4.2: Malicious PDF from Suspicious USB

Objective

The primary goal of this digital forensics task was to investigate a suspicious USB device suspected of containing a malicious file. The USB contained a README.pdf file and an autorun.inf configuration that could potentially trigger malicious actions automatically upon being inserted into a Windows machine. The investigation aimed to:

- Examine the nature of the files present on the USB.
- Determine whether the PDF is malicious or benign.
- Identify any embedded executables or scripts.
- Analyse the behaviour of the file in static and dynamic environments.
- Understand how such an attack could exploit Windows autorun functionality.
- Provide evidence-backed conclusions on the threat and its intended behavior.

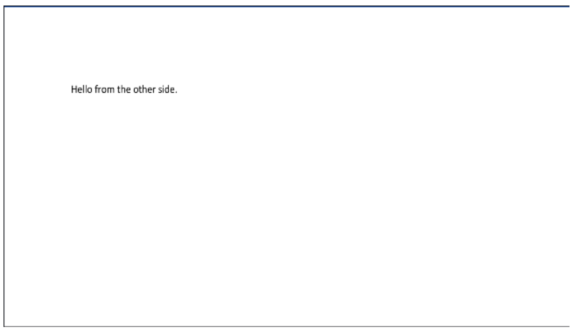
Initial Evidence and Setup

The evidence was provided in the form of a zip archive named USB.zip. Using the known password btlo, the contents were extracted as follows:

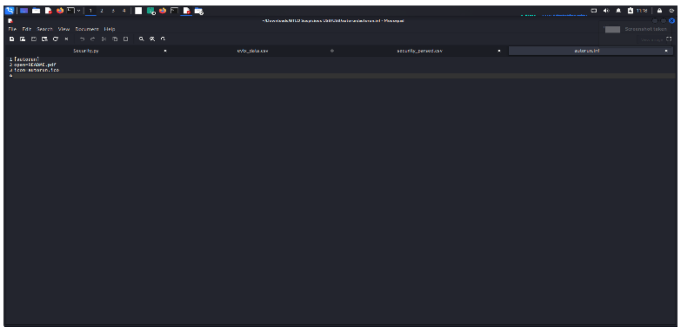
```
unzip USB.zip
```

Extracted Files:

- README.pdf — a seemingly benign PDF file.
- autorun.inf — a configuration file used in older versions of Windows to execute files automatically.



Hello from the other side.



```
[autorun]
open=README.pdf
icon=autorun.ico
```

This was the starting point for a multi-step forensic investigation that included both static and dynamic analysis.

Step-by-Step Analysis

1. Initial Inspection of autorun.inf

```
cat autorun.inf
```

- The file contained directives like:
- [autorun]
- open=README.pdf

- **Interpretation:**
  On systems where autorun is enabled (e.g., older Windows OS), this file instructs the OS to open README.pdf automatically upon USB insertion.
- **Security Implication:**
  The attacker relies on the autorun mechanism to trigger the malicious PDF without requiring user interaction.

