Static Analysis

1. Virus Total Analysis

Hash Analysis

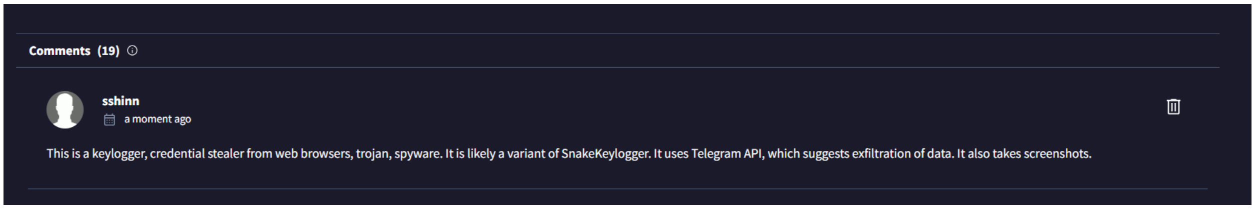
- File Hash: [Insert MD5, SHA-1, SHA-256 hash value]
  - MD5: 624401070bba5e7be70e21e3da0a1b45
  - SHA-1: c4ea9df10fc2043b917d6276e39b7bb4639c251d
  - SHA-256: 668e51b1f586bbd6b1d35890299933ea7cfa2cd62c47febb11ed2db5726bebfc
- Method of hash acquisition: [Describe process] VirusTotal
- Number of vendors flagging as malicious: [X/Y]
  - 59/72
- Analysis of vendor results:
  - [Discuss patterns in detection]
  - [Common malware names identified]
    - Trojan, SnakeKeylogger, MSIL Heracles, PWSX, GrayWare, SpyAgentAES. PwStealer, Spy.Echelon, Spyware, HEUR.Trojan, Trojan.YakbeexMSIL.ZZ4, Artemis!624401070BBA
  - [Notable vendor disagreements]
    - Some say Trojan, some say Spyware, some say KeyLogger. It could be all of those things, or not

File History

- First Submission Date: [Date]
  - First Submission: 2025-02-10 22:02:09 UTC
- File Creation Date from Windows: [Date]
  - Creation Time: 2023-07-16 00:32:11 UTC
- Analysis of submission timeline:
  - [Discussion of file age]: The file is 1.5 years old
  - [Notable resubmissions or changes]: It was created by MichaelGalde, not sure how it was flagged as a Snake Keylogger at some point in the submission history

Community Score

- [Link to your VirusTotal community contribution]:
  - https://www.virustotal.com/gui/file/668e51b1f586bbd6b1d35890299933ea7cfa2cd62c47febb11ed2db5726bebfc/community
  - Username: sshinn
- Summary of initial findings posted to the community:



Comments (19)

sshinn

a moment ago

This is a keylogger, credential stealer from web browsers, trojan, spyware. It is likely a variant of SnakeKeylogger. It uses Telegram API, which suggests exfiltration of data. It also takes screenshots.

- [Key observations] variant of snakekeylogger, logs keystrokes and takes screenshots, trojan, spyware
- [Potential indicators of compromise]: it is detected by antivirus

2. Detect It Easy (DIE) Analysis

File information

- File type: [Type]: PE32
- Architecture: [Architecture]: i386
- Compiler: [Compiler information]: VB.NET
- Additional relevant information:
  - [List notable file characteristics]
    - Library: .NET Framework(v4.0, CLR v4.0.30319)
    - File Size: 184.00 - 200 KiB
    - Operating system:Windows 95
  - [Unusual headers or structures]
    - .text, .rsrc., .reloc (packed)
    - mscore.dll import

