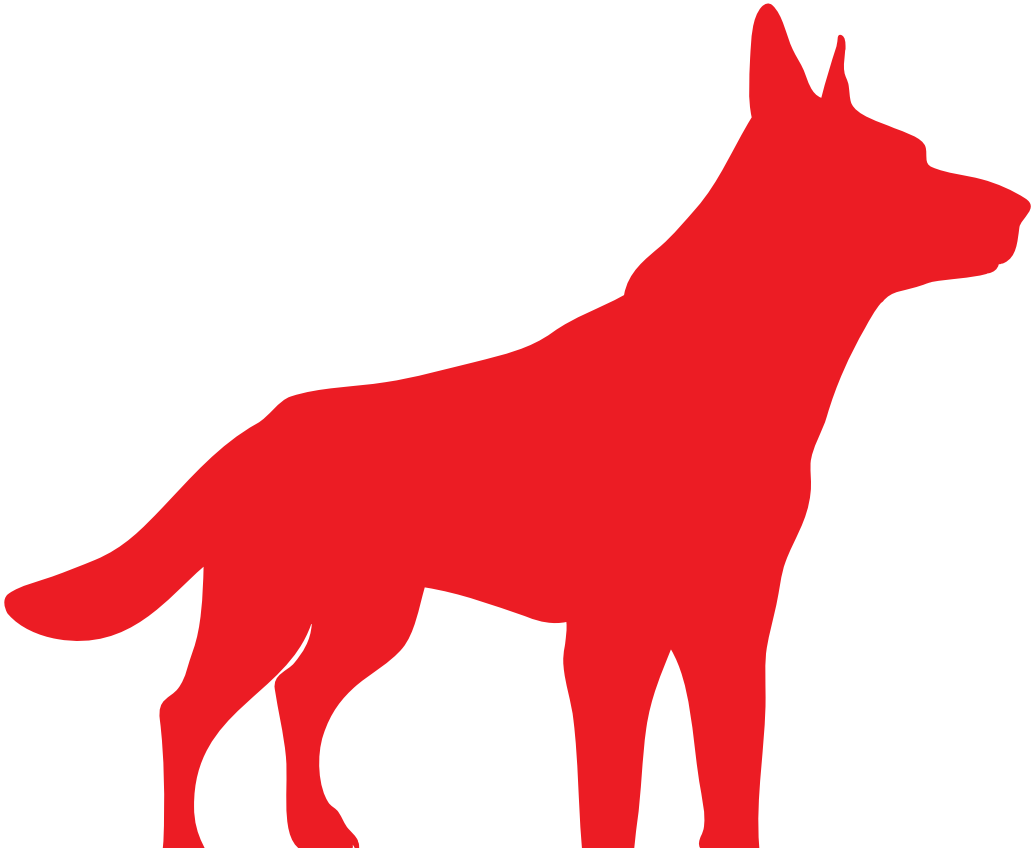
Uroburos

Paul Rascagneres
Senior Threat Researcher
GDATA SecurityLabs



Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Uroburos is a rookit revealed to the public by G DATA in February 2014. The purpose of the rootkit is to maintain remote access to the infected machine and steal sensitive data.

Here are the features of this rootkit:

- use of function hooking, to hide its activities
- Deep Packet Inspection (DPI), to monitor the network
- bypass kernel protection, to load and execute the driver
- use of virtual file system, to store configuration and data
- …

