

Malware analysis “sample2.exe”

Filippo Contro VR437055

Michele Martini VR437056

9 Gennaio 2020

Summary

Simple calculator app



But under the hood...

Summary

- Post requests
- Security center deactivation
- Infection


Summary

- Post requests
- Security center deactivation
- Infection → Polymorphic malware


Static analysis


Static analysis











First look

 pestudio 8.70 - Malware Initial Assessment - www.winator.com

File Help



 c:\users\malware\Desktop\sample

- add indicators (wait...)
-  virustotal (offline)
- ☐ dos-stub (176 bytes)
- ☐ file-header (Aug. 2001)
- ☐ optional-header (GUI)
- ☐ directories (5)
- ☐ sections (self-modifying)
- ☐ libraries (6)
-  imports (132/0/21)
-  exports (0)
-  tls-callbacks (n/a)
-  resources (26)
-  strings (wait...)
-  debug (Aug. 2001)
-  manifest (missing Trust Info)
-  version (CALC.EXE)
-  certificate (n/a)
- ☐ overlay (wait...)

| property | value |
|--------------------|---|
| md5 | F83C765FB553146712FCF2C6066670B5 |
| sha1 | A6B849E7A8312F5D7E3D7C96501887F39E3BE512 |
| sha256 | 34558AC3BFAB17CA1A1FF70860B35296395F1DF7FA8D86B39C56FAECF9C3CFFC |
| first-bytes (hex) | 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 40 00 00 00 00 00 00 |
| first-bytes (text) | M Z |
| size | 626688 bytes |
| entropy | 7.189 |
| imphash | 08F6A1B121DA8CEDDE2D1089D0906ED8 |
| cpu | 32-bit |
| signature | n/a |
| entry-point (hex) | 50 90 51 52 90 53 90 54 55 56 57 55 89 E5 83 EC 7C |
| file-version | 5.1.2600.0 (xpclient.010817-1148) |
| file-description | Windows Calculator application file |
| file-type | executable |
| subsystem | GUI |
| compiler-stamp | Fri Aug 17 21:52:32 2001 |
| debugger-stamp | Fri Aug 17 21:52:32 2001 |

Static analysis

First look

pestudio 8.70 - Malware Initial Assessment - www.winitor.com

File Help

Icons: [File Icon] [Folder Icon] [Close Icon] [Print Icon] [Help Icon]

Left pane (Tree view):

- c:\users\malware\desktop\samp1
 - indicators (wait..)
 - virustotal (offline)
 - dos-stub (176 bytes)
 - file-header (Aug.2001)
 - optional-header (GUI)
 - directories (5)
 - sections (self-modifying)
 - libraries (6)
 - imports (132/0/21)
 - exports (0)
 - tls-callbacks (n/a)
 - resources (26)
 - strings (wait..)
 - debug (Aug.2001)
 - manifest (missing Trust Info)
 - version (CALC.EXE)
 - certificate (n/a)
 - overlay (wait..)

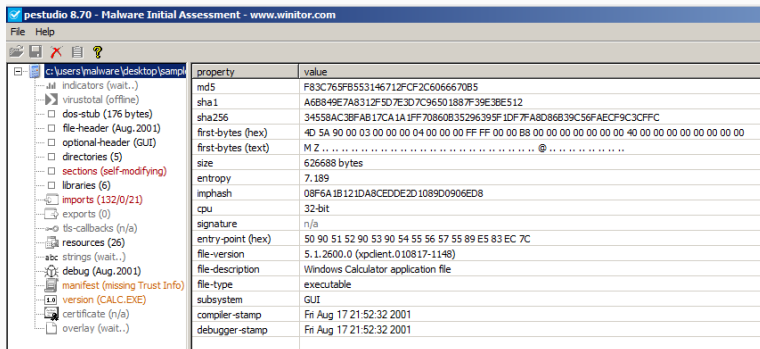
Right pane (Table view):

