



Win32/Duqu: involution of Stuxnet

Aleksandr Matrosov

Eugene Rodionov



ИНВОЛЮ́ЦИЯ (от лат. *involutio* — свёртывание) — редукция или утрата в процессе эволюции отдельных органов, упрощение их организации и функций









14.10

19.10

01.11

03.11

4.11

?

CrySyS Lab
share info
about Duqu
on public

Duqu: the
precursor
to the next
Stuxnet

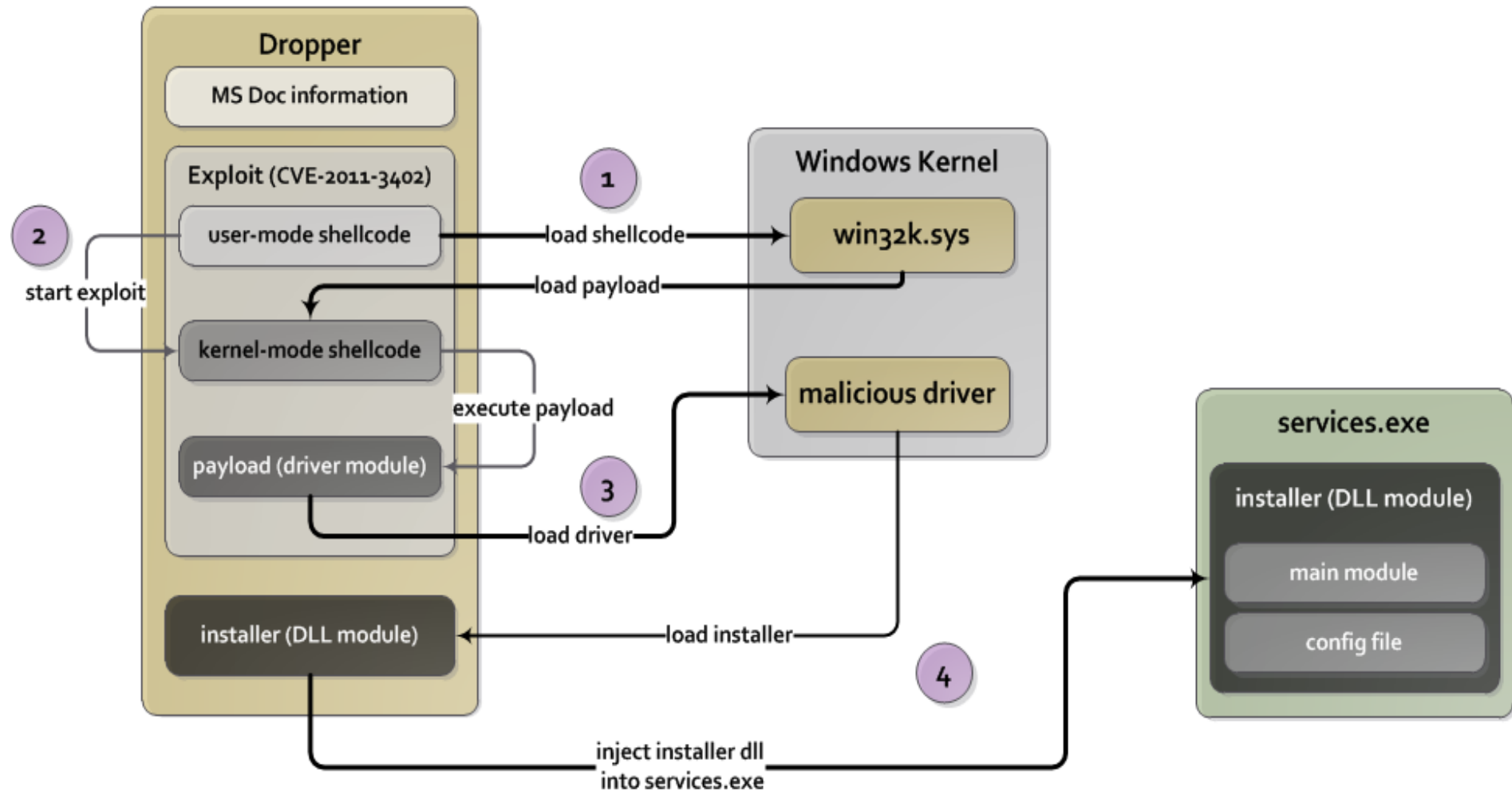
Dropper
found and
0-day
confirmed
CVE-2011-3402

Microsoft
Security
Advisory
(2639658)

MS share
info about
exploit on
MAPP

What the
next?