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Uroburos' name

Uroburos is a direct reference to the Greek word Ouroboros (Οὐροβόρος). The Ouroboros is an ancient symbol depicting a serpent or dragon eating its own tail.

```
80FA FFFF E0C9 B909 80FA FFFF 30CE B909 80FA FFFF 2C62 B909  €úÿÿÄÈ¹.€úÿÿàÉ¹.€úÿÿÒÎ¹.€úÿÿ,b¹.
80FA FFFF 2CEE B909 80FA FFFF D4CB B909 80FA FFFF 54C6 B909  €úÿÿ`Ä¹.€úÿÿ,î¹.€úÿÿÔÊ¹.€úÿÿTÆ¹.
5572 3062 5572 2829 7347 6F54 794F 7523 0000 0000 0000 0000  €úÿÿ....UrObUr()sGoTyOu#.......
80FA FFFF 2C41 B509 80FA FFFF 34CC B909 80FA FFFF FCCC B909  ´... X¹.€úÿÿ,Aµ.€úÿÿ4Ì¹.€úÿÿüÌ¹.
80FA FFFF 08CD B909 80FA FFFF 94CD B909 80FA FFFF D0EF B909  €úÿÿÐÐ¹.€úÿÿ.Í¹.€úÿÿ"Í¹.€úÿÿÐï¹.
80FA FFFF 58D3 B909 80FA FFFF 38D8 B909 80FA FFFF C4DD B909  €úÿÿ Î¹.€úÿÿXÓ¹.€úÿÿ8Ø¹.€úÿÿÄÝ¹.
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Rootkit composition

The rootkit is composed of two files:

- .sys file (the Microsoft Windows driver 32/64 bits)
- .dat file (the encrypted virtual file system)

Uroburos rootkit

The driver

The loaded driver:

```
kd> !pool 859e84f0
Pool page 859e84f0 region is Nonpaged pool
*85980000 : large page allocation, Tag is NtFs, size is 0x92000 bytes
                Pooltag NtFs : StrucSup.c, Binary : ntfs.sys
kd> db 85980000 L0x100
85980000  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
85980010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
85980020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
85980030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
85980040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
85980050  00 00 00 00 00 00 00 00-61 6d 20 63 61 6e 6e 6f  ........am canno
85980060  74 20 62 65 20 72 75 6e-20 69 6e 20 44 4f 53 20  t be run in DOS 
85980070  6d 6f 64 65 2e 0d 0d 0a-24 00 00 00 00 00 00 00  mode....$.......
85980080  00 f8 30 25 44 99 5e 76-44 99 5e 76 44 99 5e 76  ..0%D.^vD.^vD.^v
85980090  44 99 5f 76 94 99 5e 76-1d ba 4d 76 4d 99 5e 76  D._v..^v..MvM.^v
859800a0  63 5f 23 76 46 99 5e 76-63 5f 2f 76 31 99 5e 76  c_#vF.^vc_/v1.^v
859800b0  63 5f 26 76 45 99 5e 76-52 69 63 68 44 99 5e 76  c_&vE.^vRichD.^v
859800c0  00 00 00 00 00 00 00 00-50 45 00 00 4c 01 04 00  ........PE..L...
859800d0  95 41 04 4e 00 00 00 00-00 00 00 00 e0 00 02 21  .A.N...........!
859800e0  0b 01 08 00 00 8e 06 00-00 62 02 00 00 00 00 00  .........b......
859800f0  e0 d2 00 00 00 10 00 00-00 90 06 00 00 00 01 00  ................
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

The driver

The loaded driver:

```
kd> !object \driver\
Object: 8985ea70  Type: (84841e90) Directory
    ObjectHeader: 8985ea58 (new version)
    HandleCount: 0  PointerCount: 92
    Directory Object: 89805e28  Name: Driver

    Hash Address  Type          Name
    ---- -------  ----          ----
     00  85ae0530 Driver        rdpbus
         8576a1d8 Driver        Beep
         855b74b0 Driver        NDIS
         [...]
         85a3d310 Driver        Wanarpv6
     28  85a51030 Driver        discache
         8576a3f8 Driver        Null
     29  85a7aa38 Driver        VBoxVideo
         [...]
         855e6610 Driver        rdyboost
         8487e780 Driver        intelide
```