| property | value |
|--------------------|--|
| md5 | F83C765FB553146712FCF2C6066670B5 |
| sha1 | A6B849E7A8312F5D7E3D7C96501887F39E3BE512 |
| sha256 | 34558AC3BFAB17CA1A1FF70860B35296395F1DF7FA8D86B39C56FAECF9C3CFFC |
| first-bytes (hex) | 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 88 00 00 00 00 00 00 40 00 00 00 00 00 00 00 |
| first-bytes (text) | M Z |
| size | 626688 bytes |
| entropy | 7.189 |
| imphash | 08F6A1B121DA8CEDDE2D1089D0906ED8 |
| cpu | 32-bit |
| signature | n/a |
| entry-point (hex) | 50 90 51 52 90 53 90 54 55 56 57 55 89 E5 83 EC 7C |
| file-version | 5.1.2600.0 (xpclient.010817-1148) |
| file-description | Windows Calculator application file |
| file-type | executable |
| subsystem | GUI |
| compiler-stamp | Fri Aug 17 21:52:32 2001 |
| debugger-stamp | Fri Aug 17 21:52:32 2001 |

High entropy

Static analysis

First look



High entropy \rightarrow Obfuscation or Packing?

Let's look at the sections

Sections

pestudio 8.70 - Malware Initial Assessment - www.winitor.com

File Help

c:\users\malware\desktop\sampl

| property | value | value | value | value |
|------------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| name | .text | .data | .rsrc | .vmp0 |
| md5 | 179745C927697911BAA6... | 8E8381392A4F163121AB... | 86CBAEA46AB7F1C62572... | SFC458D95F5306F811C5... |
| file-ratio (99.84 %) | 12.09 % | 0.41 % | 5.64 % | 81.70 % |
| virtual-size (1843244 bytes) | 75440 bytes | 4124 bytes | 35168 bytes | 1728512 bytes |
| virtual-address | 0x00001000 | 0x00014000 | 0x00016000 | 0x0001F000 |
| raw-size (625664 bytes) | 75776 bytes | 2560 bytes | 35328 bytes | 512000 bytes |
| raw-address | 0x00000400 | 0x00012C00 | 0x00013600 | 0x0001C000 |
| cave (496 bytes) | 336 bytes | 0 bytes | 160 bytes | 0 bytes |
| entropy | 6.195 | 3.587 | 4.984 | 7.107 |
| entry-point (0x00012475) | x | - | - | - |
| bladdisted | - | - | - | x |
| writable | - | x | - | x |
| executable | x | - | - | x |
| shareable | - | - | - | - |
| discardable | - | - | - | - |
| cacheable | x | x | x | x |
| pageable | x | x | x | x |
| initialized-data | - | x | x | - |
| uninitialized-data | - | - | - | - |
| readable | x | x | x | x |

The first three sections are OK! But **.vmp0** NO!

Sections

pestudio 8.70 - Malware Initial Assessment - www.wintit.com

File Help

PE Explorer

File: C:\Users\malware\Desktop\sample.exe

Properties

Sections (self-modifying)

| property | value | value | value | value |
|------------------------------|-------------------------|------------------------|-------------------------|-------------------------|
| name | .text | .data | .bss | .rsrc |
| md5 | 170745C9226079118AA6... | 8E381392A4F563121AB... | 86CB4EA46A87F1C62572... | 9FC458D99F5306F811C5... |
| file-ratio (99.84 %) | 12.09 % | 0.41 % | 5.64 % | 81.70 % |
| virtual-size (1843244 bytes) | 75440 bytes | 4124 bytes | 35168 bytes | 1728512 bytes |
| virtual-address | 0x00001000 | 0x00014000 | 0x00026000 | 0x0003F000 |
| raw-size (625664 bytes) | 75776 bytes | 2560 bytes | 35128 bytes | 512000 bytes |
| raw-address | 0x0000400 | 0x00012C00 | 0x00013600 | 0x0001C000 |
| code (496 bytes) | 336 bytes | 0 bytes | 160 bytes | 0 bytes |
| entry | 6.195 | 3.587 | 4.984 | 7.107 |
| entry-point (0x00012475) | X | - | - | - |
| Madddicted | - | - | - | X |
| writeable | - | X | - | X |
| executable | X | - | - | X |
| shareable | - | - | - | - |
| discardable | - | - | - | - |
| cacheable | X | X | X | X |
| pageable | X | X | X | X |
| initialized-data | - | - | - | - |
| uninitialized-data | - | - | - | - |
| readable | X | X | X | X |

