a4,
    int a5)
{
    unsigned int bytesStreamLength; // ecx
    unsigned int res; // eax

    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
    {
        [...]
    }
}
```

- **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 $\geq 0x20000$
 - else *rx_buffer* member is used
 - *rx_buffer* is not erased after use

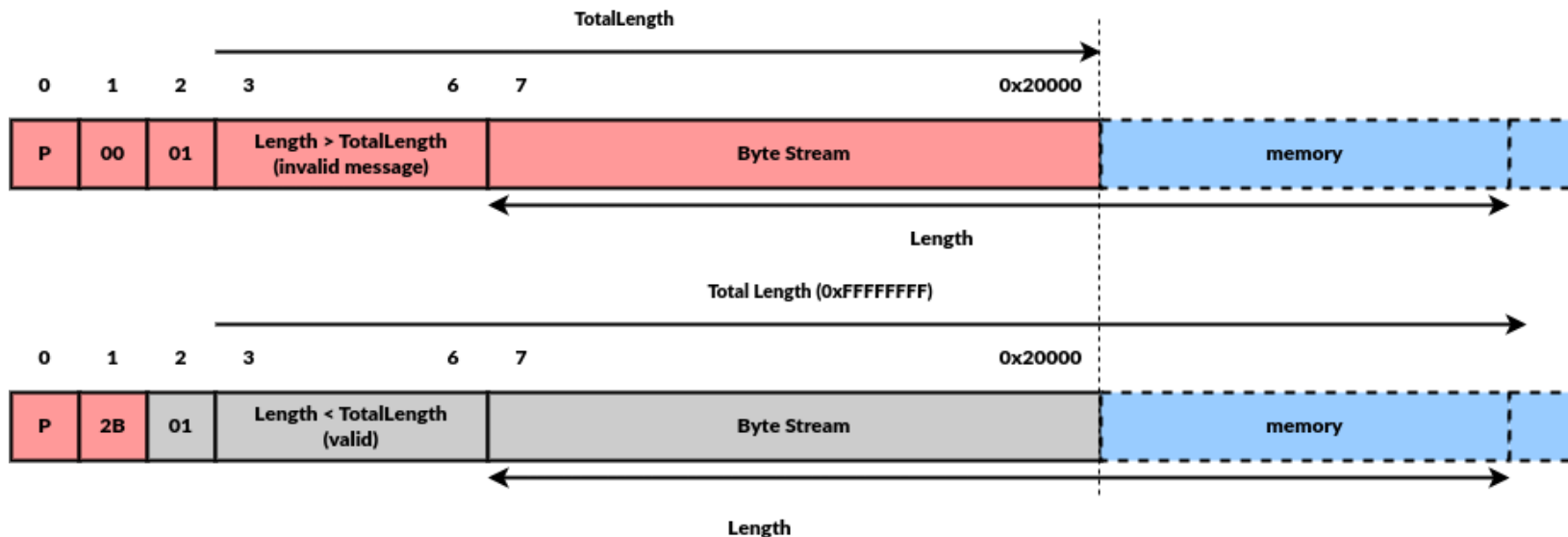
Transform into OOB

- Send first invalid message to initialise buffer



Transform into OOB

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

