

# PebbleDash - Lazarus / HiddenCobra RAT

By malwarenailed

Published: 2020-06-01 · Archived: 2026-04-05 18:33:47 UTC

Hi folks. I was analyzing the PebbleDash malware used by Lazarus APT group. While analyzing the original sample (Md5: d2de01858417fa3b580b3a95857847d5), I was able to find out the C2 server and the port, where it intends to communicate to. I also found an interesting technique it uses to identify the OS version of the victim machine.

During static analysis, I observed interesting strings were starting with "Zip-bug", as can be seen below. Using yara rules I was able to discover some other samples uploaded to HA (Hybrid Analysis) with the same strings embedded. These samples seemed to be not related to d2de01858417fa3b580b3a95857847d5. However, they communicated to South Korea and China.

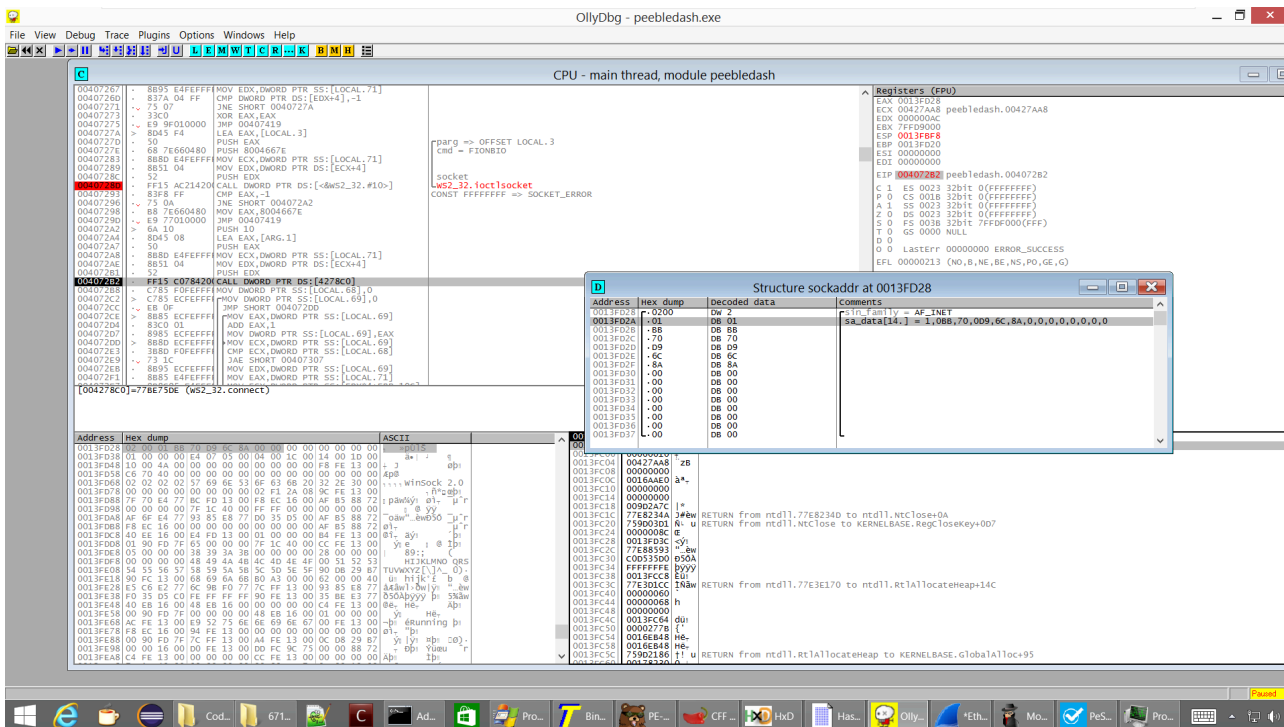
Type	Size	Blacklisted ...	Value
ascii	25	-	Couldn't create/open file
ascii	25	-	Failed to allocate memory
ascii	21	-	Error writing to file
ascii	29	-	File not found in the zipfile
ascii	24	-	Still more data to unzip
ascii	35	-	Zipfile is corrupt or not a zipfile
ascii	18	-	Error reading file
ascii	24	-	Caller: faulty arguments
ascii	52	-	Caller: the file had already been partially unzipped
ascii	47	-	Caller: can only get memory of a memory zipfile
ascii	53	-	Caller: not enough space allocated for memory zipfile
ascii	34	-	Caller: there was a previous error
ascii	52	-	Caller: additions to the zip have already been ended
ascii	42	-	Caller: mixing creation and opening of zip
ascii	46	-	Zip-bug: internal initialisation not completed
ascii	38	-	Zip-bug: trying to seek the unseekable
ascii	46	-	Zip-bug: the anticipated size turned out wrong
ascii	46	-	Zip-bug: tried to change mind, but not allowed
ascii	41	-	Zip-bug: an internal error during flation
ascii	4	-	kU'9
ascii	4	-	HMXB
ascii	4	-	???

While performing dynamic analysis, I observed that the sample uses the API call IsProcessorFeaturePresent to determine the version of the victim OS. The PF\_FLOATING\_POINT\_PRECISION\_ERRATA feature is explicitly set to FALSE in x86 version 6.1 and higher.