- High entropy
- Big portion of code (87%)
- Both writable and executable

Sections

pentestio R.70 - Hardware Initial Assessment - www.wintor.com

File Help

File Explorer view of `C:\Users\malware\Desktop\pamp\resources\bin\strings`.

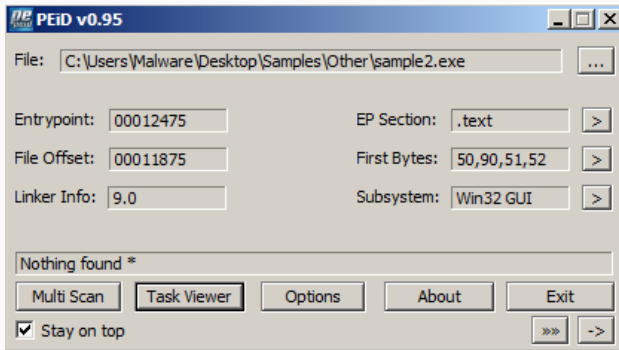
| property | value | value | value | value |
|-----------------------------|-------------------------|------------------------|------------------------|------------------------|
| name | .text | .data | .rsrc | |
| md5 | 179745C927697011BA8A... | BE8381392A7F36321AB... | 86C8AEE46AB7F1C5C75... | SFC458D957306F811C5... |
| file-rot (99.84 %) | 12.09 % | 0.41 % | 5.64 % | 81.70 % |
| virtual-size (184324 bytes) | 75440 bytes | 4124 bytes | 35588 bytes | 1738512 bytes |
| virtual-address | 0x00001000 | 0x00014000 | 0x00016000 | 0x0000F000 |
| raw-size (825664 bytes) | 75776 bytes | 2560 bytes | 35328 bytes | 512000 bytes |
| raw-address | 0x00000400 | 0x00001200 | 0x00013600 | 0x00001C00 |
| cave (406 bytes) | 336 bytes | 0 bytes | 360 bytes | 0 bytes |
| entry-point (0x0012475) | X | 3.587 | X | 7.307 |
| blacklisted | - | - | - | X |
| inritable | - | X | - | X |
| executable | X | - | - | X |
| shareable | - | - | - | X |
| discardable | - | - | - | X |
| cacheable | X | X | X | X |
| pageable | X | X | X | X |
| initialized-data | - | - | - | X |
| uninitialized-data | - | - | - | X |
| readable | X | X | X | X |

- High entropy
- Big portion of code (87%)
- Both writable and executable → Very suspicious

The presence of the 3 main sections (text, data, resources) suggests the absence of packing.

Obfuscation

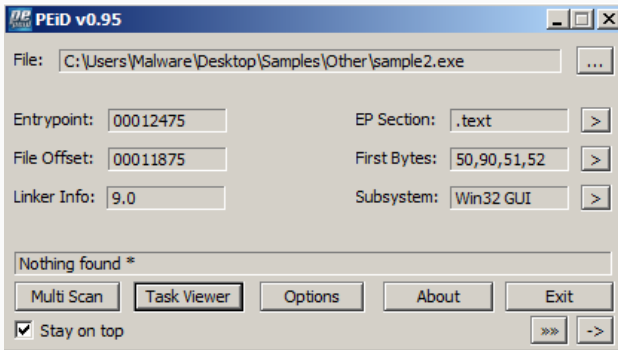
The presence of the 3 main sections (text, data, resources) suggests the absence of packing.