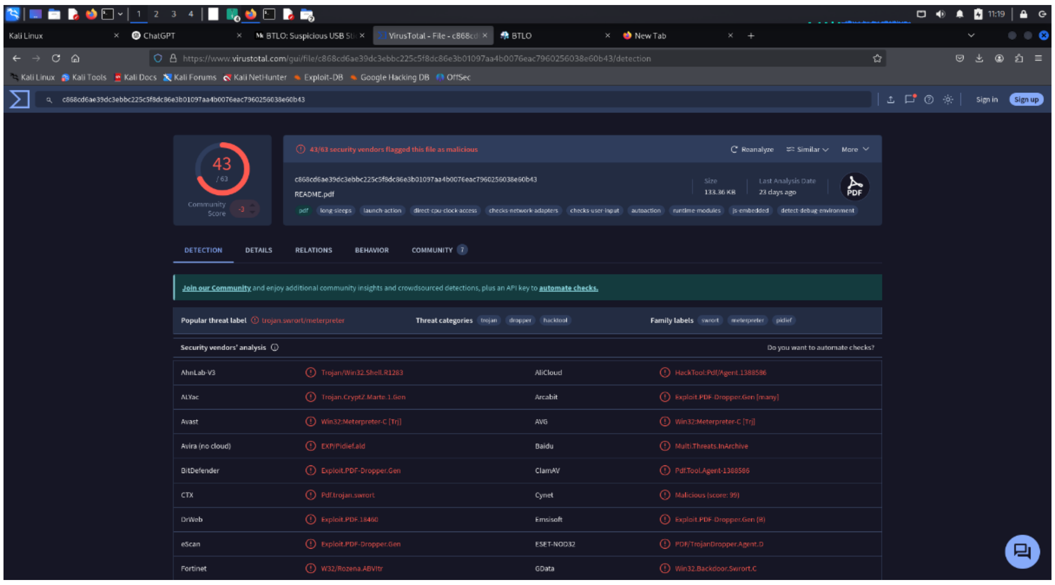
2. Check File Type and Magic Number (PDF Validation)

xxd README.pdf | head

- Output (hex view) shows:
- 00000000: 2550 4446 2d31 2e34 0a25 c4e5 f2e5 eba7 %PDF-1.4.%......
- %PDF = 25 50 44 46 → confirms it is a legitimate PDF file based on its magic number.
- **Purpose of step**:
  Attackers sometimes disguise executables as PDFs, so verifying the file signature helps eliminate this possibility.

3. Upload to Virus Total (Static Analysis)

- File uploaded to VirusTotal.
- Several antivirus engines flagged the file as **malicious**.
- Common detections: "PDF.Dropper", "Exploit.PDF.CVE-Embedded", etc.
- **Inference:**
  The file matches known malware signatures and poses a potential risk.



https://www.virustotal.com/gui/file/c868cd6ae39dc3ebbc225c5f8dc86e3b01097aa4b0076eac7960256038e60b43/detection

43 / 63 Community Score -3

43/63 security vendors flagged this file as malicious

c868cd6ae39dc3ebbc225c5f8dc86e3b01097aa4b0076eac7960256038e60b43
README.pdf
Size 133.36 KB
Last Analysis Date 23 days ago
pdf long-sleeps launch-action direct-cpu-clock-access checks-network-adapters checks-user-input autoaction runtime-modules js-embedded detect-debug-environment

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

Popular threat label trojan.swrort/meterpreter
Threat categories trojan dropper hacktool
Family labels swrort meterpreter pidief

Security vendors' analysis — Do you want to automate checks?

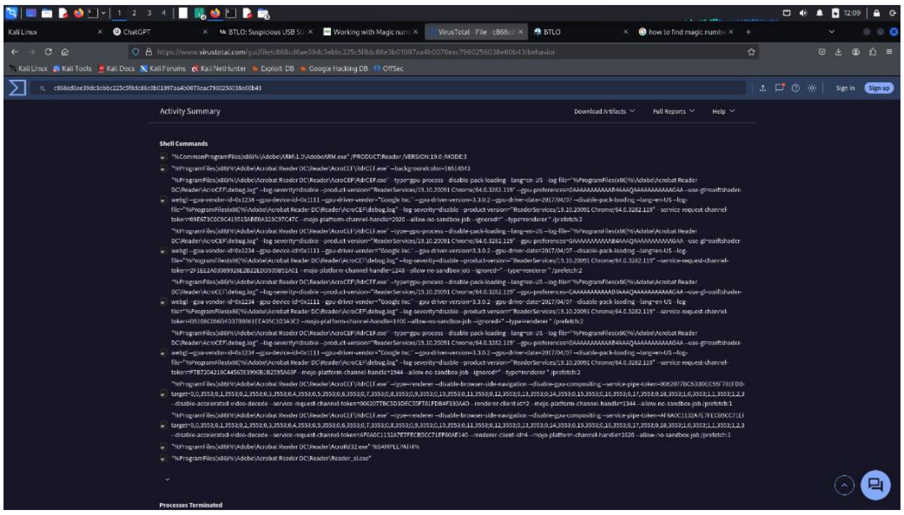
AhnLab-V3	Trojan/Win32.Shell.R1283	AliCloud	HackTool:Pdf/Agent.1388586
ALYac	Trojan.CryptZ.Marte.1.Gen	Arcabit	Exploit.PDF-Dropper.Gen [many]
Avast	Win32:Meterpreter-C [Trj]	AVG	Win32:Meterpreter-C [Trj]
Avira (no cloud)	EXP/Pidief.ald	Baidu	Multi.Threats.InArchive
BitDefender	Exploit.PDF-Dropper.Gen	ClamAV	Pdf.Tool.Agent-1388586
CTX	Pdf.trojan.swrort	Cynet	Malicious (score: 99)
DrWeb	Exploit.PDF.18460	Emsisoft	Exploit.PDF-Dropper.Gen (B)
eScan	Exploit.PDF-Dropper.Gen	ESET-NOD32	PDF/TrojanDropper.Agent.D
Fortinet	W32/Rozena.ABVItr	GData	Win32.Backdoor.Swrort.C

