[Mal Series #13] Darkside Ransom

By GhouLSec

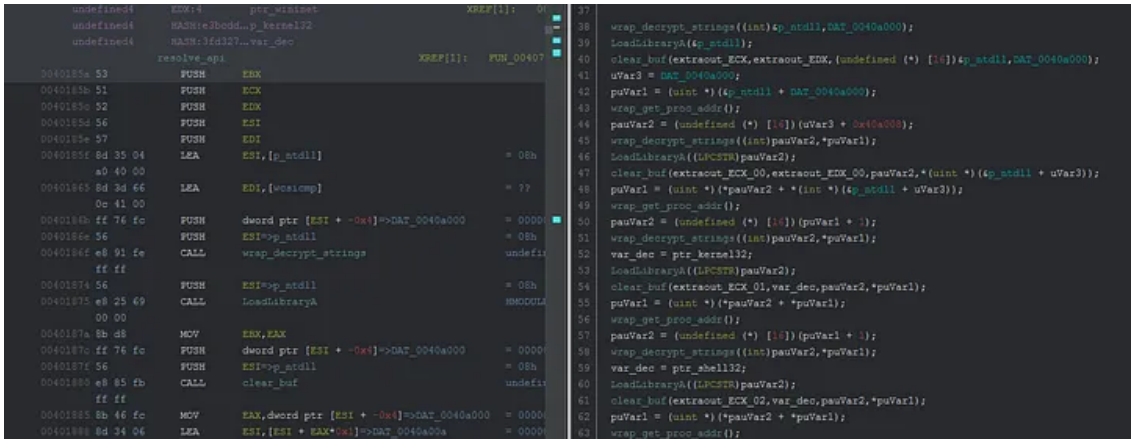
Published: 2021-05-01 · Archived: 2026-04-05 20:06:54 UTC

Here is my analysis of the Darkside ransomware.
Will attach more screenshot regarding of my analysis this time 😏

> Didn't connect to the C2 during the analysis

Dynamically Resolve Windows API

Press enter or click to view image in full size



Elevate Privilege (If running in Non-Admin privilege)

Utilizing COM bypass UAC privilege (When Access Token Method Failed)

```
Elevation:Administrator!new:%s
```

Get access token from admin process (e.g. Explorer.exe)

Press enter or click to view image in full size

```
local_8 = 0;
local_c = (int *)0x0;
iVar1 = (*_OpenProcessToken)(0xffffffff,0x28,&local_8);
if (iVar1 != 0) {
  (*_GetTokenInformation)(local_8,3,&local_c,4,&local_10);
  local_c = (int *)(*_RtlAllocateHeap)(dat_PEB_ProcHeap,8,local_10);
  iVar1 = (*_GetTokenInformation)(local_8,3,local_c,local_10,&local_10);
  if (iVar1 != 0) {
    piVar2 = local_c + 1;
    iVar1 = *local_c;
    do {
      if (piVar2[2] == 0) {
        piVar2[2] = 2;
      }
      piVar2 = piVar2 + 3;
      iVar1 = iVar1 + -1;
    } while (iVar1 != 0);
    (*_AdjustTokenPriviledge)(local_8,0,local_c,0,0,0);
  }
}
```

Adjust Privilege Token

Hash Generation File Extention, Mutex, Victim’s ID

```
do {
  bVar2 = *crc_hash >> 4;
  bVar4 = *crc_hash & 0xf;
  if (bVar2 < 10) {
    bVar2 = bVar2 + 0x30;
  }
  uVar3 = (ushort)bVar2;
  if (('\t' < bVar2) && (bVar2 < 16)) {
    uVar3 = (ushort)(byte)(bVar2 + 'W');
  }
  if (bVar4 < 10) {
    bVar4 = bVar4 + '0';
  }
  if ((9 < bVar4) && (bVar4 < 16)) {
    bVar4 = bVar4 + 87;
  }
  puVar1 = param_3 + 1;
  *param_3 = uVar3;
  param_3 = param_3 + 2;
  *puVar1 = (ushort)bVar4;
  i_4 = i_4 + -1;
  crc_hash = crc_hash + 1;
} while (i_4 != 0);
*param_3 = 0;
return;
```

