NEW YORK INSTITUTE
OF TECHNOLOGY

Lab 6 - Malware Detection
Zhijun Jiang 1339481

Task 1: Download the malware

Download the provided malware sample from the assignment. Follow the following tasks to complete your lab.

Task 2: Analyse the malware sample

perform static analysis on your sample and find important information about your malware. Search for a detailed analysis report of the given malware to check what's important in the malware sample. Report your findings about the malware and what the purpose of malware is. Explain at least on incident that uses this malware and how they leverage the malware.

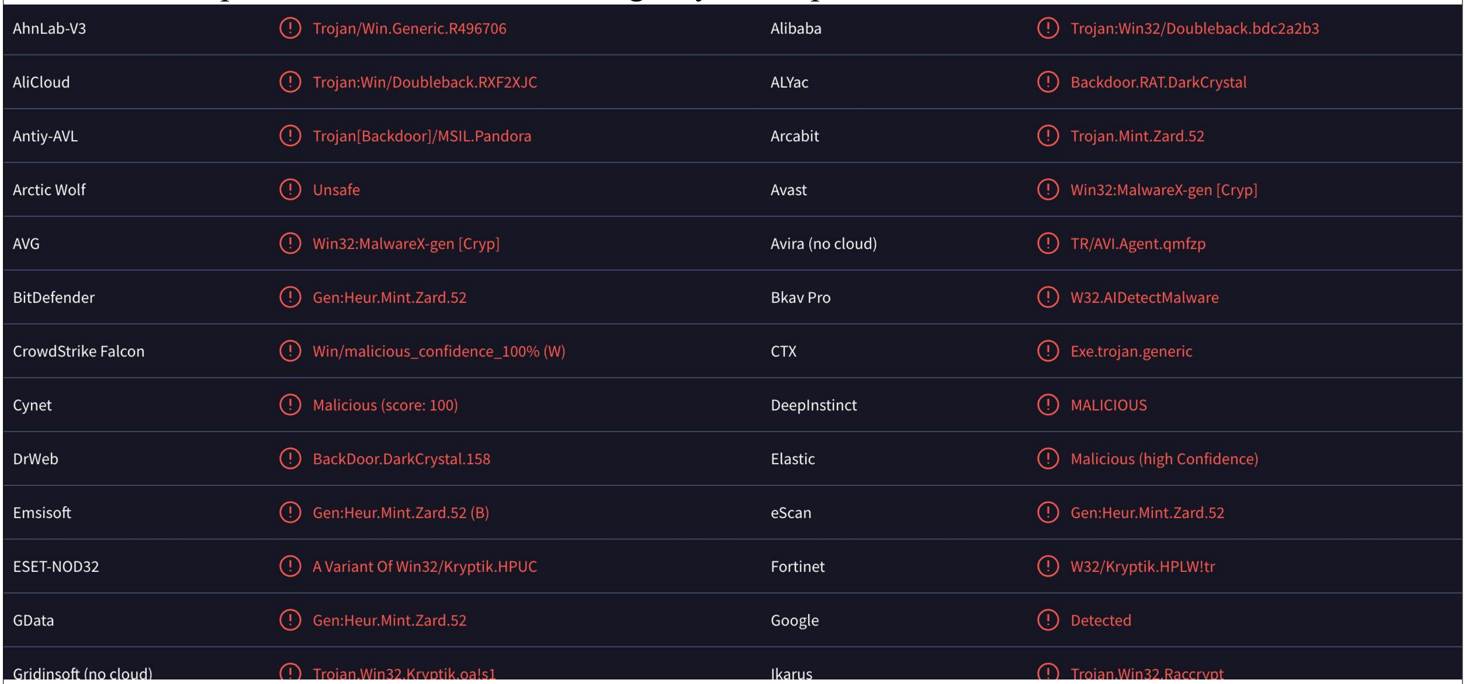
For this task you can use a combination of the tools mentioned and/or websites such as VirusTotal.