4. Behavioral Analysis via Hybrid Analysis (Dynamic Sandbox)

- File uploaded to Hybrid Analysis
- Execution environment: **Windows 10 64-bit**

Key Observations:

- cmd.exe process was spawned.
- Network behaviour: no outbound connections, indicating a payload or shell may be locally executed.
- Memory dump showed command-line execution traces.
- **Conclusion:**
  The file attempts to execute system-level commands via cmd.exe once opened in a Windows environment.



Activity Summary

Shell Commands

```
"%CommonProgramFiles(x86)%\Adobe\ARM\1.0\AdobeARM.exe" /PRODUCT:Reader /VERSION:19.0 /MODE:3
"%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\RdrCEF.exe" --backgroundcolor=16514043
"%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\RdrCEF.exe" --type=gpu-process --disable-pack-loading --lang=en-US --log-file="%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\debug.log" --log-severity=disable --product-version="ReaderServices/19.10.20091 Chrome/64.0.3282.119" --gpu-preferences=GAAAAAAAAAAAB4AAAQAAAAAAAAAAGAA --use-gl=swiftshader-webgl --gpu-vendor-id=0x1234 --gpu-device-id=0x1111 --gpu-driver-vendor="Google Inc." --gpu-driver-version=3.3.0.2 --gpu-driver-date=2017/04/07 --disable-pack-loading --lang=en-US --log-file="%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\debug.log" --log-severity=disable --product-version="ReaderServices/19.10.20091 Chrome/64.0.3282.119" --service-request-channel-token=09E673C0C9C419515ABE0A323C97C47C --mojo-platform-channel-handle=2020 --allow-no-sandbox-job --ignored=" --type=renderer " /prefetch:2
"%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\RdrCEF.exe" --type=gpu-process --disable-pack-loading --lang=en-US --log-file="%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\debug.log" --log-severity=disable --product-version="ReaderServices/19.10.20091 Chrome/64.0.3282.119" --gpu-preferences=GAAAAAAAAAAAB4AAAQAAAAAAAAAAGAA --use-gl=swiftshader-webgl --gpu-vendor-id=0x1234 --gpu-device-id=0x1111 --gpu-driver-vendor="Google Inc." --gpu-driver-version=3.3.0.2 --gpu-driver-date=2017/04/07 --disable-pack-loading --lang=en-US --log-file="%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\debug.log" --log-severity=disable --product-version="ReaderServices/19.10.20091 Chrome/64.0.3282.119" --service-request-channel-token=2F1EE2A03089928E2B22ED0509B51A01 --mojo-platform-channel-handle=1248 --allow-no-sandbox-job --ignored=" --type=renderer " /prefetch:2
"%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\RdrCEF.exe" --type=gpu-process --disable-pack-loading --lang=en-US --log-file="%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\debug.log" --log-severity=disable --product-version="ReaderServices/19.10.20091 Chrome/64.0.3282.119" --gpu-preferences=GAAAAAAAAAAAB4AAAQAAAAAAAAAAGAA --use-gl=swiftshader-webgl --gpu-vendor-id=0x1234 --gpu-device-id=0x1111 --gpu-driver-vendor="Google Inc." --gpu-driver-version=3.3.0.2 --gpu-driver-date=2017/04/07 --disable-pack-loading --lang=en-US --log-file="%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\debug.log" --log-severity=disable --product-version="ReaderServices/19.10.20091 Chrome/64.0.3282.119" --service-request-channel-token=B5106C056B4037B8861EEA05C103A3E2 --mojo-platform-channel-handle=1400 --allow-no-sandbox-job --ignored=" --type=renderer " /prefetch:2
