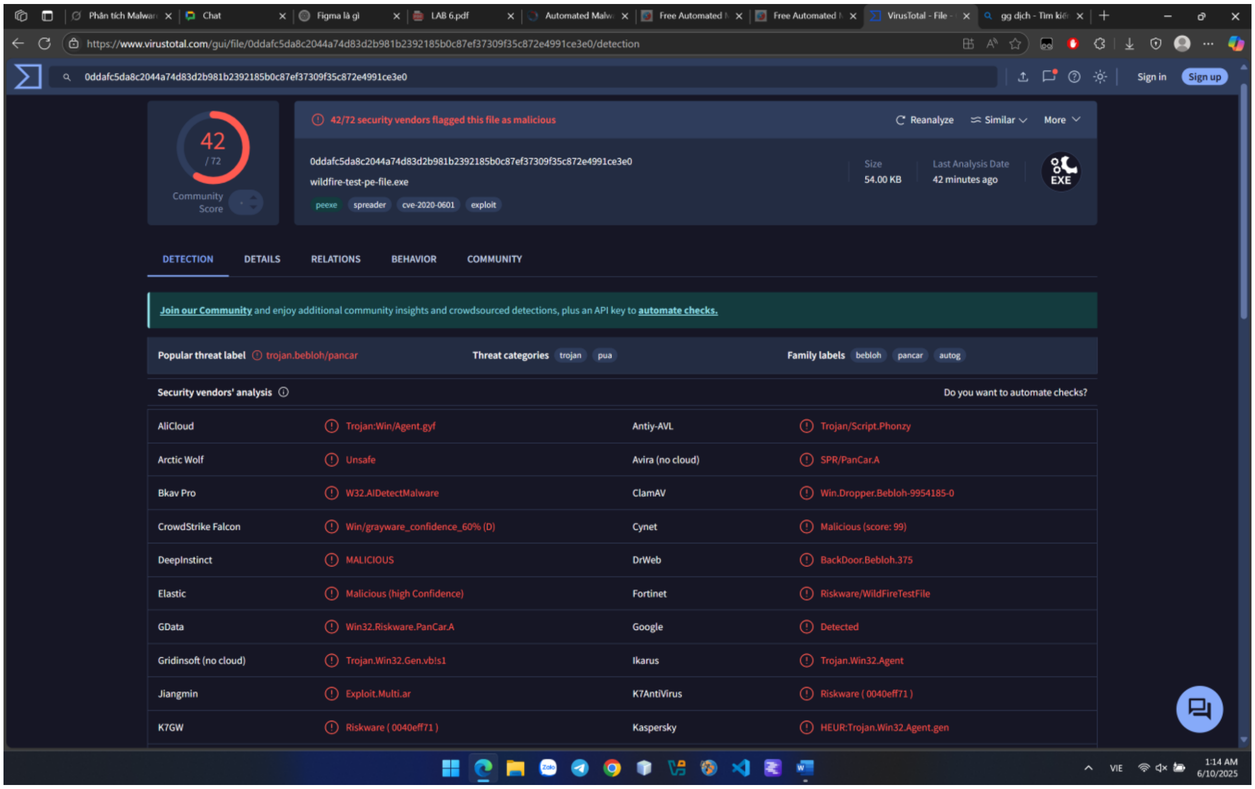
Nguyen Nam Khanh – HE191159 – IA1902

LAB 6: Public AV Scanners (VirusTotal, JoeSandbox)

VIRUS TOTAL

Upload https://wildfire.paloaltonetworks.com/publicapi/test/pe file to check.

Detection: almost tools detected Trojan(42/72 flag)



0ddafc5da8c2044a74d83d2b981b2392185b0c87ef37309f35c872e4991ce3e0

42 / 72

42/72 security vendors flagged this file as malicious

0ddafc5da8c2044a74d83d2b981b2392185b0c87ef37309f35c872e4991ce3e0

wildfire-test-pe-file.exe

peexe spreader cve-2020-0601 exploit

Size 54.00 KB

Last Analysis Date 42 minutes ago

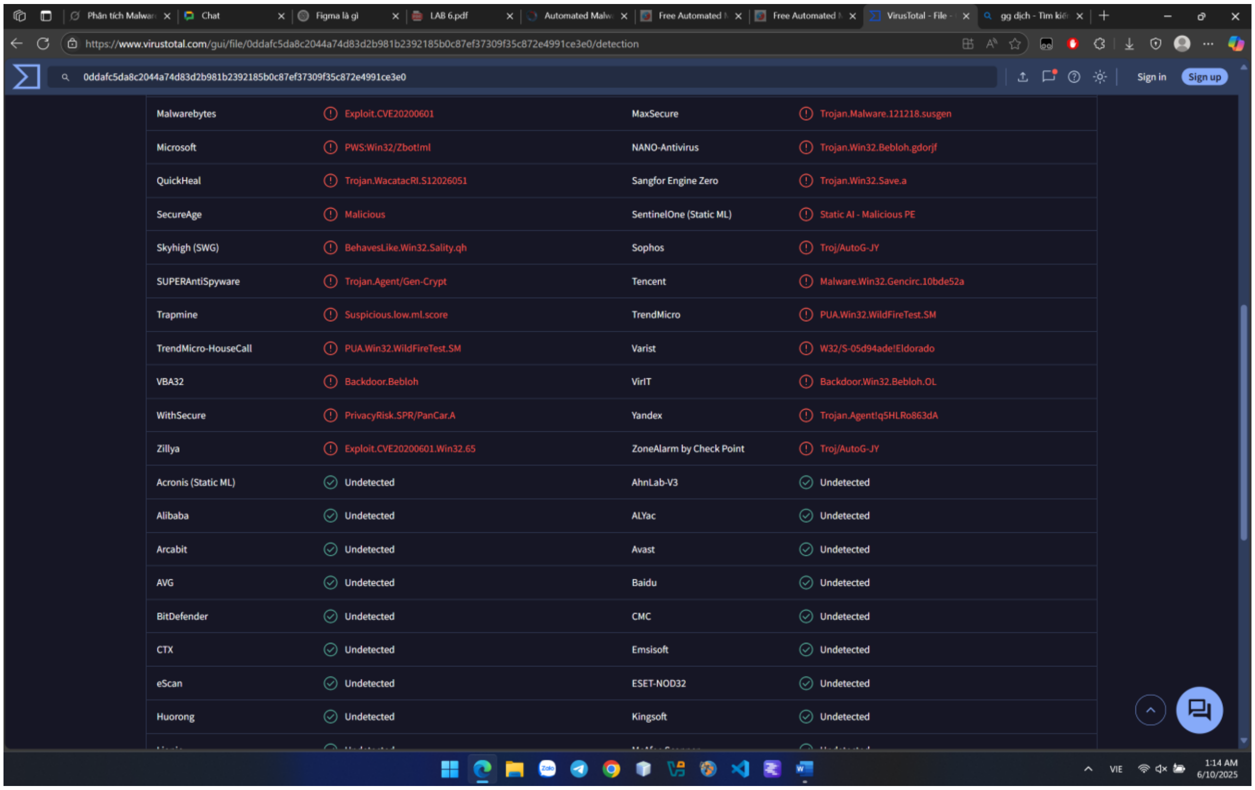
EXE

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

Popular threat label trojan.bebloh/pancar | Threat categories trojan pua | Family labels bebloh pancar autog