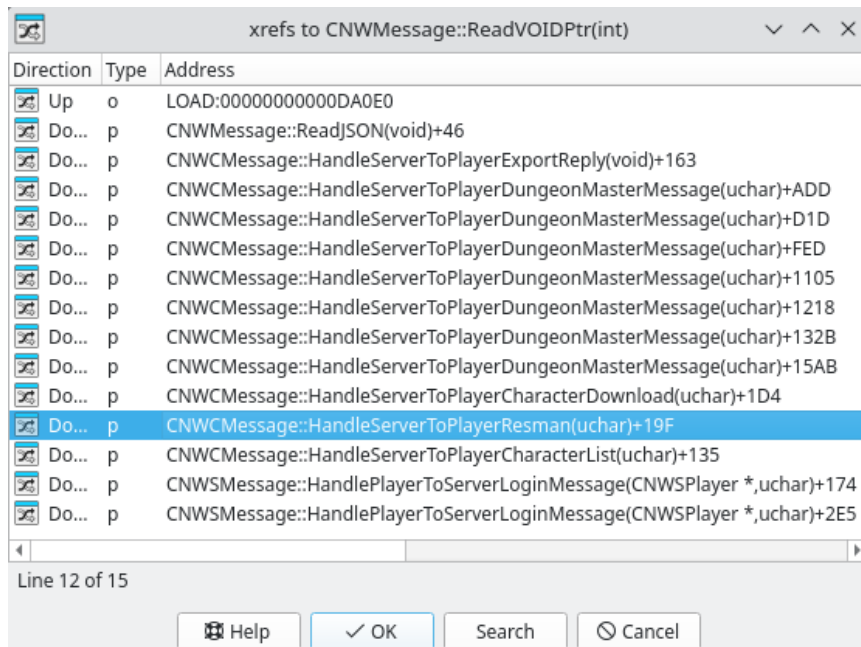


Transform into OOBR

■ CNWMessage::ReadVOIDPtr

- return a pointer to buffer of arbitrary size

■ Usage of CNWMessage::ReadVOIDPtr



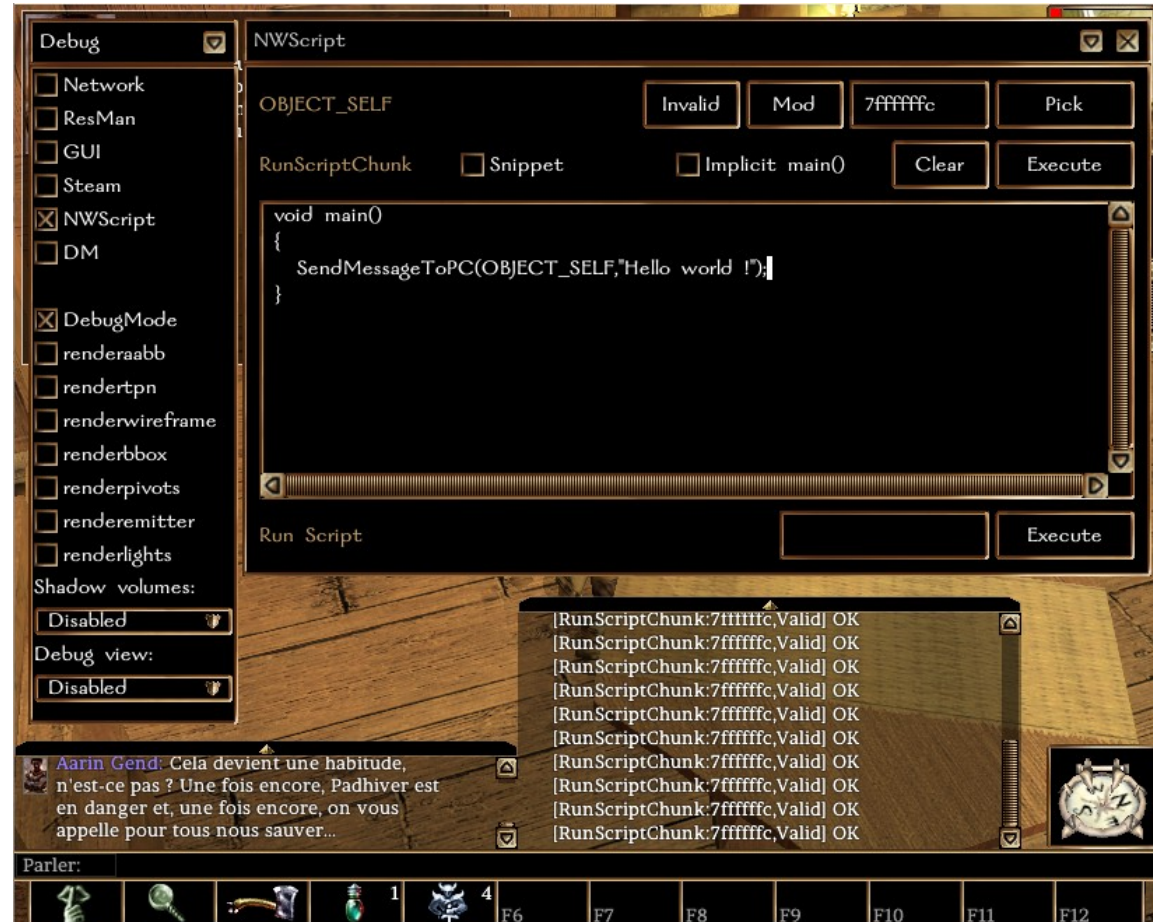
Transform into OOB

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

```
__int64 __fastcall CNWMessage::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

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



Portal feature

- 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.