PeiD confirms our supposition.

Obfuscation

The presence of the 3 main sections (text, data, resources) suggests the absence of packing.



PeiD confirms our supposition. The name **vmp0** is given by **VM-Protect**.

Imports

| pestudio 8.70 - Malware Initial Assessment - www.winitor.com | | | | | | | | |
|--|------------------|----------------|---------------|----------------|------------------|----------------|--------------|--|
| File Help | | | | | | | | |
| c:\users\malware\desktop\samples | | | | | | | | |
| symbol (132) | group (disabled) | blacklist (21) | anonymous (0) | anti-debug (0) | undocumented (1) | deprecated (0) | library (6) | |
| GetModuleHandleA | - | x | - | - | - | - | kernel32.dll | |
| LoadLibraryA | - | x | - | - | - | - | kernel32.dll | |
| GetProcAddress | - | x | - | - | - | - | kernel32.dll | |
| GlobalCompact | - | x | - | - | x | - | kernel32.dll | |
| Sleep | - | x | - | - | - | - | kernel32.dll | |
| WriteProfileStringW | - | x | - | - | - | - | kernel32.dll | |
| GetStartupInfoA | - | x | - | - | - | - | kernel32.dll | |
| CreateThread | - | x | - | - | - | - | kernel32.dll | |
| GetCommandLineW | - | x | - | - | - | - | kernel32.dll | |
| GetProfileIntW | - | x | - | - | - | - | kernel32.dll | |
| CallWindowProcW | - | x | - | - | - | - | user32.dll | |
| WinHelpW | - | x | - | - | - | - | user32.dll | |
| PostQuitMessage | - | x | - | - | - | - | user32.dll | |
| IsClipboardFormatAvailable | - | x | - | - | - | - | user32.dll | |
| GetDesktopWindow | - | x | - | - | - | - | user32.dll | |
| OpenClipboard | - | x | - | - | - | - | user32.dll | |
| GetClipboardData | - | x | - | - | - | - | user32.dll | |
| CloseClipboard | - | x | - | - | - | - | user32.dll | |
| SendMessageW | - | x | - | - | - | - | user32.dll | |
| SetWindowLongW | - | x | - | - | - | - | user32.dll | |
| SystemParametersInfoW | - | x | - | - | - | - | user32.dll | |

- getModuleHandleA
- loadLibraryA
- getProcAddressA
- getStartupInfoA
- getCommandLineW

Imports

| pestudio 8.70 - Malware Initial Assessment - www.winitor.com | | | | | | | | |
|--|----------------------------|------------------|----------------|---------------|----------------|------------------|----------------|--------------|
| File Help | | | | | | | | |
| c:\users\malware\desktop\samples | | | | | | | | |
| | symbol (132) | group (disabled) | blacklist (21) | anonymous (0) | anti-debug (0) | undocumented (1) | deprecated (0) | library (6) |
| indicators (5/26) | GetModuleHandleA | - | x | - | - | - | - | kernel32.dll |
| virustotal (offline) | LoadLibraryA | - | x | - | - | - | - | kernel32.dll |
| dos-stub (This program cannot | GetProcAddress | - | x | - | - | - | - | kernel32.dll |
| file-header (Aug.2001) | GlobalCompact | - | x | - | - | x | - | kernel32.dll |
| optional-header (GUI) | Sleep | - | x | - | - | - | - | kernel32.dll |
| directories (5) | WriteProfileStringW | - | x | - | - | - | - | kernel32.dll |
| sections (self-modifying) | GetStartupInfoA | - | x | - | - | - | - | kernel32.dll |
| libraries (6) | CreateThread | - | x | - | - | - | - | kernel32.dll |
| imports (132/0/21) | GetCommandLineW | - | x | - | - | - | - | kernel32.dll |
| exports (0) | GetProfileIntW | - | x | - | - | - | - | kernel32.dll |
| tls-callbacks (n/a) | CallWindowProcW | - | x | - | - | - | - | user32.dll |
| resources (26) | WinHelpW | - | x | - | - | - | - | user32.dll |
| strings (38/23/3/14499) | PostQuitMessage | - | x | - | - | - | - | user32.dll |
| debug (Aug. 2001) | IsClipboardFormatAvailable | - | x | - | - | - | - | user32.dll |
| manifest (missing Trust Info) | GetDesktopWindow | - | x | - | - | - | - | user32.dll |
| version (CALC.EXE) | OpenClipboard | - | x | - | - | - | - | user32.dll |
| certificate (n/a) | GetClipboardData | - | x | - | - | - | - | user32.dll |
| overlay (n/a) | CloseClipboard | - | x | - | - | - | - | user32.dll |
| | SendMessageW | - | x | - | - | - | - | user32.dll |
| | SetWindowLongW | - | x | - | - | - | - | user32.dll |
| | SystemParametersInfoW | - | x | - | - | - | - | user32.dll |