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

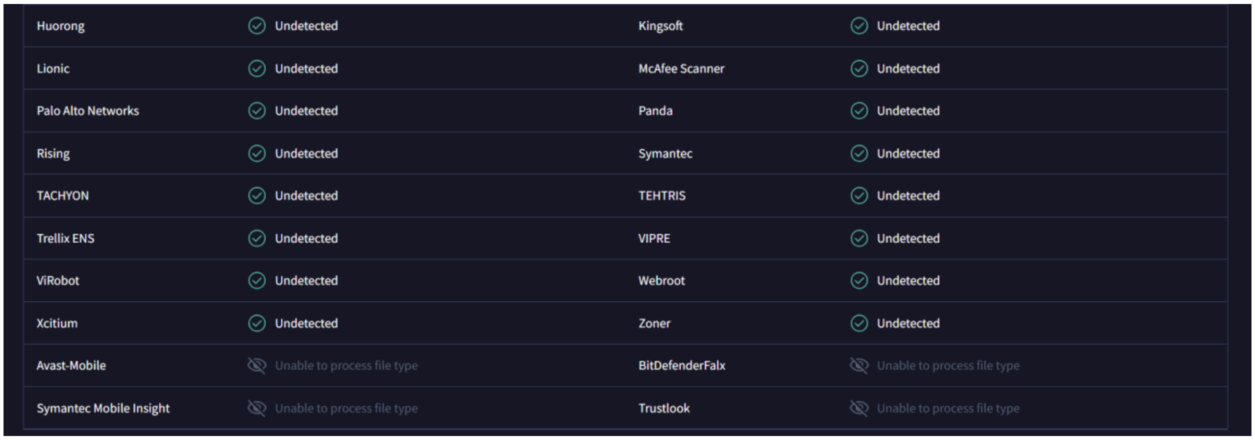
AliCloud	Trojan:Win/Agent.gyf	Antiy-AVL	Trojan/Script.Phonzy
Arctic Wolf	Unsafe	Avira (no cloud)	SPR/PanCar.A
Bkav Pro	W32.AIDetectMalware	ClamAV	Win.Dropper.Bebloh-9954185-0
CrowdStrike Falcon	Win/grayware_confidence_60% (D)	Cynet	Malicious (score: 99)
DeepInstinct	MALICIOUS	DrWeb	BackDoor.Bebloh.375
Elastic	Malicious (high Confidence)	Fortinet	Riskware/WildFireTestFile
GData	Win32.Riskware.PanCar.A	Google	Detected
Gridinsoft (no cloud)	Trojan.Win32.Gen.vb!s1	Ikarus	Trojan.Win32.Agent
Jiangmin	Exploit.Multi.ar	K7AntiVirus	Riskware (0040eff71)
K7GW	Riskware (0040eff71)	Kaspersky	HEUR:Trojan.Win32.Agent.gen



https://www.virustotal.com/gui/file/0ddafc5da8c2044a74d83d2b981b2392185b0c87ef37309f35c872e4991ce3e0/detection

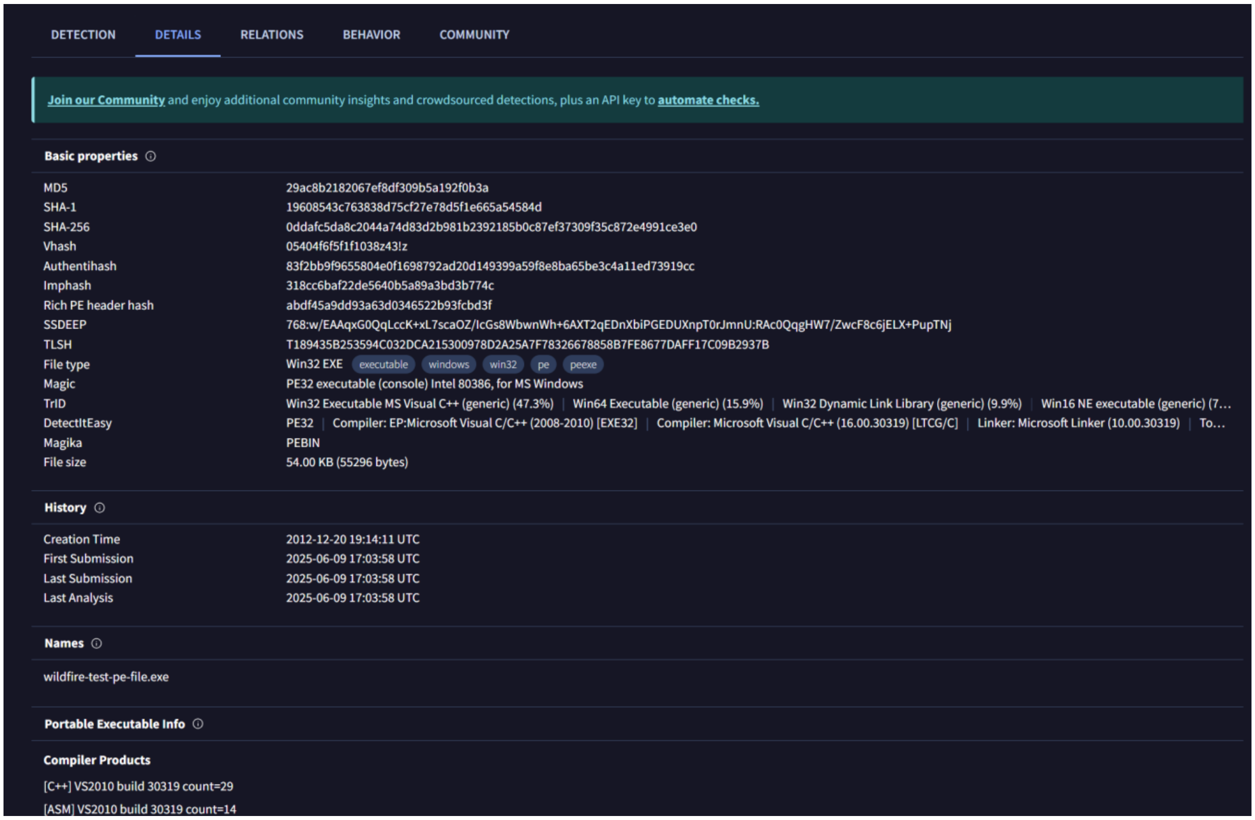
0ddafc5da8c2044a74d83d2b981b2392185b0c87ef37309f35c872e4991ce3e0

Malwarebytes	ⓘ Exploit.CVE20200601	MaxSecure	ⓘ Trojan.Malware.121218.susgen
Microsoft	ⓘ PWS:Win32/Zbot!ml	NANO-Antivirus	ⓘ Trojan.Win32.Bebloh.gdorjf
QuickHeal	ⓘ Trojan.WacatacRI.S12026051	Sangfor Engine Zero	ⓘ Trojan.Win32.Save.a
SecureAge	ⓘ Malicious	SentinelOne (Static ML)	ⓘ Static AI - Malicious PE
Skyhigh (SWG)	ⓘ BehavesLike.Win32.Sality.qh	Sophos	ⓘ Troj/AutoG-JY
SUPERAntiSpyware	ⓘ Trojan.Agent/Gen-Crypt	Tencent	ⓘ Malware.Win32.Gencirc.10bde52a
Trapmine	ⓘ Suspicious.low.ml.score	TrendMicro	ⓘ PUA.Win32.WildFireTest.SM
TrendMicro-HouseCall	ⓘ PUA.Win32.WildFireTest.SM	Varist	ⓘ W32/S-05d94ade!Eldorado
VBA32	ⓘ Backdoor.Bebloh	VirIT	ⓘ Backdoor.Win32.Bebloh.OL
WithSecure	ⓘ PrivacyRisk.SPR/PanCar.A	Yandex	ⓘ Trojan.Agent!q5HLRo863dA
Zillya	ⓘ Exploit.CVE20200601.Win32.65	ZoneAlarm by Check Point	ⓘ Troj/AutoG-JY
Acronis (Static ML)	✓ Undetected	AhnLab-V3	✓ Undetected
Alibaba	✓ Undetected	ALYac	✓ Undetected
Arcabit	✓ Undetected	Avast	✓ Undetected
AVG	✓ Undetected	Baidu	✓ Undetected
BitDefender	✓ Undetected	CMC	✓ Undetected
CTX	✓ Undetected	Emsisoft	✓ Undetected
eScan	✓ Undetected	ESET-NOD32	✓ Undetected
Huorong	✓ Undetected	Kingsoft	✓ Undetected



Huorong	✓ Undetected	Kingsoft	✓ Undetected
Lionic	✓ Undetected	McAfee Scanner	✓ Undetected
Palo Alto Networks	✓ Undetected	Panda	✓ Undetected
Rising	✓ Undetected	Symantec	✓ Undetected
TACHYON	✓ Undetected	TEHTRIS	✓ Undetected
Trellix ENS	✓ Undetected	VIPRE	✓ Undetected
ViRobot	✓ Undetected	Webroot	✓ Undetected
Xcitium	✓ Undetected	Zoner	✓ Undetected
Avast-Mobile	Unable to process file type	BitDefenderFalx	Unable to process file type
Symantec Mobile Insight	Unable to process file type	Trustlook	Unable to process file type

Details about Hash properties and metadata.



