eternalrelayx.py — Non-Admin NTLM Relaying & ETERNALBLUE Exploitation

Kory Findley (k0fin) Follow
Jul 2 · 7 min read

Introduction

In this post, we will cover how to perform the **EternalRelay** attack, an attack technique which reuses **non-Admin** SMB connections during an **NTLM Relay** attack to launch **ETERNALBLUE** against hosts running affected versions of the Windows operating system. This attack provides an attacker with the potential to achieve remote code execution in the privilege context of **SYSTEM** against vulnerable Windows hosts without the need for local Administrator privileges or credentials.

Attack Scenario

- Connected to a local internal network segment.
- Can send broadcast protocol traffic to hosts within our local internal network segment.
- No credentials for local or domain accounts.
- Attacker machine is not joined to Active Directory.
- Targets affected by ETERNALBLUE are accessible within our network segment.

Advantages

- No credentials or hash cracking required.
- No socks proxy or proxychains required.
- No need to be joined to domain.
- Non-Admin credentials can still be abused to obtain RCE in SYSTEM privilege context against affected hosts.

- Easy to use any type of payload we want for 3rd-party post-exploitation/C2.
- Exploitation capabilities can be extended to support all variations of ETERNALBLUE for additional Windows version affected by the MS17–010 SMBv1 vulnerability.

How It Works

1. Arbitrarily relay poisoned client's credentials to ETERNALBLUE target(s).
2. If the relay victim is a non-Admin (if the relay target returns an **rpc_access_denied** message for the relayed credentials), the authenticated SMB connection for the relayed user is used to perform a **TreeConnectAndX** setup to the target's **IPC$** share and create a **Tree ID** (**TID**).
3. The TID is used to perform ETERNALBLUE exploitation against the target. After this portion of the exploit is sent, a packet is evaluated for a specific retStatus.

4. If the retStatus condition is met, we should be able to continue the rest of the exploit and achieve RCE.
5. Our shellcode is sent to the target before being triggered and executed.
6. Disconnect the TID, logoff, and close the reused SMB connection.
7. The SMB thread is completed.

How It’s Made

The **eternalrelayx.py** script is included within a modified version of impacket found within the eternalrelayx project folder. It is simply a modified version of the **ntlmrelayx.py** script written by **dirkjanm**, which can be found in the official version of impacket maintained by **SecureAuthCorp** on Github. The ETERNALBLUE functionality added to the eternalrelayx project relies on the **MS17–010** exploit project by **sleepya.**

Supported Targets

Currently, eternalrelayx.py is a proof-of-concept to demonstrate the attack method, and supports ETERNALBLUE exploitation for affected releases of

Windows 7 SP1 and Windows Server 2008 R2. The actual exploit code used in the PoC is a slightly modified copy of **Worawit Wang's eternalblue_win7.py** code (however, the **zzz_exploit.py** exploit code included within his MS17–010 project is currently being integrated into eternalrelayx to drastically increase the reliability and practical use of this tool across all affected versions of Windows).

Attack Demo

Below, we can walk through a simple demonstration of using the eternalrelayx.py PoC against an (extremely simple) Windows 7 SP1 / Windows Server 2008 R2 Active Directory setup.

Finding Relay Targets

First, we'll find hosts within our segment which do not enforce SMB signing. Below, **CrackMapExec** (**cme**) is used to scan our local subnet for available SMB servers. Using cme with the **--gen-relay-list** argument will generate a list of IPv4 addresses for SMB servers which do not enforce SMB signing.

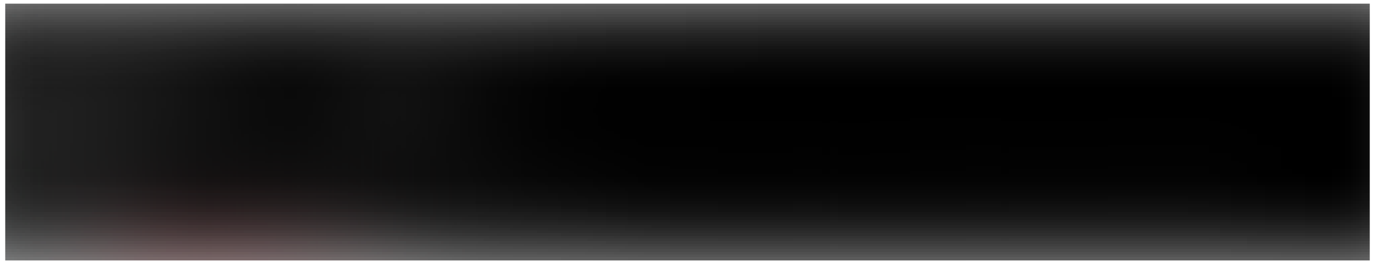
NOTE: Only CME 4.0.1+ supports the **--gen-relay-list** argument, so be sure to download the latest version from **byt3bl33d3r**'s Github! (Credits to @SourceFrenchy for pointing that out!). A link to the CME Github page is provided in the **References** section at the bottom of this write-up. :)

