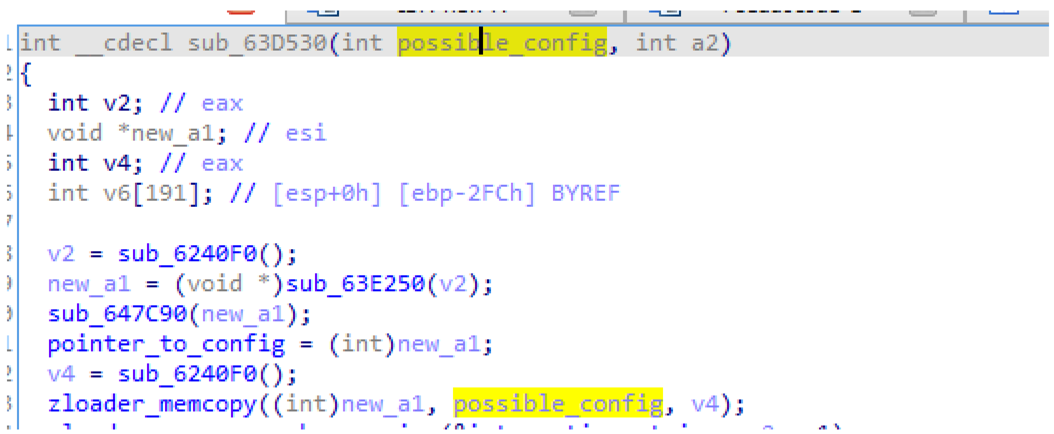
Silent Night- Zloader/Zbot

Zloader is a banking trojan, also known as a financial malware, that is designed to steal sensitive information such as login credentials, credit card numbers, and other financial information from infected machines.

Zloader is typically distributed through spam emails, malvertising, and exploit kits. Once it infects a machine, it installs itself as a service and starts communicating with its command and control (C&C) servers. It uses advanced evasion techniques to avoid detection by security software, including the use of anti-debugging and anti-virtualization techniques.

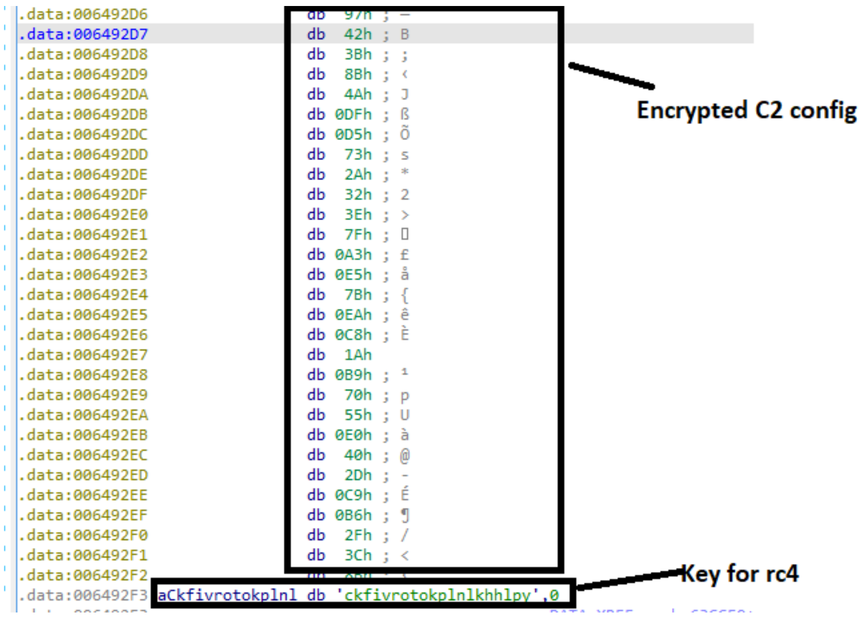
One of the key features of Zloader is its ability to update itself automatically. This means that the malware can evolve over time and become more difficult to detect and remove. Zloader also has a new encryption scheme that makes it more difficult to analyze and detect.