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

Basic properties

MD5	29ac8b2182067ef8df309b5a192f0b3a
SHA-1	19608543c763838d75cf27e78d5f1e665a54584d
SHA-256	0ddafc5da8c2044a74d83d2b981b2392185b0c87ef37309f35c872e4991ce3e0
Vhash	05404f6f5f1f1038z43!z
Authentihash	83f2bb9f9655804e0f1698792ad20d149399a59f8e8ba65be3c4a11ed73919cc
Imphash	318cc6baf22de5640b5a89a3bd3b774c
Rich PE header hash	abdf45a9dd93a63d0346522b93fcbd3f
SSDEEP	768:w/EAAqxG0QqLccK+xL7scaOZ/IcGs8WbwnWh+6AXT2qEDnXbiPGEDUXnpT0rJmnU:RAc0QqgHW7/ZwcF8c6jELX+PupTNj
TLSH	T189435B253594C032DCA215300978D2A25A7F78326678858B7FE8677DAFF17C09B2937B
File type	Win32 EXE executable windows win32 pe peexe
Magic	PE32 executable (console) Intel 80386, for MS Windows
TrID	Win32 Executable MS Visual C++ (generic) (47.3%) | Win64 Executable (generic) (15.9%) | Win32 Dynamic Link Library (generic) (9.9%) | Win16 NE executable (generic) (7…
DetectItEasy	PE32 | Compiler: EP:Microsoft Visual C/C++ (2008-2010) [EXE32] | Compiler: Microsoft Visual C/C++ (16.00.30319) [LTCG/C] | Linker: Microsoft Linker (10.00.30319) | To…
Magika	PEBIN
File size	54.00 KB (55296 bytes)

History

Creation Time	2012-12-20 19:14:11 UTC
First Submission	2025-06-09 17:03:58 UTC
Last Submission	2025-06-09 17:03:58 UTC
Last Analysis	2025-06-09 17:03:58 UTC

Names

wildfire-test-pe-file.exe

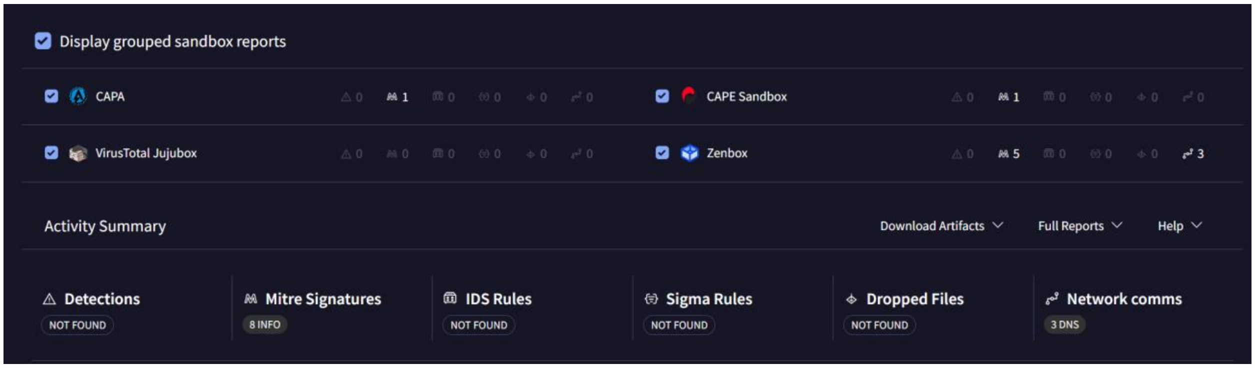
Portable Executable Info

Compiler Products

[C++] VS2010 build 30319 count=29

[ASM] VS2010 build 30319 count=14

Summary of activities on sandbox: no detections, 8 activities of MITRE signatures (Execution, Persistence, Privilege Escalation,...), 3 Network communications.



☑ Display grouped sandbox reports

☑ CAPA	0	1	0	0	0	0
☑ CAPE Sandbox	0	1	0	0	0	0
☑ VirusTotal Jujubox	0	0	0	0	0	0
☑ Zenbox	0	5	0	0	0	3

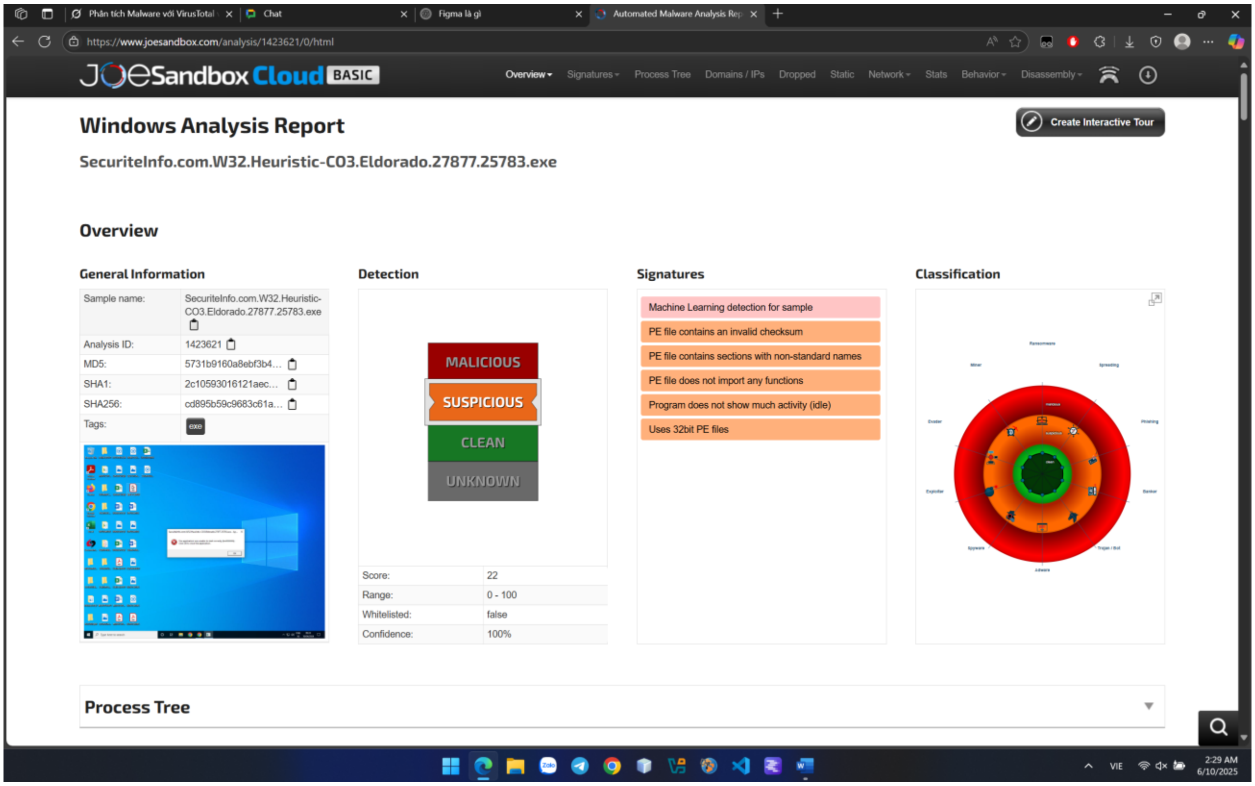
Activity Summary

Detections	Mitre Signatures	IDS Rules	Sigma Rules	Dropped Files	Network comms
NOT FOUND	8 INFO	NOT FOUND	NOT FOUND	NOT FOUND	3 DNS

JOE SANDBOX

Cause of using JOE tools needs account of company, as a student, I just have ability to inspect of some model report such as **Windows Analysis Report - Automated Malware Analysis Report for SecuriteInfo.com.W32.Heuristic-**

CO3.Eldorado.27877.25783.exe - Generated by Joe Sandbox



Windows Analysis Report

SecuriteInfo.com.W32.Heuristic-CO3.Eldorado.27877.25783.exe

Overview

General Information

Sample name:	SecuriteInfo.com.W32.Heuristic-CO3.Eldorado.27877.25783.exe
Analysis ID:	1423621
MD5:	5731b9160a8ebf3b4...
SHA1:	2c10593016121aec...
SHA256:	cd895b59c9683c61a...
Tags:	exe

Detection

MALICIOUS
SUSPICIOUS
CLEAN
UNKNOWN

Score:	22
Range:	0 - 100
Whitelisted:	false
Confidence:	100%

Signatures

- Machine Learning detection for sample
- PE file contains an invalid checksum
- PE file contains sections with non-standard names
- PE file does not import any functions
- Program does not show much activity (idle)
- Uses 32bit PE files

Classification

Process Tree

This file .exe had suspicious detection with score 20, low but more malicious and suspicious than file paloaltonetworks. With signature of invalid checkum, non-standard name and does not show much activity, these file was suspected to ransomware, spyware or trojan.