Uroburos rootkit

The driver

The loaded driver:

```
kd> !drvobj \Driver\Null
Driver object (8576a3f8) is for:
 \Driver\Null
Driver Extension List: (id , addr)

Device Object list:
864473e0  862531e0  86253748  8576a2d0
```

```
kd> !devobj 864473e0
Device object (864473e0) is for:
 FWPMCALLOUT \Driver\Null DriverObject 8576a3f8
Current Irp 00000000 RefCount 0 Type 00000000 Flags 000000c0
Dacl 8985aaf0 DevExt 00000000 DevObjExt 86447498
ExtensionFlags (0x00000800)  DOE_DEFAULT_SD_PRESENT
Characteristics (0000000000)
Device queue is not busy.

kd> !devobj 0x862531e0
Device object (862531e0) is for:
 RawDisk2 \Driver\Null DriverObject 8576a3f8
Current Irp 00000000 RefCount 0 Type 00000007 Flags 00000050
Vpb 86253158 DevExt 00000000 DevObjExt 86253298 Dope 86257008
ExtensionFlags (0x00000800)  DOE_DEFAULT_SD_PRESENT
Characteristics (0x00000001)  FILE_REMOVABLE_MEDIA
Device queue is not busy.

kd> !devobj 86253748
Device object (86253748) is for:
 RawDisk1 \Driver\Null DriverObject 8576a3f8
Current Irp 00000000 RefCount 22 Type 00000007 Flags 00000050
Vpb 862536c0 DevExt 00000000 DevObjExt 86253800 Dope 86253678
ExtensionFlags (0x00000800)  DOE_DEFAULT_SD_PRESENT
Characteristics (0x00000001)  FILE_REMOVABLE_MEDIA
Device queue is not busy.

kd> !devobj 8576a2d0
Device object (8576a2d0) is for:
 Null \Driver\Null DriverObject 8576a3f8
Current Irp 00000000 RefCount 0 Type 00000015 Flags 00000040
Dacl 8985aaf0 DevExt 00000000 DevObjExt 8576a388
ExtensionFlags (0x00000800)  DOE_DEFAULT_SD_PRESENT
Characteristics (0x00000100)  FILE_DEVICE_SECURE_OPEN
Device queue is not busy.
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Hooking

To hide its activity and its presence, the driver sets several hooks by modifying the beginning of the function with an interrupt (0x3C):

```
kd> ? IoCreateDevice
Evaluate expression: -2103684120 = 829c53e8
kd> u 829c53e8
nt!IoCreateDevice:
829c53e8 6a01            push    1
829c53ea cdc3            int     0C3h
829c53ec ec              in      al,dx
829c53ed 83e4f8          and     esp,0FFFFFFF8h
829c53f0 81ec94000000    sub     esp,94h
829c53f6 a14cda9282      mov     eax,dword ptr [nt!__security_cookie (8292da4c)]
829c53fb 33c4            xor     eax,esp
829c53fd 89842490000000  mov     dword ptr [esp+90h],eax
```

TRUST IN GERMAN SICHERHEIT

Hooking

The Interrupt Descriptor Table (idt):

```
kd> !idt
Dumping IDT: 80b95400

3194895000000030:          82c27ca4 hal!Halp8254ClockInterrupt (KINTERRUPT …)
3194895000000031:          8486b058 i8042prt!I8042KeyboardInterruptService (KINTERRUPT
3194895000000038:          82c18c6c hal!HalpRtcProfileInterrupt (KINTERRUPT …)
3194895000000039:          8486bcd8 ACPI!ACPIInterruptServiceRoutine (KINTERRUPT …)
319489500000003a:          85afd7d8 ndis!ndisMiniportIsr (KINTERRUPT 85afd780)
319489500000003b:          8486b558 ataport!IdePortInterrupt (KINTERRUPT 8486b500)
319489500000003c:          85afdcd8 i8042prt!I8042MouseInterruptService (KINTERRUPT…)
319489500000003e:          8486ba58 ataport!IdePortInterrupt (KINTERRUPT 8486ba00)
319489500000003f:          8486b7d8 ataport!IdePortInterrupt (KINTERRUPT 8486b780)
31948950000000c3:          859e84f0
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Hooking