Duqu infection scheme



primary

```

10006dc1  mov     eax, sub_1001B487
10006dc5  call   __int_prolog
10006dc8  push   ecx
10006dcc  push   ecx
10006dd4  push   ebx
10006ddc  push   esi
10006ddf  push   edi
10006de0  mov     ss:[ebp+ivar_10], esp
10006de3  mov     esi, ecx
10006de5  xor     eax, eax
10006de7  xor     ebx, ebx
10006de9  cmp    b1 ds:[esi], b1 b1
10006deb  setz   b1 al
10006ede  cmp    eax, ebx
10006ef0  jnz    loc_100060F7

```

```

10006de2  mov     ss:[ebp+ivar_14], 5
10006de9  push   sub_1001FAF8
10006dee  lea   eax, ss:[ebp+ivar_14]
10006df1  push   eax
10006df2  call  __CxxThrowException(x,x)

```

```

10006df7  cmp    b1 ss:[ebp+clientServer ], b1 b1
10006dfa  jnz    loc_10006E01

```

```

10006dfc  call  __RpcPunregister

```

```

10006e01  push   ebx
10006e02  push   ebx
10006e03  push   sub_1001E850
10006e08  call  ds:[RpcServerUnregisterif ]
10006e0e  mov   ss:[ebp+ivar_4], ebx
10006e11  call  __RpcServerRegisterif
10006e16  test  b1 al, b1 al
10006e18  jz    loc_10006E30

```

```

10006e1a  push   ss:[ebp+clientServer ]
10006e1d  mov   edi, esi
10006e1f  call  RpcServerInitialize
10006e24  test  b1 al, b1 al
10006e26  jz    loc_10006E30

```

```

10006e28  lea   ecx, ds:[esi+4]
10006e2b  mov   ds:[ecx], ecx
10006e2d  call  ds:[eax+4]
10006e30  cmp    b1 ss:[ebp+clientServer ], b1 b1
10006e33  jnz   loc_10006E3A

```

```

10006e35  call  InjectIntoServicesAndCallExport7

```

```

10006e3a  mov   b1 ds:[esi], b1 1

```

```

10006e3d  or    ss:[ebp+ivar_4], 0xFFFFFFFF
10006e41  mov   ecx, ss:[ebp+ivar_C]
10006e44  mov   fs:[0], ecx
10006e4b  pop   edi
10006e4c  pop   esi
10006e4d  pop   ebx
10006e4e  leave
10006e4f  retn  b2 4

```

secondary

```

100195a5  mov     eax, unk000011bname_705
100195a9  call   __int_prolog
100195af  sub   esp, 0xc0
100195b2  push   ebx
100195b3  push   esi
100195b4  push   edi
100195b5  mov     ss:[ebp+ivar_10], esp
100195b8  mov     esi, ecx
100195ba  xor     ebx, ebx
100195bc  mov   b1 ss:[ebp+ivar_11], b1 b1
100195bf  xor     eax, eax
100195c1  cmp    b1 ds:[esi], b1 b1
100195c3  setz   b1 al
100195c6  cmp    eax, ebx
100195c8  jnz    loc_100195E6

```

```

100195ca  mov     ss:[ebp+ivar_18], 5
100195d1  mov     ss:[ebp+ivar_1c], off_10062398
100195d8  push   sub_10068304
100195dd  lea   eax, ss:[ebp+ivar_1c]
100195e0  push   eax
100195e1  call  __CxxThrowException(x,x)

```

```

100195e6  cmp    b1 ss:[ebp+clientServer ], b1 b1
100195e9  jnz    loc_100195F0

```

```

100195eb  call  __RpcPunregister

```

```

100195f0  push   ebx
100195f1  push   ebx
100195f2  push   RpcHandle
100195f7  call  ds:[RpcServerUnregisterif ]
100195fd  mov   ss:[ebp+ivar_4], ebx
10019600  call  __RpcServerRegisterif
10019605  test  b1 al, b1 al
10019607  jz    loc_1001962f

```

```

10019609  push   ss:[ebp+clientServer ]
1001960c  push   esi
1001960d  call  RpcServerInitialize
10019612  test  b1 al, b1 al
10019614  jz    loc_1001962f

```

```

10019616  lea   ecx, ds:[esi+4]
10019619  mov   ds:[ecx], ecx
1001961b  call  ds:[eax+4]
1001961e  cmp    b1 ss:[ebp+clientServer ], b1 b1
10019621  jnz   loc_10019628

```

```

10019623  call  InjectIntoServicesAndCallExport7

```

```

10019628  mov   b1 ss:[ebp+ivar_11], b1 1
1001962c  mov   b1 ds:[esi], b1 1

```

```

1001962f  or    ss:[ebp+ivar_4], 0xFFFFFFFF
10019633  movzx eax, b1 ss:[ebp+ivar_11]
10019637  push  eax
10019638  push  dword_1005C30C
1001963d  call  __urlCreateFromPath(evlev_ssd094
10019642  pop   ecx
10019643  pop   ecx
10019644  mov   ecx, ss:[ebp+ivar_C]
10019647  mov   fs:[0], ecx
1001964e  pop   edi
1001964f  pop   esi
10019650  pop   ebx
10019651  leave
10019652  retn  b2 4