- getModuleHandleA
- loadLibraryA
- getProcAddressA
- getStartupInfoA
- getCommandLineW

- **Version:** Legit information, but no date
- **Strings:** Thousands of crypted strings
- **Certificate:** it is missing

Dynamic analysis

We used 3 tools:

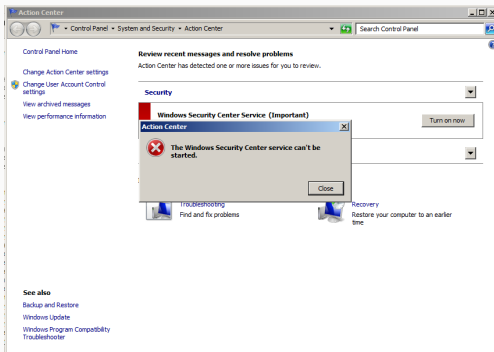
- **regshot**, to detect files and registers alterations between a time lapse;
- **procmon**, to log system functions called by the malware;
- **fakenet**, to track internet traffic in a simulated network.

In order to get consistent results we followed this schedule:

1. launch Fakenet;
2. launch and setup Procmon;
3. launch Regshot, setup path and run of its first shot;
4. start Procmon analysis and launch of the malware;
5. interaction with calculator by the GUI;
6. stop Procmon tracking
7. second Regshot shot;
8. stop Fakenet;

First run

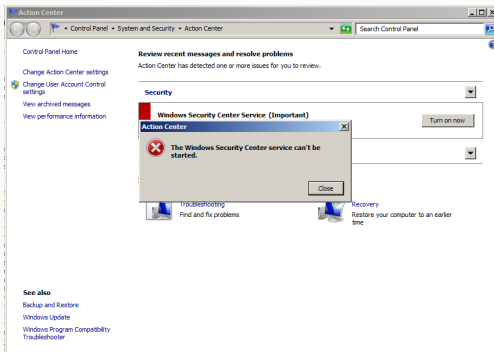
We ran the malware without any tools.



After a little bit the **Windows Security Center** was deactivated.

First run

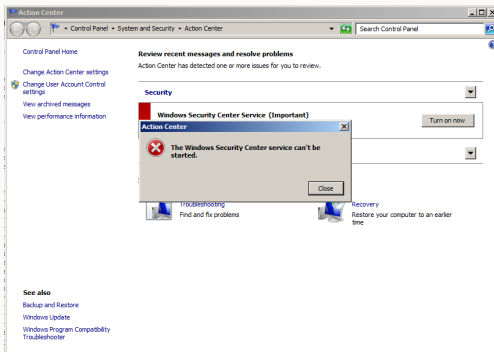
We ran the malware without any tools.



After a little bit the **Windows Security Center** was deactivated.
It could not be restarted.

