**ASSIGNMENT**

**MALWARE ANALYSIS**

Malware analysis can be described as the process of understanding the behaviour and purpose of a suspicious file or URL. The output of the process aids in detecting and mitigating any potential threat. Some key benefits that malware analysis offers are to the incident responders and security analysts. They are:

* Assesses the damage from a security threat
* Identify the source of the attack
* Identify the vulnerability of the malware, its exploitation level, and preparation to patch accordingly.
* Practically break incidents by the level of security threat
* Reveal hidden indicators of compromise that need to be blocked
* Improve the efficiency of indicators of compromise, alert and notify
* Enrich any context when hunting for threat

**OBJECTIVES:**

* **Breaks down the malware:**

A big part of malware analysis is demystifying malware and cyberthreats to increase awareness. After all, malware is only a software program written with the expressed purpose of causing harm. Understanding the code and how it works is integral to blocking malware entry or, at least, its spread across your ecosystem.

* **Investigates its characteristics:**

Every software will leave a unique digital footprint, and malware is no different. How does a specific malware variant or family approach data? How does it spread? What is its pace of replication and tactic for camouflage? Knowing the exact characteristics of malware makes it easier to detect it.

* **Unravels its functionality:**

This is a critical element of malware analysis, and it is difficult to get right. Malware will typically wait in hiding until the right time to attack. This means its functionality will not become clear to the user before it is too late. Malware analysis tries to determine the intended functionality of the software by reviewing its code.

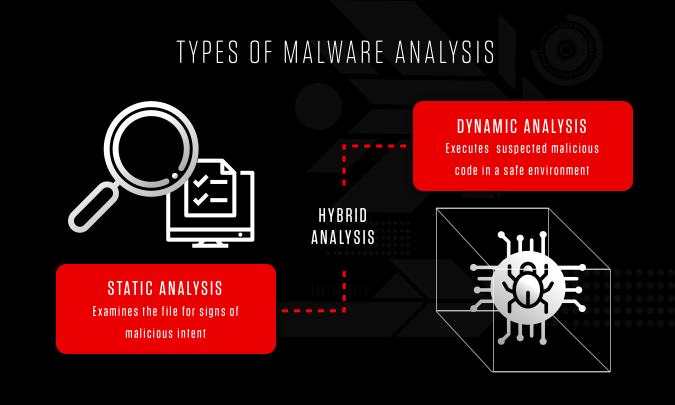
* **Traces the malware’s origin:**

Malware can be notoriously hard to trace, and hackers take advantage of this by holding data ransom for large amounts. Malware analysis tries to see beyond the anonymization of the coder and trace it back to its origin — a person, an IP, a geographic location, or even an organization, among others. This helps in the swift intervention of legal authorities during an attack.

* **Tries to predict the impact:**

By putting the above threads of investigation together, it is possible to arrive at a probable impact profile. Its functionality, nature of target systems, the pace of growth, and preferred distribution channels indicate the worst-case scenario impact of malware. This enables companies to plan and deploy mitigation procedures.

**TYPES OF MALWARE ANALYSIS:**



**Static Analysis:**

The static analysis does not analyse the code when it is running. Instead, it examines files for malicious intent. This makes it useful to identify infrastructure, packed files, and libraries. Some technical indicators can be used to determine if the file is malicious. However, since it does not run the code, it is difficult to detect sophisticated malware.

However, since static analysis does not actually run the code, sophisticated malware can include malicious runtime behavior that can go undetected. For example, if a file generates a string that then downloads a malicious file based upon the dynamic string, it could go undetected by a basic static analysis. Enterprises have turned to dynamic analysis for a more complete understanding of the behavior of the file

**Dynamic Analysis:**

Dynamic analysis executes any suspicious malicious code in a secure environment called a sandbox. It enables security professionals to watch the malware in action and not impacting the risk of infecting the system. It offers deeper visibility to reveal the true nature of the threat. It also reduces the time to rediscover a file with malicious code.

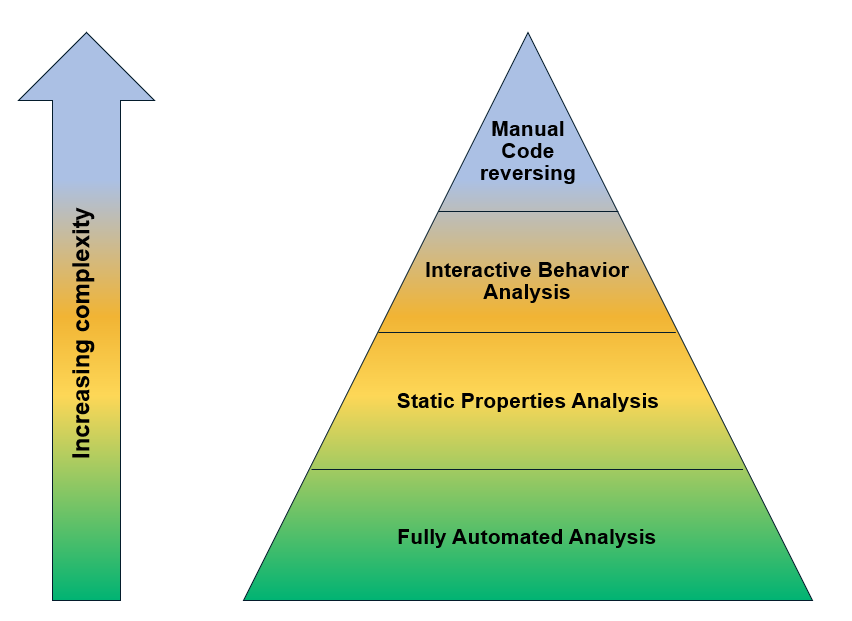
The challenge with dynamic analysis is that adversaries are smart, and they know sandboxes are out there, so they have become very good at detecting them. To deceive a sandbox, adversaries hide code inside them that may remain dormant until certain conditions are met. Only then does the code run.

**Hybrid Analysis:**

The hybrid analysis is a combination of basic and dynamic techniques to provide the best of both approaches. It detects malicious codes and extracts more indicators of compromise. It can even help detect this in sophisticated malware.

For example, one of the things hybrid analysis does is apply static analysis to data generated by behavioural analysis – like when a piece of malicious code runs and generates some changes in memory. Dynamic analysis would detect that, and analysts would be alerted to circle back and perform basic static analysis on that memory dump. As a result, more IOCs would be generated and zero-day exploits would be exposed.

**STAGES OF MALWARE ANALYSIS:**



**Static Properties Analysis:**

Static properties include strings embedded in the malware code, header details, hashes, metadata, embedded resources, etc. This type of data may be all that is needed to create IOCs, and they can be acquired very quickly because there is no need to run the program in order to see them. Insights gathered during the static properties analysis can indicate whether a deeper investigation using more comprehensive techniques is necessary and determine which steps should be taken next.

**Interactive Behavior Analysis:**

Behavioural analysis is used to observe and interact with a malware sample running in a lab. Analysts seek to understand the sample’s registry, file system, process and network activities. They may also conduct memory forensics to learn how the malware uses memory. If the analysts suspect that the malware has a certain capability, they can set up a simulation to test their theory. Behavioral analysis requires a creative analyst with advanced skills. The process is time-consuming and complicated and cannot be performed effectively without automated tools.

**Fully Automated Analysis:**

Fully automated analysis quickly and simply assesses suspicious files. The analysis can determine potential repercussions if the malware were to infiltrate the network and then produce an easy-to-read report that provides fast answers for security teams. Fully automated analysis is the best way to process malware at scale.

**Manual Code Reversing:**

In this stage, analysts reverse-engineer code using debuggers, disassemblers, compilers and specialized tools to decode encrypted data, determine the logic behind the malware algorithm and understand any hidden capabilities that the malware has not yet exhibited. Code reversing is a rare skill, and executing code reversals takes a great deal of time. For these reasons, malware investigations often skip this step and therefore miss out on a lot of valuable insights into the nature of the malware.

**MALWARE ANALYSIS PROCESS:**

Here’s a more in-depth insight into the malware analysis process:

**Step 1: Capture the malware**

Before the actual analysis, you need access to a malicious piece of code in an uncompressed format. You can use a tool like HoneyDB to attract malware and capture it in an investigation-friendly environment.

**Step 2: Build a malware lab**

A malware analysis lab is a safe environment where you can test different malware functionalities without any risk to nearby files. Typically, malware labs rely on virtual machines (VMs) to sandbox the entire exercise.

