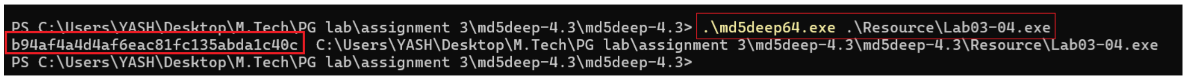
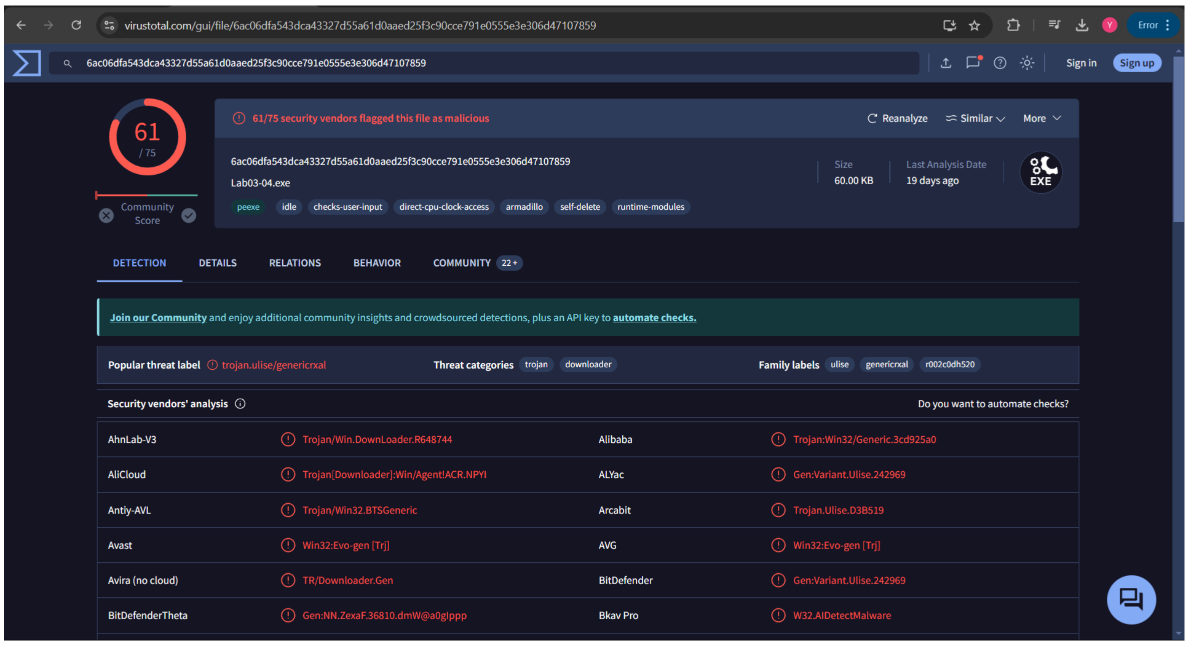
PG LAB assignment 3

1) Answer -



```
PS C:\Users\YASH\Desktop\M.Tech\PG lab\assignment 3\md5deep-4.3\md5deep-4.3> .\md5deep64.exe .\Resource\Lab03-04.exe
b94af4a4d4af6eac81fc135abda1c40c  C:\Users\YASH\Desktop\M.Tech\PG lab\assignment 3\md5deep-4.3\md5deep-4.3\Resource\Lab03-04.exe
PS C:\Users\YASH\Desktop\M.Tech\PG lab\assignment 3\md5deep-4.3\md5deep-4.3>
```



virustotal.com/gui/file/6ac06dfa543dca43327d55a61d0aaed25f3c90cce791e0555e3e306d47107859

61 / 75

61/75 security vendors flagged this file as malicious

6ac06dfa543dca43327d55a61d0aaed25f3c90cce791e0555e3e306d47107859

Lab03-04.exe

Size 60.00 KB

Last Analysis Date 19 days ago

peexe idle checks-user-input direct-cpu-clock-access armadillo self-delete runtime-modules

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

Popular threat label trojan.ulise/genericrxal — Threat categories trojan downloader — Family labels ulise genericrxal r002c0dh520

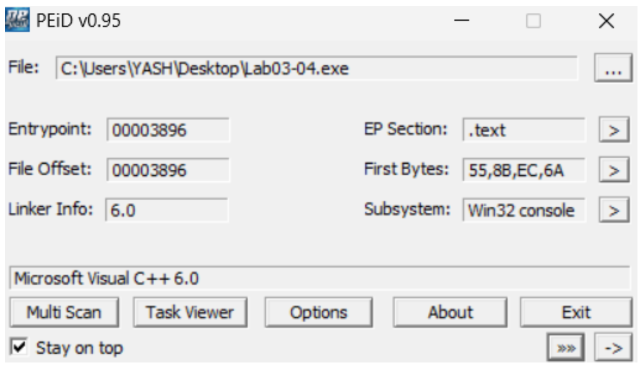
Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Trojan/Win.DownLoader.R648744	Alibaba	Trojan:Win32/Generic.3cd925a0
AliCloud	Trojan[Downloader]:Win/Agent!ACR.NPYI	ALYac	Gen:Variant.Ulise.242969
Antiy-AVL	Trojan/Win32.BTSGeneric	Arcabit	Trojan.Ulise.D3B519
Avast	Win32:Evo-gen [Trj]	AVG	Win32:Evo-gen [Trj]
Avira (no cloud)	TR/Downloader.Gen	BitDefender	Gen:Variant.Ulise.242969
BitDefenderTheta	Gen:NN.ZexaF.36810.dmW@a0glppp	Bkav Pro	W32.AIDetectMalware