HYBRID ANALYSIS

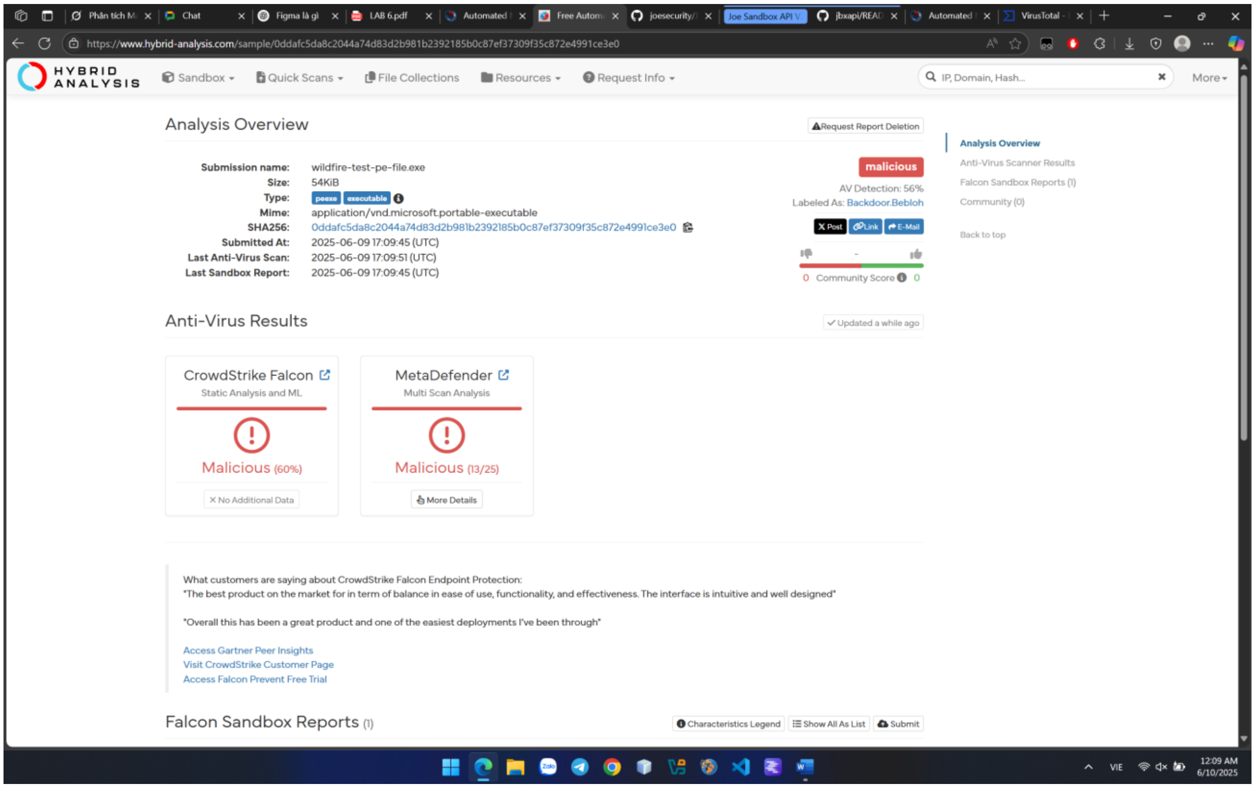
Upload https://wildfire.paloaltonetworks.com/publicapi/test/pe file to check

Overview Anit-Virus Results.

(Static analysis according to hash and signature)

CrowdStrike Falcon: 60% Malicious

MetaDenfender: 13/25 Malicious



Analysis Overview

Request Report Deletion

Submission name:	wildfire-test-pe-file.exe
Size:	54KiB
Type:	peexe executable
Mime:	application/vnd.microsoft.portable-executable
SHA256:	0ddafc5da8c2044a74d83d2b981b2392185b0c87ef37309f35c872e4991ce3e0
Submitted At:	2025-06-09 17:09:45 (UTC)
Last Anti-Virus Scan:	2025-06-09 17:09:51 (UTC)
Last Sandbox Report:	2025-06-09 17:09:45 (UTC)

malicious

AV Detection: 56%

Labeled As: Backdoor.Bebloh

X Post Link E-Mail

0 Community Score 0

Anti-Virus Results

✓ Updated a while ago

CrowdStrike Falcon
Static Analysis and ML
Malicious (60%)
✕ No Additional Data

MetaDefender
Multi Scan Analysis
Malicious (13/25)
More Details

What customers are saying about CrowdStrike Falcon Endpoint Protection:
"The best product on the market for in term of balance in ease of use, functionality, and effectiveness. The interface is intuitive and well designed"

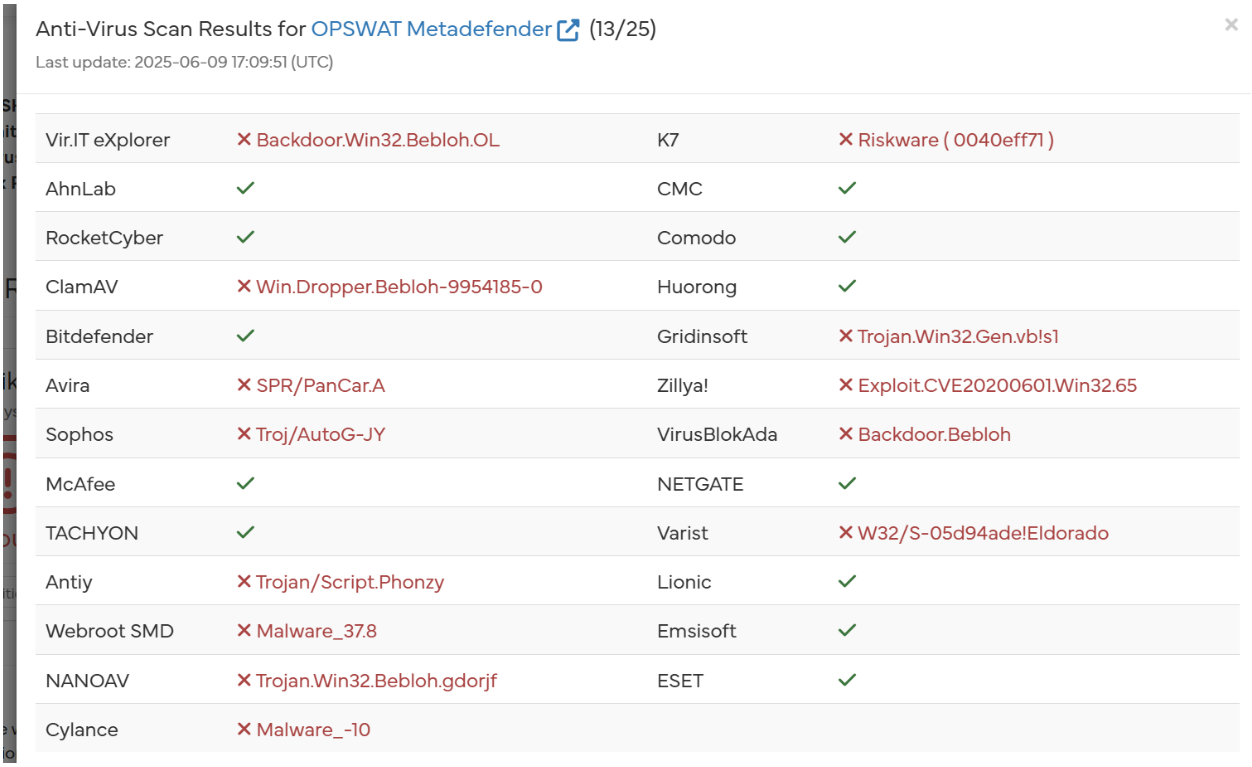
"Overall this has been a great product and one of the easiest deployments I've been through"

Access Gartner Peer Insights
Visit CrowdStrike Customer Page
Access Falcon Prevent Free Trial

Falcon Sandbox Reports (1)