Code available at 0x859e84f0:

```
kd> u 859e84f0  L0x16
859e84f0 90              nop
859e84f1 90              nop
859e84f2 90              nop
859e84f3 90              nop
859e84f4 90              nop
859e84f5 90              nop
859e84f6 90              nop
859e84f7 90              nop
859e84f8 90              nop
859e84f9 90              nop
859e84fa 90              nop
859e84fb 90              nop
859e84fc 90              nop
859e84fd 90              nop
859e84fe 90              nop
859e84ff 90              nop
859e8500 6a08            push    8
859e8502 6808859e85      push    859E8508h
859e8507 cb              retf
859e8508 fb              sti
859e8509 50              push    eax
859e850a 51              push    ecx
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Hooking

Python script to list the hooks:

```
import pykd

output = pykd.dbgCommand("x nt!*").split("\n")
for i in output:
  if i != "":
    addr=i.split()[0]
    name=i.split()[1]
    opcode=pykd.dbgCommand("db %(addr)s+2 L2" % {'addr': addr}).split()
    if (opcode[1] == "cd") and (opcode[2] == "c3"):
      print "Hook: "+name
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Hooking

The list of the ntoskrnl.exe hooked functions (the hooked feature):

```
nt!NtCreateKey                (registry)
nt!NtQueryInformationProcess  (process)
nt!NtQuerySystemInformation   (system information)
nt!ObOpenObjectByName         (driver)
nt!NtClose                    (file/process/event/…)
nt!IoCreateDevice             (driver)
nt!NtEnumerateKey             (registry)
nt!NtShutdownSystem           (system)
nt!NtTerminateProcess         (process)
nt!IofCallDriver              (driver)
nt!NtQueryKey                 (registry)
nt!NtCreateUserProcess        (process)
nt!NtCreateThread             (process)
nt!NtSaveKey                  (registry)
nt!NtReadFile                 (file system)
```

Uroburos rootkit

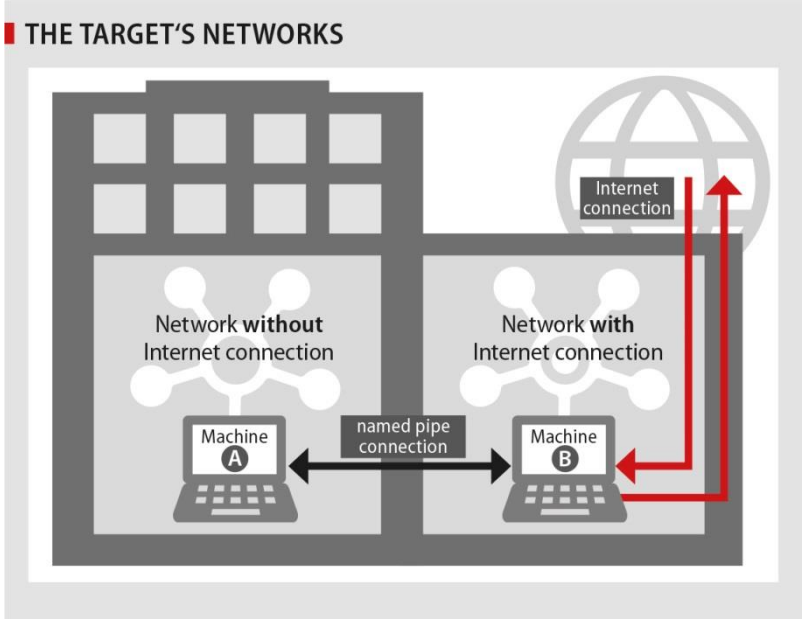
TRUST IN GERMAN SICHERHEIT

Windows Filtering Platform (WFP)

The WFP is a set of API and system services which provides a platform for creating network filtering applications. In our case, the rootkit uses this technology to perform Deep Packet Inspection (DPI) and modifications of the network flow. The purpose of this device is to intercept relevant data as soon as a connection to the Command & Control server or other local infected machines used as relay is established and to receive commands.

Uroburos rootkit

Windows Filtering Platform (WFP)



Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Windows Filtering Platform (WFP)

The filter parses HTTP and SMTP traffic (other protocols can easily be supported).
To identify the Uroburos traffic, the rootkit decrypts the network flow and looks for data starting with:

- `0xDEADBEEF`
- `0xC001BA5E`

The intercepted data is forwarded to the user land by using named pipe.

