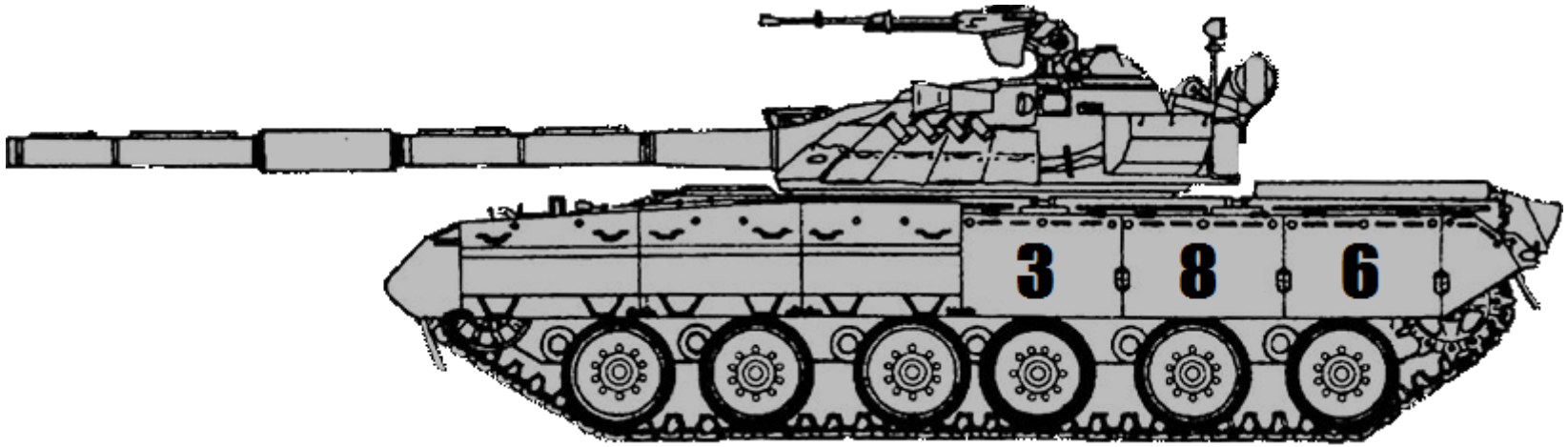


Signal (messenger)

Windows 7 CRACK

https://cracklab.team/PAunlock/

https://github.com/Blaukovitch/GOOGLE_CHROME_Windows_7

*20/02/2026, Head office of **386 Team WorkGroup**, re-engineer ELF*

What is this?

This is a binary hack of the officially distributed Signal messenger (EN). This project is a continuation and fork of the more well-known and initial Google Chrome Windows 7 crack project, from which the key based principles of hacking the application were borrowed.

Signal is open source software, a similar hack could be achieved by editing the program's source code, but «501c3» itself could do this. To a greater extent, the goal of this project is to show:

1) how insignificant the differences between Windows 7 and Windows 10 are, in order to undermine Microsoft;

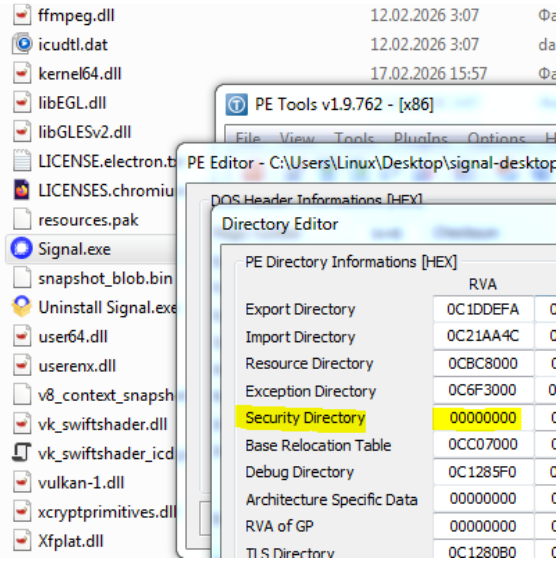
2) how lazy and bad software corporations are, refusing in the most absurd way to support Windows 7 / Windows Server 2012, even though there are simply no technical obstacles to doing so;

The authors of this hack rightly consider Windows 7 to be one of the best operating systems ever released by Microsoft.

Be advice! Google and Microsoft have already blocked these GitHub repositories in accordance with DMCA enforcements.