CME is used to discover an applicable Windows target which support SMBv1 and does not enforce SMB signing.

Finding ETERNALBLUE Targets

The generated list of NTLM Relay targets can be used to conduct further recon/enumeration and find targets which are (most likely) vulnerable to MS17–010. The output from cme above shows a Windows 7 Professional SP1 (7601) workstation which doesn't enforce SMB signing and supports communications over SMBv1.

Although this workstation looks like the perfect target for our attack, let's try to get a second opinion (if possible) using a module for MS17–010 detection included in Metasploit.

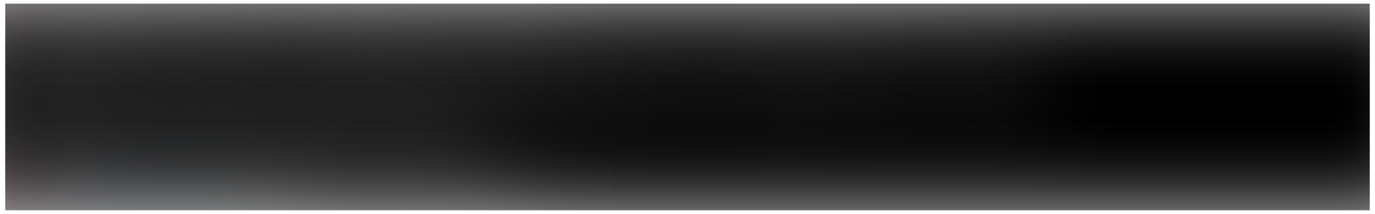


Generating a Payload

Since eternalrelayx.py will utilize a single BIN payload, any raw shellcode generated from 3rd party tools should be usable (such as a Cobalt Strike Beacon), and shouldn't exceed **2000** bytes. The following example uses demo instructions for Meterpreter included in **the eternalblue_sc_merge.py** script written by **Worawit Wang**. Generating the payload looks something like this-

- Generate raw 32-bit and 64-bit Windows Meterpreter shellcode into BIN files.
- Compile 32-bit and 64-bit Windows kernel shellcode with NASM.
- Concatenate user-mode and kernel-mode payloads into a single payload for each processor architecture.

- Merge concatenated 32-bit and 64-bit payloads into a final payload.


A simple Python script included in the eternalrelayx project is used to generate a Meterpreter payload for the EternalRelay attack.

Configuring a Payload Handler

A callback server / payload handler needs to be configured. For the purpose of this walk-through, we will use Metasploit to create and configure a Meterpreter handler. An RC file is included with the eternalrelayx project which can be used to automatically start our payload handler.

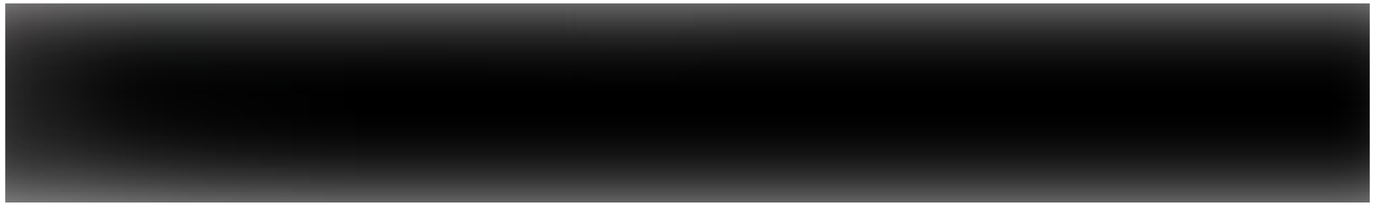
The RC file included with eternalrelayx.py is loaded into Metasploit (msfconsole) to configure and launch our payload handler.