Memory Map Analysis

- Section breakdown:
  - [.text section analysis]
    - unpacked: File Offset:00000200, Virtual address:00402000, Size: 0001e000
  - [.data section analysis]
    - no data section
  - [.rsrc section analysis]
    - packed: File offset:0001e200, Virtual address:00420000, Size: 0000fc00
    - Unpacked:

```
1 VERSIONINFO
FILEVERSION 1,0,0,0
PRODUCTVERSION 1,0,0,0
FILEOS 0x4
FILETYPE 0x1
{
BLOCK "StringFileInfo"
{
      BLOCK "000004B0"
      {
            VALUE "Comments", "Add Money to your Bank Account"
            VALUE "CompanyName", "United Kingdoom of America"
            VALUE "FileDescription", "2025 February Malware Analysis"
            VALUE "FileVersion", "1.0.0.0"
            VALUE "InternalName", "Week4.exe"
            VALUE "LegalCopyright", "Copyright \xA9  2025 Michael Galde"
            VALUE "LegalTrademarks", "Bank of America"
            VALUE "OriginalFilename", "Week4.exe"
            VALUE "ProductName", "Bank Analysis Software"
            VALUE "ProductVersion", "1.0.0.0"
            VALUE "Assembly Version", "1.0.0.0"
      }
}

BLOCK "VarFileInfo"
{
      VALUE "Translation", 0x0000 0x04B0
}
}


MANIFEST:
<?xml version="1.0" encoding="utf-8"?>
<assembly manifestVersion="1.0" xmlns="urn:schemas-microsoft-com:asm.v1">
  <assemblyIdentity version="1.0.0.0" name="MyApplication.app"/>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
    <security>
      <requestedPrivileges xmlns="urn:schemas-microsoft-com:asm.v3">
        <!-- UAC Manifest Options
             If you want to change the Windows User Account Control level 
replace the 
             requestedExecutionLevel node with one of the following.

        <requestedExecutionLevel  level="asInvoker" uiAccess="false" />
        <requestedExecutionLevel  level="requireAdministrator" 
uiAccess="false" />
        <requestedExecutionLevel  level="highestAvailable" uiAccess="false" />

            Specifying requestedExecutionLevel element will disable file and 
registry virtualization.
            Remove this element if your application requires this 
virtualization for backwards
            compatibility.
        -->
        <requestedExecutionLevel level="asInvoker" uiAccess="false" />
      </requestedPrivileges>
    </security>
  </trustInfo>

  <compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">
    <application>
      <!-- A list of the Windows versions that this application has been 
tested on and is
           is designed to work with. Uncomment the appropriate elements and
```

```
Windows will
           automatically selected the most compatible environment. -->

      <!-- Windows Vista -->
      <!--<supportedOS Id="{e2011457-1546-43c5-a5fe-008deee3d3f0}" />-->

      <!-- Windows 7 -->
      <!--<supportedOS Id="{35138b9a-5d96-4fbd-8e2d-a2440225f93a}" />-->

      <!-- Windows 8 -->
      <!--<supportedOS Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}" />-->

      <!-- Windows 8.1 -->
      <!--<supportedOS Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}" />-->

      <!-- Windows 10 -->
      <!--<supportedOS Id="{8e0f7a12-bfb3-4fe8-b9a5-48fd50a15a9a}" />-->

    </application>
  </compatibility>

  <!-- Indicates that the application is DPI-aware and will not be
automatically scaled by Windows at higher
       DPIs. Windows Presentation Foundation (WPF) applications are
automatically DPI-aware and do not need
       to opt in. Windows Forms applications targeting .NET Framework 4.6
that opt into this setting, should
       also set the 'EnableWindowsFormsHighDpiAutoResizing' setting to 'true'
in their app.config. -->
  <!--
  <application xmlns="urn:schemas-microsoft-com:asm.v3">
    <windowsSettings>
      <dpiAware
xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">true</dpiAware>
    </windowsSettings>
  </application>
  -->

  <!-- Enable themes for Windows common controls and dialogs (Windows XP and
later) -->
  <!--
  <dependency>
    <dependentAssembly>
      <assemblyIdentity
          type="win32"
          name="Microsoft.Windows.Common-Controls"
          version="6.0.0.0"
          processorArchitecture="*"
          publicKeyToken="6595b64144ccf1df"
          language="*"
        />
    </dependentAssembly>
  </dependency>
  -->

</assembly>
```

    - [Other relevant sections]
    - reloc:
        - unpacked: file offset:0002de00, Virtual address:00430000, Size: 00000200
- Notable findings:
    - [Unusual section permissions]
    - [Section size anomalies]
    - I found a lot of interesting information in the .rsrc section. I found out that United Kingdoom of America is the organization that created the malware banking service. I found out that "adding money to your bank account" is the function of the program. I found xml code, most likely for a .NET executable, which can be modified to modify the program. I modified the program to have highest privileges, and decided to run it to see what would happen.

String Analysis

- Notable strings discovered:
    - cmd.exe
    - software\microsoft\windows\currentversion\run
    - https://api.telegram.org/bot
    - /sendMessage?chat_id=
    - Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; .NET CLR1.0.3705;)
    - http://checkip.dyndns.org/
    - Clipboard
    - $%TelegramDv$
    - /sendDocument?chat_id=
    - &caption=
    - | Snake Tracker Clipboard |
    - | Snake
    - application/x-ms-dos-executable
    - Screenshot
    - \SnakeKeylogger
    - \SnakeKeylogger\
    - - Screenshot Logs ID -
    - Screenshot |
    - | Snake Tracker Screenshot |
    - - keystroke Logs ID -
    - Keystrokes
    - Keylogger |
    - | Snake Tracker Keylogger |
    - SnakeKeylogger
    - - Passwords ID -
    - Passwords
    - User