"%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\RdrCEF.exe" --type=gpu-process --disable-pack-loading --lang=en-US --log-file="%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\debug.log" --log-severity=disable --product-version="ReaderServices/19.10.20091 Chrome/64.0.3282.119" --gpu-preferences=GAAAAAAAAAAAB4AAAQAAAAAAAAAAGAA --use-gl=swiftshader-webgl --gpu-vendor-id=0x1234 --gpu-device-id=0x1111 --gpu-driver-vendor="Google Inc." --gpu-driver-version=3.3.0.2 --gpu-driver-date=2017/04/07 --disable-pack-loading --lang=en-US --log-file="%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\debug.log" --log-severity=disable --product-version="ReaderServices/19.10.20091 Chrome/64.0.3282.119" --service-request-channel-token=F7B7204218C4456593996B1B2595A63F --mojo-platform-channel-handle=1944 --allow-no-sandbox-job --ignored=" --type=renderer " /prefetch:2
"%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\RdrCEF.exe" --type=renderer --disable-browser-side-navigation --disable-gpu-compositing --service-pipe-token=0062077BC5D3DEC55F781FDB
target=0,0,3553;0,1,3553;0,2,3553;0,3,3553;0,4,3553;0,5,3553;0,6,3553;0,7,3553;0,8,3553;0,9,3553;0,10,3553;0,11,3553;0,12,3553;0,13,3553;0,14,3553;0,15,3553;0,16,3553;0,17,3553;0,18,3553;1,0,3553;1,1,3553;1,2,3
--disable-accelerated-video-decode --service-request-channel-token=0062077BC5D3DEC55F781FDB4F330A4D --renderer-client-id=2 --mojo-platform-channel-handle=1344 --allow-no-sandbox-job /prefetch:1
"%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroCEF\RdrCEF.exe" --type=renderer --disable-browser-side-navigation --disable-gpu-compositing --service-pipe-token=AF6A0C1132A7E7FECB5CC71EF
target=0,0,3553;0,1,3553;0,2,3553;0,3,3553;0,4,3553;0,5,3553;0,6,3553;0,7,3553;0,8,3553;0,9,3553;0,10,3553;0,11,3553;0,12,3553;0,13,3553;0,14,3553;0,15,3553;0,16,3553;0,17,3553;0,18,3553;1,0,3553;1,1,3553;1,2,3
--disable-accelerated-video-decode --service-request-channel-token=AF6A0C1132A7E7FECB5CC71EF60AE140 --renderer-client-id=4 --mojo-platform-channel-handle=1620 --allow-no-sandbox-job /prefetch:1
"%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\AcroRd32.exe" %SAMPLEPATH%
"%ProgramFiles(x86)%\Adobe\Acrobat Reader DC\Reader\Reader_sl.exe"
```

Processes Terminated

5. Deep Dive Using pdf-parser.py (Static PDF Object Analysis)

Check for automatic action triggers:

python3 pdf-parser.py README.pdf -s /OpenAction

python3 pdf-parser.py README.pdf -s /Launch

Output:

- Found in **Object 28**:
- /OpenAction << /S /Launch /F (cmd.exe) >>
- **Explanation:**
  - /OpenAction tells the PDF reader to take action when the document is opened.
  - /Launch with /F (cmd.exe) instructs to launch the command prompt.
- **Conclusion:**
  The PDF is not just a passive document — it contains an embedded instruction to launch cmd.exe, making it a **PDF dropper**.