NOTE: The default **LPORT** values in the RC file will need to be changed for each payload if the Meterpreter default port values for the payload being used are different (**4444** for **64**-bit and **4445** for **32**-bit by default).

Poisoning the Network

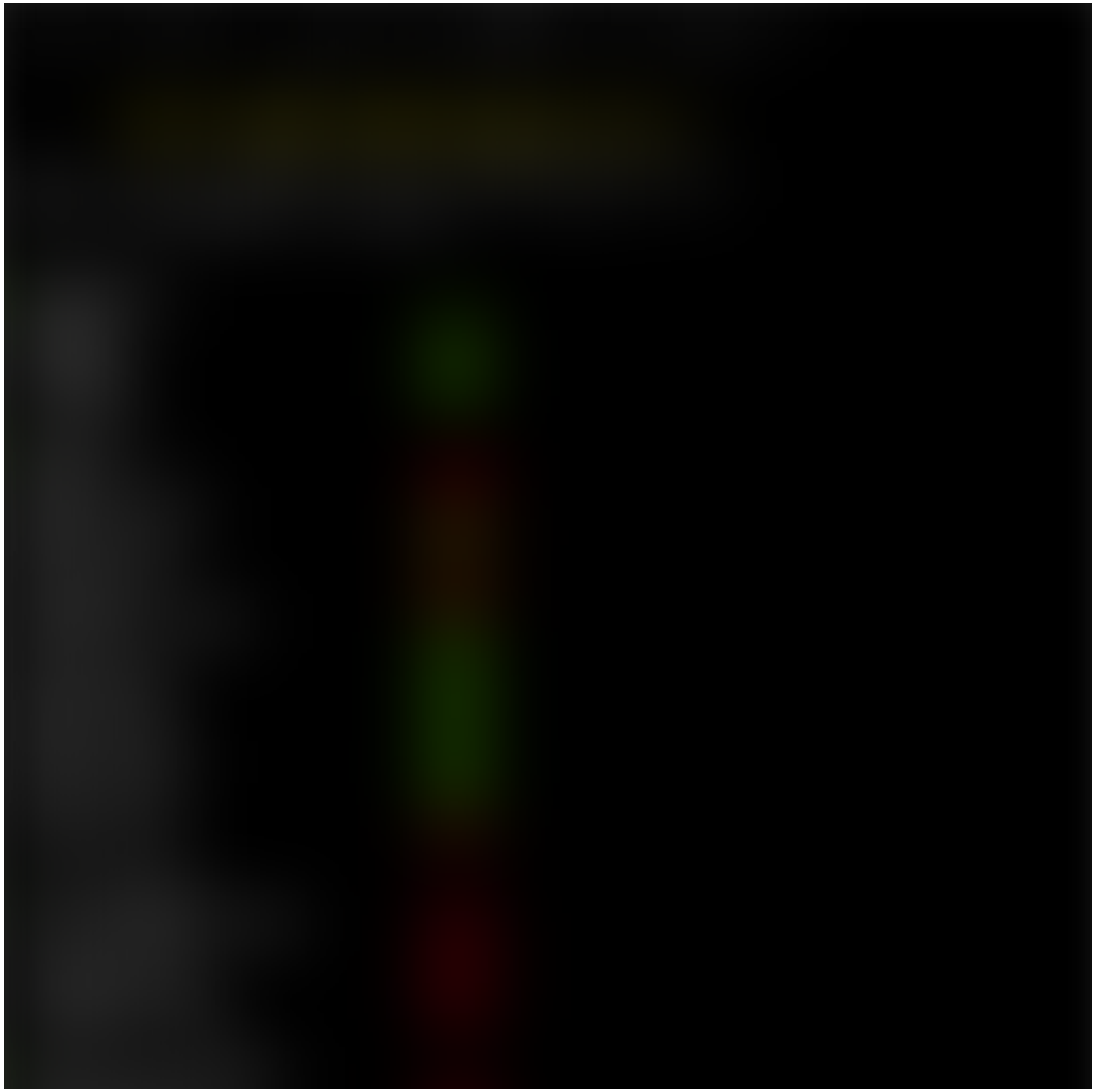
Just like a typical NTLM relay attack, we can start poisoning the network to coerce relay targets to connect to our relay server. Responder is used for this walk-through, although there are other tools and methods of accomplishing this.