Following Zloader route holds c2 config



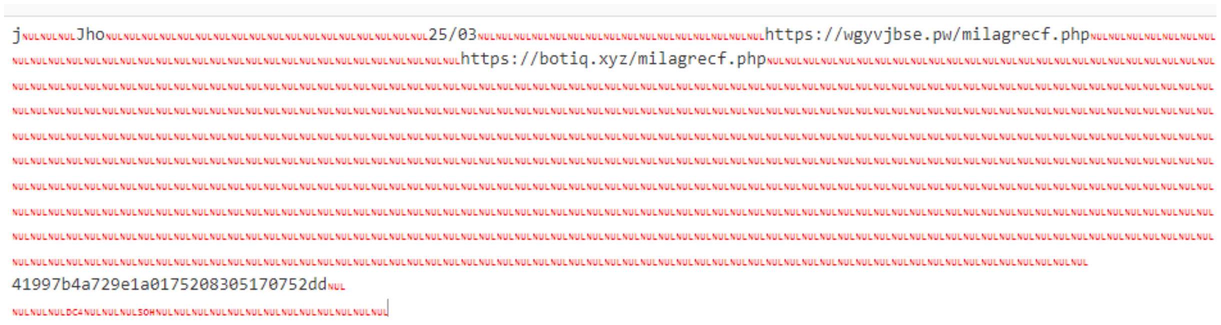
```
int __cdecl sub_63D530(int possible_config, int a2)
{
  int v2; // eax
  void *new_a1; // esi
  int v4; // eax
  int v6[191]; // [esp+0h] [ebp-2FCh] BYREF

  v2 = sub_6240F0();
  new_a1 = (void *)sub_63E250(v2);
  sub_647C90(new_a1);
  pointer_to_config = (int)new_a1;
  v4 = sub_6240F0();
  zloader_memcopy((int)new_a1, possible_config, v4);
```



```
.data:006492D6                 db  97h ; —
.data:006492D7                 db  42h ; B
.data:006492D8                 db  3Bh ; ;
.data:006492D9                 db  8Bh ; ‹
.data:006492DA                 db  4Ah ; J
.data:006492DB                 db 0DFh ; ß
.data:006492DC                 db 0D5h ; Õ
.data:006492DD                 db  73h ; s
.data:006492DE                 db  2Ah ; *
.data:006492DF                 db  32h ; 2
.data:006492E0                 db  3Eh ; >
.data:006492E1                 db  7Fh ; 
.data:006492E2                 db 0A3h ; £
.data:006492E3                 db 0E5h ; å
.data:006492E4                 db  7Bh ; {
.data:006492E5                 db 0EAh ; ê
.data:006492E6                 db 0C8h ; È
.data:006492E7                 db  1Ah
.data:006492E8                 db 0B9h ; ¹
.data:006492E9                 db  70h ; p
.data:006492EA                 db  55h ; U
.data:006492EB                 db 0E0h ; à
.data:006492EC                 db  40h ; @
.data:006492ED                 db  2Dh ; -
.data:006492EE                 db 0C9h ; É
.data:006492EF                 db 0B6h ; ¶
.data:006492F0                 db  2Fh ; /
.data:006492F1                 db  3Ch ; <
.data:006492F2
.data:006492F3 aCkfivrotokplnl db 'ckfivrotokplnlkhhlpy',0
```