Characteristics Legend | Show All As List | Submit

MetaDenfer Multi Scan Analysis Detail: almost related to Trojan and Backdoor

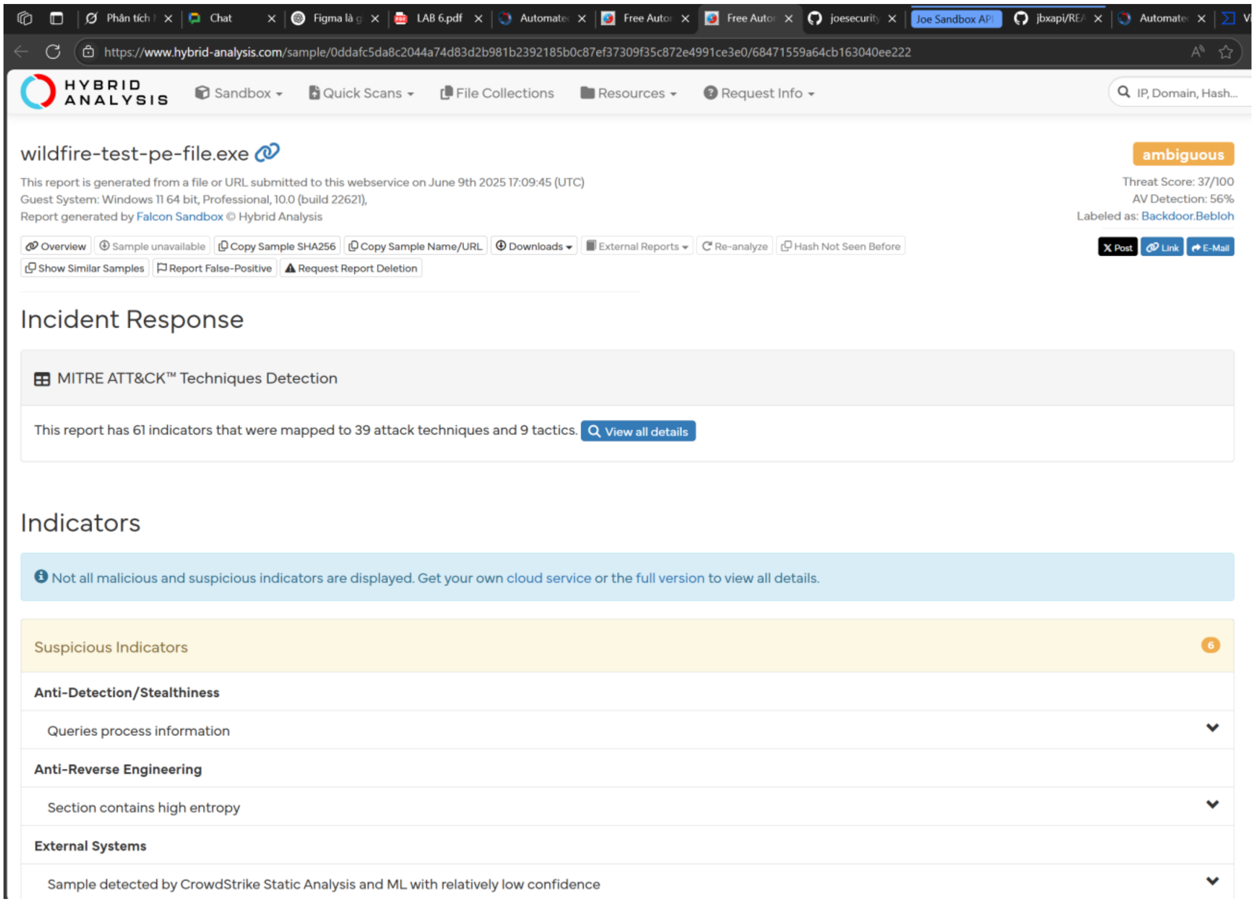


Anti-Virus Scan Results for OPSWAT Metadefender (13/25)

Last update: 2025-06-09 17:09:51 (UTC)

Vir.IT eXplorer	✕ Backdoor.Win32.Bebloh.OL	K7	✕ Riskware (0040eff71)
AhnLab	✓	CMC	✓
RocketCyber	✓	Comodo	✓
ClamAV	✕ Win.Dropper.Bebloh-9954185-0	Huorong	✓
Bitdefender	✓	Gridinsoft	✕ Trojan.Win32.Gen.vb!s1
Avira	✕ SPR/PanCar.A	Zillya!	✕ Exploit.CVE20200601.Win32.65
Sophos	✕ Troj/AutoG-JY	VirusBlokAda	✕ Backdoor.Bebloh
McAfee	✓	NETGATE	✓
TACHYON	✓	Varist	✕ W32/S-05d94ade!Eldorado
Antiy	✕ Trojan/Script.Phonzy	Lionic	✓
Webroot SMD	✕ Malware_37.8	Emsisoft	✓
NANOAV	✕ Trojan.Win32.Bebloh.gdorjf	ESET	✓
Cylance	✕ Malware_-10		

Intergrated Falcon Sandbox report detail: Threat score 37/100 with 61 indicators mapped 39 attack techniques and 9 tatics. (dynamic analysis with monitoring operation of process in virtual enviroment)



wildfire-test-pe-file.exe

This report is generated from a file or URL submitted to this webservice on June 9th 2025 17:09:45 (UTC)
Guest System: Windows 11 64 bit, Professional, 10.0 (build 22621),
Report generated by Falcon Sandbox © Hybrid Analysis

ambiguous

Threat Score: 37/100
AV Detection: 56%
Labeled as: Backdoor.Bebloh

Incident Response

MITRE ATT&CK™ Techniques Detection

This report has 61 indicators that were mapped to 39 attack techniques and 9 tactics.

Indicators

Not all malicious and suspicious indicators are displayed. Get your own cloud service or the full version to view all details.