First run

We ran the malware without any tools.



After a little bit the **Windows Security Center** was deactivated.

It could not be restarted.

The malware needs time to perform those actions.

With fakenet we registered many POST requests.
More than 900, to 300 different destinations.

With fakenet we registered many POST requests.
More than 900, to 300 different destinations.

Every request contains:

- Request type
- Destination URL
- Protocol
- User agent

With fakenet we registered many POST requests.
More than 900, to 300 different destinations.

Every request contains:

- Request type
- Destination URL
- Protocol
- User agent

Here's an example:

```
User-Agent: Mozilla/4.0  
(compatible; MSIE 28;...
```

With fakenet we registered many POST requests.
More than 900, to 300 different destinations.

Every request contains:

- Request type
- Destination URL
- Protocol
- User agent

Here's an example:

```
User-Agent: Mozilla/4.0  
(compatible; MSIE 28;...
```

There are not imported libraries to send HTTP requests

With fakenet we registered many POST requests.
More than 900, to 300 different destinations.

Every request contains:

- Request type
- Destination URL
- Protocol
- User agent

Here's an example:

```
User-Agent: Mozilla/4.0  
(compatible; MSIE 28;...
```

There are not imported libraries to send HTTP requests
→ *dynamically imported*

We used procmon to keep track of every action made by the malware
Dividing them in 3 categories:

- DLL
- Registry
- Files

We measured 46 different dll files loaded with the *LoadImage* primitive.

Among them the most interesting are:

- **cryptbase** - **crypt32**: to handle cryptography
- **ws2_32**: to manage web socket

We saw many open-read-close actions on many system registers, but only a few write

We saw many open-read-close actions on many system registers, but only a few write

The only keys modified were:

- Language list, which has no interesting effects
- Windows internet zones set to 0 which means *Allow anything* for each network type

We detect the infection of other files watching the “WriteFile” operations and the amount of bytes written.

We obtained the sequence of actions that the malware implement to infect other files

We detect the infection of other files watching the “WriteFile” operations and the amount of bytes written.

We obtained the sequence of actions that the malware implement to infect other files

- Read the *.exe* victim file.
- Write of the content plus the infected part in a *.vir* file with the same name.
- Copy of the content of the *.vir* file to the *.exe* one changing the EOF location.
- Set of fake information on the executable such as creation and last access time.
- Delete the *.vir* file.

The infected files were many, and in different location.

They were mainly common executables, run frequently by the average user.

The infected files were many, and in different location.

They were mainly common executables, run frequently by the average user.

The main ones were:

- Windows Media Player
- Internet Explorer
- Windows Defender
- Windows Mail
- Windows Photo Viewer
- and more...

We inspect the infected files with pestudio and we found a new section called **.vmp0**

With regshot we had a confirmation of all the actions tracked with procmon.

The fact that caught our attention was the registry change related to the Windows Security Center.

```
HKLM\System\CurrentControlSet\services\wscsvc\Start = 4
```

The value 4 means disabled.

With regshot we had a confirmation of all the actions tracked with procmon.

The fact that caught our attention was the registry change related to the Windows Security Center.

```
HKLM\System\CurrentControlSet\services\wscsvc\Start = 4
```

The value 4 means disabled.

The weird thing is that this value change has not been made by the malware.

We discovered that the value was changed by **services.exe**

Reverse engineering

The reverse engineering was divided in 2 phases:

- Code rebuilding
- Debugging

We explored the cfg of the start function created by IDA, and we built a pseudo code for the first part, which deals with the decryption of the obfuscated zone.

We explored the cfg of the start function created by IDA, and we built a pseudo code for the first part, which deals with the decryption of the obfuscated zone.

The `.vmp0` section is decrypted through a cycle that perform an arithmetic xor of the code with a certain key.

The cycle is repeated 7 times, but during the last one the key is incremented by one.

The key is 0x58.