First, we need to edit the Responder configuration file (**Responder.conf**) for Responder to work with relay attack tools. Normally, Responder will host its own arbitrary server listeners for various protocols (SMB, HTTP, etc.) while poisoning hosts within a target network. Since we want to relay authentication to remote targets within the network (rather than just intercept and (attempt to) crack NetNTLMv1 or NetNTLMv2 hashes), we will disable SMB, HTTP,and HTTPS servers for Responder.



Responder is configured to run without its SMB, HTTP, and HTTPS servers enabled. This allows Responder to be used for poisoning attacks without handling server connections or authentication attempts.

After saving the configuration file, we can start up Responder and begin listening for broadcast protocol traffic which can be abused to poison local network segment clients.



Responder is started in poisoning mode (-wrf) and listens for broadcast protocol traffic which can be poisoned to coerce clients into connecting to our relay server.

Once Responder successfully poisons a victim, the victim will attempt to connect and authenticate to our relay server, which will then forward the authentication to the relay target. We can demonstrate this by performing a lookup for a non-existent host within the local network.

An attempted connection to a host name which does not exist in the local network triggers an NBNS lookup.

Launching Eternalrelayx.py

Once poisoning is configured and running properly, eternalrelayx.py is used to wait for victim client authentication. Once a relay victim's non-Admin privilege context is confirmed, the attack is launched against the relay target(s).

A non-Admin user is relayed using eternalrelayx.py, which successfully launches the attack against our relay target.

Meterpreter Callback and Code Execution

After the attack completes, our payload handler receives a callback from the victim and a Meterpreter session is created with SYSTEM privileges.

Commands such as sysinfo and getuid can be used to gather basic information about the currently compromised system.

The Meterpreter payload handler receives a callback from the payload on port 4444.

Additional Payloads

Currently, I've only fully tested eternalrelayx.py with Meterpreter. Other payload methods for eternalrelayx.py are currently being tested and documented for added C2 and post-exploitation framework support (Cobalt Strike, Empire, etc.).

Download

The eternalrelayx project can be found here-

https://www.github.com/k0fin/eternalrelayx

Features in Progress

- Integrating **zzz_exploit.py** for more reliable and practical exploitation across all affected Windows versions
- Ability to use custom 1-liner payloads (PowerShell, etc.) for EternalRelay RCE
- Using NTLM relay victims to simply scan / detect the presence of the MS17–010 SMBv1 vulnerability affecting Windows relay targets.

Credits

- **Skip Duckwall (passingthehash)** for the original idea!

- **Tyler Robinson** for passing the idea along and encouraging me to try it out!
- **Worawit Wang (sleepya)** for his work around **MS17–010** and **ETERNALBLUE**!
- **Dirk-jan Mollema (dirkjanm)** and **Alberto Solino (agsolino)** for their work around **impacket** and **ntlmrelayx.py**!

Special Thanks & Shout-Outs

- **LETHAL Security — West Coast Hackers**
- **illMob**
- pr3c0re
- Tyler Robinson
- 0ray
- illwill
- cheetz

- Skip Duckwall
- Vijay Bolina
- Devin Ertel
- 3nc0d3r
- tactical_intel
- dc0de
- Robert CurtinSeufert
- johnhsawyer
- p00r0ne
- jaybeale
- haxorthematrix
- TheBlindHacker
- PermaNull

References

Microsoft Security Bulletin MS17-010 - Critical

Published: March 14, 2017 Version: 1.0 This security update resolves vulnerabilities in Microsoft Windows. The most...

docs.microsoft.com

SecureAuthCorp/impacket

Impacket is a collection of Python classes for working with network protocols. - SecureAuthCorp/impacket

github.com

Eternalblue | The NSA-developed Exploit That Just Won't Die

By SentinelOne - You will undoubtedly recall the names Shadow Brokers, who back in 2017 were dumping software exploits...

www.sentinelone.com

worawit/MS17-010

MS17-010. Contribute to worawit/MS17-010 development by creating an account on GitHub.

github.com

Playing with Relayed Credentials | SecureAuth

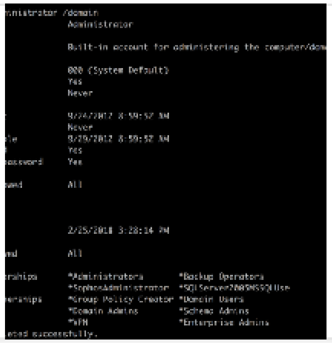
During penetration testing exercises, the ability to make a victim connect to an attacker's controlled host provides an…

www.secureauth.com

Top Five Ways I Got Domain Admin on Your Internal Network before Lunch (2018 Edition)