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Virtual file systems

Uroburos uses two virtual file systems: FAT32 & NTFS. During our analysis, the first one was never used (maybe a legacy mode). The second one is the decrypted .dat file (CAST-128 encryption).

The volume can be accessed by: `\\.\Hd1\`

The file system contains a queue file, log files, additional tools (reconnaissance tools)…

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Virtual file systems

```
kd> !devobj Rawdisk1
Device object (86253748) is for:
 RawDisk1 \Driver\Null DriverObject 8576a3f8
Current Irp 00000000 RefCount 22 Type 00000007 Flags 00000050
Vpb 862536c0 DevExt 00000000 DevObjExt 86253800 Dope 86253678
ExtensionFlags (0x00000800)  DOE_DEFAULT_SD_PRESENT
Characteristics (0x00000001)  FILE_REMOVABLE_MEDIA
Device queue is not busy.

kd> !vpb 862536c0
Vpb at 0x862536c0
Flags: 0x1 mounted
DeviceObject: 0x86259020
RealDevice:   0x86253748
RefCount: 22
Volume Label:

kd> !devobj 0x86259020
Device object (86259020) is for:
  \FileSystem\Ntfs DriverObject 8516e558
Current Irp 00000000 RefCount 0 Type 00000008 Flags 00040000
DevExt 862590d8 DevObjExt 86259fb0
ExtensionFlags (0x00000800)  DOE_DEFAULT_SD_PRESENT
Characteristics (0000000000)
AttachedDevice (Upper) 86253020 \FileSystem\FltMgr
Device queue is not busy.
```

Uroburos rootkit

Virtual file systems

```
kd> !devhandles \device\Rawdisk1

Checking handle table for process 0x8483c8f0
Kernel handle table at 89801be0 with 411 entries in use

PROCESS 8483c8f0  SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 00185000  ObjectTable: 89801be0  HandleCount: 411.
    Image: System

02bc: Object: 8625b6e8  GrantedAccess: 0012019f Entry: 89803578
Object: 8625b6e8  Type: (848bd3f8) File
    ObjectHeader: 8625b6d0 (new version)
        HandleCount: 1  PointerCount: 2
        Directory Object: 00000000  Name: \$Extend\$RmMetadata\$TxfLog\$TxfLog.blf {RawDisk1}

[...]

PROCESS 8483c8f0  SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 00185000  ObjectTable: 89801be0  HandleCount: 411.
    Image: System

02f0: Object: 8626b6f0  GrantedAccess: 0012019f Entry: 898035e0
Object: 8626b6f0  Type: (848bd3f8) File
    ObjectHeader: 8626b6d8 (new version)
        HandleCount: 1  PointerCount: 10
        Directory Object: 00000000  Name: \queue {RawDisk1}

PROCESS 8483c8f0  SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 00185000  ObjectTable: 89801be0  HandleCount: 411.
    Image: System

0344: Object: 8626f400  GrantedAccess: 00100004 Entry: 89803688
Object: 8626f400  Type: (848bd3f8) File
    ObjectHeader: 8626f3e8 (new version)
        HandleCount: 1  PointerCount: 1
        Directory Object: 00000000  Name: \klog {RawDisk1}

[...]

PROCESS 86248a00  SessionId: 0  Cid: 01f0    Peb: 7ffd8000  ParentCid: 01a8
    DirBase: 7ec9b080  ObjectTable: 82374a98  HandleCount: 288.
    Image: services.exe
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Virtual file systems

```
1  net use z: \\fileserver-1\Arbeitsgruppen /u:Administrator P**********g
2  \\.\Hd1\\rar.exe a -y -ta20130624 \\.\Hd1\\backup.rar z:\\
3  net use z: /delete
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Queue file

On the virtual file system we have a particularly interesting file: `\\.\Hd1\queue`

This file contains the rootkit configuration, encryption key, addition dll, ex-filtrated data…

These dll are injected in user land by the rootkit (for example in the browsers to steal sensitive information).