Suspicious Indicators 6

Anti-Detection/Stealthiness

Queries process information

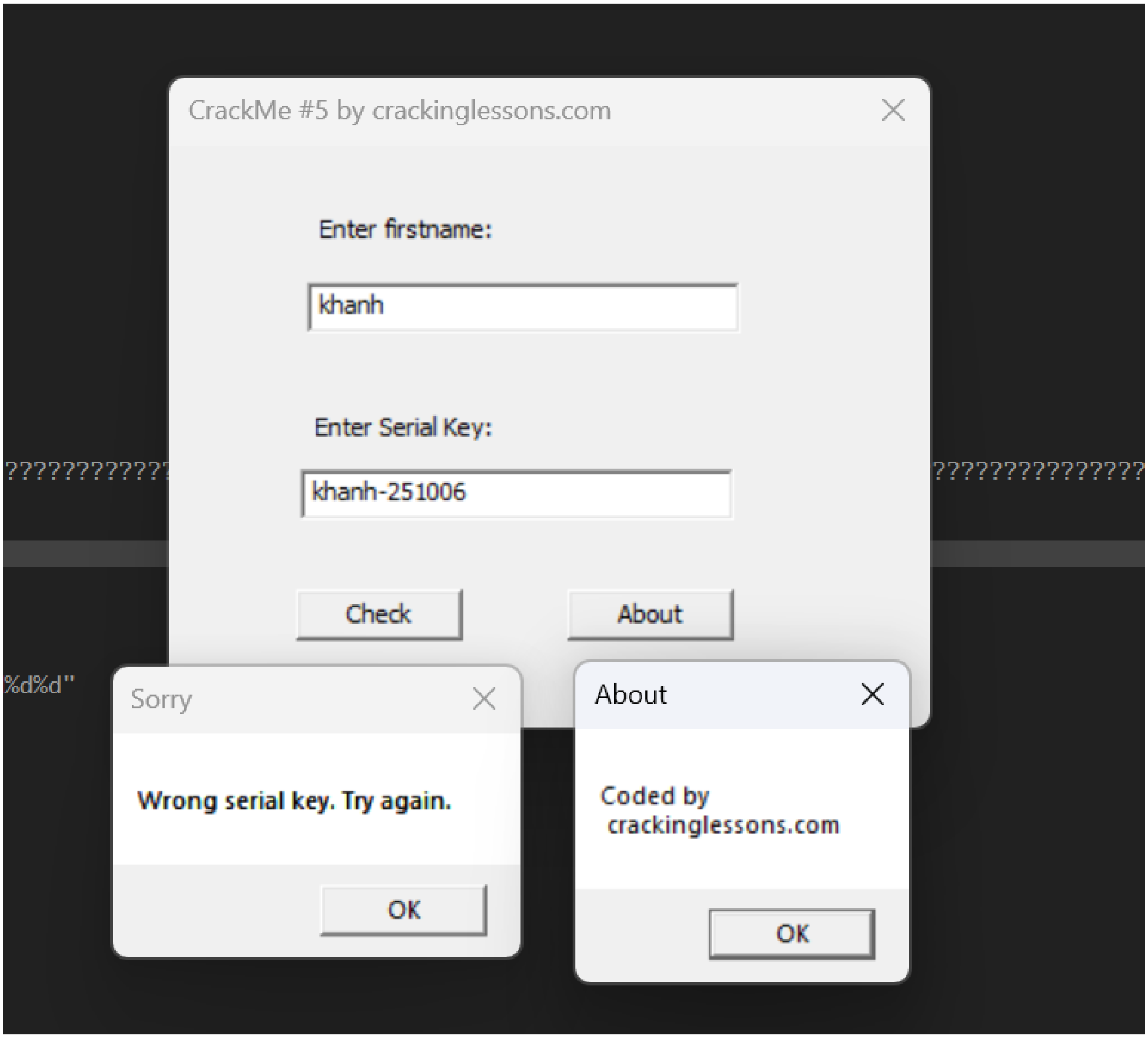
Anti-Reverse Engineering

Section contains high entropy

External Systems

Sample detected by CrowdStrike Static Analysis and ML with relatively low confidence

CRACK ME 5



CrackMe #5 by crackinglessons.com

Enter firstname:

khanh

Enter Serial Key:

khanh-251006

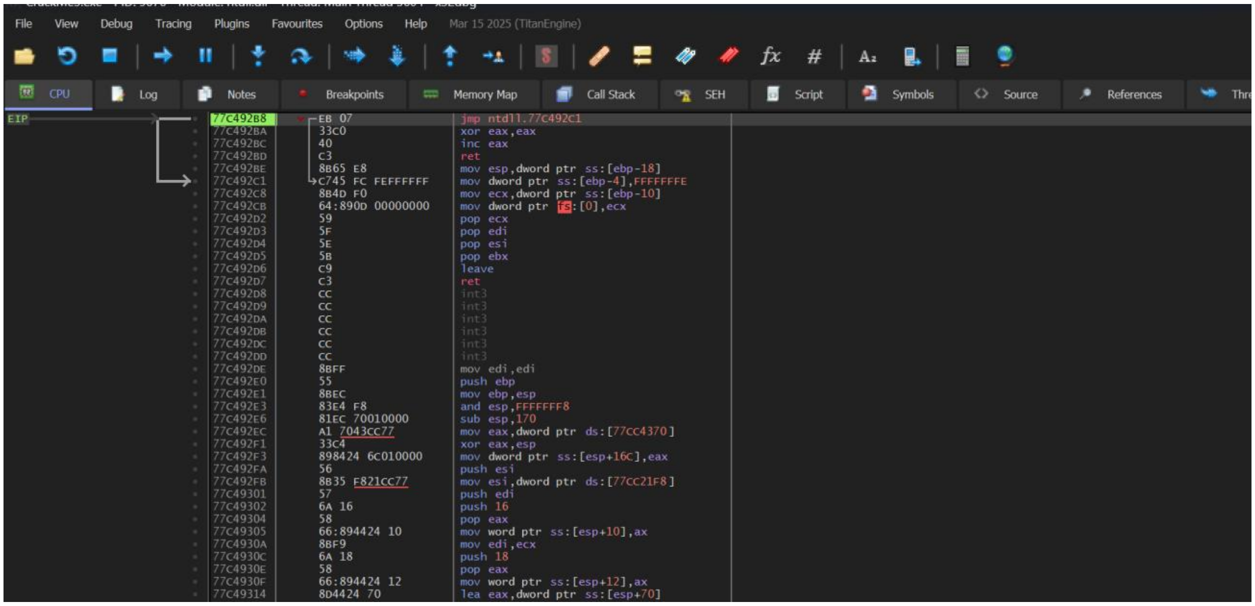
Sorry

Wrong serial key. Try again.

About

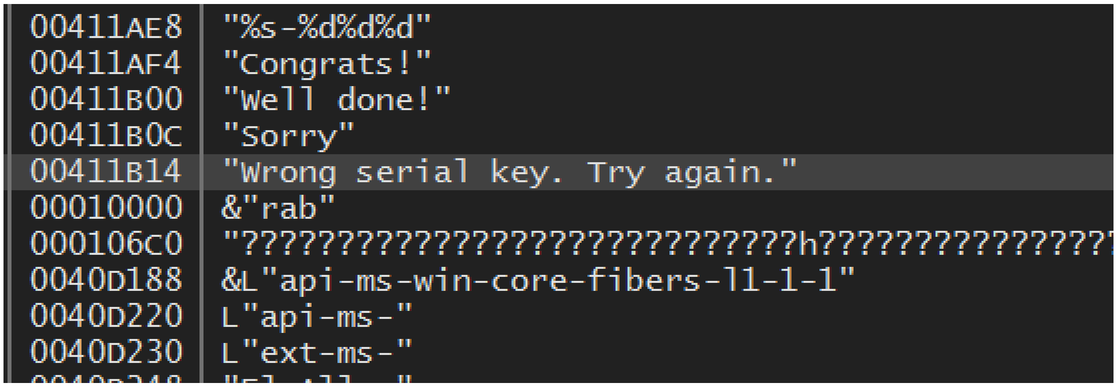
Coded by
crackinglessons.com

Inspect this file .exe in x32dbg tool.

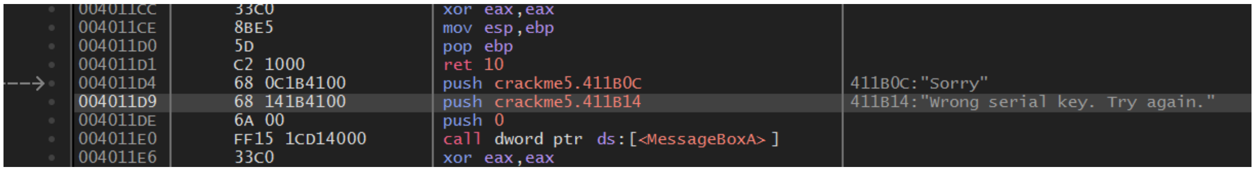


```
77C492B8   EB 07              jmp ntdll.77c492c1
77C492BA   33C0               xor eax,eax
77C492BC   40                 inc eax
77C492BD   C3                 ret
77C492BE   8B65 E8            mov esp,dword ptr ss:[ebp-18]
77C492C1   C745 FC FEFFFFFF   mov dword ptr ss:[ebp-4],FFFFFFFE
77C492C8   8B4D F0            mov ecx,dword ptr ss:[ebp-10]
77C492CB   64:890D 00000000   mov dword ptr fs:[0],ecx
77C492D2   59                 pop ecx
77C492D3   5F                 pop edi
77C492D4   5E                 pop esi
77C492D5   5B                 pop ebx
77C492D6   C9                 leave
77C492D7   C3                 ret
77C492D8   CC                 int3
77C492D9   CC                 int3
77C492DA   CC                 int3
77C492DB   CC                 int3
77C492DC   CC                 int3
77C492DD   CC                 int3
77C492DE   8BFF               mov edi,edi
77C492E0   55                 push ebp
77C492E1   8BEC               mov ebp,esp
77C492E3   83E4 F8            and esp,FFFFFFF8
77C492E6   81EC 70010000      sub esp,170
77C492EC   A1 7043CC77        mov eax,dword ptr ds:[77CC4370]
77C492F1   33C4               xor eax,esp
77C492F3   898424 6C010000    mov dword ptr ss:[esp+16C],eax
77C492FA   56                 push esi
77C492FB   8B35 F821CC77      mov esi,dword ptr ds:[77CC21F8]
77C49301   57                 push edi
77C49302   6A 16              push 16
77C49304   58                 pop eax
77C49305   66:894424 10       mov word ptr ss:[esp+10],ax
77C4930A   8BF9               mov edi,ecx
77C4930C   6A 18              push 18
77C4930E   58                 pop eax
77C4930F   66:894424 12       mov word ptr ss:[esp+12],ax
77C49314   8D4424 70          lea eax,dword ptr ss:[esp+70]
```

Find string reference to string “Wrong serial key. Try again.”. Click to jump to the line of code of these string.