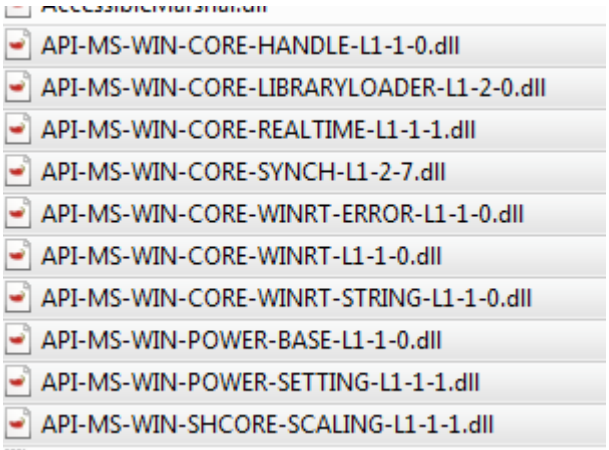
Security (anti-virus) warnings:

Most PE COFF files in the folder have had their digital signatures removed. Since binary changes have been made, the checksum of such a file is naturally violated and the signature becomes invalid. Since antivirus software may consider such a file to be infected in this case, the contents of the field in the security section of the PE file are reset to zero (as shown in the example of the exe Signal.exe, which is opened in the PeTools hacker utility):



If you have any antivirus issues, please report them as a regular technical issue on GitHub or contact me by any available means.

How this work?



For the most part, the issue lies in the missing WinAPI functions for the Windows 7 operating system. Similar technical problems are already being solved by projects such as VxKex (VxNext). In the case of our project, we use standard dynamic libraries–wrappers[1] over Windows system libraries that emulate the missing WinAPI. Fortunately, this is a fairly simple process, and there are no more than 20 missing WinAPI functions. Naturally, it was necessary to edit the PE COFF file import table for correct calling in the Windows 7 environment.

A binary code change has also been made, mostly affecting UNICODE strings in "bcryptprimitives.dll" (a Windows system library used to generate random numbers when creating cryptographic algorithms), various GUIDs located in the .rdata section of PE COFF files, and the disabling of certain functions related to the creation of COM objects (WinAPI CoCreateInstance), which is necessary to eliminate the crashes found.

More technical Signal Windows 7 CRACK information is available in the section below.

[1] **API-MS-WIN_XP open source project** https://github.com/Blaukovitch/API-MS-WIN_XP

How to start?

1) Recommended BACKUP you current Signal profiles in folder "%APPDATA%\Signal"

To quickly navigate to the Profiles folder, you can use a batch script file !Signal_Profiles.bat

2) You can use any method convenient for you:
   a. to replace the existing version of Signal in the folder "%APPDATA%\Signal"
   b. or simply copy it to any other separate folder. In this case, the only difference will be which of the two browsers you want to set as the default to launch;
3) Launch Signal.exe

Known issues:

1. **Windows 7 N editions. Install Windows 7 Media pack[2] update Windows6.1-KB968211-x64-RefreshPkg.msu[3] (x86) first for enable video decode options.**
2. **WebGPU is supported.**

Technical crashes report:

1. **Send me screenshot of violate**
2. **Step-by-step instructions on how to reproduce the crash**
3. **Exact version of the operating system (e.g., Windows 7 Professional SP1, build 7601)**
4. **Attach %APPDATA%\..\Roaming\Signal\logs non-empty files;**
5. **Also, you can manual extract NODE.JS archive from resources folder:**

Name of batch file	Affected modules
!list.BAT	**app.asar** list of files (NODE.JS)
!UNPACK.BAT	**Unpack (extract) app.asar file in to folder** (NODE.JS)
!PACK.BAT	**Pack folder back to app.asar archive** (NODE.JS)
asar-win7.exe	**NODE JS unpacker main executable with Win7 compatible patch**

[2] https://learn.microsoft.com/en-us/troubleshoot/windows-client/shell-experience/windows-media-feature-pack-for-windows-7

[3] https://github.com/Blaukovitch/GOOGLE_CHROME_Windows_7/releases/download/discord_new/Windows6.1-KB968211-x64-RefreshPkg.msu

Best regards from:

1. **cristianadam** synchronize code
   https://github.com/cristianadam/api-ms-win-core-synch-Win7/blob/main/api-ms-win-core-synch-l1-2-0.c
2. **i486** for any help and reports https://github.com/i486
3. **CRACKLAB.TEAM support** (dj.zlo, sendersu, Fettel, alzcore397)
4. https://github.com/SpotX-Official/SpotX
5. **Also, any peoples and companies who help crack this:**