6. Extract and Search for Executable Strings

strings README.pdf | grep .exe

- Output:
- cmd.exe

xxd README.pdf | grep 'cmd.exe'

- Confirms the presence of cmd.exe in raw hex.
- **Why This Matters:**
  Even if /Launch wasn't detected, raw binary or ASCII references to executables are red flags in forensic PDF inspection.

7. Correlation of Autorun + Malicious PDF

Putting it all together:

- autorun.inf triggers README.pdf.
- README.pdf triggers cmd.exe on file open.
- This creates a stealthy **two-stage execution** attack: USB → PDF → cmd.exe

Command Summary Used in This Investigation

# View autorun configuration

cat autorun.inf

# Confirm PDF magic number

xxd README.pdf | head

# Find any executable references

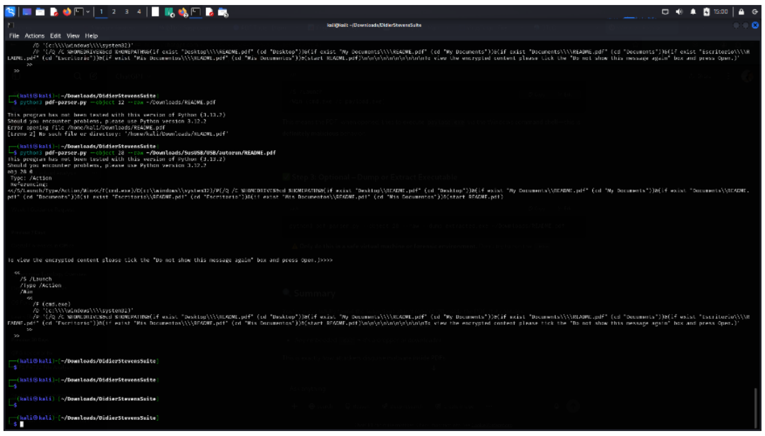
strings README.pdf | grep .exe

xxd README.pdf | grep 'cmd.exe'