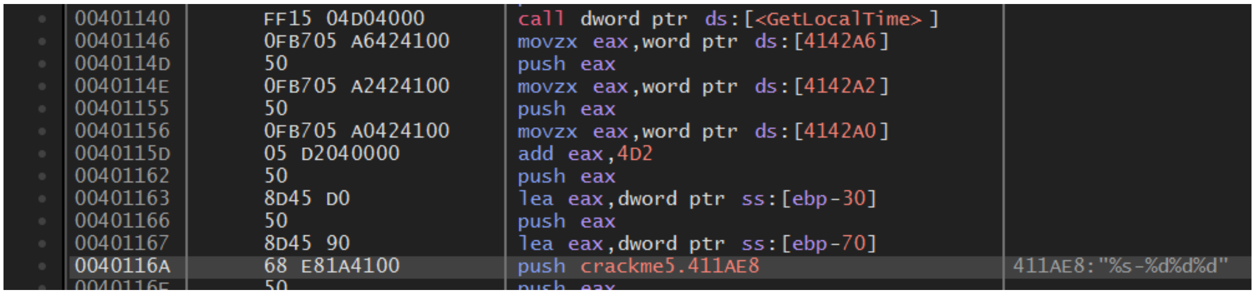


```
00411AE8   "%s-%d%d%d"
00411AF4   "Congrats!"
00411B00   "Well done!"
00411B0C   "Sorry"
00411B14   "Wrong serial key. Try again."
00010000   &"rab"
000106C0   "????????????????????????????h???????????????
0040D188   &L"api-ms-win-core-fibers-l1-1-1"
0040D220   L"api-ms-"
0040D230   L"ext-ms-"
```



```
004011CC   33C0          xor eax,eax
004011CE   8BE5          mov esp,ebp
004011D0   5D            pop ebp
004011D1   C2 1000       ret 10
004011D4   68 0C1B4100   push crackme5.411B0C             411B0C:"Sorry"
004011D9   68 141B4100   push crackme5.411B14             411B14:"Wrong serial key. Try again."
004011DE   6A 00         push 0
004011E0   FF15 1CD14000 call dword ptr ds:[<MessageBoxA>]
004011E6   33C0          xor eax,eax
```

Trace back to the above code to see what operates, leading to this “Sorry” code.

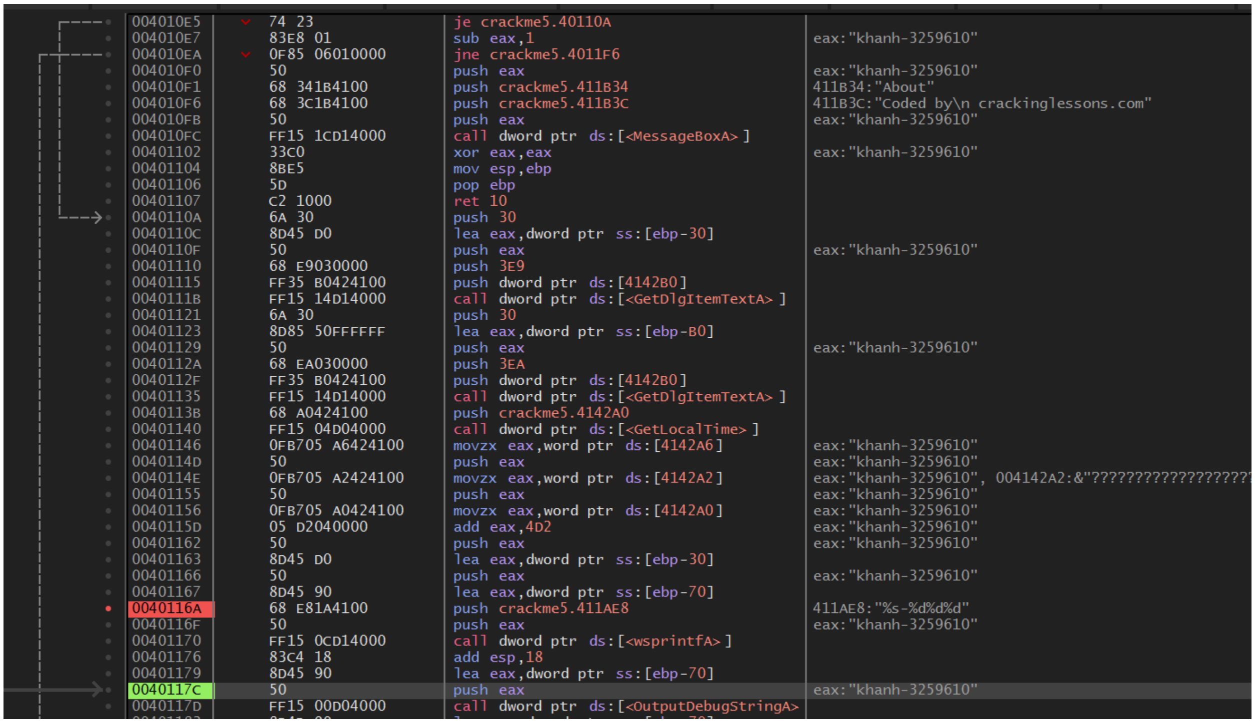


```
00401140   FF15 04D04000   call dword ptr ds:[<GetLocalTime>]
00401146   0FB705 A6424100 movzx eax,word ptr ds:[4142A6]
0040114D   50              push eax
0040114E   0FB705 A2424100 movzx eax,word ptr ds:[4142A2]
00401155   50              push eax
00401156   0FB705 A0424100 movzx eax,word ptr ds:[4142A0]
0040115D   05 D2040000     add eax,4D2
00401162   50              push eax
00401163   8D45 D0         lea eax,dword ptr ss:[ebp-30]
00401166   50              push eax
00401167   8D45 90         lea eax,dword ptr ss:[ebp-70]
0040116A   68 E81A4100     push crackme5.411AE8             411AE8:"%s-%d%d%d"
0040116E   50              push eax
```

Notice to this code part, from function GetLocalTime, analysing variabe EAX. May be, EAX represents for serial key. But now, ignoring it for a bit, we add breakpoint to code line of function GetLocalTime and step into to deeply analyse.

After many times for tracing and putting breakpoint, I get some detail to see.

At the address 0040116A, we have the comment: “411AE8: “%s-%d%d%d””, below this is hint “khanh-325910”, this is likely to be serial key.