Inside gen_hash_val

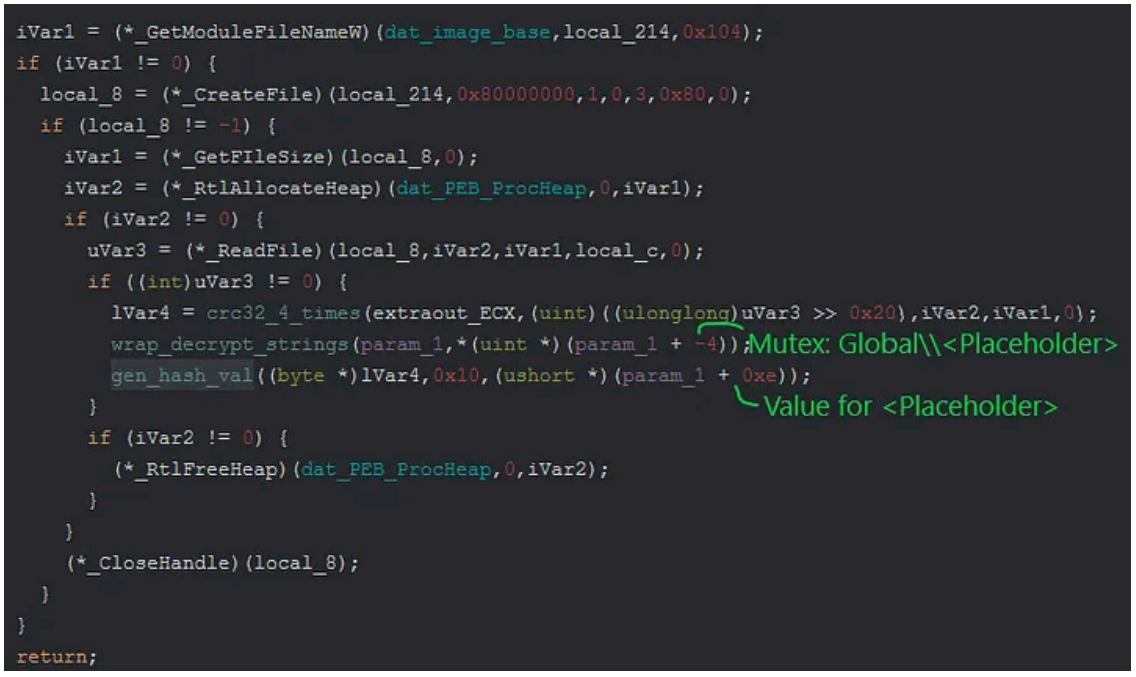
Press enter or click to view image in full size

```
uVar2 = wrap_wrap_dec_strings(&DAT_0040b79a);
iVar1 = (*_RegOpenKeyW)(0x80000002,(int)uVar2,0,0x101,&local_8);
if (iVar1 == 0) {
  local_c = 1;
  local_10 = 0x80;
  uVar3 = wrap_wrap_dec_strings(&DAT_0040b7de);
  iVar1 = (*_RegQueryValueExW)(local_8,(int)uVar3,0,&local_c,local_d0,&local_10);
  if (iVar1 == 0) {
    uVar4 = (*_WideCharToMultiByte)(0,0,local_d0,0xffffffff,local_50,0x40,0,0);
    lVar5 = crc32_4_times(extraout_ECX,(uint)((ulonglong)uVar4 >> 0x20),local_50,(int)uVar4,0);
    lVar5 = crc32_4_times(extraout_ECX_00,(uint)((ulonglong)lVar5 >> 0x20),(int)lVar5,0x10,1);
    lVar5 = crc32_4_times(extraout_ECX_01,(uint)((ulonglong)lVar5 >> 0x20),(int)lVar5,0x10,1);
    lVar5 = crc32_4_times(extraout_ECX_02,(uint)((ulonglong)lVar5 >> 0x20),(int)lVar5,0x10,1);
    *param_1 = 0x2e;
    gen_hash_val((byte *)lVar5,4,param_1 + 1);
  }
  (*_RtlFreeHeap)(dat_PEB_ProcHeap,0,(int)uVar3);
  (*_RegCloseKey)(local_8);
}
(*_RtlFreeHeap)(dat_PEB_ProcHeap,0,(int)uVar2);
return;
```

Machine GUID

File Extension Name

Press enter or click to view image in full size




```
iVar1 = (*_GetModuleFileNameW)(dat_image_base,local_214,0x104);
if (iVar1 != 0) {
  local_8 = (*_CreateFile)(local_214,0x80000000,1,0,3,0x80,0);
  if (local_8 != -1) {
    iVar1 = (*_GetFileSize)(local_8,0);
    iVar2 = (*_RtlAllocateHeap)(dat_PEB_ProcHeap,0,iVar1);
    if (iVar2 != 0) {
      uVar3 = (*_ReadFile)(local_8,iVar2,iVar1,local_c,0);
      if ((int)uVar3 != 0) {
        lVar4 = crc32_4_times(extraout_ECX,(uint)((ulonglong)uVar3 >> 0x20),iVar2,iVar1,0);
        wrap_decrypt_strings(param_1,*(uint *)(param_1 + -4));
        gen_hash_val((byte *)lVar4,0x10,(ushort *)(param_1 + 0xe));
      }
      if (iVar2 != 0) {
        (*_RtlFreeHeap)(dat_PEB_ProcHeap,0,iVar2);
      }
    }
    (*_CloseHandle)(local_8);
  }
}
return;
```