Yes it's still easy to get Domain Admin "before lunch" as it was when I first started.

medium.com



```
nistrator /domain
                 Administrator

                 Built-in account for administering the computer/dom

                 000 (System Default)
                 Yes
                 Never

                 9/24/2012 8:59:57 AM
                 Never
le               9/29/2012 8:59:57 AM
                 Yes
password         Yes

wed              All



                 2/25/2018 3:28:14 PM

wed              All

rships           *Administrators       *Backup Operators
                 *SophosAdministrator  *SQLServer2005MSSQLUse
erships          *Group Policy Creator *Domain Users
                 *Domain Admins        *Schema Admins
                 *VPN                  *Enterprise Admins
eted successfully.
```

byt3bl33d3r/CrackMapExec

A swiss army knife for pentesting networks. Contribute to byt3bl33d3r/CrackMapExec development by creating an account…

github.com

Thanks to Tyler Robinson.

Ntlm Relay Eternalblue Red Team Penetration Testing Active Directory

28 claps

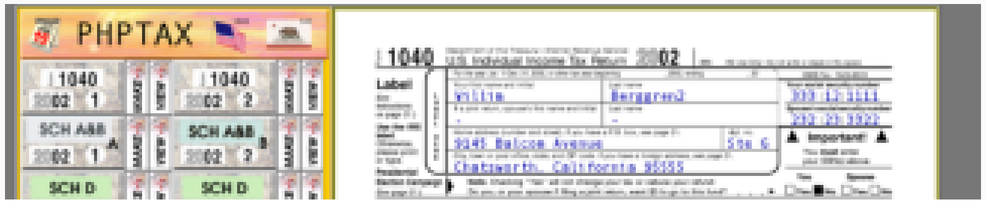
WRITTEN BY

Kory Findley (k0fin)