	MS EDGE 128.0.2739.42
	Spotify 1.2.37.701 Logon is fixed!
	Brave 123.1.64.109
	Discord app-1.0.9193
	Opera 110.0.5130.49
	Opera 099.0.4788.88
	Electron Mail v5.3.0
	Vivaldi 6.8.3381.44
	ungoogled-chromium 128.0.6613.119
	Tor Browser 14.0.4

Contancts

6. via Github


https://github.com/Blaukovitch/GOOGLE_CHROME_Windows_7

7. via SourceForge

https://sourceforge.net/projects/google-chrome-windows-7/files/

8. mail to official 80_PA SecuROM keygen mailbox:

securom80pa[dot]gmail.com

9. CRACKLAB.TEAM forum

https://cracklab.team/index.php?threads/1037/

10. Call me) +7 91….

Extended technical info of crack(void[4]):

80_PA SecuROM keygen

****

Signal Windows 7 CRACK, 2026

..-..-..-..-..-..-..-..-

https://cracklab.team/PAunlock

[PE improt patch]

@signalapp+libsignal-client.node

libringrtc-x64.node

---

kernel32.dll > KERNEL64.DLL

bcryptprimitives.dll > xcryptprimitives.dll

api-ms-win-core-synch-l1-2-0.dll > api-ms-win-core-synch-l1-2-7.dll

@signalapp+sqlcipher.node

---

api-ms-win-core-synch-l1-2-0.dll > api-ms-win-core-synch-l1-2-7.dll

[PE x64/x86 assembly patch]