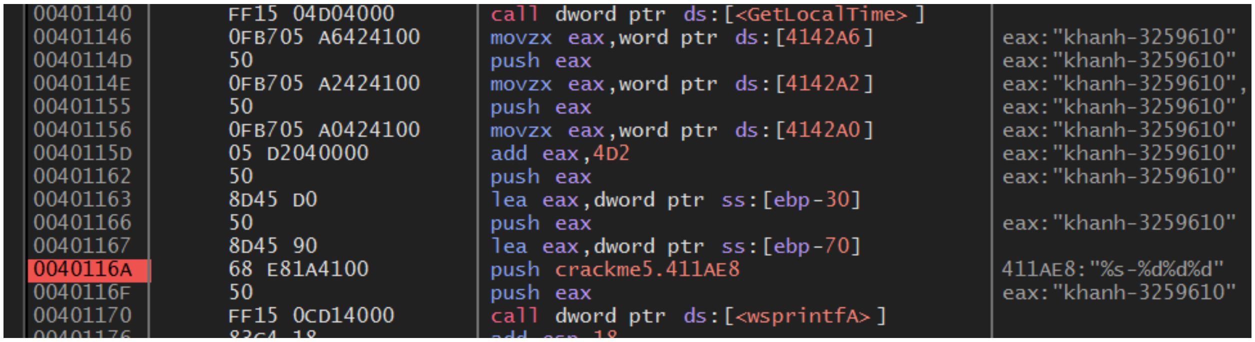


```
004010E5   74 23             je crackme5.40110A
004010E7   83E8 01           sub eax,1                                  eax:"khanh-3259610"
004010EA   0F85 06010000     jne crackme5.4011F6
004010F0   50                push eax                                   eax:"khanh-3259610"
004010F1   68 341B4100       push crackme5.411B34                       411B34:"About"
004010F6   68 3C1B4100       push crackme5.411B3C                       411B3C:"Coded by\n crackinglessons.com"
004010FB   50                push eax                                   eax:"khanh-3259610"
004010FC   FF15 1CD14000     call dword ptr ds:[<MessageBoxA>]
00401102   33C0              xor eax,eax                                eax:"khanh-3259610"
00401104   8BE5              mov esp,ebp
00401106   5D                pop ebp
00401107   C2 1000           ret 10
0040110A   6A 30             push 30
0040110C   8D45 D0           lea eax,dword ptr ss:[ebp-30]
0040110F   50                push eax                                   eax:"khanh-3259610"
00401110   68 E9030000       push 3E9
00401115   FF35 B0424100     push dword ptr ds:[4142B0]
0040111B   FF15 14D14000     call dword ptr ds:[<GetDlgItemTextA>]
00401121   6A 30             push 30
00401123   8D85 50FFFFFF     lea eax,dword ptr ss:[ebp-B0]
00401129   50                push eax                                   eax:"khanh-3259610"
0040112A   68 EA030000       push 3EA
0040112F   FF35 B0424100     push dword ptr ds:[4142B0]
00401135   FF15 14D14000     call dword ptr ds:[<GetDlgItemTextA>]
0040113B   68 A0424100       push crackme5.4142A0
00401140   FF15 04D04000     call dword ptr ds:[<GetLocalTime>]
00401146   0FB705 A6424100   movzx eax,word ptr ds:[4142A6]             eax:"khanh-3259610"
0040114D   50                push eax                                   eax:"khanh-3259610"
0040114E   0FB705 A2424100   movzx eax,word ptr ds:[4142A2]             eax:"khanh-3259610", 004142A2:&"??????????????????
00401155   50                push eax                                   eax:"khanh-3259610"
00401156   0FB705 A0424100   movzx eax,word ptr ds:[4142A0]             eax:"khanh-3259610"
0040115D   05 D2040000       add eax,4D2                                eax:"khanh-3259610"
00401162   50                push eax                                   eax:"khanh-3259610"
00401163   8D45 D0           lea eax,dword ptr ss:[ebp-30]
00401166   50                push eax                                   eax:"khanh-3259610"
00401167   8D45 90           lea eax,dword ptr ss:[ebp-70]
0040116A   68 E81A4100       push crackme5.411AE8                       411AE8:"%s-%d%d%d"
0040116F   50                push eax                                   eax:"khanh-3259610"
00401170   FF15 0CD14000     call dword ptr ds:[<wsprintfA>]
00401176   83C4 18           add esp,18
00401179   8D45 90           lea eax,dword ptr ss:[ebp-70]
0040117C   50                push eax                                   eax:"khanh-3259610"
0040117D   FF15 00D04000     call dword ptr ds:[<OutputDebugStringA>
```

But why we have this form of serial key? The regex “%s-%d%d%d” equal to khanh-325910.

%s is for khanh (the firstname) combine with -%d%d%d to -3259610.

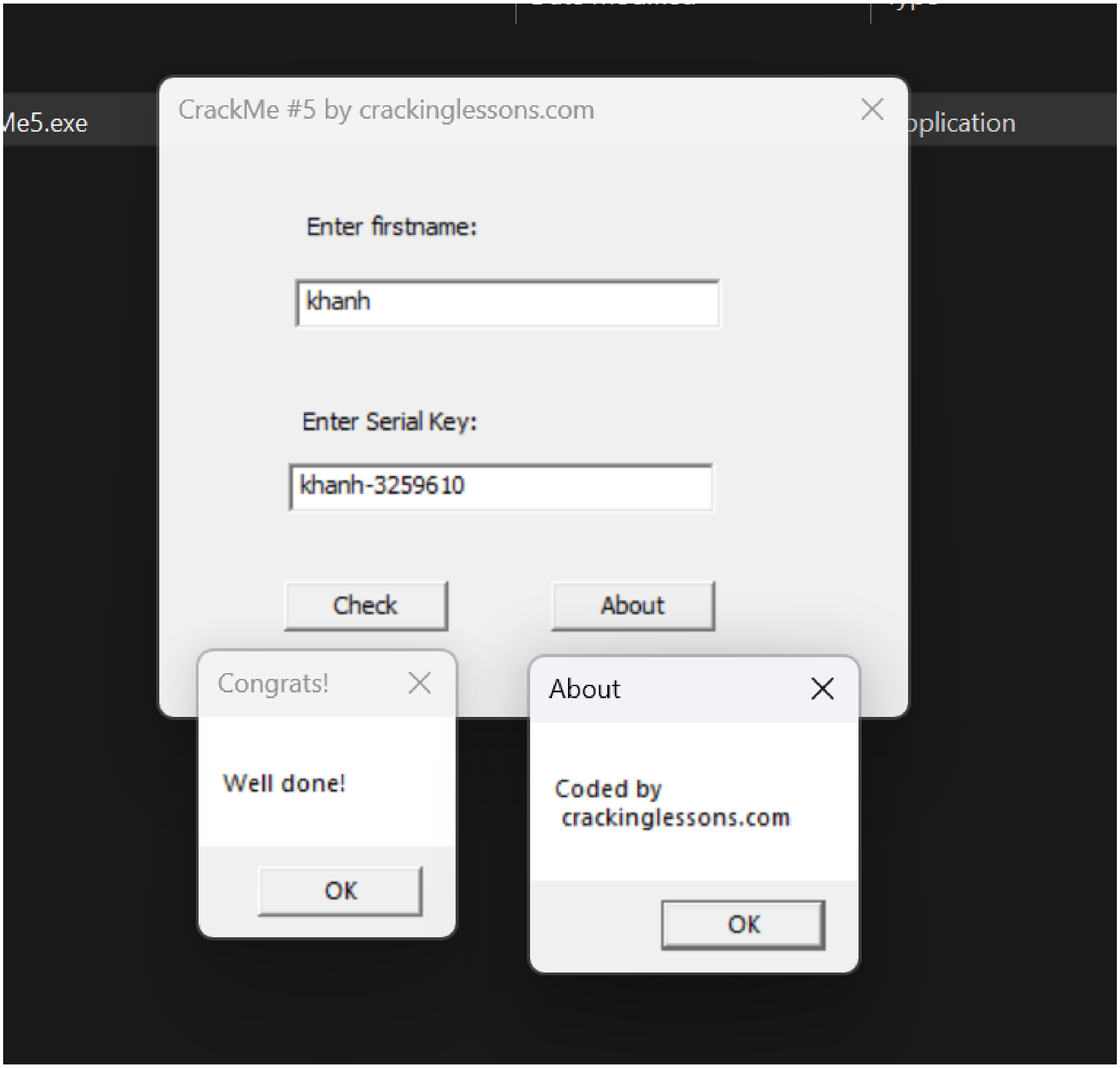
Let trace back again to this part of code below:



```
00401140   FF15 04D04000     call dword ptr ds:[<GetLocalTime>]
00401146   0FB705 A6424100   movzx eax,word ptr ds:[4142A6]     eax:"khanh-3259610"
0040114D   50                push eax                           eax:"khanh-3259610"
0040114E   0FB705 A2424100   movzx eax,word ptr ds:[4142A2]     eax:"khanh-3259610",
00401155   50                push eax                           eax:"khanh-3259610"
00401156   0FB705 A0424100   movzx eax,word ptr ds:[4142A0]     eax:"khanh-3259610"
0040115D   05 D2040000       add eax,4D2                        eax:"khanh-3259610"
00401162   50                push eax                           eax:"khanh-3259610"
00401163   8D45 D0           lea eax,dword ptr ss:[ebp-30]
00401166   50                push eax                           eax:"khanh-3259610"
00401167   8D45 90           lea eax,dword ptr ss:[ebp-70]
0040116A   68 E81A4100       push crackme5.411AE8               411AE8:"%s-%d%d%d"
0040116F   50                push eax                           eax:"khanh-3259610"
00401170   FF15 0CD14000     call dword ptr ds:[<wsprintfA>]
```

System call GetLocalTime this means get the time schedule of system, at this time is 10th of june 2025. If mapping to regex, we have -2025610, but serial key is -3259610.

Notice to the address 0040115D, after assigning and pushing numbers of month and day to EAX. System assigns number of year to EAX, and at this line of code, adding EAX with 4D2 in hexa (1234 in decimal). 2025 adds with 1234 euqal to 3259. Hence, we have Serial Key: khanh-3259610.



CrackMe #5 by crackinglessons.com

Enter firstname: khanh

Enter Serial Key: khanh-3259610

Congrats!

Well done!

About

Coded by crackinglessons.com

Typing firstname and try the cracked serial key to check.

DONE!!!