





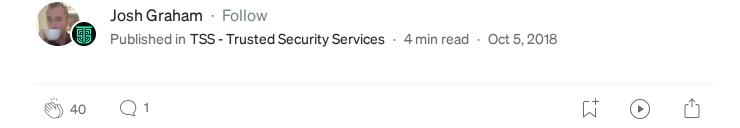
X

Sign in



X

# Pentesting and .hta (bypass PowerShell Constrained Language Mode)

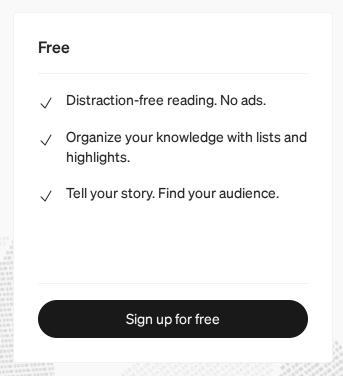


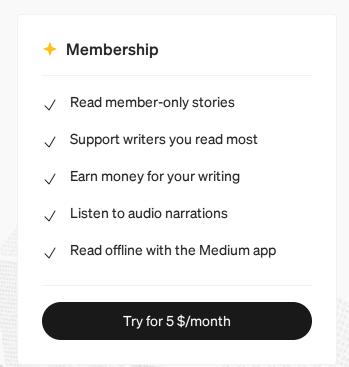
When I'm on an engagement and I'm given a SOE and a domain account, I usually want to use a tool like *PowerShell Empire* to remotely control the SOE and run all my other pentesting tools. When the client has strong security controls in place, it can be frustrating to get that initial foothold on the SOE. In these situations I find .hta files very useful. This blog post will have some of the interesting ways you can use .hta files in pentesting.

## **Application whitelist bypass**

'hta' stands for 'HTML Application' and is basically the same as a regular HTML page except that it is run using *mshta.exe* and supports extra scripting languages. *mshta.exe* is a built in windows application and is often allowed through application whitelisting rules and we can do lots of interesting things using the extra scripting languages, specifically *vbscript*.

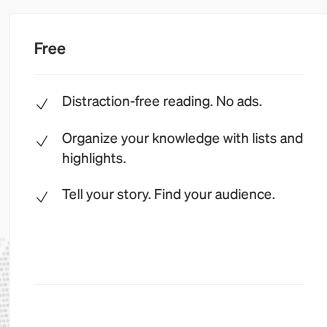
# Medium

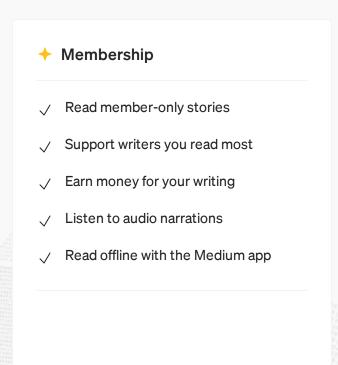




The above hta file isn't great...The command opens in another window and disappears immediately after it is finished making it difficult to read the output. We can make this a bit better by capturing the output of the command and displaying inside the HTML application something like this:

# Medium





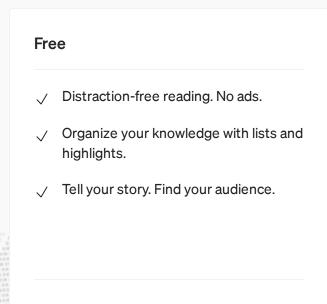
dir don't work and you can't do compound commands (i.e. ping 1.1.1.1 && pause) so lets try to do better.

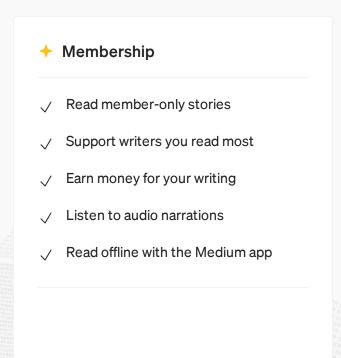
## Running PowerShell/C# with .hta

To run PowerShell commands with powershell.exe blocked we can use C#'s <u>Pipeline</u> class. To run arbitrary C# commands on a locked down system we will use a technique I learned at NOTSOSECURE's *Advanced Infrastructure Hacking* course. The bypass works by using *InstallUtil.exe* (built in windows binary) to run an uninstall function in an arbitrary executable. We are going to create said executable using another built in windows binary, *csc.exe*, which can be used to compile arbitrary C# code (this resource is useful for application whitelist bypasses). Doing this manually is a pain and too hard to remember so we use a .hta to wrap the mundane steps. The hta will perform the following steps:

- 1. Write some C# to a file, the C# contains an uninstall hook
- 2. Compile the C# from step 1 into a .exe using csc.exe
- 3. Write the PowerShell commands you want to run to a file. The C# uninstall hook will read these commands and execute them inside a *Pipeline*
- 4. Execute uninstallUtil.exe on the exe from step 2 using a WScript.Shell object
- 5. Display the output from uninstallUtil in the HTML application.