**Step 3: Install your tools**

You can use several tools to analyse malware, including open-source and paid options. There is plenty of support to be found on GitHub. You could also use Cuckoo Sandbox and other equivalent analysis enablers. These tools must be installed in your VMs.

**Step 4: Record the baseline**

Before running the malware, assess the operating environment and document it as your baseline. The tools installed in the VMs will help here — running these same tools later (after the malware is activated) indicates malware behavior and impact.

**Step 5: Commence your investigation**

There are several phases involved in the investigation step. Some require intense manual involvement, while others can gain from automation tools. Take the malware apart before initiating these phases to reveal its properties at every layer.

**Step 6: Document the results**

Depending on the tools you’re using, you will have detailed information on malware behavior, tendencies, and interaction patterns with its surrounding digital environment. Consolidate these results into an exhaustive document that forms the deliverable for your malware analysis exercise.

Malware analysis is at the heart of cybersecurity innovation today. Analysts can work with governments, non-profit organizations, research institutions, and corporates to develop the body of knowledge around malware.

**MALWARE ANALYSIS TOOLS:**

****

1. **PESTUDIO:**

PeStudio is a portable tool that performs malware assessments on executable files. Since the target file is never launched during the course of the investigation, you can safely evaluate the file, in addition to malware, without risk.

1. **PROCESS HACKER:**

Process Hacker allows a malware analyst to see what processes are running on a device. This can be useful when detonating a piece of malware to see what new processes are created by the malware and where these are being run from on disk. Malware will often try to hide by copying itself to a new location and then renaming itself, Process Hacker will display this activity occurring making it easy to identify how the malware is attempting to hide. This tool is also useful for pulling information from the memory of a process. This means that if a piece of malware is detonated then Process Hacker can be used to inspect the memory for strings, the strings found in memory will often return useful information such as IP addresses, domains, and user agents that are being used by the malware.

1. **PROCESS MONITOR:**

ProcMon is a powerful tool from Microsoft which records live filesystem activity such as process creations and registry changes. This is really handy when used in tandem with Process Hacker as a new process may be created and then quickly killed, this process can then be reviewed in the ProcMon capture. Using the prebuilt filters or process tree an analyst can quickly identify what processes were created, where the executable was run from, and the parent/child dependencies.

1. **PROC DOT:**

ProcDot allows a malware analyst to ingest the output from ProcMon and automatically generate a graphical representation of the captured data. Simply upload the csv into ProcDot and select the process name of the malware. Rather than creating filters and navigating hundreds of thousands of events you are now able to navigate a visual diagram of what recorded malware activity

1. **AUTORUN:**

Autoruns is another Microsoft tool that will display any installed software on a device that is set to launch when a machine is powered on. Malware can hide but ultimately it has to run and in order to survive a reboot a piece of malware must create a persistence mechanism

1. **FIDDLER:**

Malware will often use HTTP/HTTPS to contact its C2 servers and download additional malware or exfiltrate data. Using a tool such as Fiddler which acts as a web proxy allows this traffic to be captured and analyzed. This can prove useful when analysing a malicious document which incorporates macros to download a malicious payload, running fiddler allows a malware analyst to identify the domains that are hardcoded into the document and will be used to download the hosted malware.

1. **WIRESHARK:**

Malware will often use HTTP/HTTPS to contact its C2 servers and download additional malware or exfiltrate data. Using a tool such as Fiddler which acts as a web proxy allows this traffic to be captured and analyzed. This can prove useful when analysing a malicious document which incorporates macros to download a malicious payload, running fiddler allows a malware analyst to identify the domains that are hardcoded into the document and will be used to download the hosted malware.

1. **X64dbg:**

This tool is for manually debugging and reverse engineering malware samples, you need to have an understanding of assembly code to use this tool however once that learning curve has plateaued it allows a malware analyst to manually unpack.

1. **GHIDRA:**

Using Ghidra you are able to navigate the assembly code functions like in x64dbg, however, the key difference is that the code is not executed, it is disassembled so that it can be statically analyzed. Another key difference from x64dbg is that Ghidra will attempt to decompile the code into a human-readable output that is close to what the malware author will have written when creating the malware.

1. **RADARE2:**

Radare2 is an open-source reverse engineering framework used for static and dynamic analysis, digital forensics or software exploitation supporting multiple platforms, architectures and binary formats. Radare2 is shipped with a few other important tools for file metadata extraction, base conversion, unified binary diffing and many others.

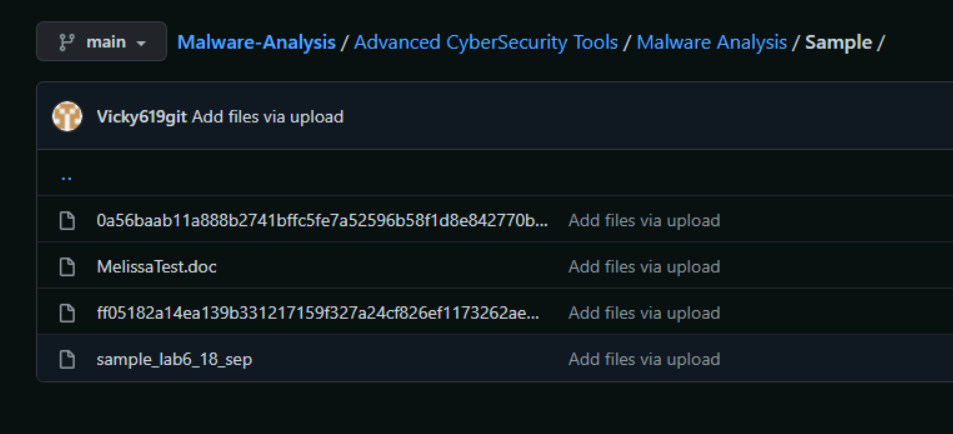
1. **CUCKOO SANDBOX:**

A Cuckoo Sandbox is a tool that is used to launch malware in a secure and isolated environment, the idea is the sandbox fools the malware into thinking it has infected a genuine host. The sandbox will then record the activity of the malware and then generate a report on what the malware has attempted to do while in this secure environment.

**MALWARE ANALYSIS – CASE STUDY**

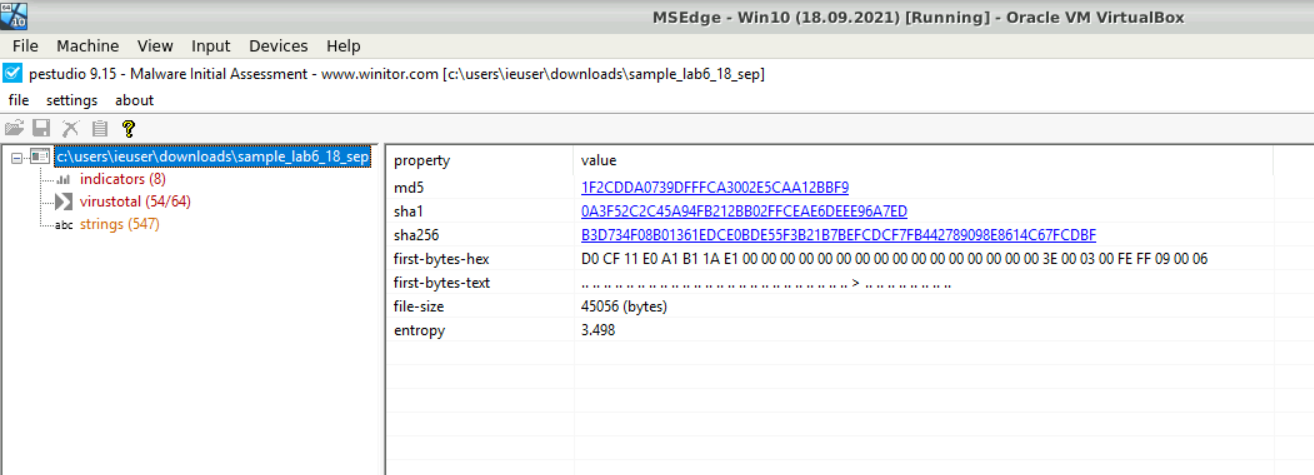
The sample used for this case study is available in the Git repository

<https://github.com/Vicky619git/Malware-Analysis/tree/main/Advanced%20CyberSecurity%20Tools/Malware%20Analysis/Sample>



**STATIC ANALYSIS USING PESTUDIO:**

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.