- | Snake Tracker PW |
- SnakePW
- oken
- ProtectTrue
- E-Mail:
- PSWD:
- IMAP Password
- POP3 Password
- HTTP Password
- SMTP Password
- Software\Microsoft\Office\15.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676
- Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676
- Software\Microsoft\Windows Messaging Subsystem\Profiles\9375CFF0413111d3B88A00104B2A6676
- Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676
- \Accounts\Account.rec0
- Account
- POP3Account
- Password
- POP3Password
- !
- E-Mail: {0}
- PSWD: {0}
- \Local State
- "encrypted_key":"(.*?)"
- \Kinza\User Data\Default\Login Data
- logins
- origin_url
- username_value
- password_value
- abcdefghijklmnopqrstuvwxyz1234567890_-.~!@#$%^&*()[{]}\|';:,<>/?+=
- All User Profile * : (?.*)
- after
- {0}{1}{2}{3}{4}
- -------- Snake Tracker --------
- Found From: Connected Wifi
- WiFi Name:
- Password:
- wlan show profile name="
- " key=clear
- netsh
- wlan show profile
- Key Content * : (?.*)
- Open Network
- \discord\Local Storage\leveldb\
- .log
- N/A
- -------- Snake Tracker -------- Found From: Discord Token:
- ---------------------------------
- (
- UNIQUE

- table
- Mozilla\Firefox\Profiles
- logins.json
- -------- Snake Tracker -------- Found From: Firefox Host:
- Waterfox\Profiles
- -------- Snake Tracker -------- Found From: WaterFox Host:
- Thunderbird\Profiles\
- -------- Snake Tracker -------- Found From: Thunderbird Host:
- Mozilla\SeaMonkey\Profiles
- -------- Snake Tracker -------- Found From: SeaMonkey Host:
- Comodo\IceDragon\Profiles
- -------- Snake Tracker -------- Found From: Ice Dragon Host:
- 8pecxstudios\Cyberfox\Profiles
- -------- Snake Tracker -------- Found From: CyberFox Host:
- FlashPeak\SlimBrowser\Profiles
- -------- Snake Tracker -------- Found From: Slim Host:
- PostboxApp\Profiles
- -------- Snake Tracker -------- Found From: PostBox Host:
- Moonchild Productions\Pale Moon\Profiles
- -------- Snake Tracker -------- Found From: PaleMoon Host:
- NSS_Shutdown
- PROGRAMFILES
- \Mozilla Thunderbird\
- \Mozilla Firefox\
- \SeaMonkey\
- \Comodo\IceDragon\
- \Cyberfox\
- \Pale Moon\
- \Waterfox Current\
- \SlimBrowser\
- \Postbox\
- \mozglue.dll
- \nss3.dll
- NSS_Init
- PK11SDR_Decrypt
- WrapNonExceptionThrows
- YFGGCVyufgtwfyuTGFWTVFAUYVF
- 256d2426-b4cc-4996-9a99-c8e915357eef
- 1.0.0.0
- MyTemplate
- 11.0.0.0
- My.Computer
- My.Application
- My.User
- My.Forms
- My.WebServices
- System.Windows.Forms.Form
- Create__Instance__
- Dispose__Instance__

    - My.MyProject.Forms
    - System.Resources.Tools.StronglyTypedResourceBuilder
    - 17.0.0.0
    - _CorExeMain
    - mscoree.dll
    - AHA-SOFT-LARGE-BUSINESS-CASH-REGISTER.512
    - VS_VERSION_INFO
    - StringFileInfo
    - 000004B0
    - Comments
    - Add Money to your Bank Account
    - CompanyName
    - United Kingdoom of America
    - FileDescription
    - 2025 February Malware Analysis
    - FileVersion
    - 1.0.0.0
    - InternalName
    - Week4.exe
    - LegalCopyright
    - Copyright © 2025 Michael Galde
    - LegalTrademarks
    - Bank of America
    - OriginalFilename
    - Week4.exe
    - ProductName
    - Bank Analysis Software
    - ProductVersion
    - 1.0.0.0
    - Assembly Version
    - 1.0.0.0
- Analysis of string findings:
    - [Potential functionality indicated]
    - [Suspicious patterns]
    - I found multiple “snake keylogger” and “keylogger” strings
    - I found many password strings
    - I found many suspicious looking username type strings such as CyberFox, PaleMoon, and SeaMonkey

Entropy Analysis

- Overall entropy score: [Score]
    - 6.81090
- Section-specific entropy:
    - [List sections with unusual entropy]
    - 2.68576 - PE Header
    - 5.85631 - .text section
    - 7.94006 - .rsrc section (packed)
    - 0.10473 - .reloc section

- Packing analysis:
    - [Packed/Unpacked determination]:The executable is packed, the unpac.me analysis determined this file to be a SnakeKeylogger. I
    - [Packer identified (if applicable)]: (Heur) packer
    - [Unpacking methodology (if attempted)]: resource hacker
    - [Alternative unpacking approaches (if needed)]: I tried using unpac.me as well as de4dot. unpac.me did not work. I ended up using a combination of de4dot and resource hacker and that worked, and I was able to figure out how to alter the code embedded in the .rsrc section of the executable. de4dot was a necessary first step before using resource hacker, only a combination of both methods worked, neither method worked alone.