# **Medium**



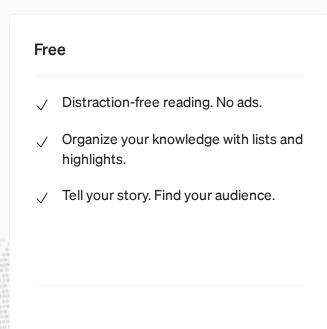


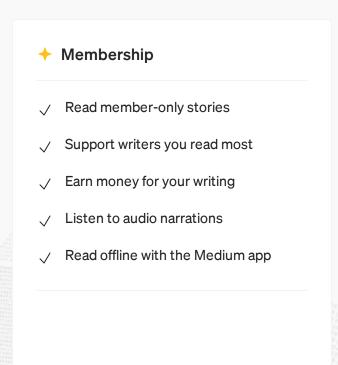
Although there won't be a visible window, the mshta.exe process will be present in the task manager which *should* stick out like a sore thumb...

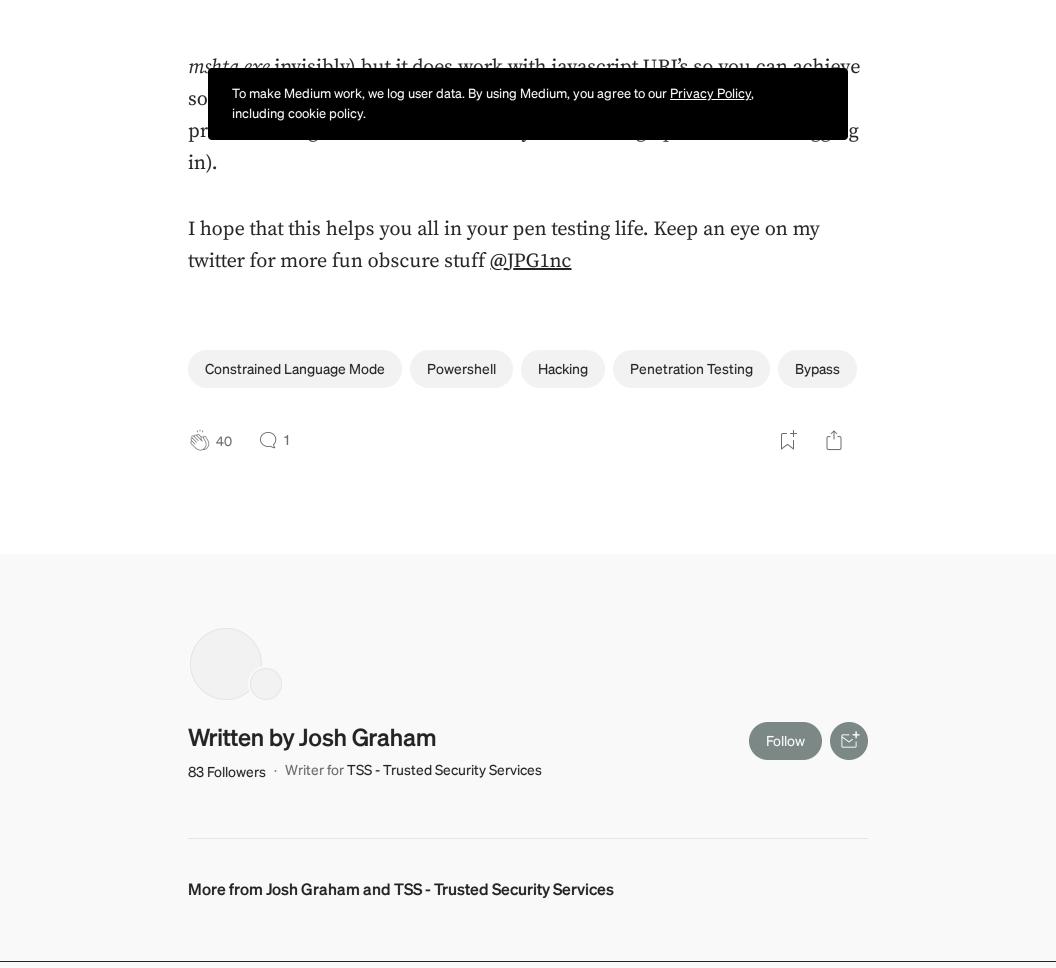
#### File-less hta

You don't even need a hta file at all! mshta.exe understands *JavaScript URIs* so you can use a command similar to the one below to run your pentesting