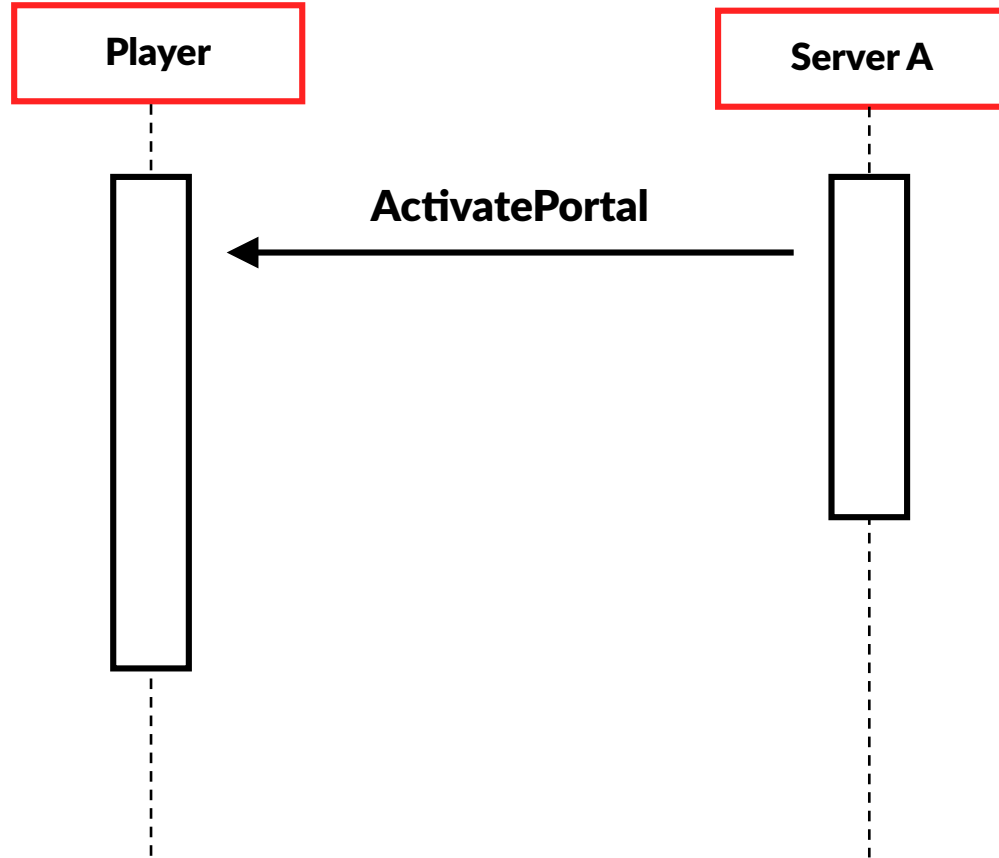
```
void ActivatePortal(  
    object oTarget,  
    string sIPAddress = "",  
    string sPassword = "",  
    string sWaypointTag = "",  
    int bSeamless = FALSE  
);
```