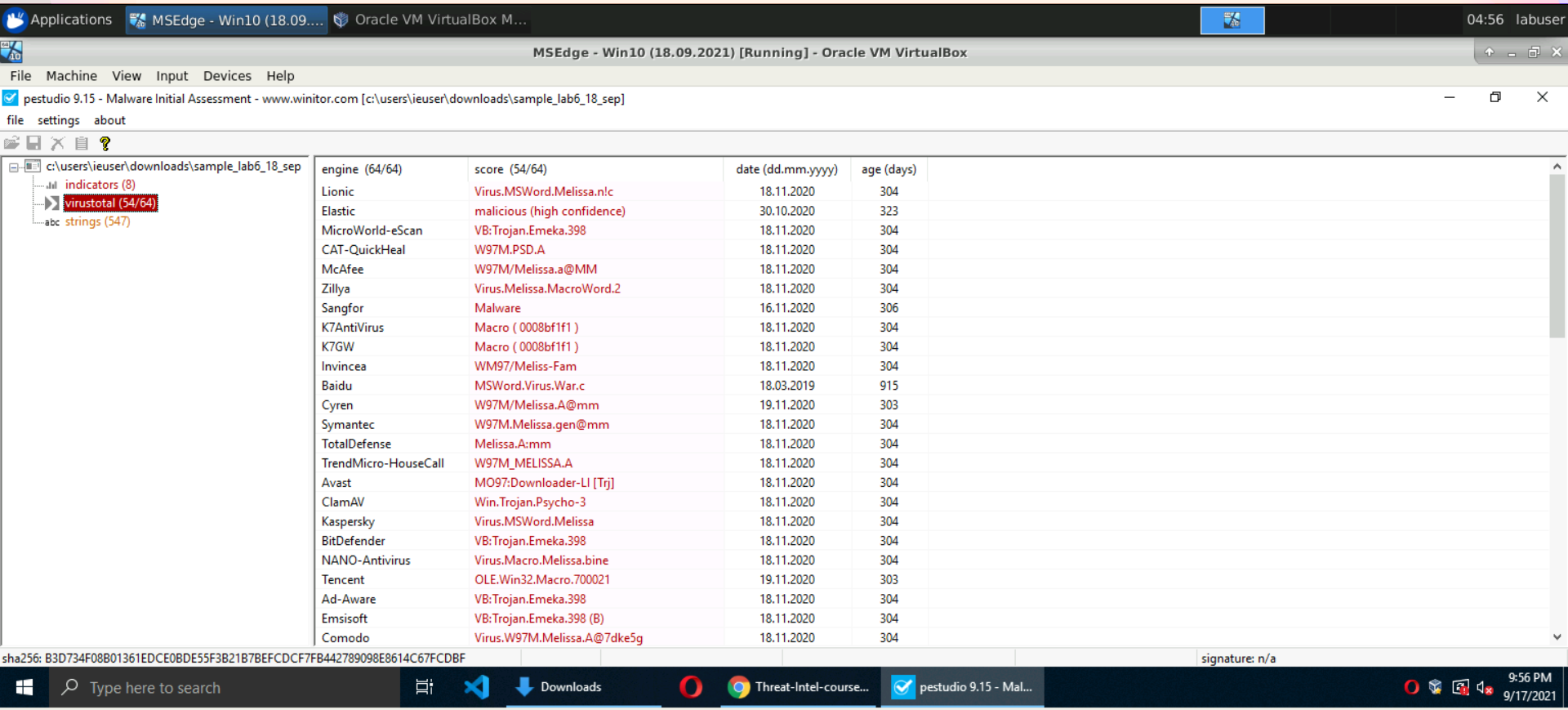


First-bytes: D0 CF

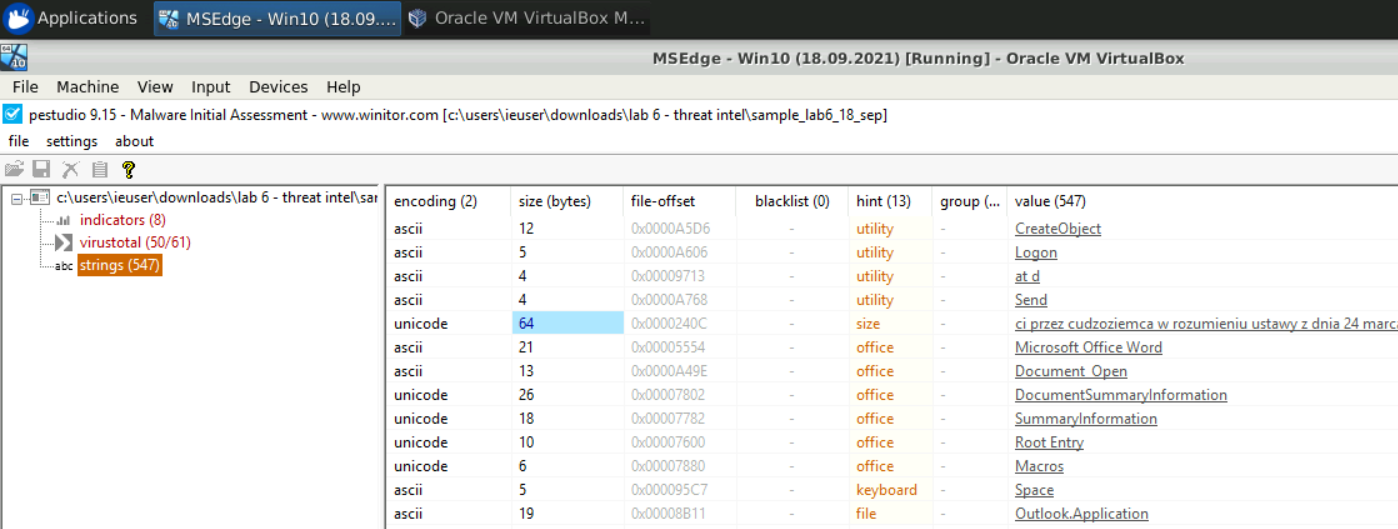
File type: (doc, xls, ppt, msg) Compound File Binary Format, a container format used for document by older versions of Microsoft Office. It is however an open format used by other programs as well. MS Word Document. (<https://en.wikipedia.org/wiki/List_of_file_signatures>)

SHA-256: b3d734f08b01361edce0bde55f3b21b7befcdcf7fb442789098e8614c67fcdbfs

File size: 44.00 KB (45056 bytes)