Generator Mutex String

Press enter or click to view image in full size

```
uVar3 = wrap_wrap_dec_strings(&DAT_0040b79a);
uVar1 = (undefined4)uVar3;
iVar2 = (*_RegOpenKeyW)(0x80000002,uVar1,0,0x101,&local_8);
if (iVar2 == 0) {
  local_c = 1;
  local_10 = 0x80;
  uVar3 = wrap_wrap_dec_strings(&DAT_0040b7de);
  iVar2 = (*_RegQueryValueExW)(local_8,(int)uVar3,0,&local_c,local_d0,&local_10);
  if (iVar2 == 0) {
    uVar4 = (*_WideCharToMultiByte)(0,0,local_d0,0xffffffff,local_50,0x40,0,0);
    lVar5 = crc32_4_times(extraout_ECX,(uint)((ulonglong)uVar4 >> 0x20),local_50,(int)uVar4,0);
    gen_hash_val((byte *)lVar5,10,param_1);
    iVar2 = (*_wcslen)(param_1);
    unaff_EBX = iVar2 << 1;
  }
  (*_RtlFreeHeap)(dat_PEB_ProcHeap,0,(int)uVar3);
  (*_RegCloseKey)(local_8);
}
(*_RtlFreeHeap)(dat_PEB_ProcHeap,0);
return CONCAT44(uVar1,unaff_EBX);
```

Victim ID: Get first 10 bytes from CRC32 block of Machine GUID

File Drop

Drop ransomware icon file in %APPDATA% and create Regkey for it.

```
uVar4 = wrap_wrap_dec_strings(&DAT_0040bd9c);
uVar5 = (*_RtlAllocateHeap)(dat_PEB_ProcHeap,0,DAT_0040bd98 << 6);
pbVar1 = (byte *)uVar5;
uVar5 = anther_decryption(extraout_ECX,(int)((ulonglong)uVar5 >> 0x20),(byte *
wrap_CreateFile(local_41c,pbVar1,(int)uVar5);
(*_RtlFreeHeap)(dat_PEB_ProcHeap,0,(byte *)uVar4);
(*_RtlFreeHeap)(dat_PEB_ProcHeap,0,pbVar1);
if (_ImpersonateLoggedOnUser != 0) {
  (*_RevertToSelf)();
}
iVar2 = (*_RegCreateKey)(0x80000000,param_1,0,0,0,0x2000000,0,&local_8,0);
if (iVar2 == 0) {
  iVar2 = (*_wcslen)(iVar3);
  iVar2 = (*_RegSetValueExW)(local_8,&DAT_00410700,0,1,iVar3,iVar2 * 2 + 2);
  if (iVar2 == 0) {
    (*_RegCloseKey)(local_8);
```

Create File -> RegCreateKey -> RegSetValueExW

Service Enumeration and Delete

Enumerate and compare with these services,

```
vss,sql,svc$,memtas,mepocs,sophos,veeam,backup
```

If found then delete the service.

```
local_8 = 0;
local_10 = (undefined4 *)0x0;
local_8 = (*_OpenSCManager)(0,0,4);
if (local_8 != 0) {
  local_14 = 0;
  (*_EnumServiceStatusExW)(local_8,0,0x30,3,0,0,&local_14,&local_18,0,0);
  local_10 = (undefined4 *)(*_RtlAllocateHeap)(dat_PEB_ProcHeap,8,local_14);
  iVar2 = (*_EnumServiceStatusExW)(local_8,0,0x30,3,local_10,local_14,&local_14,&loc
  psVar3 = DAT_004108a4;
  puVar4 = local_10;
  while (DAT_004108a4 = psVar3, iVar2 != 0) {
    bVar1 = false;
    do {
      if (!bVar1) {
        (*_wcslwr)(*puVar4);
        bVar1 = true;
      }
      iVar2 = (*_wcsstr)(*puVar4,psVar3);
      if (iVar2 != 0) {
        uVar5 = (*_OpenService)(local_8,*puVar4,0x10020);
        local_c = (int)uVar5;
        if (local_c != 0) {
          clear_buf(extraout_ECX,(int)((ulonglong)uVar5 >> 0x20),(undefined (*) [16]
                    );
          (*_ControlService)(local_c,1,local_34);
          (*_DeleteService)(local_c);
          (*_CloseServiceHandle)(local_c);
          break;
        }
```