@indutny+simple-windows-notific_unloaded

RoGetAGileIntarface

[4] void duplicated in the file _firefox{64/86}_CRACK.log

```
$+17E0      | 48:894C24 08          | mov qword ptr ss:[rsp + 0x8],rcx                          | [rsp+08]:"=::=::\\"
$+17E5      | 55                    | push rbp                                                  |
$+17E6      | 53                    | push rbx                                                  | rbx:napi_register_module_v1+189D0
$+17E7      | 57                    | push rdi                                                  | rdi:napi_register_module_v1+189E0
$+17E8      | 48:8BEC               | mov rbp,rsp                                               |
$+17EB      | 48:83EC 50            | sub rsp,0x50                                              |



$+3BE0      | 48:895C24 08          | mov qword ptr ss:[rsp + 0x8],rbx                          | [rsp+08]:&L"h-MO_radstr"
$+3BE5      | 48:897424 10          | mov qword ptr ss:[rsp + 0x10],rsi                         | [rsp+10]:protected: void __cdecl
v8::internal::StrongRootAllocatorBase::deallocate_impl(unsigned __int64 *, unsigned __int64)+2A27D1
$+3BEA      | 48:897C24 18          | mov qword ptr ss:[rsp + 0x18],rdi                         |
$+3BEF      | 4C:896424 20          | mov qword ptr ss:[rsp + 0x20],r12                         |
$+3BF4      | B8 01000000           | mov eax,0x1                                               |
$+3BF9      | C3                    | ret                                                       |
$+3BFA      | 48:81EC 30030000      | sub rsp,0x330                                             |
$+3C01      | 48:8B05 F8930200      | mov rax,qword ptr ds:[0x7FEF217E000]                      |



signal.exe
__int64 __fastcall sub_1450EA910(__int64 a1, __int64 *a2)
{
  __int64 v4; // rcx

  v4 = *a2;
```

```
  if ( a1 )
  {
    if ( v4 )
    {
      *a2 = 0;
      (*(*v4 + 16LL))(v4);
    }
    return RoGetAgileReference(0, &unk_14A49E570, a1, a2);
  }
  else
  {
    *a2 = 0;
    if ( v4 )
      (*(*v4 + 16LL))(v4);
    return 0;
  }
}
$+50E9910      | 31C0              | xor eax,eax                                  | eax:BaseThreadInitThunk
$+50E9912      | C3                | ret                                          |
$+50E9913      | 90                | nop                                          |
$+50E9914      | 90                | nop                                          |
$+50E9915      | 90                | nop                                          |
$+50E9916      | 48:89D6           | mov rsi,rdx                                  | rdx:Cr_z_crc32_z+659A40
$+50E9919      | 48:89CF           | mov rdi,rcx                                  |
$+50E991C      | 48:8B0A           | mov rcx,qword ptr ds:[rdx]                   | rdx:Cr_z_crc32_z+659A40
```

```
$+50E991F      | 48:85FF                | test rdi,rdi                                                    |
$+50E9922      | 74 35                  | je signal.1443CA959                                               |



//remove REG_NOTIFY_THREAD_AGNOSTIC flag
$+2154253      | C74424 20 01000000     | mov dword ptr ss:[rsp + 0x20],0x1                                 |
$+215425B      | 48:89D9                | mov rcx,rbx                                                     |
$+215425E      | BA 01000000            | mov edx,0x1                                                       |
$+2154263      | 41:B8 0F000000         | mov r8d,0xF                                                       |
$+2154269      | FF15 71651D0A          | call qword ptr ds:[0x14B60B7E0]                                   |

$+3BF2484      | 48:8B4F 08             | mov rcx,qword ptr ds:[rdi + 0x8]                                  |
$+3BF2488      | 4C:8B4F 10             | mov r9,qword ptr ds:[rdi + 0x10]                                  | r9:Cr_z_crc32_z+659A40
$+3BF248C      | C74424 20 01000000     | mov dword ptr ss:[rsp + 0x20],0x1                                 |
$+3BF2494      | BA 01000000            | mov edx,0x1                                                       |
$+3BF2499      | 41:B8 05000000         | mov r8d,0x5                                                       |
$+3BF249F      | FF15 3B837308          | call qword ptr ds:[0x14B60B7E0]                                   |
$+3BF24A5      | 85C0                   | test eax,eax                                                    | eax:BaseThreadInitThunk
$+3BF24A7      | 0F85 C4FEFFFF          | jne signal.142ED3371                                              |
$+3BF24AD      | 48:8B4F 20             | mov rcx,qword ptr ds:[rdi + 0x20]                                 |
$+3BF24B1      | 4C:8B4F 28             | mov r9,qword ptr ds:[rdi + 0x28]                                  | r9:Cr_z_crc32_z+659A40
$+3BF24B5      | C74424 20 01000000     | mov dword ptr ss:[rsp + 0x20],0x1                                 |
$+3BF24BD      | BA 01000000            | mov edx,0x1                                                       |
$+3BF24C2      | 41:B8 05000000         | mov r8d,0x5                                                       |
```

```
$+3BF24C8       | FF15 12837308             | call qword ptr ds:[0x14B60B7E0]                                  |


$+3BF250D       | C74424 20 01000000        | mov dword ptr ss:[rsp + 0x20],0x1                                |
$+3BF2515       | BA 01000000               | mov edx,0x1                                                      |
$+3BF251A       | 41:B8 05000000            | mov r8d,0x5                                                      |
$+3BF2520       | FF15 BA827308             | call qword ptr ds:[0x14B60B7E0]                                  |
$+3BF2526       | 85C0                      | test eax,eax                                                     | eax:BaseThreadInitThunk
$+3BF2528       | 75 22                     | jne signal.142ED354C                                             |
$+3BF252A       | 48:8B4E 20                | mov rcx,qword ptr ds:[rsi + 0x20]                                |
$+3BF252E       | 4C:8B4E 28                | mov r9,qword ptr ds:[rsi + 0x28]                                 | r9:Cr_z_crc32_z+659A40
$+3BF2532       | C74424 20 01000000        | mov dword ptr ss:[rsp + 0x20],0x1                                |
$+3BF253A       | BA 01000000               | mov edx,0x1                                                      |