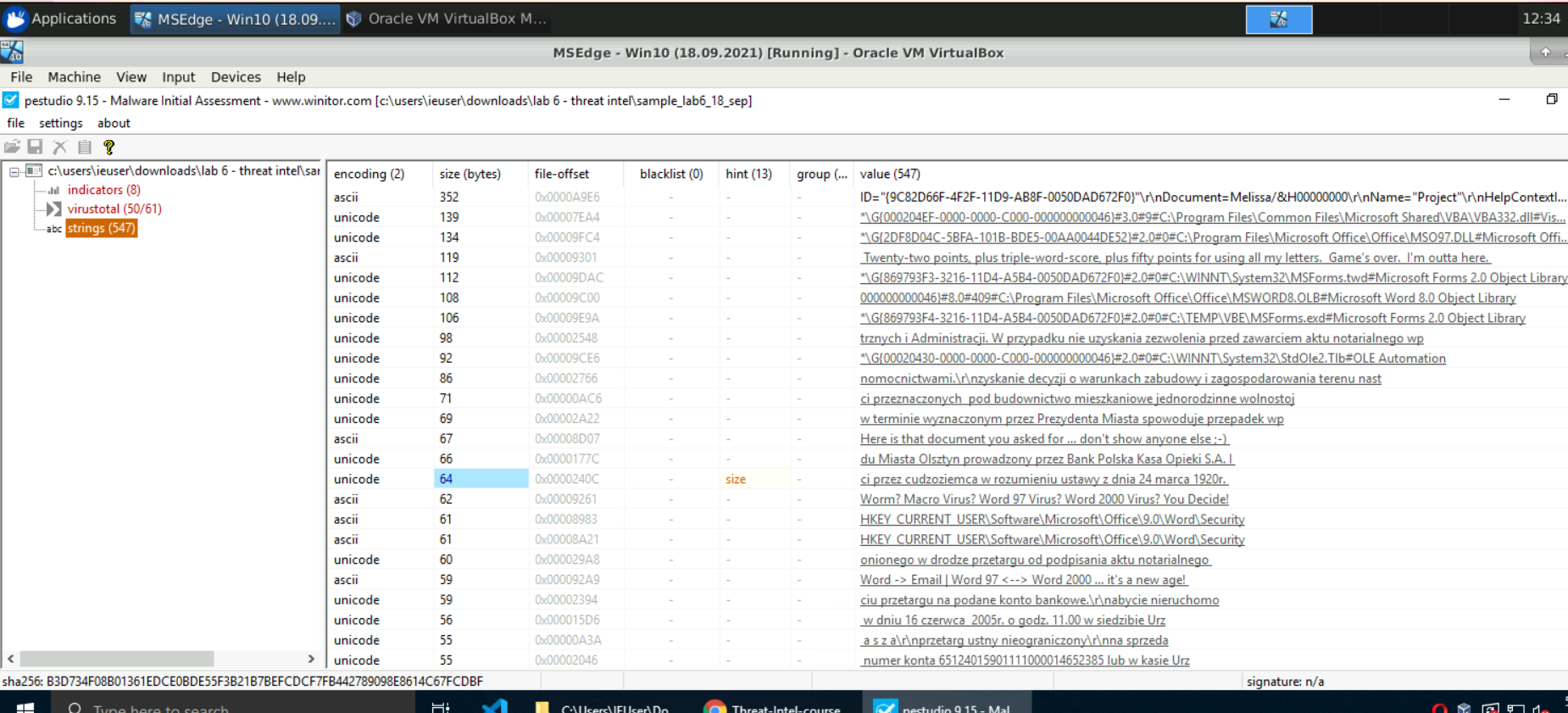


54 out of 64 confirm this sample to be malicious and Majority of AV engines specify the malware to be Melissa (W97M/Melissa).



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.



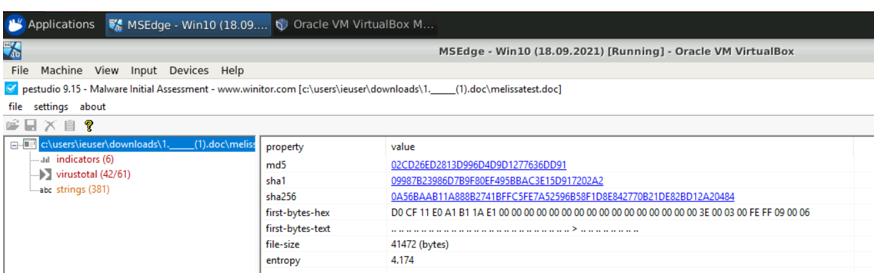
In the above screenshot, there are some strings which were displayed during the execution of the malware. Some of them are like

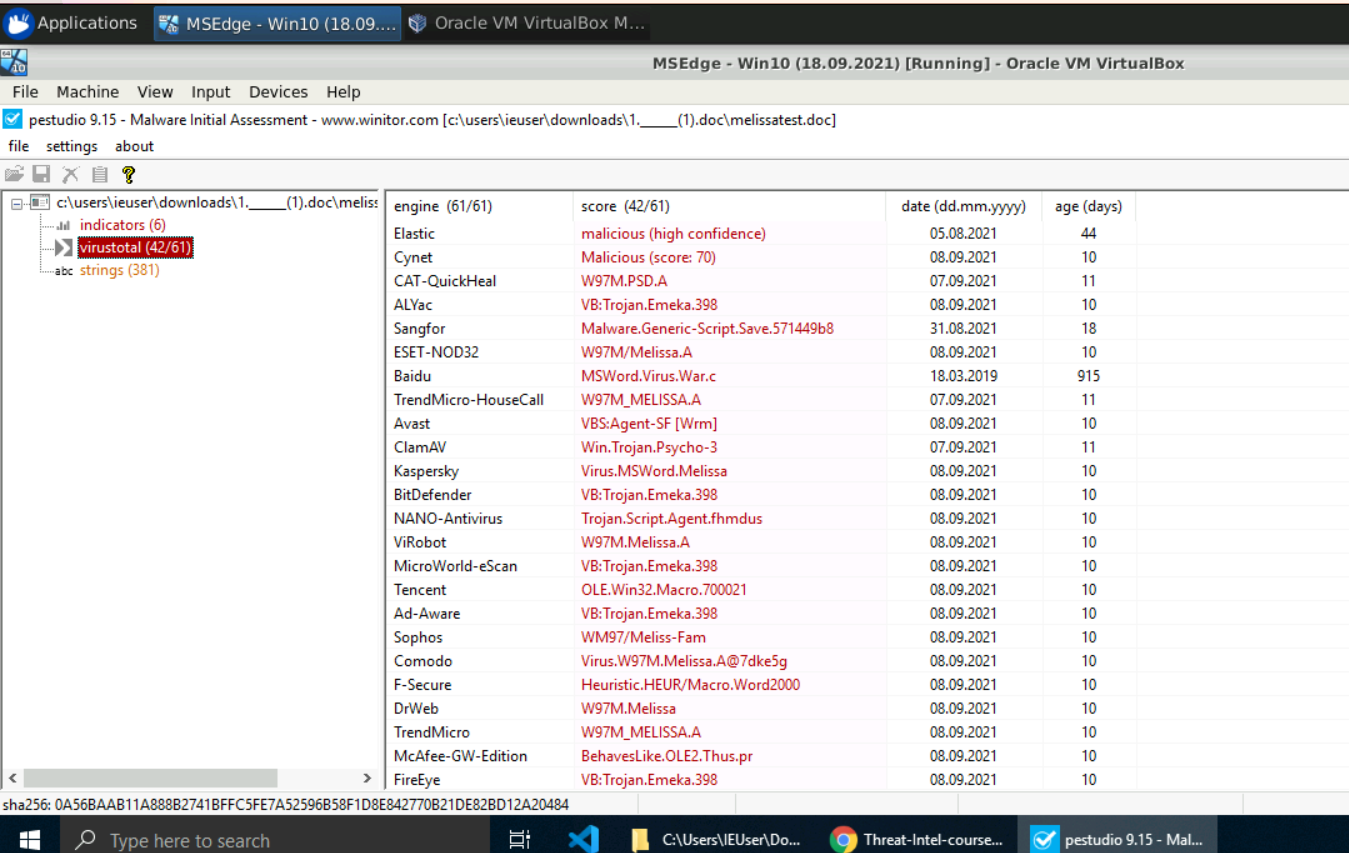
* Twenty-two points, plus triple-word-score, plus fifty points for using all my letters. Game's over. I'm outta here.
* Here is that document you asked for ... don't show anyone else ;-)
* Worm? Macro Virus? Word 97 Virus? Word 2000 Virus? You Decide!
* Word -> Email | Word 97 <--> Word 2000 ... it's a new age!
* WORD/Melissa written by Kwyjibo
* Important Message From

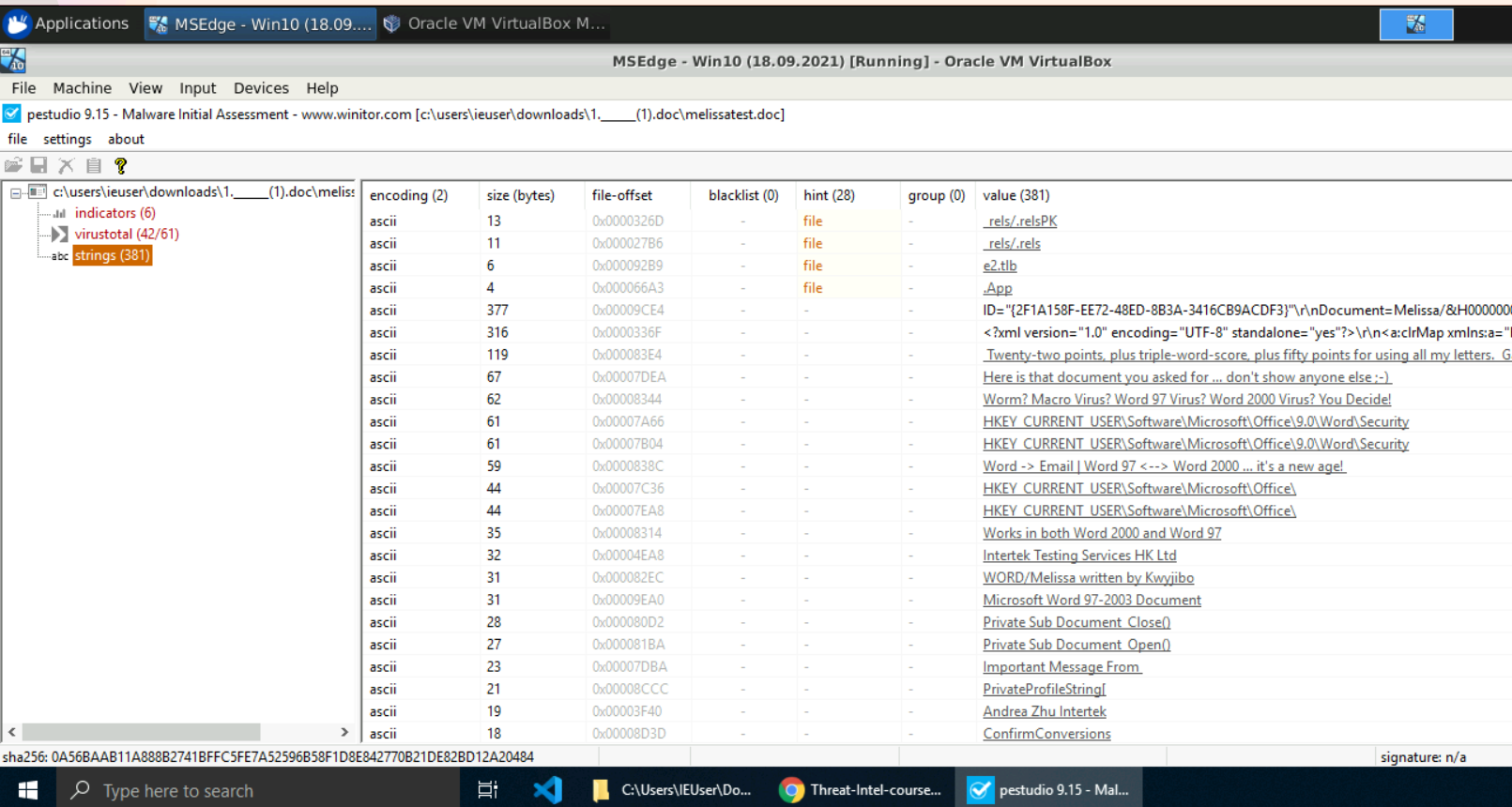
Similar analysis is done on another sample:

I was able to download one sample file from Anyrun website

MelissaTest.doc (MD5 - 02cd26ed2813d996d4d9d1277636dd91)





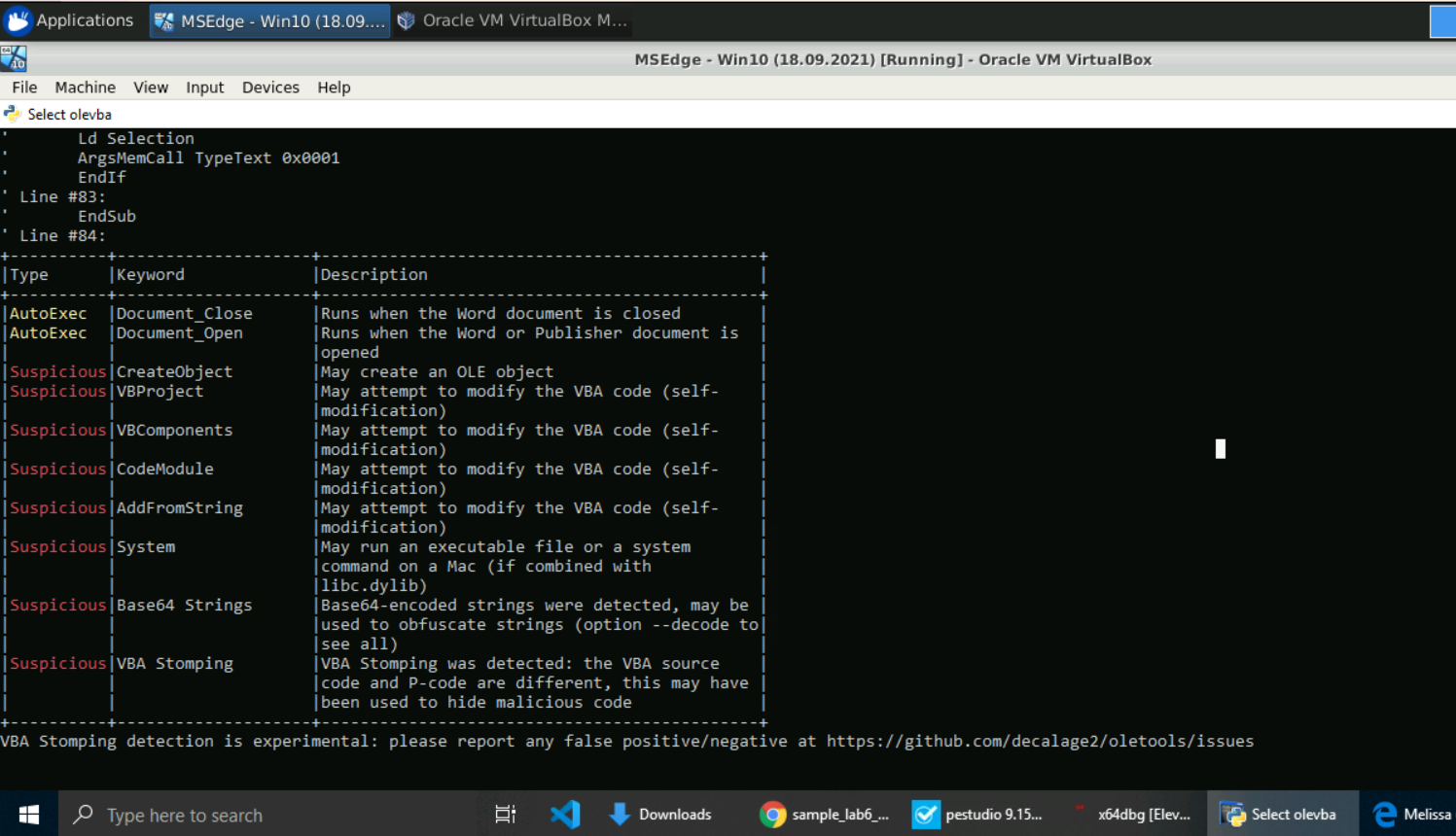


The string patterns are mostly similar in both the samples taken

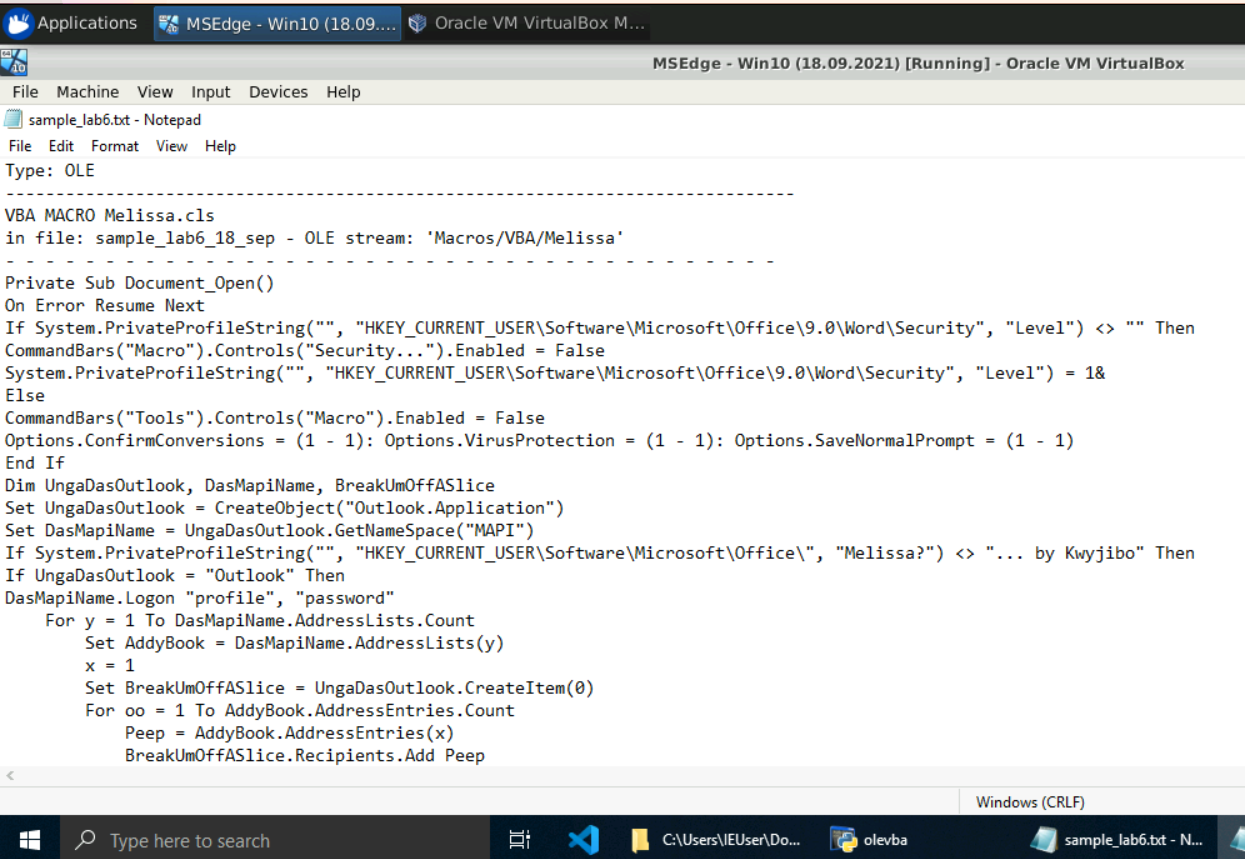
**OLEVBA:**

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)