Gather Victim Info

```
uVar6 = get_drive_and_capacity(local_2d8);
if ((int)uVar6 != 0) {
  local_c = 0xf;
  (*_GetUserName)(local_50,&local_c);
  if (local_c != 0) {
    iVar2 = local_c * 2;
    local_c = 0x1f;
    (*_GetComputerName)(local_90,&local_c);
    if (local_c != 0) {
      iVar3 = local_c * 2;
      uVar7 = get_language(local_30);
      if ((int)uVar7 != 0) {
        uVar8 = get_net_info(local_4e0);
        if ((int)uVar8 != 0) {
          uVar9 = get_win_os_ver(local_d0);
          if ((int)uVar9 != 0) {
            uVar10 = get_machine_guid(&DAT_004108e8);
            if ((int)uVar10 != 0) {
              uVar11 = get_sys_arch(extraout_ECX,(uint)((ulonglong)uVar10
```

Victim’s info gather function

```
{
    "bot":{
    "ver":"1.8.5.8",
    "uid":"<>"
    },
    "os":{
    "lang":"<>",
    "username":"<>",
    "hostname":"<>",
    "domain":"<>",
    "os_type":"<>",
    "os_version":"<>",
    "os_arch":">?",
    "disks":"<>",
    "id":"<>"
    }
}
```

Output of Victim’s Info

Get DriverType & Size

```
uVar1 = (*_GetLogicalDriveStringsW)(0x104,local_21c);
if (uVar1 != 0) {
  root_path_name = local_21c;
  uVar1 = uVar1 >> 2;
  puVar4 = param_1;
  do {
    drive_type_code = (*_GetDriveTypeW)(root_path_name);
    if (((drive_type_code == 3) || (drive_type_code == 2)) &&
       (drive_type_code = (*_GetFreeDiskSpaceW)(root_path_name,0,&remaining,&total)
        drive_type_code != 0)) {
      *puVar4 = *root_path_name;
      uVar2 = (*_alldiv)(remaining,local_8,0x40000000,0);
      uVar2 = (*_alldiv)(total,local_10,0x40000000,0,uVar2);
      drive_type_code = (*_swprintf)(puVar4 + 1,u_%u/%u|_004106c0,uVar2);
      puVar4 = (undefined4 *)((int)(puVar4 + 1) + drive_type_code * 2);
    }
    root_path_name = root_path_name + 2;
    uVar1 = uVar1 - 1;
  } while (uVar1 != 0);
  puVar3 = (undefined2 *)(*_wcsrchr)(param_1,'|');
  *puVar3 = 0;
  drive_type_code = (*_wcslen)(param_1);
  uVar1 = drive_type_code << 1;
}
return CONCAT44(unaff_EDI,uVar1);
```