# Static PDF analysis for suspicious behavior

python3 pdf-parser.py README.pdf -s /OpenAction

python3 pdf-parser.py README.pdf -s /Launch


kali@kali: ~/Downloads/DidierStevensSuite
File Actions Edit View Help

```
        /D '(c:\\\\windows\\\\system32)'
        /P '(/Q /C %HOMEDRIVE%&cd %HOMEPATH%&(if exist "Desktop\\\\README.pdf" (cd "Desktop"))&(if exist "My Documents\\\\README.pdf" (cd "My Documents"))&(if exist "Documents\\\\README.pdf" (cd "Documents"))&(if exist "Escritorio\\\\README.pdf" (cd "Escritorio"))&(if exist "Mis Documentos\\\\README.pdf" (cd "Mis Documentos"))&(start README.pdf)\n\n\n\n\n\n\n\n\nTo view the encrypted content please tick the "Do not show this message again" box and press Open.)'
      >>
  >>

┌──(kali㉿kali)-[~/Downloads/DidierStevensSuite]
└─$ python3 pdf-parser.py --object 12 --raw ~/Downloads/README.pdf
This program has not been tested with this version of Python (3.13.2)
Should you encounter problems, please use Python version 3.12.2
Error opening file /home/kali/Downloads/README.pdf
[Errno 2] No such file or directory: '/home/kali/Downloads/README.pdf'

┌──(kali㉿kali)-[~/Downloads/DidierStevensSuite]
└─$ python3 pdf-parser.py --object 28 --raw ~/Downloads/SusUSB/USB/autorun/README.pdf
This program has not been tested with this version of Python (3.13.2)
Should you encounter problems, please use Python version 3.12.2
obj 28 0
 Type: /Action
 Referencing:
<</S/Launch/Type/Action/Win<</F(cmd.exe)/D(c:\\windows\\system32)/P(/Q /C %HOMEDRIVE%&cd %HOMEPATH%&(if exist "Desktop\\README.pdf" (cd "Desktop"))&(if exist "My Documents\\README.pdf" (cd "My Documents"))&(if exist "Documents\\README.pdf" (cd "Documents"))&(if exist "Escritorio\\README.pdf" (cd "Escritorio"))&(if exist "Mis Documentos\\README.pdf" (cd "Mis Documentos"))&(start README.pdf)

To view the encrypted content please tick the "Do not show this message again" box and press Open.)>>>>

  <<
    /S /Launch
    /Type /Action
    /Win
      <<
        /F (cmd.exe)
        /D '(c:\\\\windows\\\\system32)'
        /P '(/Q /C %HOMEDRIVE%&cd %HOMEPATH%&(if exist "Desktop\\\\README.pdf" (cd "Desktop"))&(if exist "My Documents\\\\README.pdf" (cd "My Documents"))&(if exist "Documents\\\\README.pdf" (cd "Documents"))&(if exist "Escritorio\\\\README.pdf" (cd "Escritorio"))&(if exist "Mis Documentos\\\\README.pdf" (cd "Mis Documentos"))&(start README.pdf)\n\n\n\n\n\n\n\n\nTo view the encrypted content please tick the "Do not show this message again" box and press Open.)'
      >>
  >>

┌──(kali㉿kali)-[~/Downloads/DidierStevensSuite]
└─$

┌──(kali㉿kali)-[~/Downloads/DidierStevensSuite]
└─$

┌──(kali㉿kali)-[~/Downloads/DidierStevensSuite]
└─$

┌──(kali㉿kali)-[~/Downloads/DidierStevensSuite]
└─$
```

8. Conclusion

This investigation revealed that the README.pdf file was a **malicious dropper** embedded inside a USB, designed to automatically execute Windows shell commands (cmd.exe) using embedded PDF actions. The attack is a **USB-delivered malware technique** that exploits legacy features like Windows autorun and PDF reader scripting.

Key Evidence:

Artifact	**Observation**
autorun.inf	Triggers README.pdf automatically
README.pdf	Valid PDF with embedded /OpenAction + /Launch
VirusTotal	Multiple antivirus detections
Hybrid Analysis	Spawned cmd.exe, simulating command execution
pdf-parser.py	Found embedded cmd.exe call in Object 28

Attack Summary

- **Vector:** USB
- **Trigger:** autorun.inf → PDF open
- **Payload:** Embedded call to cmd.exe
- **Goal:** Execute system-level commands without user awareness
- **Classification:** USB-based PDF Dropper Malware

9. Learning Outcomes

Through this hands-on challenge, I gained practical insights and skills in the following areas:

- **USB forensics** and how removable devices can be weaponized.
- **PDF malware analysis**, especially how scripts and embedded actions can be exploited.
- Use of **static** and **dynamic** tools:
  - pdf-parser.py for object-level PDF dissection.
  - VirusTotal and Hybrid Analysis for reputation and behavior scanning.
- Familiarity with **hex analysis**, file signatures, and CLI tools (xxd, strings, cat, grep) in Linux.
- How Windows autorun can be a serious security risk and why it's disabled in modern versions.

10. References

- Security Blue Team. *BTLO Challenge – Suspicious USB Stick*.
  https://blueteamlabs.online
- Phil Harvey. *ExifTool – Read and edit image metadata*.
  https://exiftool.org
- Steghide Project. *Steghide – Steganography tool*.
  https://steghide.sourceforge.net
- Linux Commands Used: exiftool, steghide, strings, file, binwalk
  Referenced from: https://linux.die.net/man/