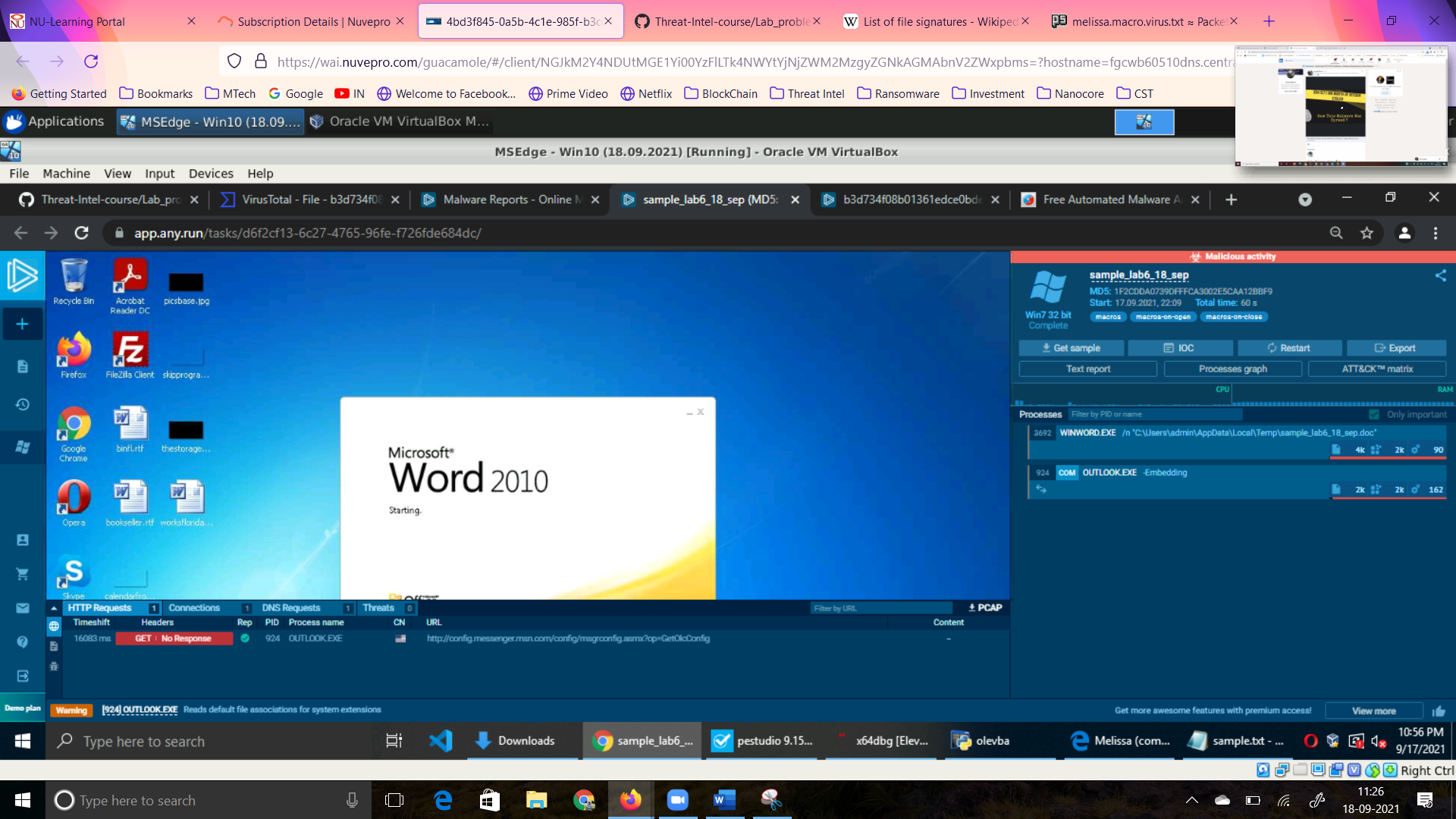
Since the samples are macro embedded, we are going to use OLEVBA to analyse the samples.