Use cyber checf to decrpt the c2 config

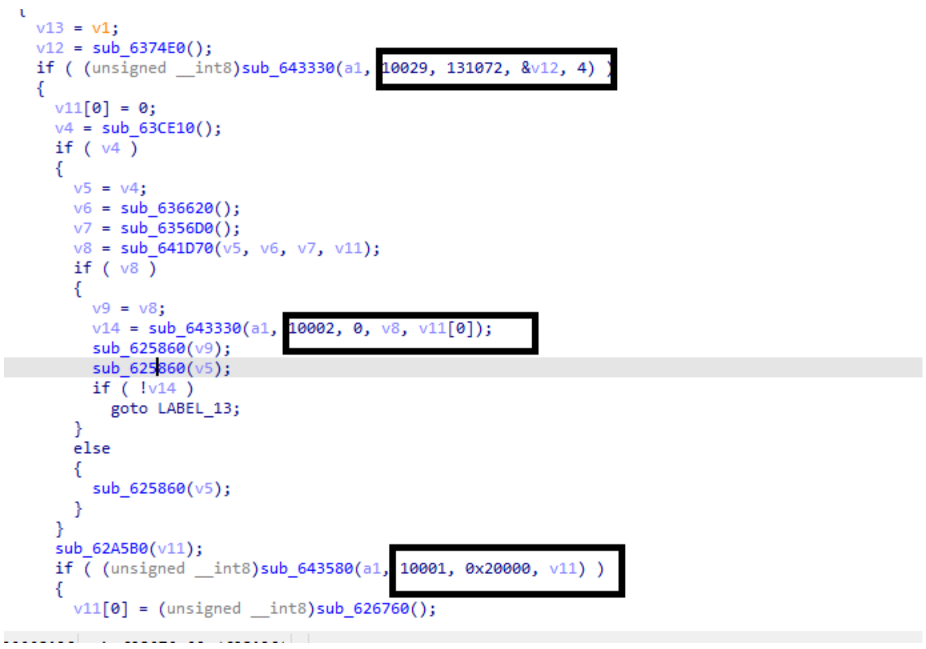


```
jNULNULNULJhoNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL25/03NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULhttps://wgyvjbse.pw/milagrecf.phpNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULhttps://botiq.xyz/milagrecf.phpNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
41997b4a729e1a0175208305170752ddNUL
NULNULNULDC4NULNULNULSOHNULNULNULNULNULNULNULNULNULNULNULNULNUL
```

Zloader-Zues Link:

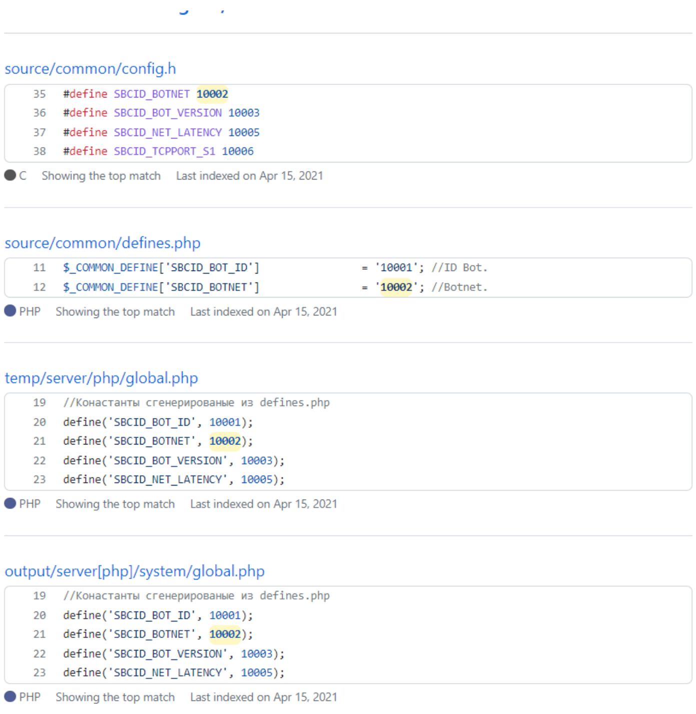
Zloader is a newer and more advanced variant of Zeus . Zeus and Zloader are both banking trojans that share a common ancestry and have some code similarities. However, Zloader is a newer variant of Zeus and has been updated with new features and functionality.