Format “<Drive Name>:<Remaining Disk Space>/<Total Disk Space>” e.g. C:30/50

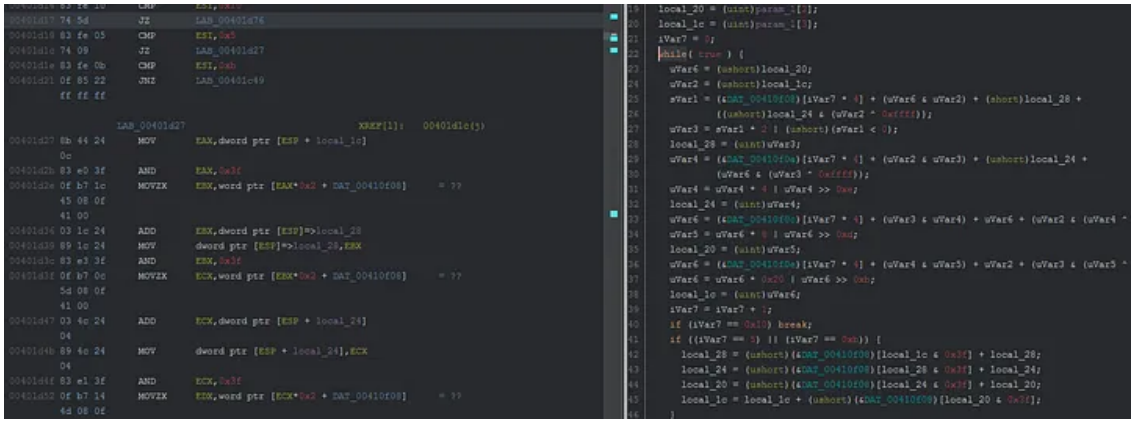
Language

> HKCU = HKEY_CURRENT_USER = 0*x80000001* *Details*

HKCU/Control Panel/Desktop/MuiCached/MachinePreferredUILanguage

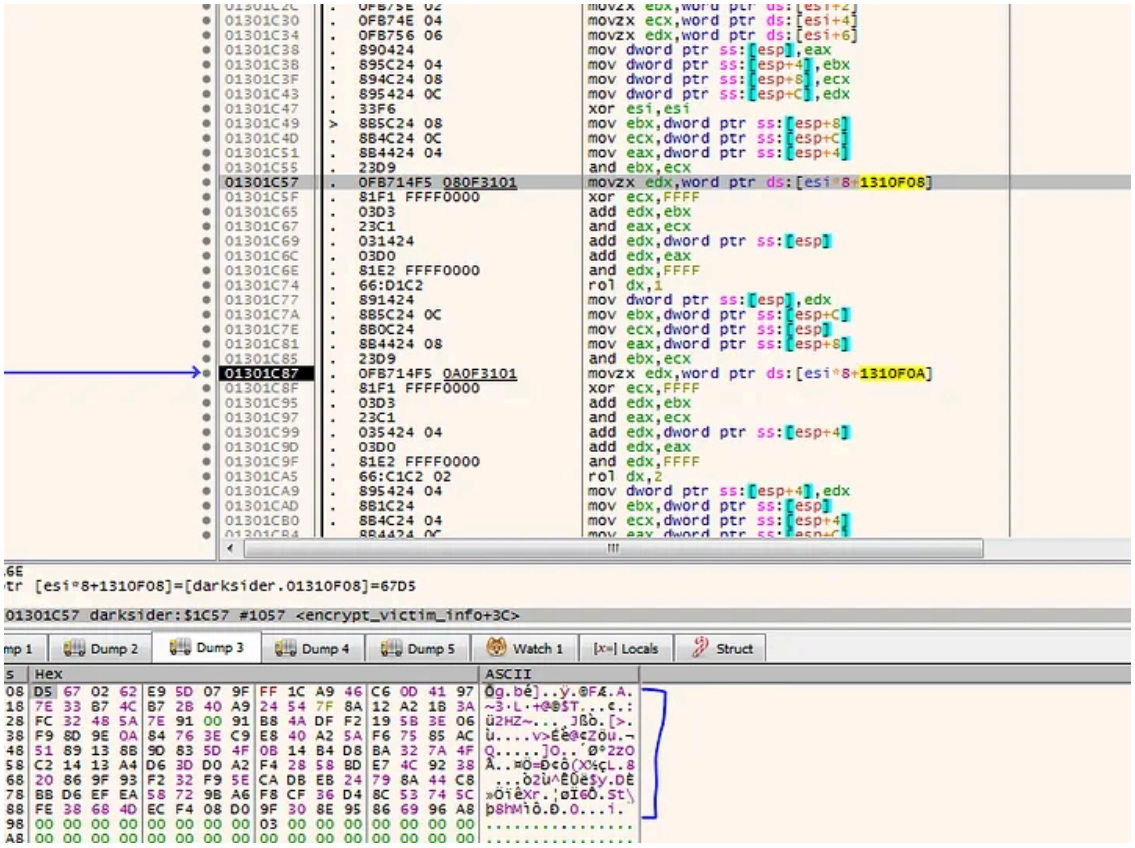
Encrypt Victim’s Info

Press enter or click to view image in full size



Encryption Routine: Encrypt 8 bytes for one function call

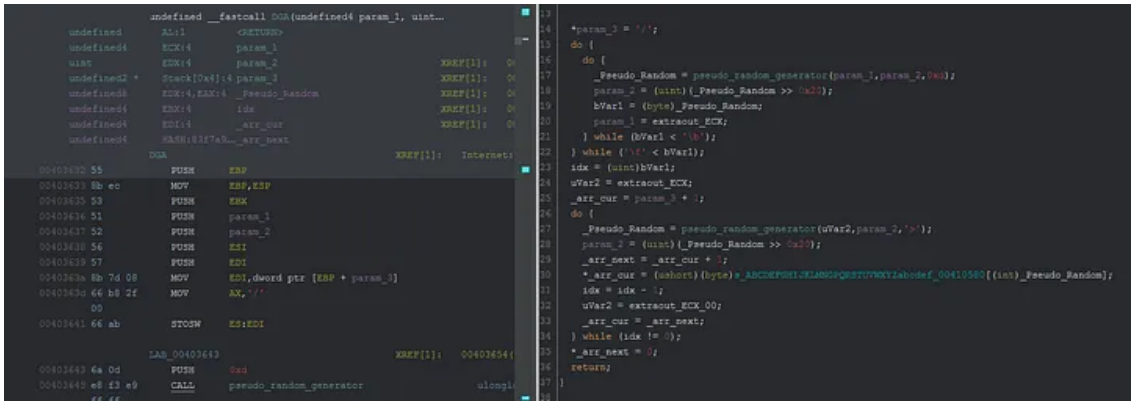
Press enter or click to view image in full size