■ CNWCMessage::HandleServerToPlayerPortalActivatePortal

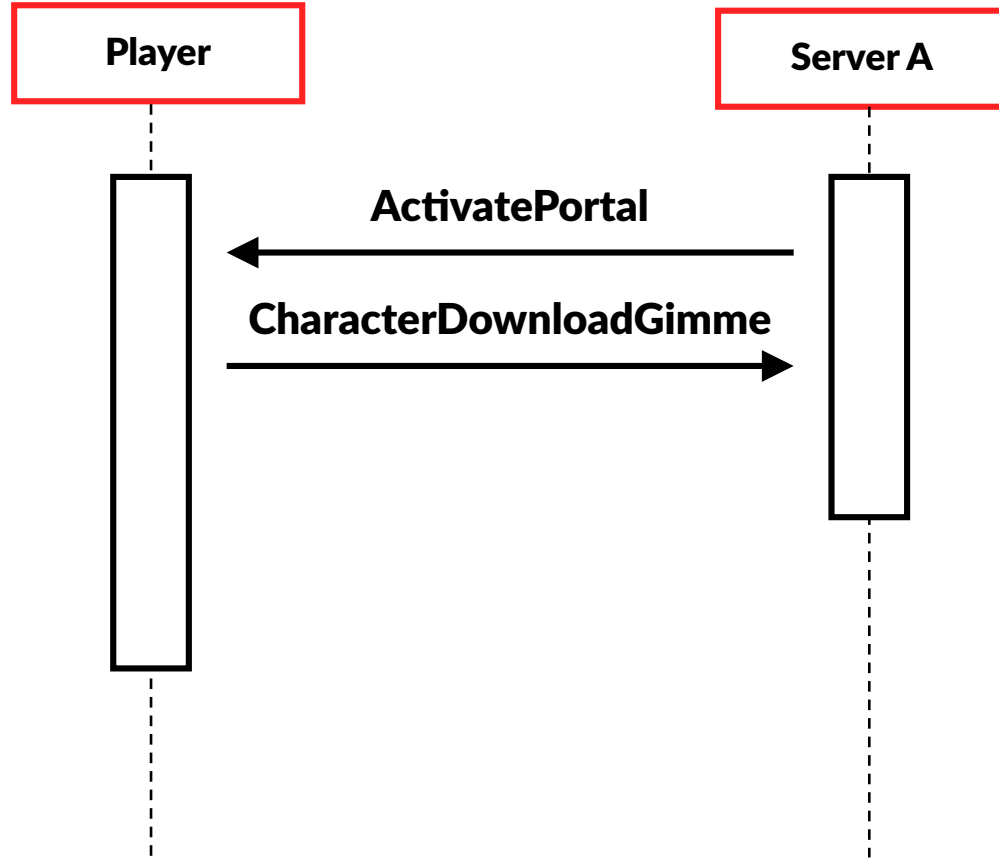
- Major/Minor 2A/01
- Work with DebugMode disabled
- bSeamless => Automatic transfert without dialog box



