**103 – A suspicious developer**

|  |
| --- |
| **Team Information** |

**Team Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Team Member \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Email Address \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| **Instructions** |

**Description** The auditing department of a software development company received an internal whistleblower report stating that a developer from the development team had outsourced a project to another company for execution. The auditing department initiated an investigation to determine whether there had been a violation of internal labor and security regulations. In response, that developer claimed to have personally developed the project and submitted the program source files and the resulting executable files to the audit team as evidence. The auditing department obtained the previous deliverables (executable files) that the developer had submitted by retrieving them from company’s project management server. Verify the accuracy of the internal whistleblower’s claims.

|  |  |
| --- | --- |
| Target | Hash (SHA1) |
| Files.zip | 26B11C75AD1F469B35284A29D973B716C030C71B |

**Questions**

# Please solve all problems based on UTC+9 time zone.

1. Write the items indicating the build tool version information stored in the given two PE format executable files in the format of “[ProductID].[BuildID].[Count]”. (80 points)
   * Write 9 items per file and do so for both two files. (40 points each)
2. Write the build folder paths for the given two executable files. (20 points)
   * Write the build folder paths for both files. (10 points each)

Teams must:

* Develop and document the step-by-step approach used to solve this problem to allow another examiner to replicate team actions and results.
* Specify all tools used in deriving the conclusion(s).

|  |
| --- |
| **Tools used:** |

|  |  |  |  |
| --- | --- | --- | --- |
| Name: |  | Publisher**:** |  |
| Version: |  | | |
| URL: |  | | |

|  |
| --- |
| **Step-by-step methodology:** |

1. Write the items indicating the build tool version information stored in the given two PE format executable files in the format of “[ProductID].[BuildID].[Count]”. (80 points)

제공된 파일을 HXD로 열어준다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

DOS코드와 NT헤더 사이에 hvxh’Rich’hvxh로 작성된 구간이 있다.

\*Rich 헤더는 PE 파일의 DOS와 NT 헤더 사이의 사용되지 않는 공간에 저장되는 데이터이다.

실행에 영향을 주지 않으며 프로그램이 빌드된 환경에 대한 정보가 저장되어 사이버 위협 인텔리전스에서 활용이 된다고 한다. ex) Kaspersky lab의 GReAT팀이 Rich Header 분석을 통해 공격 그룹을 밝혀낸 사례가 있음

이를 알아보기 위해 PE-Bear의 Rich 헤더 분석 기능을 사용해보았다.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

PE-bear라는 툴을 사용하였고, Rich Hdr란에서 [ProductID].[BuildIB].[Count] 값을 발견할 수 있었다.

|  |  |
| --- | --- |
| **FromBuildServer.exe** | **FromDeveloper.exe** |
| **[Utc1310\_C].[4035].[7]** | **[Utc1400\_C].[50727].[7]** |
| **[Implib710].[4035].[23]** | **[Implib800].[50727].[23]** |
| **[Import0].[0].[504]** | **[Import0].[0].[505]** |
| **[Masm800].[50727].[25]** | **[Masm900].[21022].[26]** |
| **[Utc1400\_C].[50727].[136]** | **[Utc1500\_C].[21022].[138]** |
| **[Utc1400\_CPP].[50727].[133]** | **[Utc1500\_CPP].[21022].[135]** |
| **[Utc1400\_LTCG\_CPP].[50727].[3]** | **[Utc1500\_LTCG\_CPP].[21022].[3]** |
| **[Cvtres800].[50727].[1]** | **[Cvtres900].[21022].[1]** |
| **[Linker800].[50727].[1]** | **[Linker900].[21022].[1]** |

2. Write the build folder paths for the given two executable files. (20 points)

\*PE 파일을 디버깅했을 때, 디버깅 정보에서 ‘빌드 폴더 경로’를 찾을 수 있으며, PDB 파일은 PE 파일에 대한 디버깅 관련 정보를 담고 있다.

1. FromBuildServer.exe

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

자동 생성된 설명

PE-bear에서 Debug 목록을 찾아 들어가면 경로를 찾을 수 있다.

**d:\BusinessDev\DFC\_Company\DFC2023\Release\DFC2023.pdb**

1. FromDeveloper.exe

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**C:\Users\dskm\Documents\Visual Studio 2008\Projects\DFC2023\Release\DFC2023.pdb**

**The deadline for this problem is July 31.**

**Please do not post your write-up before the deadline for fair competition!**

**\* Delete this box when submitting your answer.**