```


RPC Function	Stuxnet	Duqu
Rpc 1 – return version of the worm	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rpc 2 – load module in into a new process and execute export function	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rpc 3 – load module into existing process and execute export #1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rpc 4 – load module in a process and execute its entry point	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rpc 5 – Build the worm dropper	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rpc 6 – run specified application (calling CreateProcess API)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rpc 7 – read data from specified file	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rpc 8 – write data into specified file	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rpc 9 – delete specified file	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rpc 10 – work with target files	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Finding exact date of infection

```
EnterCfgData(&CfgMutex);
v8 = 0;
bDeleteItself = GetConfigBuffer()->bCheckSelfDelete;
v8 = -1;
result = LeaveCfgData(&CfgMutex);
if ( !bDeleteItself )
{
    EnterCfgData(&_CfgMutex);
    v8 = 1;
    EnterCfgData(&CfgMutex);
    LOBYTE(v8) = 2;
    v3 = GetConfigBuffer();
    v4 = GetConfigBuffer();
    bTimeElapsed = IsTimeElapsed(&v4->InfectionDate, v3->DaysToLive); // check time to remove itself from the system
    LOBYTE(v8) = 1;
    LeaveCfgData(&CfgMutex);
    v8 = -1;
    result = LeaveCfgData(&_CfgMutex);
    if ( bTimeElapsed )
        result = InjectAndExecuteExportFunction(2, 0); // call export 2 (remove itself from the system)
```

Config decryption algorithm

```
def decrypt(data):  
    gamma = [0x2b, 0x72, 0x73, 0x34, 0x99, 0x71, 0x98, 0xAE]  
  
    a = 0  
    for ix in xrange(len(data)):  
        data[ix] ^= gamma[a]  
        a = a + 1  
        if a == 7:  
            a = 0
```

Finding date in UTC format

11/08/2011 at 07:50:01

```
00001210: 00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00001220: 00 00 00 00.00 00 00 00.00 00 01 00.00 00 01 00
00001230: 00 00 01 00.00 00 01 00.00 00 38 31.00 00 40 13
00001240: 52 46 FB 57 CC 01 01 00.00 00 00 00.00 00 22 00
00001250: 00 00 00 00.9D 22 9F 3D CB 01 BC 02.00 00 00 00
00001260: 00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00001270: 00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
```

time of infection

days to live

36

18/08/2011 at 07:29:07

```
00001220: 00 00 00 00.00 00 00 00.00 00 01 00.00 00 01 00
00001230: 00 00 01 00.00 00 01 00.00 00 38 31.00 00 11 4E
00001240: 8D 83 78 5D CC 01 01 00.00 00 00 00.00 00 00 00
00001250: 00 00 00 80.51 9C BE A2 C5 01 FA 19.00 00 4A 81
00001260: 84 94 03 00.00 00 0B 00.00 00 14 00.00 00 00 00
00001270: 00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
```

time of infection

days to live

30



PLEASE STAND BY

References

✓ “Win32/Duqu: It’s A Date”

<http://blog.eset.com/2011/10/25/win32duqu-it’s-a-date>

✓ “Stuxnet Under the Microscope”

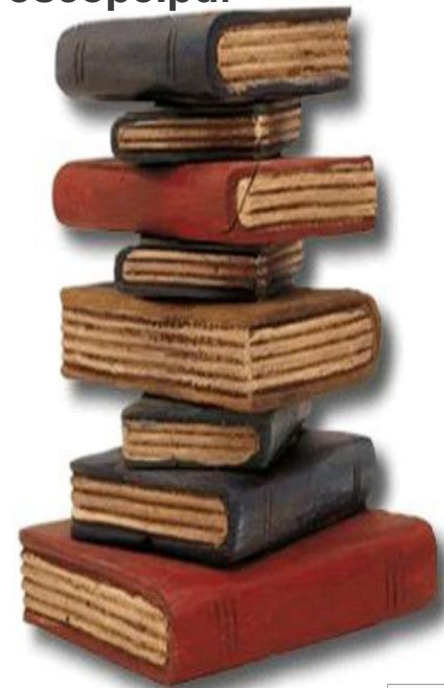
http://go.eset.com/us/resources/white-papers/Stuxnet_Under_the_Microscope.pdf

✓ “Win32/Duqu analysis: the RPC edition”

<http://blog.eset.com/2011/10/28/win32duqu-analysis-the-rpc-edition>

✓ Follow ESET Threat Blog

<http://blog.eset.com>



Thank you for your attention ;)



Aleksandr Matrosov

matrosov@eset.sk

@matrosov

Eugene Rodionov

rodionov@eset.sk

@vxradius

The ESET logo, consisting of the word 'eset' in white lowercase letters on a green rounded rectangular background, is suspended from the helicopter by a thin wire.