$+3BF253F       | 41:B8 05000000            | mov r8d,0x5                                                      |
$+3BF2545       | FF15 95827308             | call qword ptr ds:[0x14B60B7E0]                                  |


//WinAPI GetHostNameW redirection to kernel64.dll
$+3BB45CF       | 48:8D0D CE0C4708          | lea rcx,qword ptr ds:[0x14BA562A4]                               | 000000014BA562A4:"kernel64"
$+3BB45D6       | FF15 9C6F6608             | call qword ptr ds:[<GetModuleHandleA>]                           |
$+3BB45DC       | 48:85C0                   | test rax,rax                                                     |
$+3BB45DF       | 74 17                     | je signal.1435E55F8                                              |
$+3BB45E1       | 48:8D15 C70C4708          | lea rdx,qword ptr ds:[<_GetHostNameW>]                           | 000000014BA562AF:"GetHostNameW"
$+3BB45E8       | 48:89C1                   | mov rcx,rax                                                      |
$+3BB45EB       | FF15 07706608             | call qword ptr ds:[<GetProcAddress>]                             |
$+3BB45F1       | 48:8905 F83A8B08          | mov qword ptr ds:[0x14BE990F0],rax                               |
```

```
//OS version fixed
$+2D6445F      | 48:8B8C24 C0000000          | mov rcx,qword ptr ss:[rsp + 0xC0]                                  |
$+2D64467      | 48:31E1                     | xor rcx,rsp                                                        |
$+2D6446A      | E8 81CE3202                 | call signal.144AC22F0                                              |
$+2D6446F      | 48:8D0D 5A5A6E09            | lea rcx,qword ptr ds:[0x14BE7AED0]                                 |
$+2D64476      | E8 91BC3202                 | call signal.144AC110C                                              |
$+2D6447B      | 833D 4E5A6E09 FF            | cmp dword ptr ds:[0x14BE7AED0],0xFFFFFFFF                          |
$+2D64482      | 0F85 7AFFFFFF               | jne signal.142795402                                               |
$+2D64488      | E8 B37D3EFF                 | call <signal.ver_info>                                             |
$+2D6448D      | 83F8 0C                     | cmp eax,0xC                                                        | 0C:'\f'
$+2D64490      | 7F 68                       | jg signal.1427954FA                                                |
$+2D64492      | E8 A97D3EFF                 | call <signal.ver_info>                                             |
$+2D64497      | 31C0                        | xor eax,eax                                                        |
$+2D64499      | 90                          | nop                                                                |
$+2D6449A      | 83F8 05                     | cmp eax,0x5                                                        |
$+2D6449D      | 0F87 CE010000               | ja signal.142795671                                                |
$+2D644A3      | 31FF                        | xor edi,edi                                                        |
$+2D644A5      | 48:8D0D 604BFE08            | lea rcx,qword ptr ds:[0x14B77A00C]                                 |
$+2D644AC      | 48:630481                   | movsxd rax,dword ptr ds:[rcx + rax * 4]                            |
$+2D644B0      | 48:01C8                     | add rax,rcx                                                        |
$+2D644B3      | FFE0                        | jmp rax                                                            |


//WEBGPU support
```

```
$+28BD572      | 84C0                | test al,al                              |
$+28BD574      | B8 02080000         | mov eax,802                             |
$+28BD579      | B9 00000000         | mov ecx,0                               |
$+28BD57E      | 0F45C8              | cmovne ecx,eax                          |
$+28BD581      | 898C24 3C010000     | mov dword ptr ss:[rsp+13C],ecx          |
$+28BD588      | 0F57C0              | xorps xmm0,xmm0                         |
$+28BD58B      | 0F298424 00010000   | movaps xmmword ptr ss:[rsp+100],xmm0    |
...
$+28B77E8      | F78424 58010000 40001000 | test dword ptr ss:[rsp+158],100040 |
$+28B77F3      | 0F94C0              | sete al                                 |
$+28B77F6      | 48:85ED             | test rbp,rbp                            |
$+28B77F9      | 0F95C1              | setne cl                                |
$+28B77FC      | 08C1                | or cl,al                                |
$+28B77FE      | 90                  | nop                                     |
$+28B77FF      | 90                  | nop                                     |
$+28B7800      | 90                  | nop                                     |
$+28B7801      | 90                  | nop                                     |
$+28B7802      | 90                  | nop                                     |
$+28B7803      | 90                  | nop                                     |
$+28B7804      | 4C:89CB             | mov rbx,r9                              |

[strings patch]
libringrtc-x64.node
000007FEF146C5F0  78 00 63 00 72 00 79 00 70 00 74 00 70 00 72 00  x.c.r.y.p.t.p.r.
000007FEF146C600  69 00 6D 00 69 00 74 00 69 00 76 00 65 00 73 00  i.m.i.t.i.v.e.s.
```

```
[section NODE.JS]
@signal-desktop\resources\$app.asar\ts\updater\index.main.js
|> str/92
function autoUpdateDisabled() {
  return !import_electron.app.isPackaged || process.mas || !import_config.default.get("updatesEnabled"); >> return true;
}
__name(autoUpdateDisabled, "autoUpdateDisabled");


@\signal-desktop\resources\$app.asar\ts\updater\common.main.js
|> str/395
  async #checkForUpdates(checkType) {
    if ((0, import_version_std.isNotUpdatable)(import_packageJson_node.version)) {
      this.logger.info(
        "checkForUpdates: not checking for updates, this is not an updatable build"
      );
      return;
    }

>>

async #checkForUpdates(checkType) {
```

```
    this.logger.info(
      "checkForUpdates: not checking for updates, this is not an updatable build"
    );
    return;
```

```
[additional TOOLS]
@signal-desktop\resources\asar-win7.exe
for @app.asar
!list.BAT
!PACK.BAT
!UNPACK.BAT


[config]
%AppData%\Roaming\Signal\
@config.json
{
  "updatesEnabled": false
}

* Windows 10 System lib, lot of refts to RT && native components
```