Thereafter, provide the answers to the following questions about the malware along with the related screenshots:

1. What is the md5sum? What of interest does VirusTotal Report?

4825e7df93d8acb3dd236cc14c342a71

VirusTotal report this file has been flags by multiple vendors such as



AhnLab-V3	Trojan/Win.Generic.R496706	Alibaba	Trojan:Win32/Doubleback.bdc2a2b3
AliCloud	Trojan:Win/Doubleback.RXF2XJC	ALYac	Backdoor.RAT.DarkCrystal
Antiy-AVL	Trojan[Backdoor]/MSIL.Pandora	Arcabit	Trojan.Mint.Zard.52
Arctic Wolf	Unsafe	Avast	Win32:MalwareX-gen [Cryp]
AVG	Win32:MalwareX-gen [Cryp]	Avira (no cloud)	TR/AVI.Agent.qmfzp
BitDefender	Gen:Heur.Mint.Zard.52	Bkav Pro	W32.AIDetectMalware
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	CTX	Exe.trojan.generic
Cynet	Malicious (score: 100)	DeepInstinct	MALICIOUS
DrWeb	BackDoor.DarkCrystal.158	Elastic	Malicious (high Confidence)
Emsisoft	Gen:Heur.Mint.Zard.52 (B)	eScan	Gen:Heur.Mint.Zard.52
ESET-NOD32	A Variant Of Win32/Kryptik.HPUC	Fortinet	W32/Kryptik.HPLW!tr
GData	Gen:Heur.Mint.Zard.52	Google	Detected
Gridinsoft (no cloud)	Trojan.Win32.Kryptik.oa!s1	Ikarus	Trojan.Win32.Raccrypt

2. When is the file compiled?

2022-06-06 08:46:18 UTC

3. List a few imports or sets of imports and describe how the malware might use them.

Files opened

C:\Program Files (x86)\Common Files\Oracle\Java\javapath\

C:\ProgramData\Microsoft\Windows\AppRepository\Packages\microsoft.windowscommunicationsapps_16005.14326.21962.0_x64__8wekyb3d8bbwe\S-1-5-18.pckgdep

C:\ProgramData\Microsoft\Windows\AppRepository\Packages\microsoft.windowscommunicationsapps_16005.14326.21962.0_x64__8wekyb3d8bbwe\S-1-5-21-4005801669-2598574594-602355426-1001.pckgdep

C:\Users\

C:\Users\<USER>\

C:\Users\<USER>\AppData\

C:\Users\<USER>\AppData\Local\Microsoft

C:\Users\<USER>\AppData\Local\Microsoft\CLR_v4.0_32

C:\Users\<USER>\AppData\Local\Microsoft\CLR_v4.0_32\UsageLogs

C:\Users\<USER>\AppData\Local\Microsoft\CLR_v4.0_32\UsageLogs\InstallUtil.exe.log

List at least 3 common Windows APIs used. Can you spot any suspicious one among them? Please mention why do you think they're suspicious.

VirtualAlloc / VirtualAllocEx - Memory allocation (often used by malware)

LoadLibraryA / LoadLibraryW - Dynamic loading of libraries

GetProcAddress - Getting function addresses (used in API hooking)

WriteFile / CreateFileA - File operations

Windows error messages and runtime library references

It tried to call install util which is suspicious,making me suspect that it will download and install unknown software in my PC