To debug the code we used Ollydbg alongside procmon, executing instructions one by one, stepping over the function calls and keeping track of the actions performed.

We had 2 main target:

- Detect the infection function
- Detect the deactivation of the security center

Eventually we achieved a procedure to debug the infection function:

1. breakpoint in 0101273A; then after the *RET* the malware enters the obfuscated section.
2. breakpoint in 010AAB30; then there is the creation of the second thread which is the analyzed one.
3. breakpoint in 010BD0A9 which is the begin of the target function

Eventually we achieved a procedure to debug the infection function:

1. breakpoint in 0101273A; then after the *RET* the malware enters the obfuscated section.
2. breakpoint in 010AAB30; then there is the creation of the second thread which is the analyzed one.
3. breakpoint in 010BD0A9 which is the begin of the target function

In particular we discovered that the thread calls FUN_010ACABF, which then calls FUN_0109F059, which then calls iteratively FUN_010B0CA this last one contains FUN_010BD0A9 which is the target function that performs the malicious actions.

The infection function is FUN_10BD636.

It has only the target name as parameter.

The infection function is FUN_10BD636.

It has only the target name as parameter.

This procedure is huge, with too much code to reverse; it even contains a recursive call inside.

We set the procmon filters to monitor “services.exe”, and in particular the “RegSetValue” operation.

We set the procmon filters to monitor “services.exe”, and in particular the “RegSetValue” operation.

The malicious action is done at 010B10C7. The third time that this instruction is executed the security center is deactivated

This action is a call to **StartServiceA** from **ADVAPI32.dll**.

We set the procmon filters to monitor “services.exe”, and in particular the “RegSetValue” operation.

The malicious action is done at 010B10C7. The third time that this instruction is executed the security center is deactivated

This action is a call to **StartServiceA** from **ADVAPI32.dll**.

Reading the documentation we discovered that this function needs privilege to close services

We set the procmon filters to monitor “services.exe”, and in particular the “RegSetValue” operation.

The malicious action is done at 010B10C7. The third time that this instruction is executed the security center is deactivated

This action is a call to **StartServiceA** from **ADVAPI32.dll**.

Reading the documentation we discovered that this function needs privilege to close services

→ *probable privilege escalation*

Conclusions

| Static Analysis | | | |
|---------------------|--------------------|------------------------|--|
| Category | Select | Score | |
| Packed | Packed | 2 | |
| Strings | Suspicious Strings | 3 | |
| Imports | Suspicious | 2 | |
| Sections | Abnormal Sections | 1 | |
| Main Icon | Legitimate Icon | 0 | |
| Additional Icons | Legitimate | 0 | |
| Dialogs | Legitimate | 0 | |
| Version Information | Present | 0 | |
| Digital Signature | Not Present | 2 | |
| Total Score | | 10 | |
| Verdict | | Potentially Suspicious | |

| Dynamic Analysis | | | |
|----------------------------|-------------------------------|------------|--|
| Category | Select | Score | |
| Persistence | Multiple Entries | 2 | |
| File Manipulation | Affects other files | 2 | |
| Process Manipulation | Affects other processes | 2 | |
| Registry Manipulation | Registry manipulation | 1 | |
| Additional Processes | Doesn't start other processes | 0 | |
| Removal Resistance | No removal prevention | 0 | |
| Analysis Resistance | No analysis prevention | 0 | |
| Interface/Visible Activity | GUI/Interface | 0 | |
| Network Activity | Network Activity | 1 | |
| Rootkit Behaviour | Rootkit Behaviour | 2 | |
| System Calls | Suspicious system calls | 1 | |
| Behaviour | Expected behaviour | 0 | |
| Total Score | | 11 | |
| Verdict | | Suspicious | |

Summing up the result of our analysis we can describe the malware as a polymorphic one, which performs various malicious actions such as internet connections, replications on other system programs and deactivates the security center. It disguise itself as a calculator, fooling the average user.