4. Disassembly Analysis

- For this section, you will use BinaryNinja (https://cloud.binary.ninja/):
    - navigate to sub_4043ab
    - Identify what type of registers are SS, CS, FS
        - SS (Stack Segment Register): Holds the segment selector for the stack segment, which is used for stack-related operations like push, pop, call, and ret.
        - CS (Code Segment Register): Holds the segment selector for the currently executing code. Instructions are fetched from this segment.
        - FS (Segment Register): An extra segment register often used in Windows for accessing thread-local storage (TLS) or special structures like the PEB (Process Environment Block) and TEB (Thread Environment Block).
    - Identify why those are used.
        - SS is used to manage stack operations, ensuring correct execution of function calls and local variables.
        - CS is necessary for fetching and executing code from memory.
        - FS is particularly important in Windows malware analysis because it provides access to key system structures, such as:
    - What in this function can be used to identify malicious intent?
        - The function uses regparm to pass arguments in registers instead of the stack. This can be an anti-debugging or obfuscation technique to evade analysis.
    - How does adc eax, 0x342825 modify the register state? Why might ADC be used instead of ADD?
        - The use of ADC instead of ADD suggests a reliance on the carry flag (CF), which could indicate obfuscation
    - What effect does jo 0x4043cf have on execution flow? What conditions must be met for this jump to be taken?

        - JO (Jump if Overflow) will jump to 0x4043cf only if the Overflow Flag (OF) is set.
        - Anti-analysis: Malware might create alternate execution paths to evade signature-based detection.
    - Why might this function store arguments in unusual registers like xmm1 and st0 instead of on the stack?
        - Many static analysis tools and debuggers track stack-based parameters easily. Using xmm1 and st0 avoids this detection.
        - Malware can store obfuscated data in floating-point registers, making it harder to analyze manually.

4. Static Analysis Summary

- Key findings from static analysis:
    - [Major indicators of malicious behavior]
        - From the strings, the VirusTotal scan, as well as unpac.me analysis, as well as the resource hacker results, it seems this is a SnakeKeylogger that stores passwords and takes screenshots. It is spyware. It also seems that it is a Trojan, the program is disguised as a banking app
    - [Potential functionality]
        - Resource Hacker revealed xml code in the .rsrc section which indicated that the transferring money functionality could possibly be restored if some privileges are escalated manually by changing the code.
    - [Risk indicators]
        - It is a risk of collecting keystrokes, passwords, screenshots

Dynamic Analysis

1. Analysis Environment

Environment Setup

- Virtual Machine specifications:
    - [OS version] - Windows 7
    - [Memory allocation] - 8 GB
    - [Network configuration] - Disabled network adapter, I used AnyRun to see network activity
- Monitoring tools deployed:
    - [Process monitoring]
        - Ensure you use RegShot, Process Monitor, Process Explorer
        - This time I used RegShot, ProcessHacker
    - [Network monitoring]

        - Ensure you use Wireshark
        - I used Wireshark
    - [File system monitoring]
- Safety measures implemented:
    - [Network isolation]
        - Try the analysis with and without Fakenet
        - I disabled network adapter
        - 0 packets were captured by WireShark, I used AnyRun to monitor network activity
    - [Snapshot configuration]
        - I have a snapshot saved of a fresh OS with tools installed
    - [Additional protections]

2. Runtime Observations

Initial Execution

- [Immediate system changes]
    - None, program did not appear to do anything obvious. No windows were opened.
- [Process creation]
    - Smss.exe and Memory Compression are very active
    - The registry is actively logging changes
    - AnyRun: Week4.exe, svchost.exe,
- [Registry creation]
    - Keys: 399559
    - Values: 700672
    - Comparison: Keys deleted: 1, Keys added: 23
    - Values deleted:1, Values added: 58
    - Values modified: 36
    - Total changes 119
- [Network activity]:
    - Internet disabled, WIreshark did not register any network changes
    - AnyRun:http://checkip.dyndns.com
- [File system changes]
    - None that I could detect

Continued Monitoring

- [Persistent changes]
    - AnyRun: steals credentials from web browsers
- [Scheduled tasks]
    - AnyRun: snakekeylogger detected
- [Registry modifications]
    - Comparison: Keys deleted: 1, Keys added: 23
    - Values deleted:1, Values added: 58

    - Values modified: 36
    - Total changes 119
- [Additional payloads]
    - In the .rsrc section after unpacking I found some xml code that suggested a payload

3. Post-Execution Analysis

- System state changes:
    - [Permanent modifications] installed a keylogger, which also takes screenshot
    - [Persistence mechanisms] steals credentials from web browsers
    - [Data exfiltration evidence] presence of Telegram API suggests exfiltration
- Network activity summary:
    - [Connection attempts] ::http://checkip.dyndns.com
    - [Data transfers]
    - [Command & Control activity]
        - http://checkip.dyndns.com is from Brazil
        - Presence of telegram api suggests exfiltration to command and control center

Impact Analysis

1. User Impact Assessment

Home Users

- [Potential impact] could compromise accounts with stolen credentials, loss of privacy, screenshots and keylogging would cause loss of confidentiality
- [Risk level] Very high risk
- [Data compromise potential] data could be compromised with stolen credentials

Business Users

- [Operational impact] Could cause sabotage on user accounts from stolen credentials, leading to unauthorized access
- [Data security concerns]: Loss of confidentiality through screenshots and keylogging
- [Financial implications]: Lawsuits due to loss of confidentiality revealing personal information, and sabotage of operations using stolen credentials