TRUST IN GERMAN SICHERHEIT

User land injected libraries

The injected libraries are used to communicate to the Command & Control servers, steal information… These file are used to create a kind of “proxy” between the kernel land and the user land. The libraries are: `inj_snake_Win32.dll` and `inj_services_Win32.dll`.

From the user land point of view, the protocol used for the C&C communication can be:

- HTTP
- SMTP
- ICMP
- …

TRUST IN GERMAN SICHERHEIT

Bypass of the kernel protection

The first bypassed protection is the **Kernel Patch Protection** (aka PatchGuard).

This protection checks the integrity of the Windows kernel to make sure that no critical parts are modified. If a modification is detected, the `KeBugCheckEx()` (with the code 0x109 `CRITICAL_STRUCTURE_CORRUPTION`) is executed and the system is shutdown with a blue screen.

The rootkit bypasses this protection, the rootkit hooks the `KeBugCheckEx()` function to avoid handling the code 0x109.

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Bypass of the kernel protection

The second bypassed protection is the **Driver Signature Enforcement**.

To avoid loading malicious drivers, Microsoft created this technology for its 64-bit versions of Windows Vista and later versions. To load a driver, the .sys file must be signed by a legitimate publisher. The flag to identify whether the protection is enable or not is `g_CiEnabled`.

The rootkit's driver is not signed but it still loaded.

Uroburos rootkit

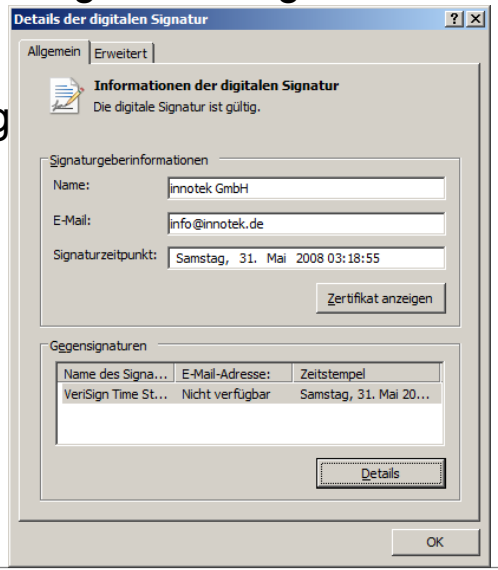
TRUST IN GERMAN SICHERHEIT

Bypass of the kernel protection

To bypass the **Signature Driver Enforcement**, the attackers use a legitimate, signed driver (in our case VirtualBox driver) and exploit a vulnerability to switch arbitrary memory address to 0. In our case, the address of the flag `g_CiEnabled` to switch off the protection. The used CVE is CVE-2008-3431. The VirtualBox driver is presently expired.

Before: `kd> dq nt!g_cienabled -> fffff800`02e45eb8 00000001`

After: `kd> dq nt!g_cienabled -> fffff800`02e45eb8 00000000`



Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Bypass of the kernel protection

The **Signature Driver Enforcement** bypass step by step:

- the malware opens the `VBoxDrv` symbolic link;
- it loads ntoskrnl.exe;
- it locates `g_CiEnabled`;
- it uses `DeviceIoControl()` to switch arbitrary address to 0