Following routine uncovers the Zloader , Zues trojan link



```
  {
    v13 = v1;
    v12 = sub_6374E0();
    if ( (unsigned __int8)sub_643330(a1, 10029, 131072, &v12, 4) )
    {
      v11[0] = 0;
      v4 = sub_63CE10();
      if ( v4 )
      {
        v5 = v4;
        v6 = sub_636620();
        v7 = sub_6356D0();
        v8 = sub_641D70(v5, v6, v7, v11);
        if ( v8 )
        {
          v9 = v8;
          v14 = sub_643330(a1, 10002, 0, v8, v11[0]);
          sub_625860(v9);
          sub_625860(v5);
          if ( !v14 )
            goto LABEL_13;
        }
        else
        {
          sub_625860(v5);
        }
      }
      sub_62A5B0(v11);
      if ( (unsigned __int8)sub_643580(a1, 10001, 0x20000, v11) )
      {
        v11[0] = (unsigned __int8)sub_626760();
```

Here 10002 and 10003 represent the BOTNET ID and Botnet version.



source/common/config.h

```
#define SBCID_BOTNET 10002
#define SBCID_BOT_VERSION 10003
#define SBCID_NET_LATENCY 10005
#define SBCID_TCPPORT_S1 10006
```

● C   Showing the top match   Last indexed on Apr 15, 2021

source/common/defines.php

```
$_COMMON_DEFINE['SBCID_BOT_ID']                = '10001'; //ID Bot.
$_COMMON_DEFINE['SBCID_BOTNET']                = '10002'; //Botnet.
```

● PHP   Showing the top match   Last indexed on Apr 15, 2021

temp/server/php/global.php

```
//Конастанты сгенерированые из defines.php
define('SBCID_BOT_ID', 10001);
define('SBCID_BOTNET', 10002);
define('SBCID_BOT_VERSION', 10003);
define('SBCID_NET_LATENCY', 10005);
```

● PHP   Showing the top match   Last indexed on Apr 15, 2021

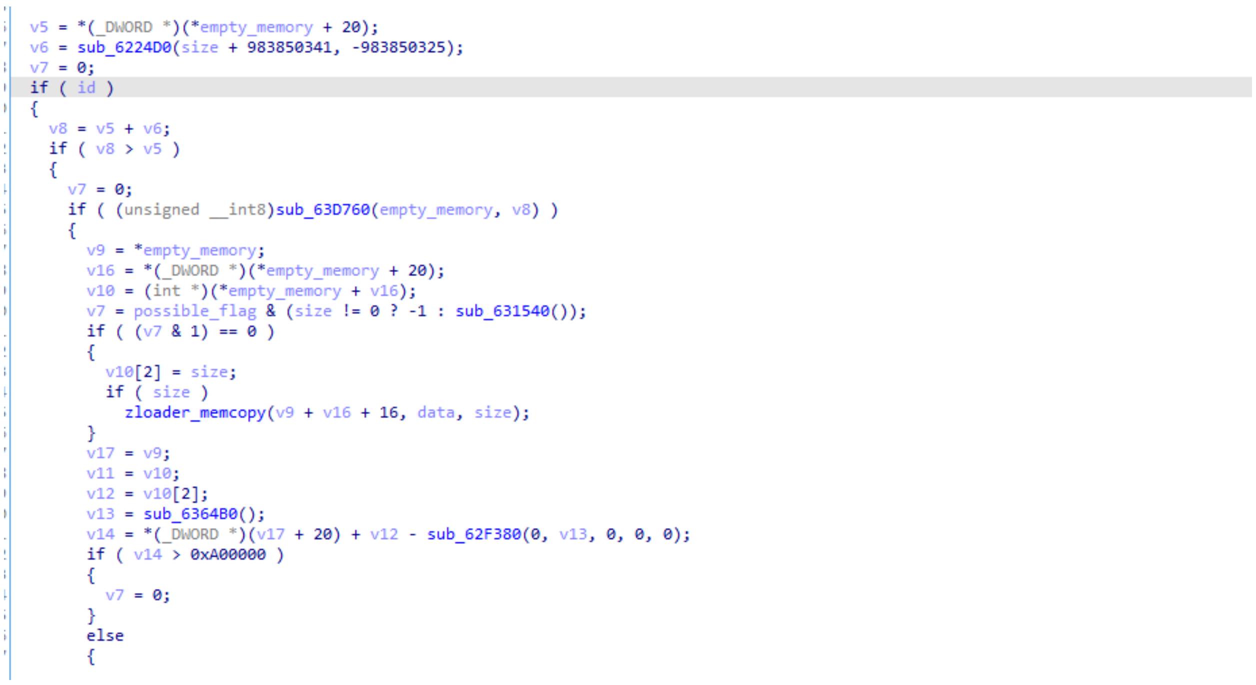
output/server[php]/system/global.php

```
//Конастанты сгенерированые из defines.php
define('SBCID_BOT_ID', 10001);
define('SBCID_BOTNET', 10002);
define('SBCID_BOT_VERSION', 10003);
define('SBCID_NET_LATENCY', 10005);
```