Government Users

- [Security implications]: Loss of confidentiality, could cause loss of access if stolen credentials are used to gain unauthorized access to systems and then passwords are changed
- [Data sensitivity concerns]: Loss of confidentiality, this could expose top secret activities with screenshots and keylogger
- [Operational disruption potential]: Loss of access to personal accounts and could -allow attackers to gain top secret intelligence that could cause damage

2. Mitigation Strategy

Immediate Response

- [Initial containment steps] isolate computer from network
- [System isolation procedures] disable network adapter
- [Data preservation methods] perform regular backups, use a backup and reinstall operating system

Long-term Prevention

- [Security control recommendations]: Do not install suspicious programs
- [Policy modifications]: Keep backups
- [Training requirements]: Do not install suspicious software, if software claims to wire you money or sounds too good to be true, it could be malicious

Conclusion

1. Analysis Reflection

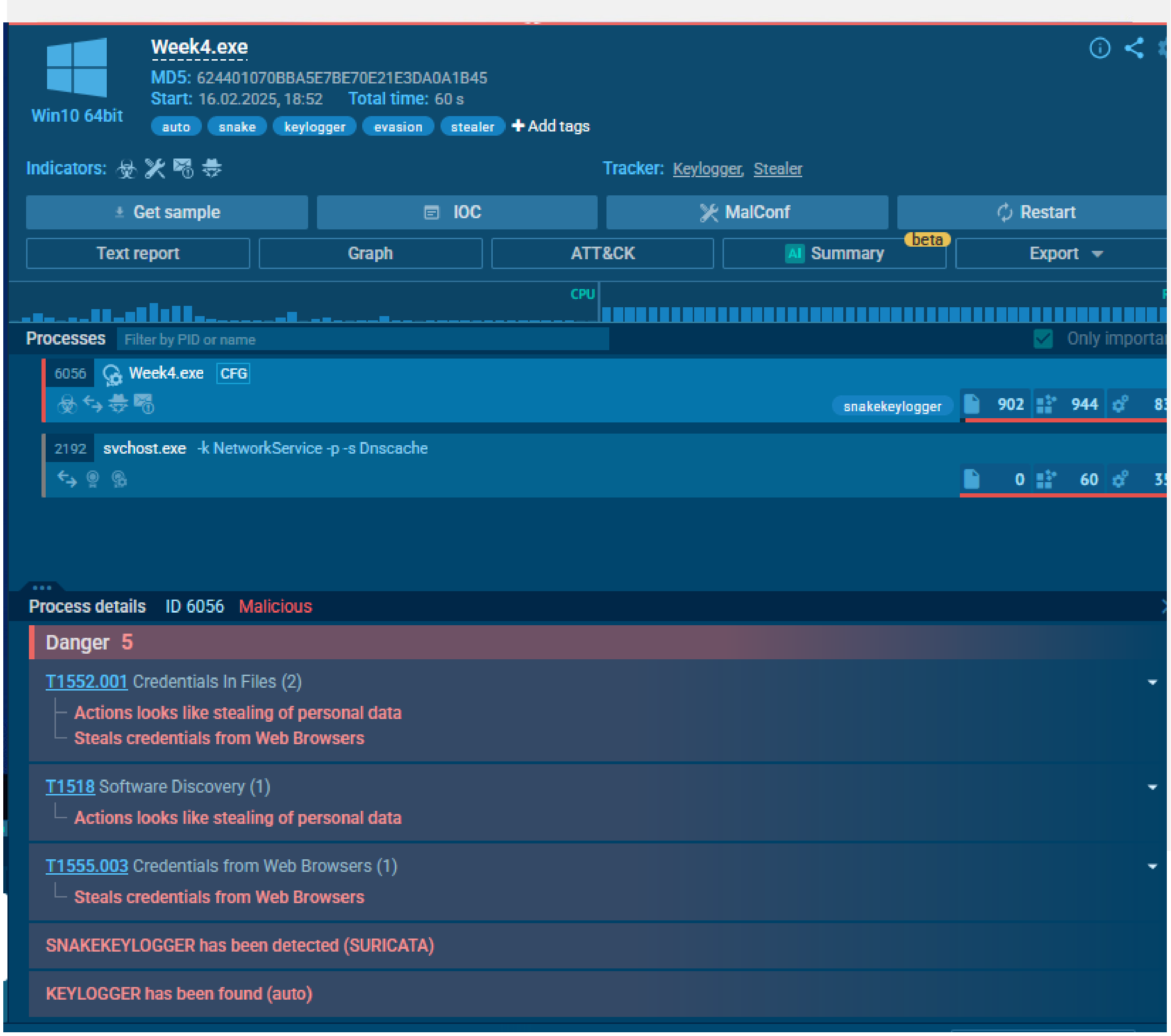
- [Summary of findings]
  - The static and dynamic analysis suggests that the executable Week4.exe is a malicious file, most likely a variant of Snake Keylogger. The high detection rate on VirusTotal (59/72 vendors) supports this conclusion. Multiple AV vendors classify it as a Trojan, Spyware, and Keylogger, which aligns with the string analysis. The presence of password stealing, and keylogging capabilities indicates that this malware is designed for credential theft.
- [Unusual characteristics]
  - The file’s PE32 format with a VB.NET compiler suggests it was developed using .NET technologies, and the .rsrc section was packed. The unusual company name (United Kingdoom of America) and misleading file description further suggest an attempt to obfuscate its true purpose. The entropy score of 7.94 in the resource section indicates that it was packed.
  - The embedded strings and entropy analysis confirm the malware’s primary function: stealing sensitive user data such as stored passwords, Discord tokens, Outlook credentials, and Wi-Fi