Encryption Key maybe? 🤔

URL Path Generator

Press enter or click to view image in full size



URL Path Generator function

Press enter or click to view image in full size

```
ulonglong __fastcall pseudo_random_generator(undefined4 param_1,uint param_2,uint param_3)

{
  int iVar1;

  iVar1 = wrap_random_ex();
  return (ulonglong)(iVar1 * 0x41c64e6d + 0x3039U & 0x7fffffff) % (ulonglong)param_3 |
         (ulonglong)param_2 << 0x20;
}
```

Pseudo Random

```
void srand(uint32_t seed) {
        size_t i;

        for (i = 0; i < 25; i++) {
                seed = seed * 0x41C64E6D + 0x3039;
                rand_state[i] = seed ^ rand_salt[i];

                /* on the off chance... */
                if (rand_state[i] == 0) {
                        rand_state[i] = rand_salt[i];
                }
        }

        rand_regen();
}
```

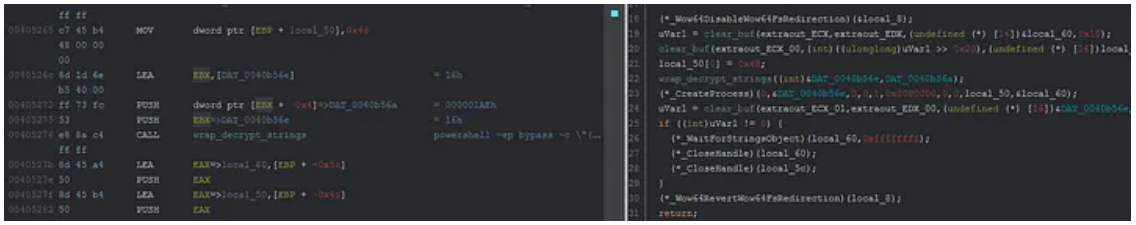
Psuedo random generator similar with srand code

Internet connection

securebestapp20[.]com/<URL Path Generator>

Encrypted Powershell runs “delete shadow copy”

Press enter or click to view image in full size



Ok, bye shadow copy

Salsa session key generation & RSA encryption on Salsa session key

The session key generated from the `RtlRandomEx` function which feeds with a hard coded seed value. The when the length == 5 it will leave 0 bytes there. (Refer to “Custom Salsa key state arrangement”)

```
void gen_key_state_salsa(int key_state_buf)

{
  int Max_length;
  undefined8 uVar1;

  Max_length = 8;
  do {
    uVar1 = wrap_random_ex();
    if (Max_length == 5) {
      uVar1 = 0;
    }
    *(int *)(key_state_buf + -4 + Max_length * 8) = (int)uVar1;
    *(int *)(key_state_buf + -8 + Max_length * 8) = (int)((ulonglong)uVar1 >> 0x20);
    Max_length = Max_length + -1;
  } while (Max_length != 0);
  return;
}
```

Salsa session key generator

```
void wrap_random_ex(void)

{
  if (_seed == 0) {
    (*_RtlRandomEx)(&seed);
  }
  (*_RtlRandomEx)(&seed);
  return;
}
```

RtlRandomEx inside wrap_random_ex()

```
00E46711  .  8D43 34        lea eax,dword ptr ds:[ebx+34]
00E46714  .  50             push eax
00E46715  .  E8 52B9FFFF    call <darksider.gen_key_stat_salsa>
00E4671A  .  6A 40          push 40
00E4671C  .  8D43 34        lea eax,dword ptr ds:[ebx+34]
00E4671F  .  50             push eax
00E46720  .  8D43 74        lea eax,dword ptr ds:[ebx+74]
00E46723  .  50             push eax
00E46724  .  E8 79ADFFFF    call <darksider.buf_cpy>
00E46729  .  8D0D 8807E500  lea ecx,dword ptr ds:[E50788]
00E4672F  .  8D81 80000000  lea eax,dword ptr ds:[ecx+80]
00E46735  .  50             push eax
00E46736  .  8D01           lea eax,dword ptr ds:[ecx]
00E46738  .  50             push eax
00E46739  .  8D43 74        lea eax,dword ptr ds:[ebx+74]
00E4673C  .  50             push eax
00E4673D  .  E8 F1BFFFFF    call <darksider.rsa>
00E46742  .  6A 00          push 0
00E46744  .  68 80000000    push 80
00E46749  .  8D43 74        lea eax,dword ptr ds:[ebx+74]
00E4674C  .  50             push eax
00E4674D  .  E8 C2B6FFFF    call <darksider.crc_calc>
00E46752  .  6A 10          push 10
00E46754  .  50             push eax
00E46755  .  8D83 F4000000  lea eax,dword ptr ds:[ebx+F4]
00E4675B  .  50             push eax
00E4675C  .  E8 41ADFFFF    call <darksider.buf_cpy>
```

Flow of the keygen -> rsa encrypt -> crc -> result buffer copy

How to identify Salsa encryption algorithm?