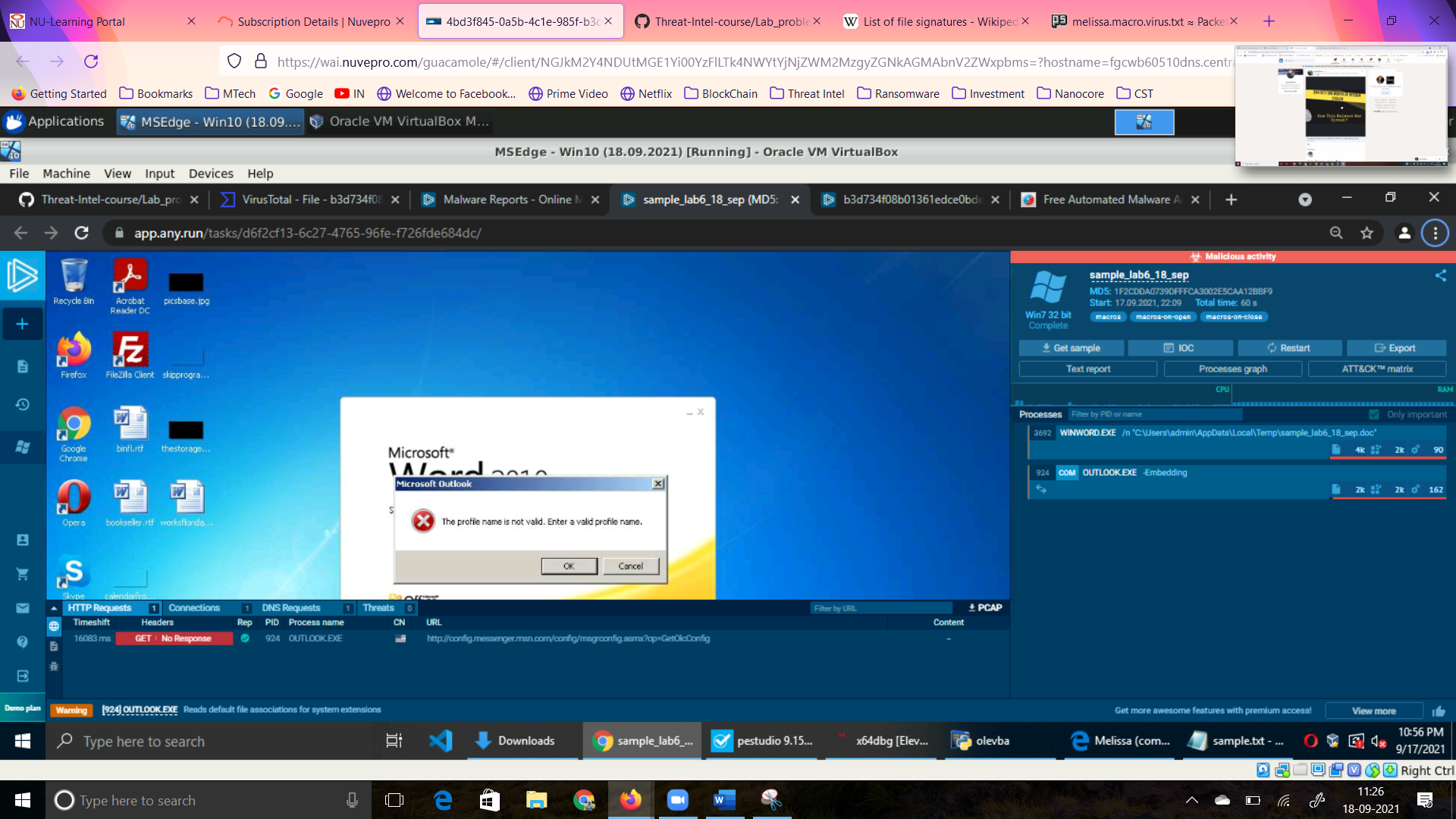
The OLEVBA Output from both the samples

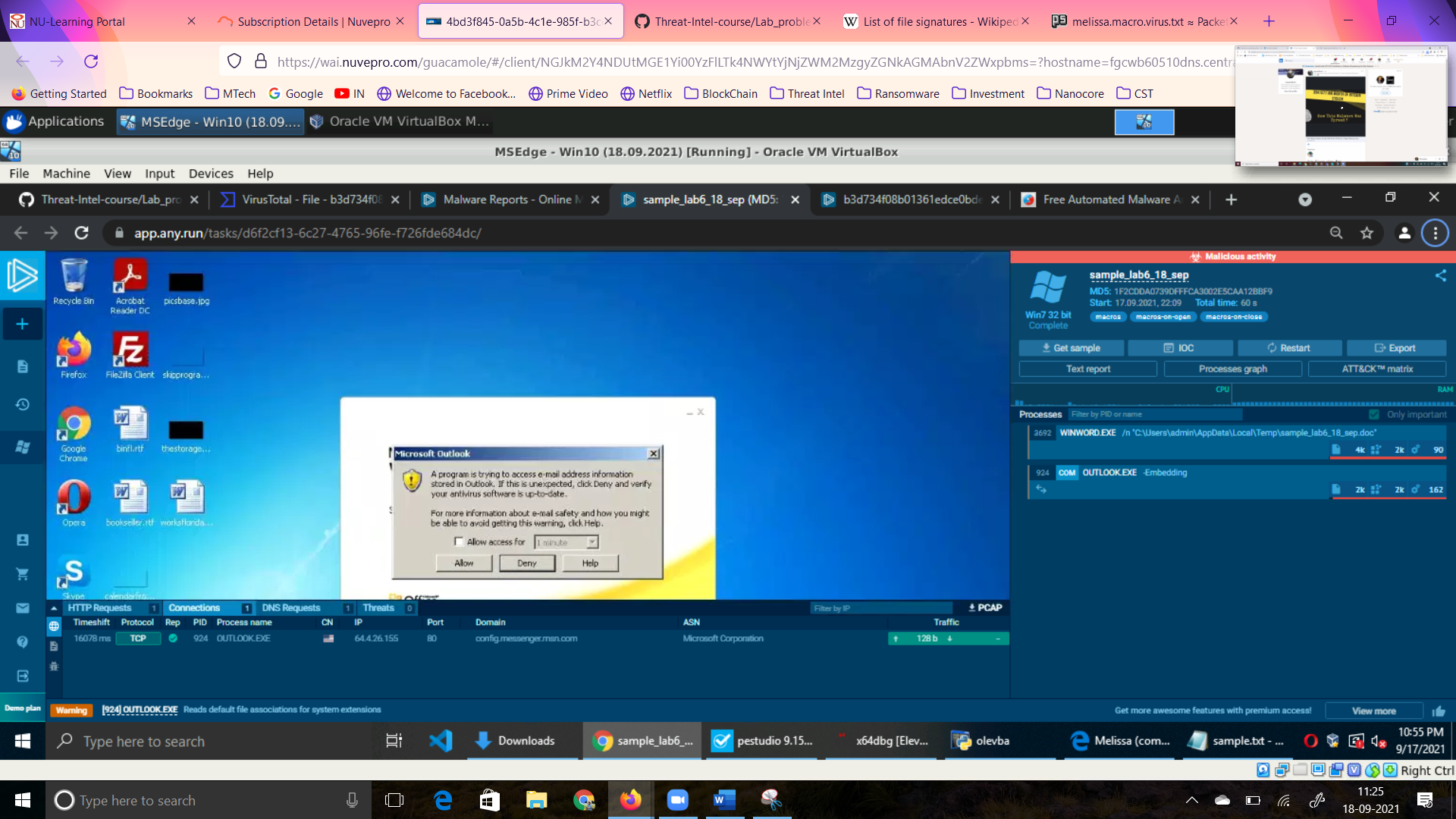


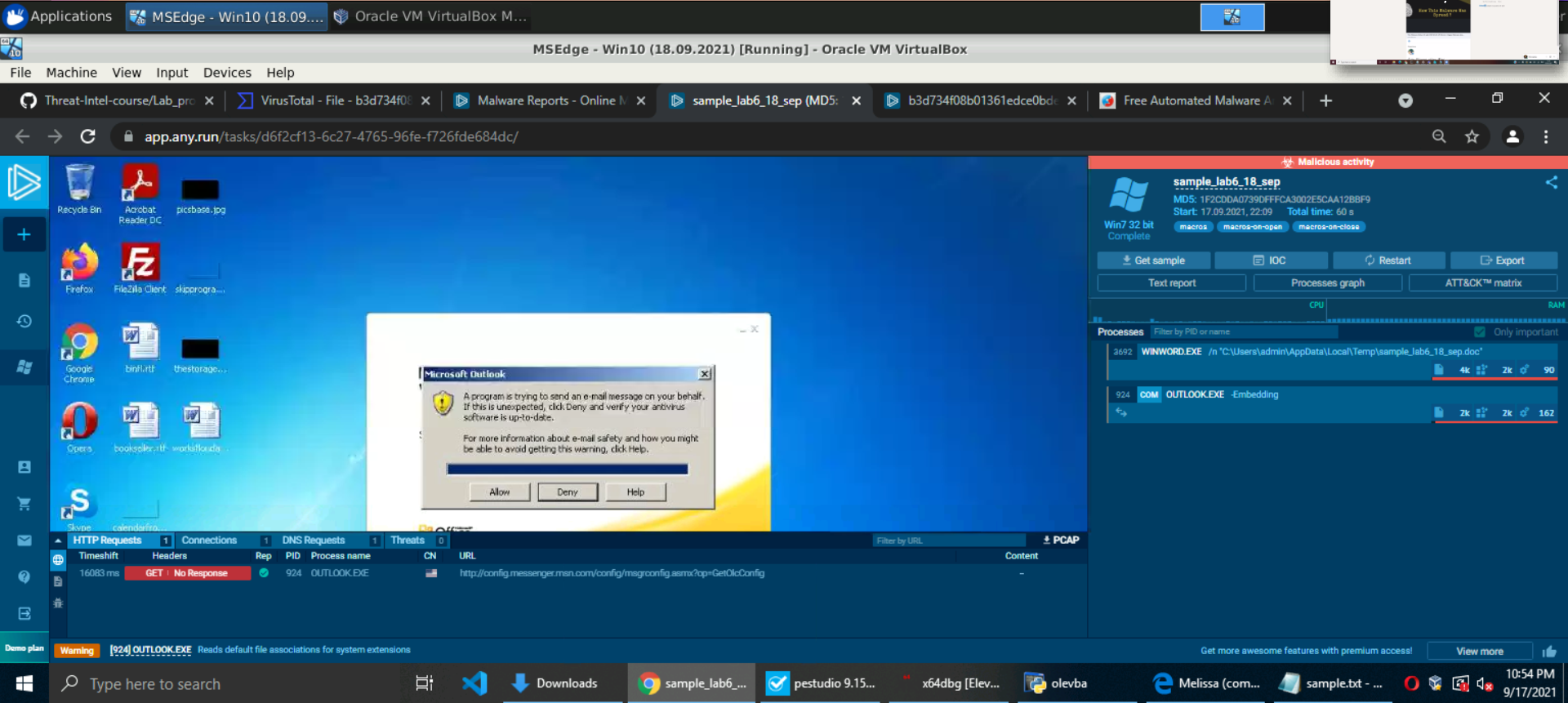


**ANYRUN:**









**MELISSA:**

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.

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.

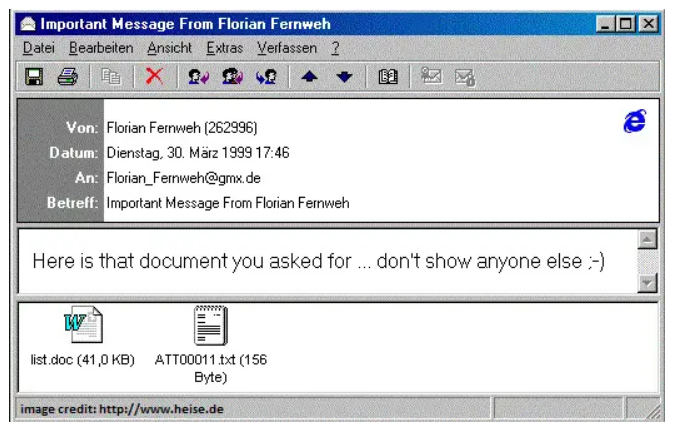
Next, the worm attempts to send itself out as an e-mail attachment. Since the mailing process is triggered once per each infected machine, the virus checks for the presence of its marker in the registry by comparing the value: HKCU\Software\Microsoft\Office\Melissa? against the string: "... by Kwyjibo".