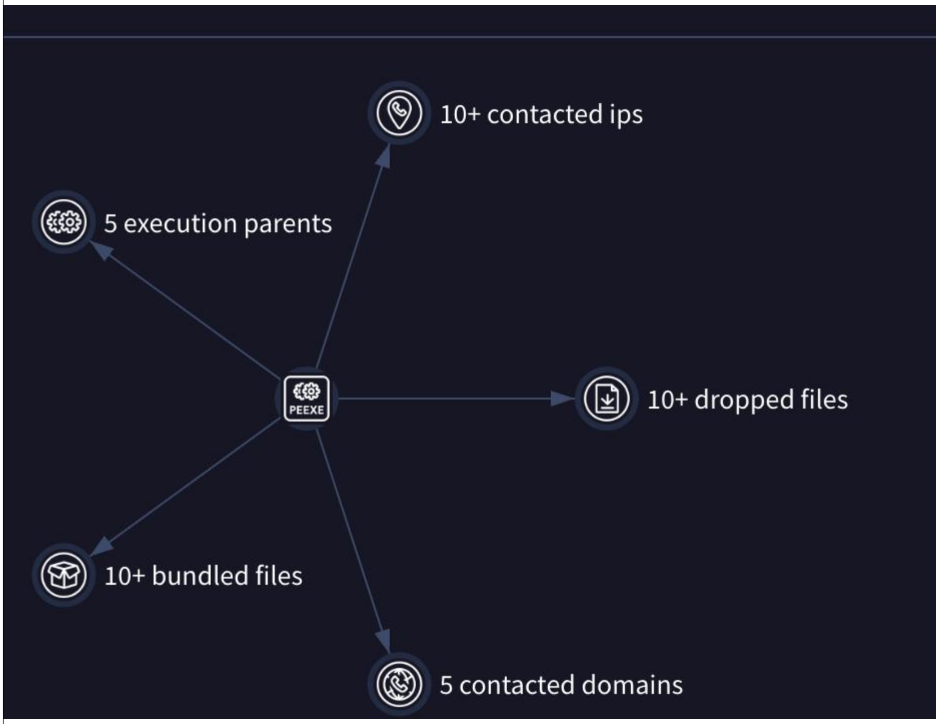
5. What are a few strings that stick out to you and why?

Matches rule HackTool - Koadic Execution by wagga, Jonhnathan Ribeiro, oscd.community at Sigma Integrated Rule Set (GitHub)

Detects command line parameters used by Koadic hack tool

Matches rule Python Initiated Connection by frack113 at Sigma Integrated Rule Set (GitHub)

Adversaries may attempt to get a listing of services running on remote hosts, including those



6. What is the language of the malware and what's the entry point address?

Entry Point : 59840

file malware.exe

malware.exe: PE32 executable (GUI) Intel 80386, for MS Windows

7. What are the sections of the malware and what's the entropy of each section?

8. Are there any indications that this malware is packed? What are they? What is it packed with?

9. Have you noticed any network communication in your research for the malware? If yes, what are the suspicious URLs, and IPs?

URLs like https://d.symcb.com/cps, http://s.symcd.com, http://ts-crl.ws.symantec.com

Certificate-related URLs suggesting potential certificate manipulation

What are some of the suspicious activities based on the report's findings. Explain why they are suspicious.

Task 3. Write and run YARA rules on the malware sample

Based on your findings on task 2, write a Yara rule to match your findings. Your signature needs to meet the following matching criteria:

- Include file properties, strings, and code patterns (if you find something).
  - File size
  - Imports and functions used: at least two
  - Static strings like interesting error messages, names, static values of variables
- The strings need to have at least one example of each of the following types:
  - Static strings
  - Binary data containing wild cards (? and ??)

Test the YARA rule against the provided sample and adjust as needed.