Get GhouLSec's stories in your inbox

Found these pattern inside the code instead of its constant.

```
b ^= (a + d) <<< 7;
c ^= (b + a) <<< 9;
d ^= (c + b) <<< 13; (0xd in Hex)
a ^= (d + c) <<< 18; (0x12 in Hex)
```

```
MOV     ESI,EAX
ADD     ESI,EDX
ROL     ESI,0x7
XOR     EBX,ESI
MOV     ESI,EBX
ADD     ESI,EAX
ROL     ESI,0x9
XOR     ECX,ESI
MOV     ESI,ECX
ADD     ESI,EBX
ROL     ESI,0xd
XOR     EDX,ESI
MOV     ESI,EDX
ADD     ESI,ECX
ROL     ESI,0x12
XOR     EAX,ESI
```

Yay, same pattern 🤗

Let's check out the key generated. Hmm… There is no constant found for the Salsa Key generated.

```
7F 57 3B 0C 05 C4 1B 56 32 D0 40 05 4D 87 FA 06
8B 81 76 DE A7 4E CA 26 0A 4D 65 92 81 34 11 4C
00 00 00 00 00 00 00 00 3C C7 D7 15 38 9C 3C 2D
FF 89 48 D1 3D 3D 8F 44 A6 49 2E AB 59 40 1A 26
```

Custom Salsa key state arrangement

Initial state of Salsa20

"expa"	Key	Key	Key
Key	"nd 3"	Nonce	Nonce
Pos.	Pos.	"2-by"	Key
Key	Key	Key	"te k"

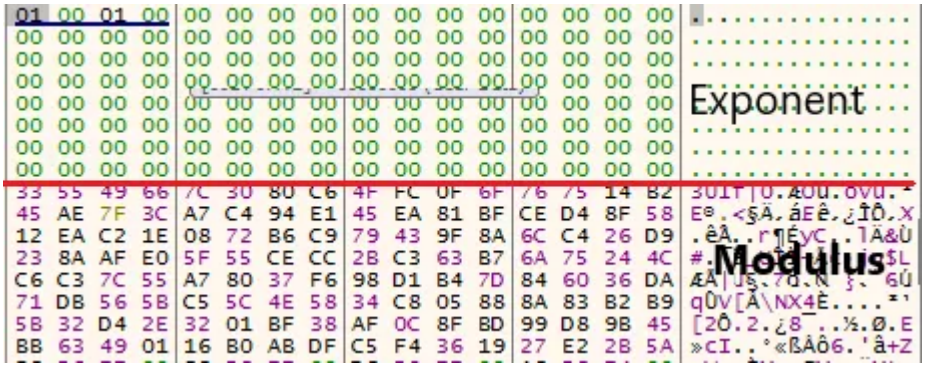
Default Salsa key state arrangement

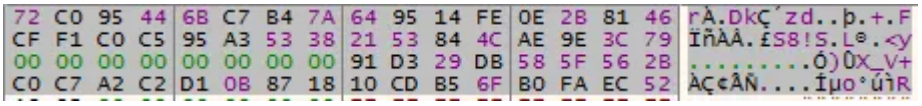
Usually “expa”, “nd 3”, “2-by”, “te k” were seen in Salsa implementation but this seems like a custom one.

RSA Public Ley Encryption

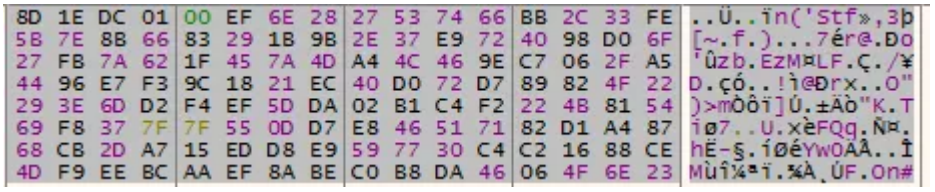
How to determine RSA?