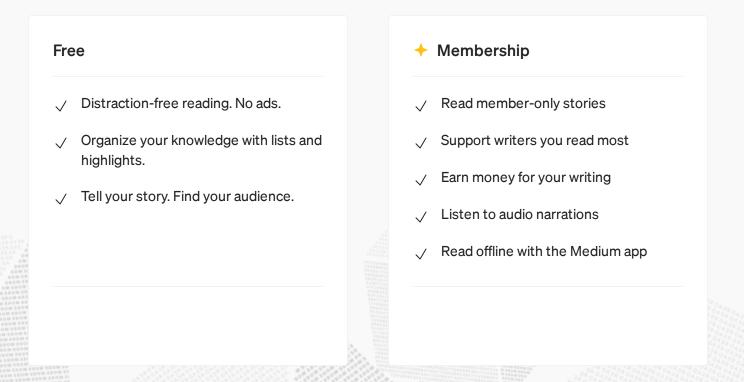
# Medium

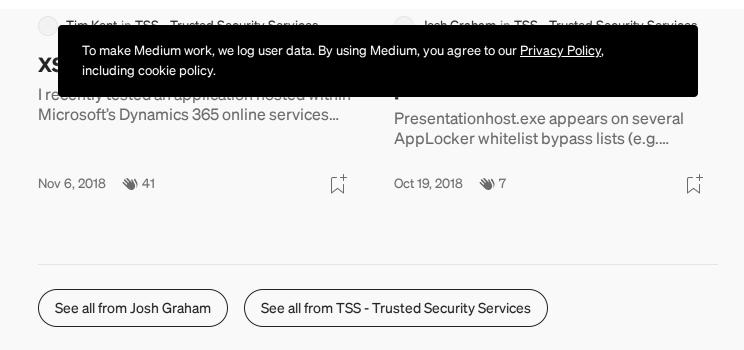






# **Medium**





## **Recommended from Medium**

Maryamshakeel in "The Algorithmic Author"

### **Ethical Hacking:**

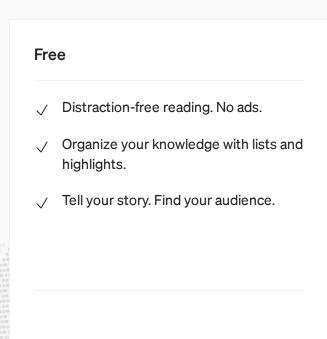
Introduction Digital systems have lately become part of life. The extent to which they...

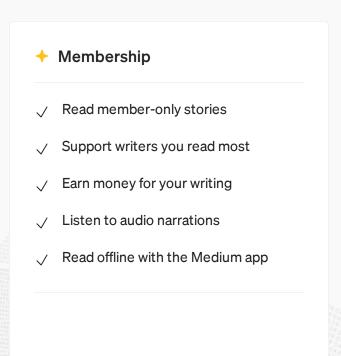
Jonathan Mondaut

# How ChatGPT Turned Me into a Hacker

Discover how ChatGPT helped me become a hacker, from gathering resources to tackling...

## **Medium**





To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy,

including cookie policy.

Sanskar Kalra Riley Pickles **Game of Active Directory: Linux Privelege Escalation | Hack Pentesting Strategies for Real-...** the Box Walkthrough | Part 1 DISCLAIMER\*\* \_This write-up is intended Introduction purely for educational purposes and to shar...  $\Box$ → Oct 17 < 6</p> Aug 16 🔌 3 🗨 1 Mavrogiannis Panagiotis backdoor **Setting Up Mythic C2: A Guide to Red Team Tactics: Exploiting Paste** Jacking with PsycheShell for... **Evading Advanced Detection...** Introduction DISCLAIMER: Using these tools and methods against hosts that you do not have explicit...  $\Box$ Jun 3 👋 59 Oct 8 **3** 1

# **Medium**