YARA Rule Example:

```
rule Advanced_Malware_Detection_Lab6_Task3
{
   meta:
      description = "Advanced malware detection based on string analysis from Lab 6-1"
      author = "Security Analyst"
      date = "2024-01-20"
      version = "1.0"
      sample_hash = "unknown"

   strings:
      // Suspicious API calls commonly used by malware
      $api1 = "VirtualAlloc" ascii
      $api2 = "VirtualAllocEx" ascii
      $api3 = "GetProcAddress" ascii
      $api4 = "LoadLibraryA" ascii
      $api5 = "LoadLibraryW" ascii
      $api6 = "WriteFile" ascii
      $api7 = "CreateFileA" ascii

      // Network-related indicators
      $url1 = "http://s.symcd.com" ascii
      $url2 = "https://d.symcb.com" ascii
      $url3 = "http://ts-crl.ws.symantec.com" ascii
      $url4 = "http://ts-ocsp.ws.symantec.com" ascii

      // Obfuscated or suspicious text patterns
      $obfus1 = "Madelimi ticataw cisa veque hopes wipap bepacet fajij fonenin pef" ascii
      $obfus2 = "Kaquajip" ascii
```

```
        $obfus3 = "Hoqui jatipo" ascii
        $obfus4 = "Codase samo jad welo bixipox" ascii

        // Runtime error patterns (potential packing indicators)
        $runtime1 = "Microsoft Visual C++ Runtime Library" ascii
        $runtime2 = "bad allocation" ascii
        $runtime3 = "HEAP CORRUPTION DETECTED" ascii
        $runtime4 = "This program cannot be run in DOS mode" ascii

        // File header
        $mz_header = { 4D 5A }  // MZ header

        // Certificate and trust network strings (potential cert manipulation)
        $cert1 = "Symantec Trust Network" ascii
        $cert2 = "VeriSign Trust Network" ascii
        $cert3 = "www.entrust.net/legal-terms" ascii

    condition:
        // Must be a Windows PE file
        $mz_header at 0 and

        // At least 3 suspicious API calls
        3 of ($api*) and

        // Either network indicators OR obfuscated strings OR certificate
manipulation
        (
            any of ($url*) or
            2 of ($obfus*) or
            any of ($cert*)
        ) and

        // Runtime indicators suggesting potential packing/obfuscation
        any of ($runtime*) and

        // File size constraints (typical malware size range)
        filesize > 100KB and filesize < 10MB
}

rule Specific_Malware_Patterns_Lab6
{
    meta:
        description = "Specific patterns found in the analyzed malware sample"
        author = "Security Analyst"
        reference = "Lab 6-1 Analysis"

    strings:
        // Specific suspicious string combinations found
        $pattern1 = "Vec rom sam wed fota vejag mejarido naqu niwefo" ascii
        $pattern2 = "Vacagil xeb saweha bema bavi watataj nejeq caf vimoxi wik" ascii
        $pattern3 = "Quec!" ascii
        $pattern4 = "l1p1t1" ascii

        // Memory allocation and manipulation
        $mem1 = "VirtualAllocEx" ascii
        $mem2 = "GetProcAddress" ascii

        // File operations
        $file1 = "CreateFileA" ascii
        $file2 = "WriteFile" ascii

    condition:
        2 of ($pattern*) and
        all of ($mem*) and
        any of ($file*)
}

rule Certificate_Manipulation_Indicator
{
    meta:
        description = "Detects potential certificate manipulation based on observed URLs"
        author = "Security Analyst"

    strings:
        $symantec1 = "https://d.symcb.com/cps" ascii
        $symantec2 = "http://ts-crl.ws.symantec.com" ascii
        $symantec3 = "http://ts-ocsp.ws.symantec.com" ascii
        $trust = "Symantec Trust Network" ascii

    condition:
        2 of them
}
```

```
rule SampleMalware {
  meta:
     description = "Detects the presence of the sample malware"
     author = "Your Name"
     reference = "Provide any relevant references"
  strings:
     $magic = ???     // PE Signature
     $suspicious_string = "evil_command" wide
     $encoded_string = { 41 42 43 44 45 }     // Example of an encoded
string
  condition:
     $magic at 0 and $suspicious_string or $encoded_string
}
```

Submission

Submit a short report about your findings with screenshots, including the YARA rule you created. Include any observations, potential indicators of compromise, and recommendations for further analysis if necessary. You need to submit your Yara signature file with .txt extension separately, as well.

Note: Exercise caution when handling malware samples. Perform the analysis in an isolated environment to prevent unintended consequences. Always adhere to ethical guidelines and legal requirements.