passwords. The presence of Telegram API references suggests that exfiltrated data is sent to a remote attacker.

- [Learning outcomes]
  - How to unpack a resource section using resource hacker, along with de4dot
- [Additional research needed]
  - More investigation with the telegram api would be useful to track down information about a possible command and control center

2. Evidence Documentation

- [Screenshot descriptions and relevance]
  - I included screenshots of the steps I took to analyze this executable
- [Tool output documentation] I used many tools, including a Windows 7 vm, virustotal, detect-it-easy, resourcehacker, unpac.me, de4dot, ghidra, binaryninja, any.run, regshot, wireshark, and process hacker.
- [Additional supporting materials]- I included the unpacked code from the resource section in my analysis in the memory map section, which includes code that could be modified to fix the banking software (elevate privileges) which could change the program’s functionality



Week4.exe

MD5: 624401070BBA5E7BE70E21E3DA0A1B45

Start: 16.02.2025, 18:52 Total time: 60 s

auto snake keylogger evasion stealer + Add tags

Indicators:

Tracker: Keylogger, Stealer

Processes Filter by PID or name

6056 Week4.exe CFG snakekeylogger 902 944 83

2192 svchost.exe -k NetworkService -p -s Dnscache 0 60 35

Process details ID 6056 Malicious

Danger 5

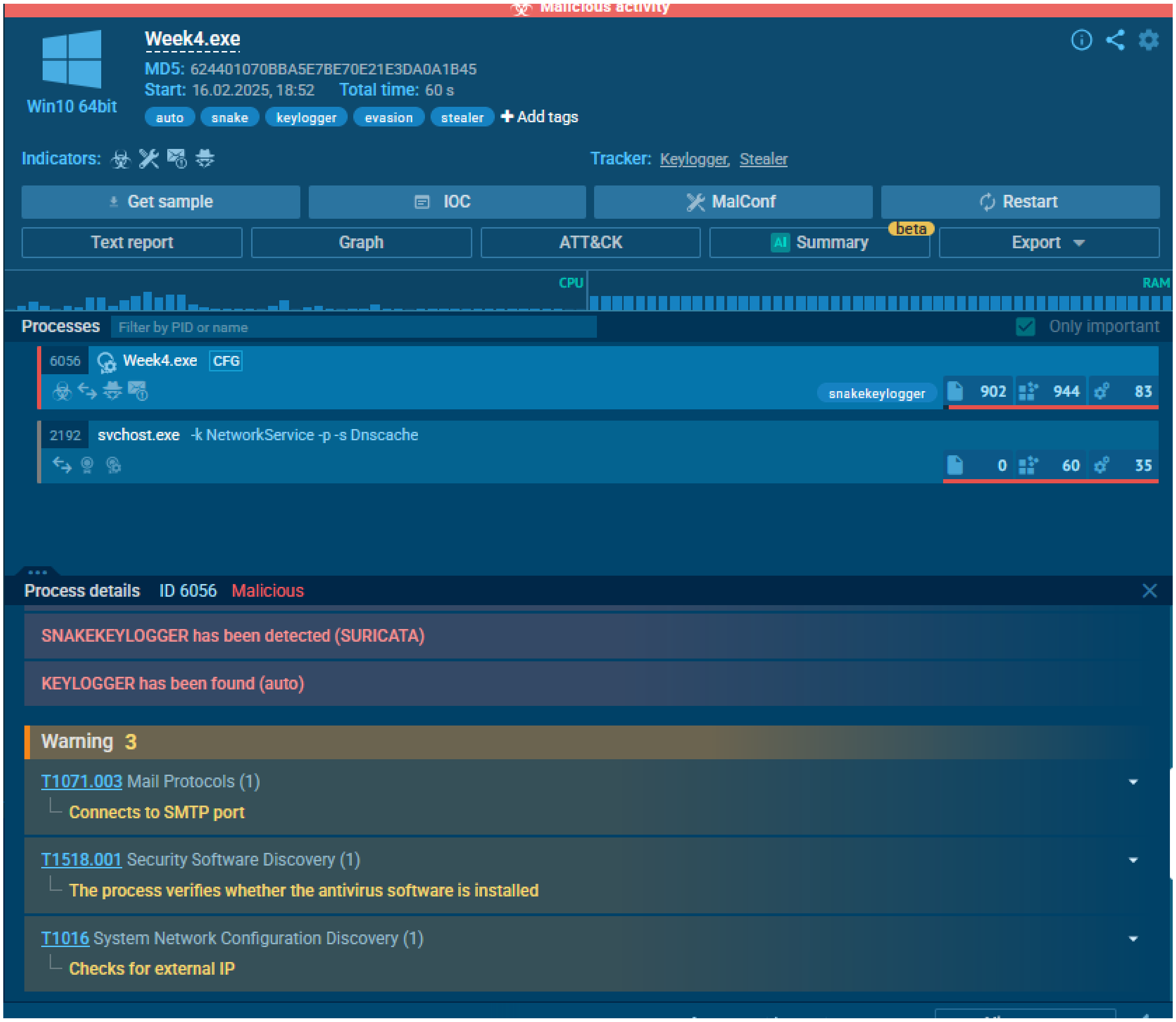
T1552.001 Credentials In Files (2)
- Actions looks like stealing of personal data
- Steals credentials from Web Browsers