For example:

```
DeviceIoControl(VBoxDrv, SUP_IOCTL_FAST_DO_NOP, g_CiEnabledAddr, 0, g_CiEnabledAddr, 0, &cb, NULL)
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Bypass of the kernel protection

The VirtualBox driver is presently expired.

What about the signature's revocation of legacy software or vulnerable software?

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Other exploits

In the dropper, we can find several resources sections. These resources contain exploits to obtain administrator privileges (to be able to install and load the driver).
For example MS09-025 or MS10-015.

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Command & Controls

The attackers seem to use two kinds of C&C:

- dedicated servers
- legitimate compromised web sites (water holing) (TYPO3 CMS)

Thanks to the use of the WFP mechanism, we can imagine infected machines without any C&C hardcoded in the malware. The filter simply waits for the network pattern. The fact that the malware uses local, infected systems as relay adds complexity, too.

For incident response point of view, the identification and containment can become a nightmare…

Uroburos rootkit

Command & Controls

bgl.serveftp.net/wordpress/wp-includes/css/img/stat

Admin panel

Enter password!

login

Admin panel

Stats | View rule | Clear Log+Exception | DELETE TASK | TASK EDITOR | CONFIG EDITOR | DELETE successful count | Syslogs | Web shell

Down

Total - 341

Interesting IP - 0

Safari	Chrome	Unknown
	21	210

				OS	Client	Country	Referer	User-agent
			R::	Windows XP or XP SP3	MSIE 8.0	CH	--	Mozilla/4.0 (compatible; Windows NT 5.1; Trident/ GTB7.4; .NET CLR 1.0.370 1.1.4322; Media Center P CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR .NET4.0C; OfficeLiveConn OfficeLivePatch.1.3)
2	2013-01-09 06:42:05	81.[redacted]	SNIFFER::	Windows XP or XP SP3	MSIE 8.0	CH	www.tb-mittelland.ch	Mozilla/4.0 (compatible; Windows NT 5.1; Trident/ GTB7.4; .NET CLR 1.0.370 1.1.4322; Media Center P CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR .NET4.0C; OfficeLiveConn OfficeLivePatch.1.3)

Source: Kaspersky

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Infection vectors

- Spear phishing e-mails with Adobe PDF exploits (CVE-2013-3346 + CVE-2013-5065)
- Social engineering to trick the user into running malware installers with ".SCR" extension, sometimes packed with RAR
- Watering hole attacks using Java exploits (CVE-2012-1723), Flash exploits (unknown) or Internet Explorer 6,7,8 exploits (unknown)
- Watering hole attacks that rely on social engineering to trick the user into running fake "Flash Player" malware installers

Source: Kaspersky

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Targets

In February 2014, we mentioned in our report: *“Due to the complexity of the Uroburos rootkit, we estimate that it was designed to target government institutions, research institutions or companies dealing with sensitive information as well as similar high-profile targets.“*

Uroburos rootkit

Targets

In May 2014:

dS De Standaard

Aanbod voor abonnees Abonneer u Kl

NIEUWS KRANT AVOND ARCHIEF+

Home > Krant > Binnenland

RUSSISCH VIRUS GEÏDENTIFICEERD

Buitenlandse Zaken besmet door 'Snake'

13/05/2014 | Van onze redacteurs Nikolas Vanhecke en Mark Eeckhaut

Het computervirus dat Buitenlandse Zaken heeft aangevallen heet 'Snake'. Het virus wordt door de veiligheidsdiensten aanzien als het middel bij uitstek van de Russen om de wereld te begluren. Bij Buitenlandse Zaken is de schoonmaak aan de gang.

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Targets

In August 2014:

Government (Ministry of interior (EU country), Ministry of trade and commerce (EU country), Ministry of foreign/external affairs (Asian country, EU country), Intelligence (Middle East, EU Country)), Embassies, Military (EU country)

Education

Research (Middle East)

Pharmaceutical companies

Source: Kaspersky

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Attribution

During our analysis we found some technical links connecting Uroburos to Agent.Btz:

- Encryption key
- Usage of the same file name
- Check whether Agent.Btz is installed on the system
- Use of Russian language and user names (vlad, gilg, urik…)

```
Resource entries
================================================================================
Name                RVA       Size      Lang          Sublang                  Type
--------------------------------------------------------------------------------
RT_VERSION          0x6e060   0x444     LANG_RUSSIAN  SUBLANG_RUSSIAN          data
```

Uroburos rootkit

TRUST IN GERMAN SICHERHEIT

Attribution

In an article published by Reuters, in 2011, the journalist mentioned that “U.S. government strongly suspects that the original attack was crafted by Russian Intelligence.”

With the last elements presented by Belgian journalists, concerning the attack against the Ministry of Foreign Affairs, the Russian roots are further confirmed.

TRUST IN GERMAN SICHERHEIT

Thank you for your attention!

Questions?