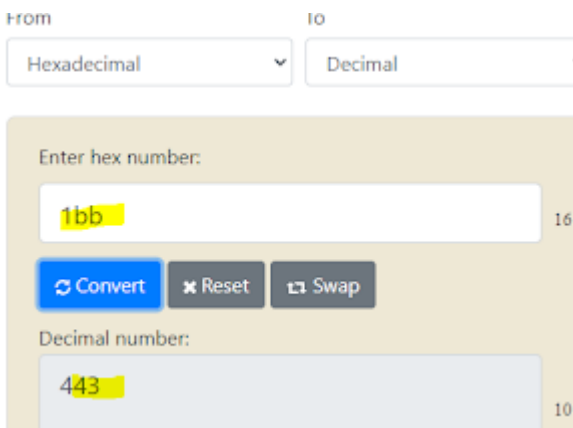
```

0041AF46 74 05 JZ SHORT 0041AF4D
0041AF48 6A 00 PUSH 0
0041AF4C FF D0 CALL EAX
0041AF4C C3 RETN
0041AF4D E9 99FFFFFF JMP 0041AE6B
0041AF52 56 PUSH ESI
0041AF53 8B7424 08 MOV ESI, DWORD PTR SS:[ARG_1]
0041AF57 0FBE06 MOVSW EAX, BYTE PTR DS:[ESI]
0041AF5A 50 PUSH EAX
0041AF5B E8 BB340000 CALL 0041E41B
0041AF60 83F8 65 CMP EAX, 65
0041AF63 59 POP ECX
0041AF64 74 2C JE SHORT 0041AF92
0041AF66 4E INC ESI
0041AF67 833D BC704201 CMP DWORD PTR DS:[4270BC], 1
EAX=75C61A54 (KERNEL32.IsProcessorFeaturePresent)
    
```





The first two bytes in the structure represents the destination port and we can see that it is 443 in this case.



The rest of the four bytes are: 0x70 0xd9 0x6C 0x8A, which translates to 112.217.108.138 (hex to decimal). This is the C2 ip address where PebbleDash communicates to. This IOC also be seen in the US-CERT advisroy.

terminate processes, and perform target system enumeration.

For a downloadable copy of IOCs, see [MAR-10288834-3.v1.stix](#).

Submitted Files (1)

aab2868a6ebc6bdee5bd12104191db9fc1950b30bcf96eab99801624651e77b6 (D2DE01858417FA3B580B3A95857847D5)

IPs (1)

112.217.108.138

Findings

aab2868a6ebc6bdee5bd12104191db9fc1950b30bcf96eab99801624651e77b6

Tags

rootkit trojan

Details

Name	D2DE01858417FA3B580B3A95857847D5
Size	167937 bytes

PebbleDash inserts fake "server name" in the TLS packet. We can see below some:

The screenshot displays assembly code from a debugger. Key instructions include:

- 0040932D: F7F1 DIV ECX
- 0040932F: 8995 50FEFFFF MOV DWORD PTR SS:[EBP-1B0],EDX
- 00409335: 8B95 50FEFFFF MOV EDX,DWORD PTR SS:[EBP-1B0]
- 0040933B: 8B0495 B45C41 MOV EAX,DWORD PTR DS:[EDX\*4+425CB4]
- 00409342: 50 PUSH EAX
- 00409343: FE15 44204201 CALL DWORD PTR DS:[<&KERNEL32!strlenA>]
- 00409349: 8985 3CFEFFFF MOV DWORD PTR SS:[EBP-0C4],EAX
- 0040934F: 66:C785 58FEFF MOV WORD PTR SS:[EBP-1A8],0
- 00409358: 8B80 3CFEFFFF MOV ECX,DWORD PTR SS:[EBP-0C4]
- 0040935E: 83C1 05 ADD ECX,5
- 00409361: 51 PUSH ECX
- 00409362: E8 D9860100 CALL <JMP.&WS2\_32.#9>
- 00409367: 66:8985 5AFEFF MOV WORD PTR SS:[EBP-1A6],AX
- 0040936E: 8B95 3CFEFFFF MOV EDX,DWORD PTR SS:[EBP-0C4]
- 00409374: 83C2 03 ADD EDX,3
- 00409377: 52 PUSH EDX
- 00409378: E8 C3860100 CALL <JMP.&WS2\_32.#9>
- 0040937D: 66:8985 5CFEFF MOV WORD PTR SS:[EBP-1A4],AX
- 00409384: C685 5EFEFFFF MOV BYTE PTR SS:[EBP-1A2],0
- 0040938B: 66:8B85 3CFEFF MOV AX,WORD PTR SS:[EBP-0C4]
- 00409392: 50 PUSH EAX
- 00409393: E8 A8860100 CALL <JMP.&WS2\_32.#9>

The hex dump at the bottom shows the following data:

Address	Hex dump	ASCII
01370248	21 00 00 00 31 00 00 00 41 50 50 44 41 54 41 3D	1 APPDATA=
01370258	43 3A 5C 55 73 65 72 73 5C 52 45 4D 5C 41 70 70	C:\users\REM\AppData
01370268	44 61 74 61 5C 52 6F 61 6D 69 6E 67 00 00 00 00	data\Roaming

Source: <https://malwarenailed.blogspot.com/2020/06/peebledash-lazarus-hiddencobra-rat.html?m=1>