T1518 Software Discovery (1)
- Actions looks like stealing of personal data

T1555.003 Credentials from Web Browsers (1)
- Steals credentials from Web Browsers

SNAKEKEYLOGGER has been detected (SURICATA)

KEYLOGGER has been found (auto)



Week4.exe

MD5: 624401070BBA5E7BE70E21E3DA0A1B45

Start: 16.02.2025, 18:52 Total time: 60 s

auto snake keylogger evasion stealer + Add tags

Tracker: Keylogger, Stealer

6056 Week4.exe CFG snakekeylogger 902 944 83

2192 svchost.exe -k NetworkService -p -s Dnscache 0 60 35

Process details ID 6056 Malicious

SNAKEKEYLOGGER has been detected (SURICATA)

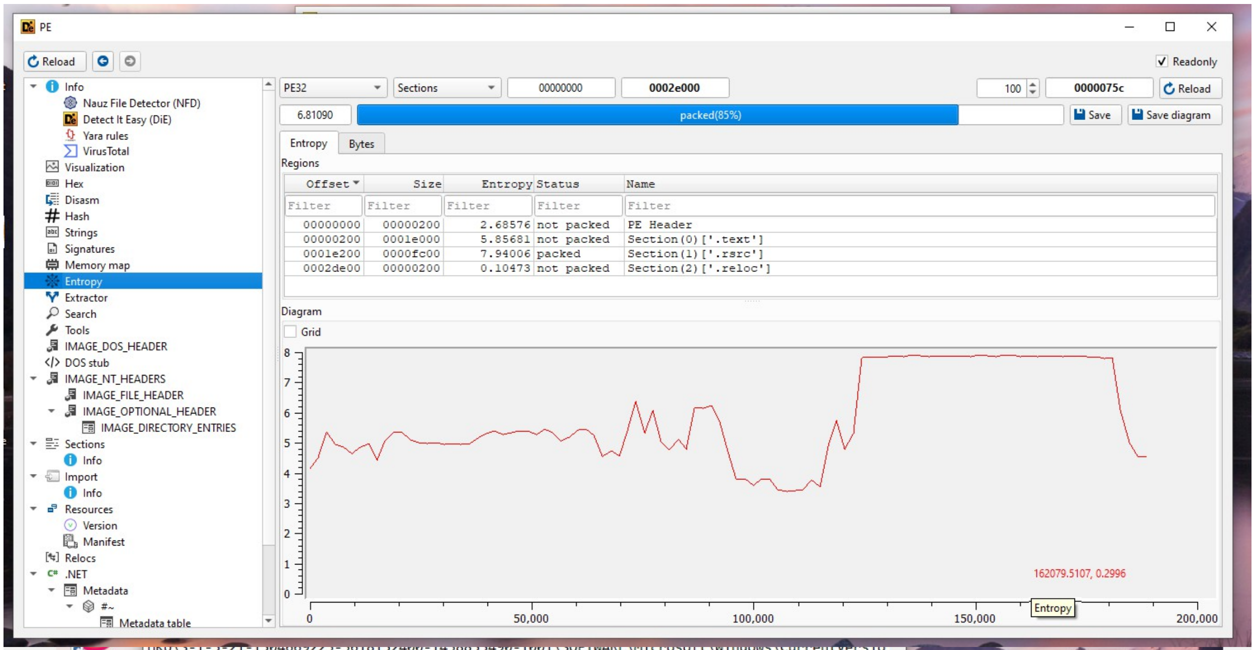
KEYLOGGER has been found (auto)

Warning 3

T1071.003 Mail Protocols (1)
└ Connects to SMTP port

T1518.001 Security Software Discovery (1)
└ The process verifies whether the antivirus software is installed