● PHP   Showing the top match   Last indexed on Apr 15, 2021

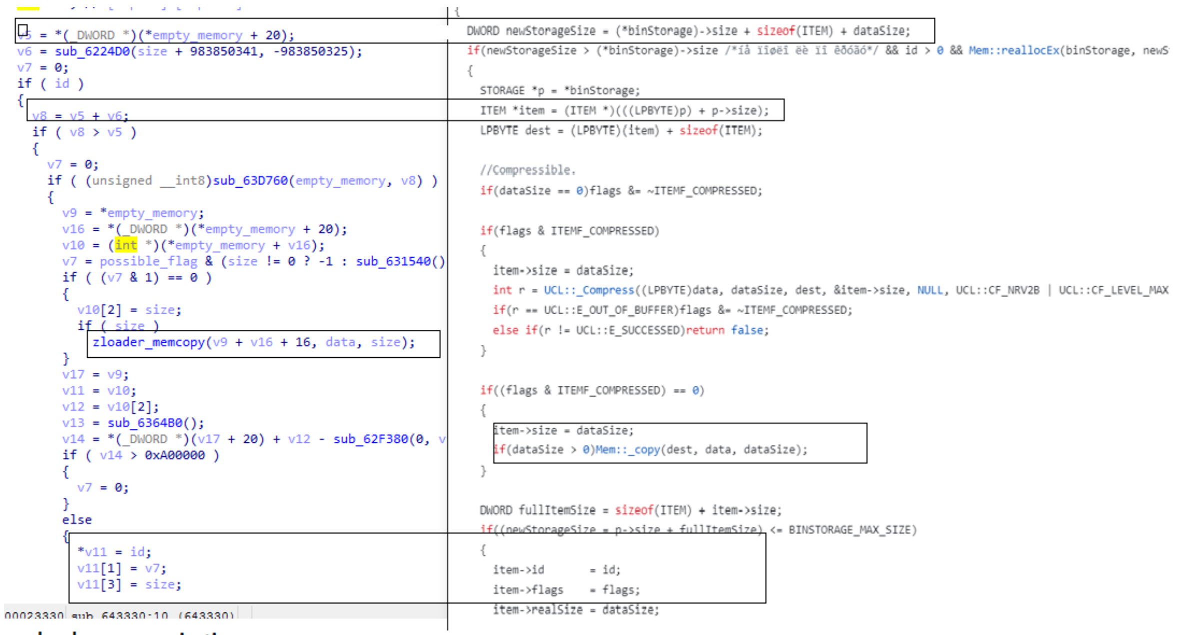
Zloader open source code https://github.com/Visgean/Zeus/search?q=10002