- Knowing the exponential (010010h LE) (10001h BE)
- Guessing the Exponential function (Here is good explanation regarding to the RSA algo)





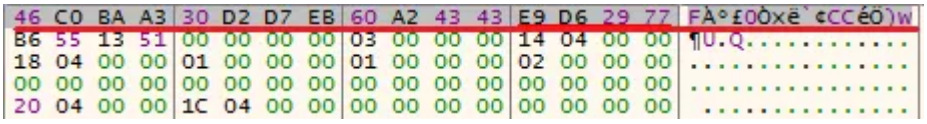
Before RSA encryption



After RSA encryption

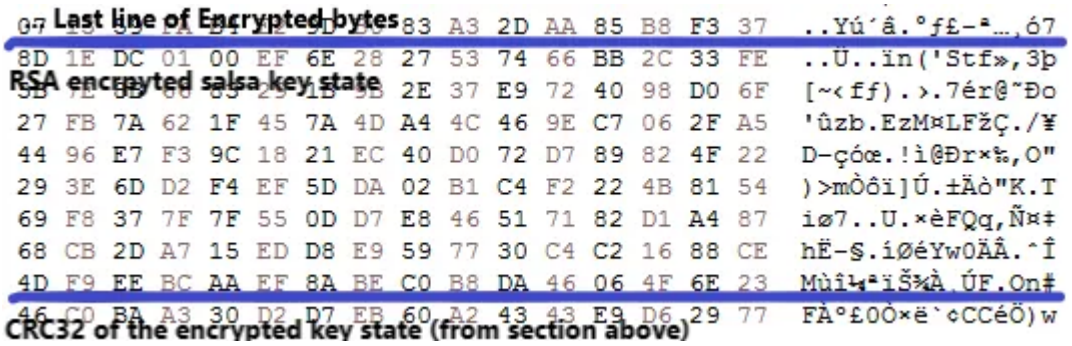
As for details like exponential and modulo function, I still cant figure it out yet. However, feels like the `rcl`, `sbb` and `adc` plays an important role both exponential and modulo operation. Maybe someone can figure this out. 🤔

Generates 16 bytes block hash by using RtlComputeCrc32.



16 bytes CRC32 block from Encrypted Salsa Key

After encrypted the byte. It will append the byte with the encrypted key and its CRC32 hash.



Encrypted file format

Excluded Folder, File and Extension

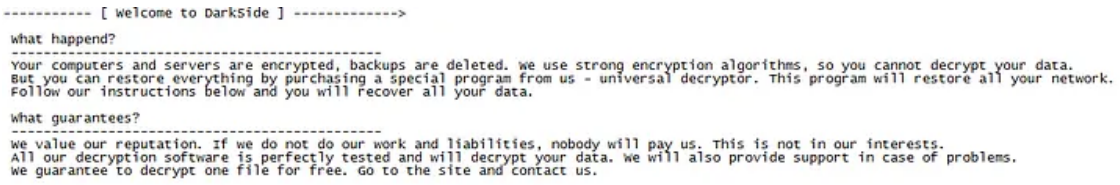
`$recycle.bin config.msi $windows.~bt $windows.~ws windows appdata application data boot google mozilla program files program files (x86) programdata system volume information tor browser windows.old intel msocache perflogs x64dbg public all users default`

`autorun.inf boot.ini bootfont.bin bootsect.bak desktop.ini iconcache.db ntldr ntuser.dat ntuser.dat.log ntuser.ini thumbs.db`

`386 adv ani bat bin cab cmd com cpl cur deskthemepack diagcab diagcfg diagpkg dll drv exe hlp icl icns ico ics idx ldf lnk mod mpa msc msp msstyles msu nls nomedia ocx prf ps1 rom rtp scr shs spl sys theme themepack wpx lock key hta msi pdb`

Ransomnote

Press enter or click to view image in full size



```
----------- [ Welcome to DarkSide ] ------------->

What happend?
---------------------------------------------
Your computers and servers are encrypted, backups are deleted. We use strong encryption algorithms, so you cannot decrypt your data.
But you can restore everything by purchasing a special program from us - universal decryptor. This program will restore all your network.
Follow our instructions below and you will recover all your data.

What guarantees?
---------------------------------------------
We value our reputation. If we do not do our work and liabilities, nobody will pay us. This is not in our interests.
All our decryption software is perfectly tested and will decrypt your data. We will also provide support in case of problems.
We guarantee to decrypt one file for free. Go to the site and contact us.
```

Sha256

afb22b1ff281c085b60052831ead0a0ed300fac0160f87851dacc67d4e158178

References:

Buy me a Pizza 🍕?

Source: https://ghoulsec.medium.com/mal-series-13-darkside-ransomware-c13d893c36a6