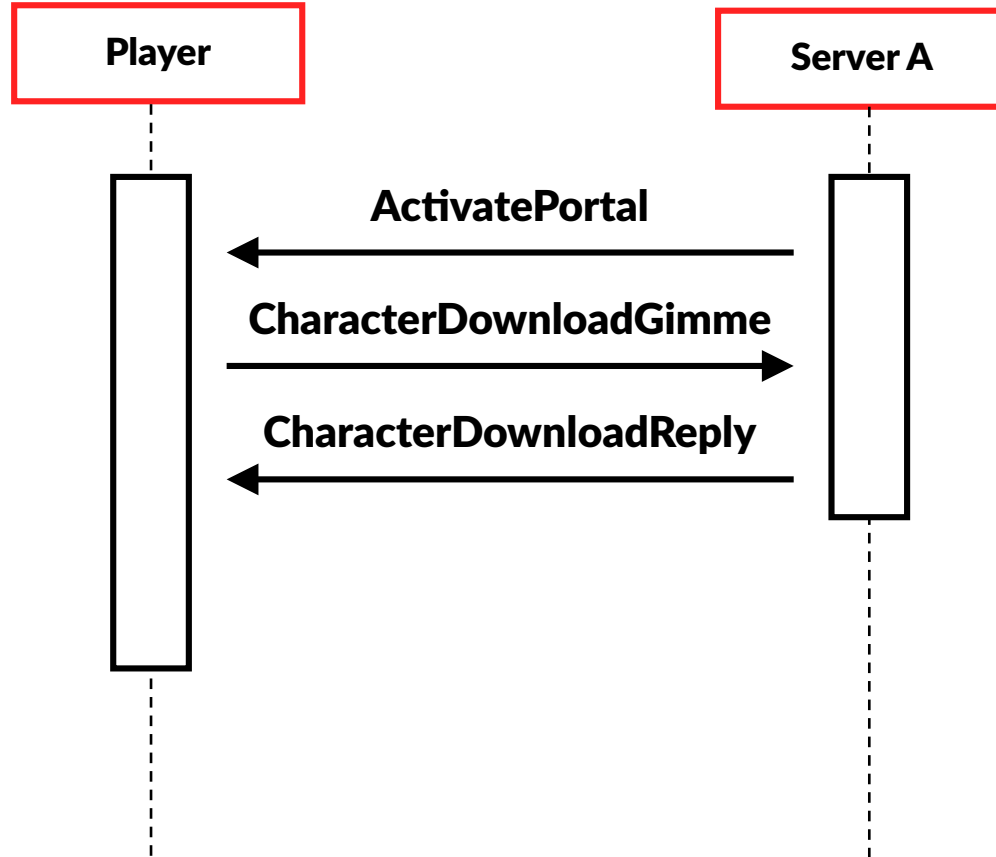
Portal feature



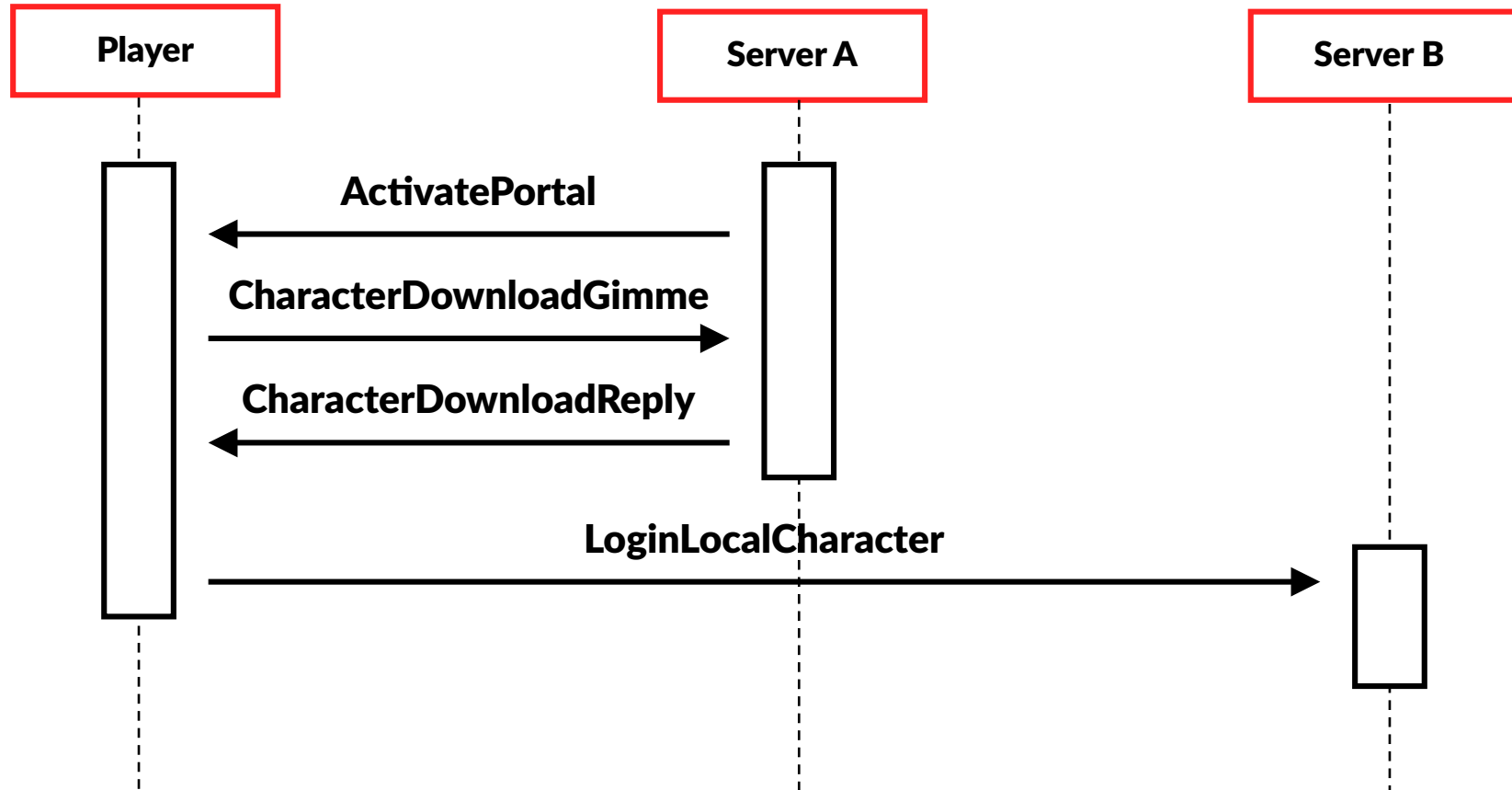
Portal feature



Portal feature



Portal feature



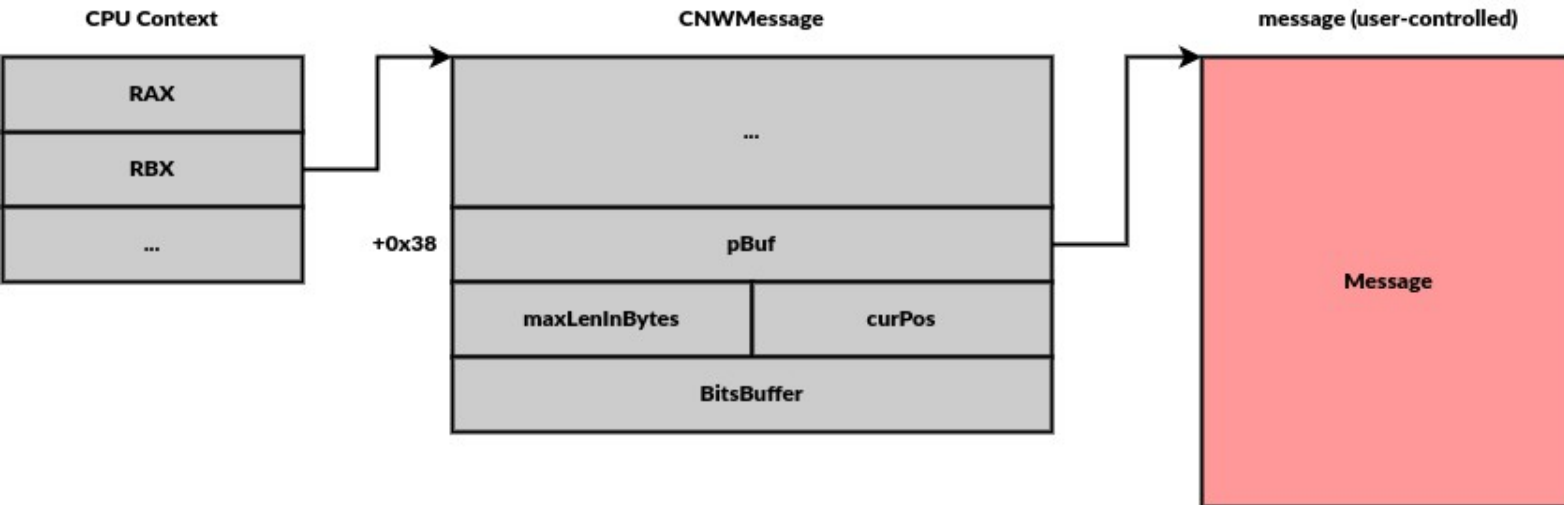
Exploitation

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

```
__int64 __fastcall CNWMessage::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);
    }
}
```