Following routine uncover how Zloader takes Zeus communication function.



```
v5 = *(_DWORD *)(*empty_memory + 20);
v6 = sub_6224D0(size + 983850341, -983850325);
v7 = 0;
if ( id )
{
  v8 = v5 + v6;
  if ( v8 > v5 )
  {
    v7 = 0;
    if ( (unsigned __int8)sub_63D760(empty_memory, v8) )
    {
      v9 = *empty_memory;
      v16 = *(_DWORD *)(*empty_memory + 20);
      v10 = (int *)(*empty_memory + v16);
      v7 = possible_flag & (size != 0 ? -1 : sub_631540());
      if ( (v7 & 1) == 0 )
      {
        v10[2] = size;
        if ( size )
          zloader_memcopy(v9 + v16 + 16, data, size);
      }
      v17 = v9;
      v11 = v10;
      v12 = v10[2];
      v13 = sub_6364B0();
      v14 = *(_DWORD *)(v17 + 20) + v12 - sub_62F380(0, v13, 0, 0, 0);
      if ( v14 > 0xA00000 )
      {
        v7 = 0;
      }
      else
      {
```

Following routine shows the Zloader and Zues code similarity for communication routine



```
v5 = *(_DWORD *)(*empty_memory + 20);
v6 = sub_6224D0(size + 983850341, -983850325);
v7 = 0;
if ( id )
{
  v8 = v5 + v6;
  if ( v8 > v5 )
  {
    v7 = 0;
    if ( (unsigned __int8)sub_63D760(empty_memory, v8) )
    {
      v9 = *empty_memory;
      v16 = *(_DWORD *)(*empty_memory + 20);
      v10 = (int *)(*empty_memory + v16);
      v7 = possible_flag & (size != 0 ? -1 : sub_631540()
      if ( (v7 & 1) == 0 )
      {
        v10[2] = size;
        if ( size )
          zloader_memcopy(v9 + v16 + 16, data, size);
      }
      v17 = v9;
      v11 = v10;
      v12 = v10[2];
      v13 = sub_6364B0();
      v14 = *(_DWORD *)(v17 + 20) + v12 - sub_62F380(0, v
      if ( v14 > 0xA00000 )
      {
        v7 = 0;
      }
      else
      {
        *v11 = id;
        v11[1] = v7;
        v11[3] = size;
00023330 sub_643330:10 (643330)
```

```
{
  DWORD newStorageSize = (*binStorage)->size + sizeof(ITEM) + dataSize;
  if(newStorageSize > (*binStorage)->size /*íå ïîøëî èè ïî êðóãó*/ && id > 0 && Mem::reallocEx(binStorage, newS
  {
    STORAGE *p = *binStorage;
    ITEM *item = (ITEM *)(((LPBYTE)p) + p->size);
    LPBYTE dest = (LPBYTE)(item) + sizeof(ITEM);

    //Compressible.
    if(dataSize == 0)flags &= ~ITEMF_COMPRESSED;

    if(flags & ITEMF_COMPRESSED)
    {
      item->size = dataSize;
      int r = UCL::_Compress((LPBYTE)data, dataSize, dest, &item->size, NULL, UCL::CF_NRV2B | UCL::CF_LEVEL_MAX
      if(r == UCL::E_OUT_OF_BUFFER)flags &= ~ITEMF_COMPRESSED;
      else if(r != UCL::E_SUCCESSED)return false;
    }

    if((flags & ITEMF_COMPRESSED) == 0)
    {
      item->size = dataSize;
      if(dataSize > 0)Mem::_copy(dest, data, dataSize);
    }

    DWORD fullItemSize = sizeof(ITEM) + item->size;
    if((newStorageSize = p->size + fullItemSize) <= BINSTORAGE_MAX_SIZE)
    {
      item->id       = id;
      item->flags    = flags;
      item->realSize = dataSize;
```

zloader communication routine

Zeus communication routine

IOCS

https://wgyvjbse.pw/milagrecf.php

https://botiq.xyz/milagrecf.php