Follow

Write the first response

More From Medium



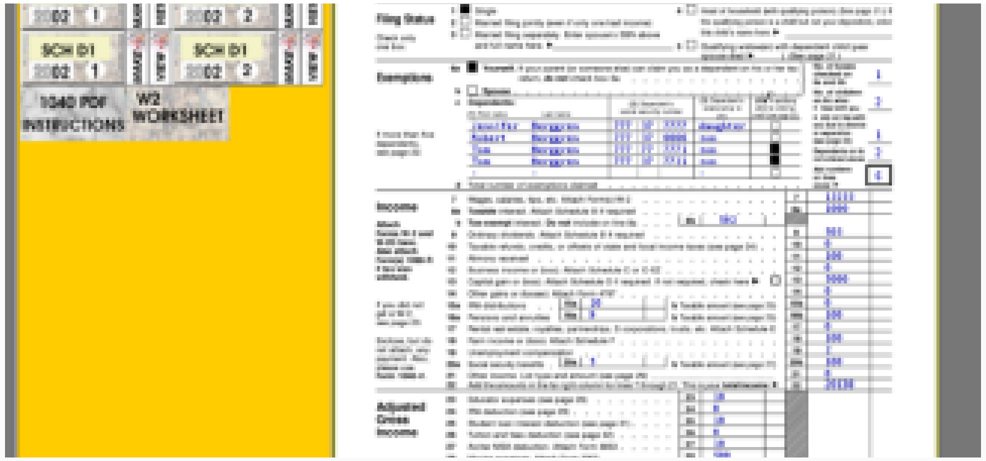
Related reads

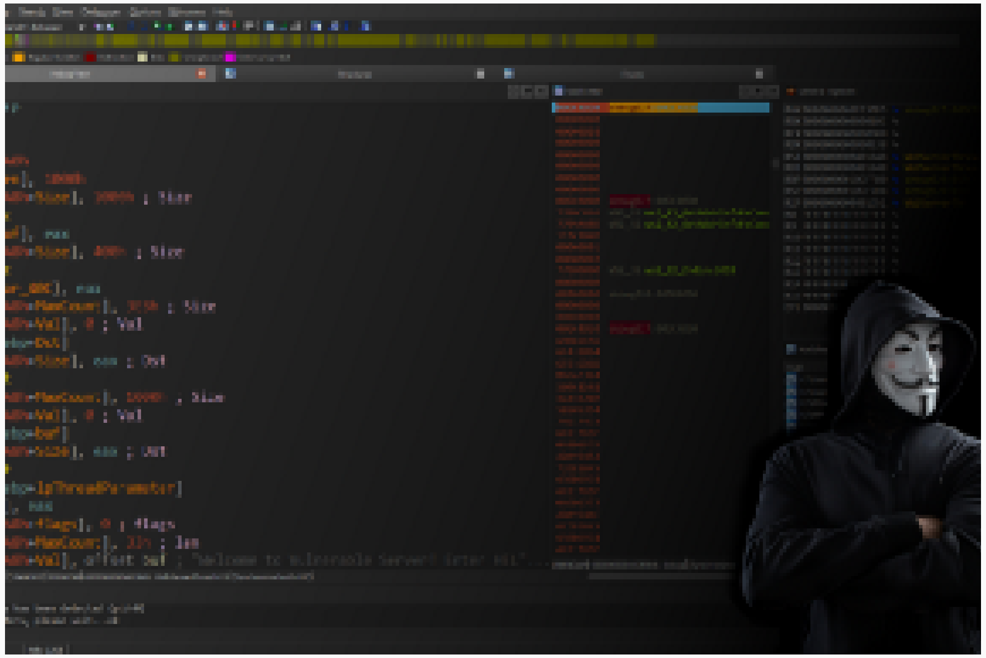
VulnHub — Kioptrix: Level 5

Mike Bond

Oct 21, 2018 · 13 min read

386





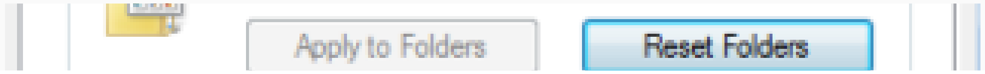
Related reads

Windows-Based Exploitation —VulnServer TRUN Command Buffer Overflow

Andreas Poyiatzis in InfoSec Write-ups

May 30 · 6 min read ★

188



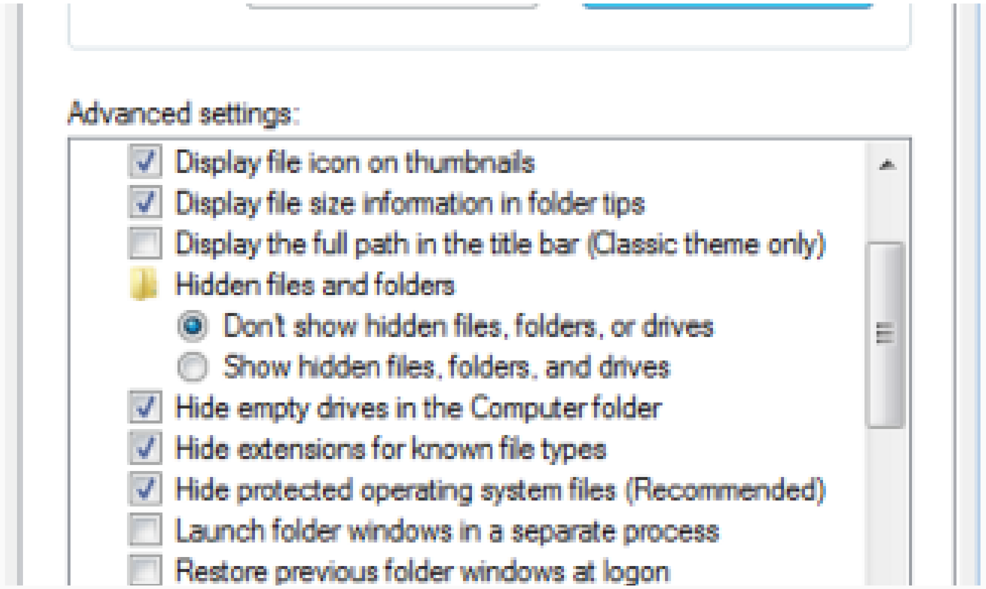
Related reads

Basic Static Analysis (Part 1)

Tstillz

Nov 19, 2018 · 11 min read ★

199



Advanced settings:

- [✓] Display file icon on thumbnails
- [✓] Display file size information in folder tips
- [ ] Display the full path in the title bar (Classic theme only)
- Hidden files and folders
  - ● Don't show hidden files, folders, or drives
  - ○ Show hidden files, folders, and drives
- [✓] Hide empty drives in the Computer folder
- [✓] Hide extensions for known file types
- [✓] Hide protected operating system files (Recommended)
- [ ] Launch folder windows in a separate process
- [ ] Restore previous folder windows at logon