T1016 System Network Configuration Discovery (1)
└ Checks for external IP

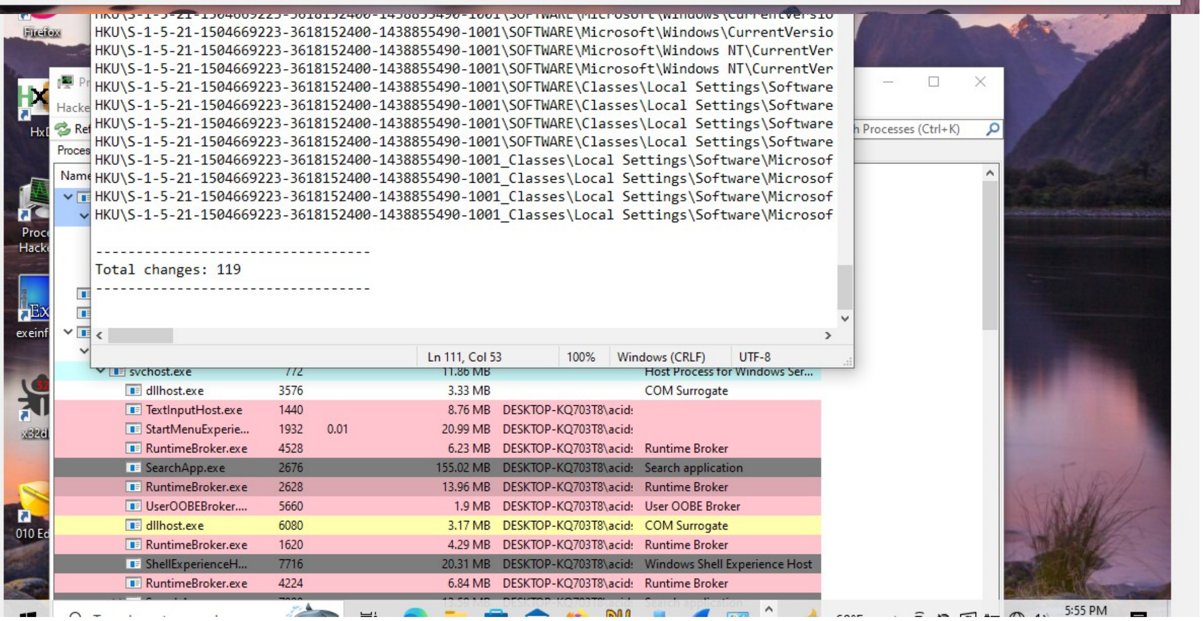


PE32 | Sections | 00000000 | 0002e000 | 100 | 0000075c

6.81090 | packed(85%)

Regions

Offset	Size	Entropy	Status	Name
00000000	00000200	2.68576	not packed	PE Header
00000200	0001e000	5.85681	not packed	Section(0)['.text']
0001e200	0000fc00	7.94006	packed	Section(1)['.rsrc']
0002de00	00000200	0.10473	not packed	Section(2)['.reloc']



```
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001\SOFTWARE\Microsoft\Windows\CurrentVersio
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001\SOFTWARE\Microsoft\Windows\CurrentVersio
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001\SOFTWARE\Microsoft\Windows NT\CurrentVer
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001\SOFTWARE\Microsoft\Windows NT\CurrentVer
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001\SOFTWARE\Classes\Local Settings\Software
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001\SOFTWARE\Classes\Local Settings\Software
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001\SOFTWARE\Classes\Local Settings\Software
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001\SOFTWARE\Classes\Local Settings\Software
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001_Classes\Local Settings\Software\Microsof
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001_Classes\Local Settings\Software\Microsof
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001_Classes\Local Settings\Software\Microsof
HKU\S-1-5-21-1504669223-3618152400-1438855490-1001_Classes\Local Settings\Software\Microsof

----------------------------------
Total changes: 119
----------------------------------
```

Ln 111, Col 53 | 100% | Windows (CRLF) | UTF-8

svchost.exe	772		11.86 MB		Host Process for Windows Ser...
dllhost.exe	3576		3.33 MB		COM Surrogate
TextInputHost.exe	1440		8.76 MB	DESKTOP-KQ703T8\acid:	
StartMenuExperie...	1932	0.01	20.99 MB	DESKTOP-KQ703T8\acid:	
RuntimeBroker.exe	4528		6.23 MB	DESKTOP-KQ703T8\acid:	Runtime Broker
SearchApp.exe	2676		155.02 MB	DESKTOP-KQ703T8\acid:	Search application
RuntimeBroker.exe	2628		13.96 MB	DESKTOP-KQ703T8\acid:	Runtime Broker
UserOOBEBroker....	5660		1.9 MB	DESKTOP-KQ703T8\acid:	User OOBE Broker
dllhost.exe	6080		3.17 MB	DESKTOP-KQ703T8\acid:	COM Surrogate
RuntimeBroker.exe	1620		4.29 MB	DESKTOP-KQ703T8\acid:	Runtime Broker
ShellExperienceH...	7716		20.31 MB	DESKTOP-KQ703T8\acid:	Windows Shell Experience Host
RuntimeBroker.exe	4224		6.84 MB	DESKTOP-KQ703T8\acid:	Runtime Broker