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

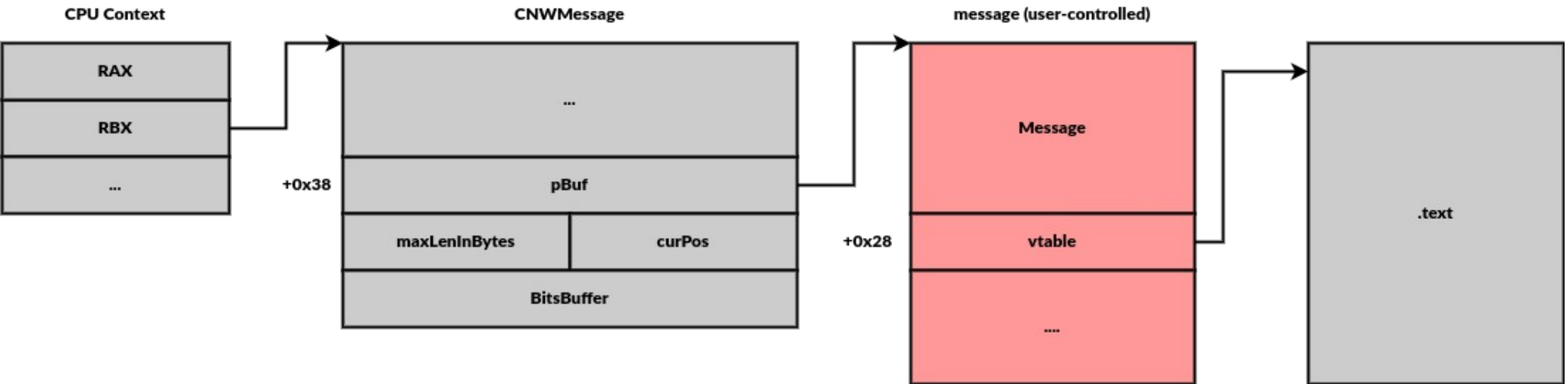
- OOBW to rewrite return address
- RBX points to CWNMessage
- Limited call to existing vtable functions





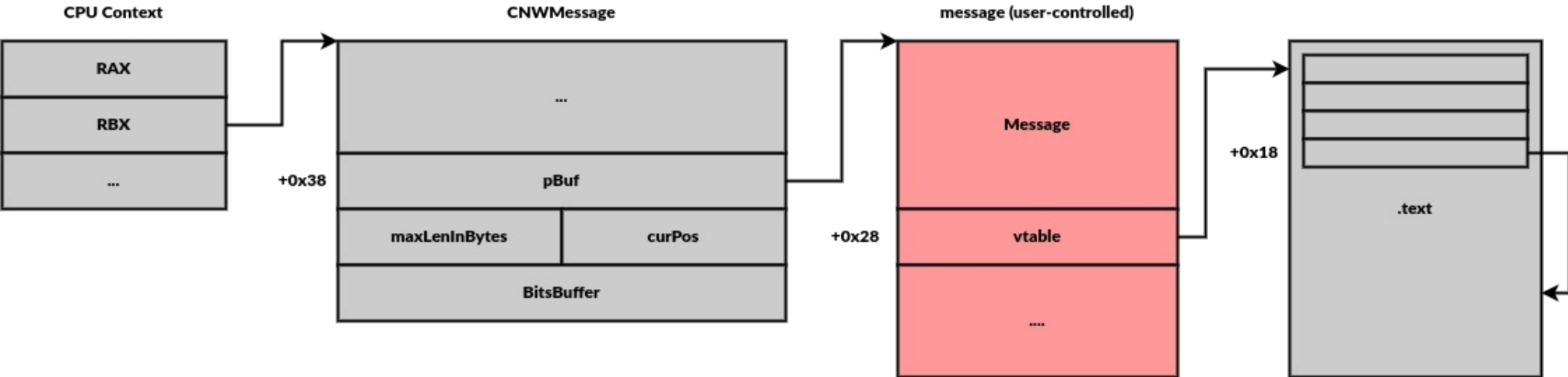
```
mov    rcx, [rbx+38h]
mov    rdi, [rcx+28h]
call   qword ptr [rdi+18h]
```

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



```
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



- **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**

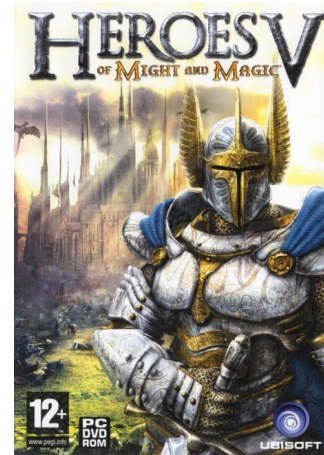
```
.text:000000001401D8850    mov     rax, rcx
.text:000000001401D8853    mov     qword ptr [rcx+10h], 0
.text:000000001401D885B    mov     rcx, [rcx+18h]
.text:000000001401D885F    xor     r8d, r8d
.text:000000001401D8862    jmp     qword ptr [rax+20h]
```

- **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
```


Conclusion

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





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



<https://twitter.com/synacktiv>



<https://synacktiv.com>