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>



(Image Source: <https://cyberhoot.com/cybrary/melissa-virus/> )

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

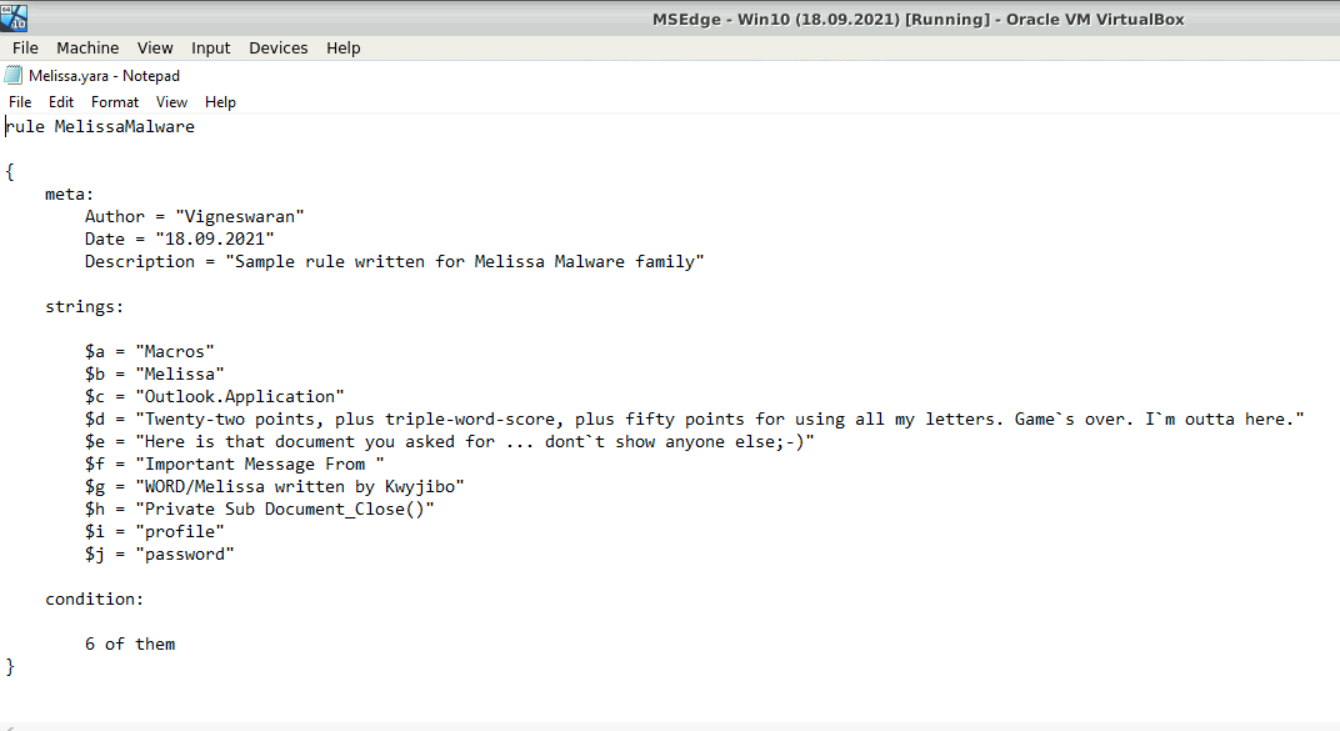
Works in both Word 2000 and Word 97

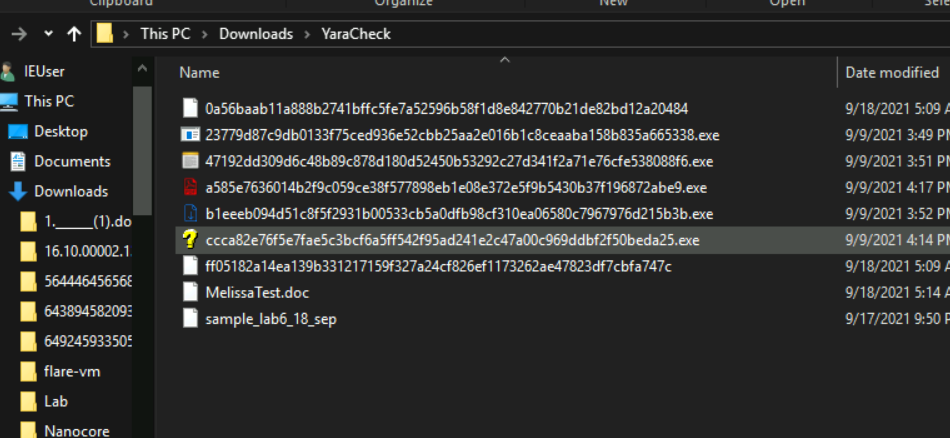
Worm? Macro Virus? Word 97 Virus? Word 2000 Virus? You Decide!

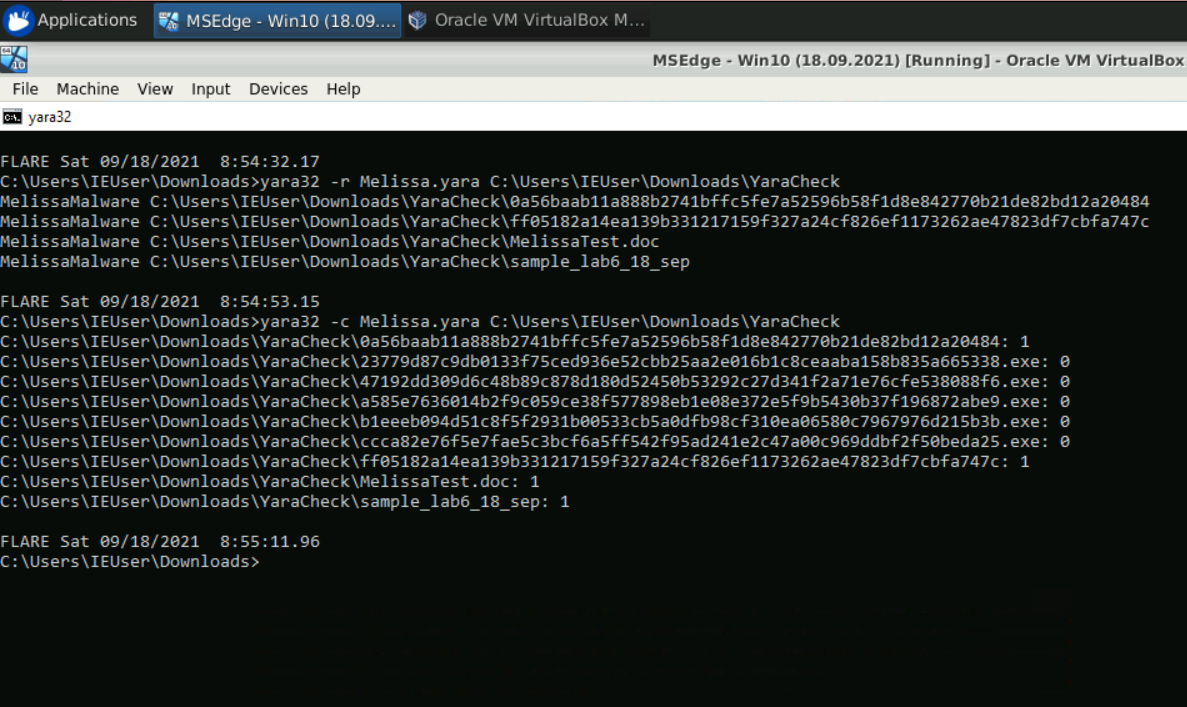
Word -> Email | Word 97 <--> Word 2000 ... it's a new age!

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 ;).”

**YARA RULE:**







**References:**

1. <https://www.f-secure.com/v-descs/melissa.shtml>
2. <https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Virus%3AW97M%2FMelissa.A>
3. <https://cyberhoot.com/cybrary/melissa-virus/>
4. <https://www.varonis.com/blog/malware-analysis-tools/>