As you can see in the screenshot I took by generating a hash() of Lab03-04.exe by using md5deep and pasting it into virustotal.com shows that the inputted file(Lab03-04.exe) is a virus labeled as **trojan** and **downloader** and is identified as a virus by 61 antivirus software out of 75. (hash value - b94af4a4d4af6eac81fc135abda1c40c)

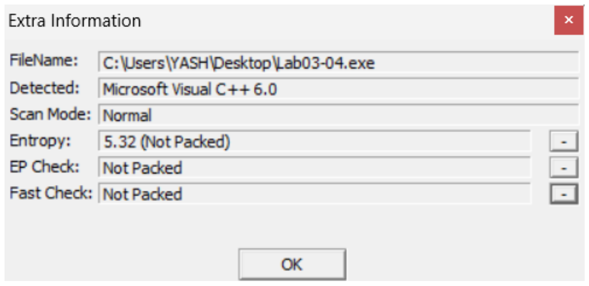
2) Answer -



PEiD v0.95

File:	C:\Users\YASH\Desktop\Lab03-04.exe		
Entrypoint:	00003896	EP Section:	.text
File Offset:	00003896	First Bytes:	55,8B,EC,6A
Linker Info:	6.0	Subsystem:	Win32 console

Microsoft Visual C++ 6.0



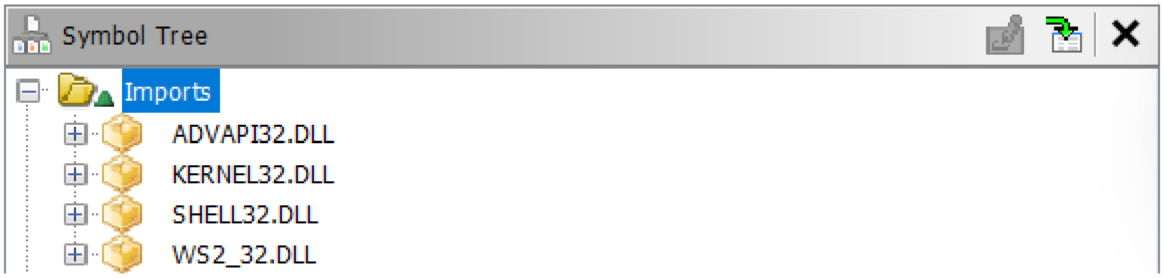
Extra Information

FileName:	C:\Users\YASH\Desktop\Lab03-04.exe
Detected:	Microsoft Visual C++ 6.0
Scan Mode:	Normal
Entropy:	5.32 (Not Packed)
EP Check:	Not Packed
Fast Check:	Not Packed

To check if the file is packed or obfuscated, I used the program called PEiD, which shows in the screenshot that the inputted file(Lab03-04.exe) is compiled by Microsoft Visual C++ 6.0, which is a typical program to compile the file from this screenshot. It does not look like this file is packed. The right-side screenshot mentions that the file(Lab03-04.exe) is not packed.

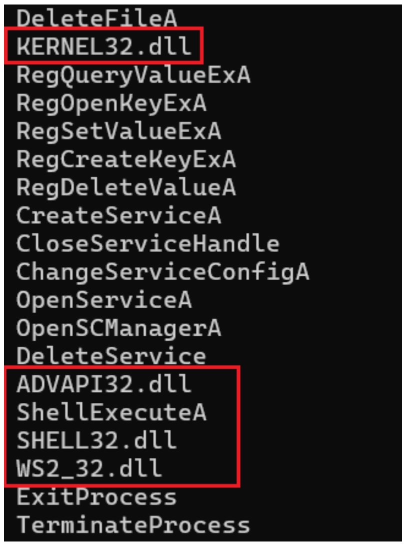
3) Answer -

In imports there are various .dll files like KERNEL32.DLL ADVAPI32.DLL SHELL32.DLL and WS2_32.DLL.



```
Symbol Tree
Imports
  ADVAPI32.DLL
  KERNEL32.DLL
  SHELL32.DLL
  WS2_32.DLL
```

I used Ghidra to get these imports but they can be viewed using strings command also.



```
DeleteFileA
KERNEL32.dll
RegQueryValueExA
RegOpenKeyExA
RegSetValueExA
RegCreateKeyExA
RegDeleteValueA
CreateServiceA
CloseServiceHandle
ChangeServiceConfigA
OpenServiceA
OpenSCManagerA
DeleteService
ADVAPI32.dll
ShellExecuteA
SHELL32.dll
WS2_32.dll
ExitProcess
TerminateProcess
```

KERNEL32.dll - Provides essential functions for the Windows operating system, including memory management, input/output operations, and process creation.

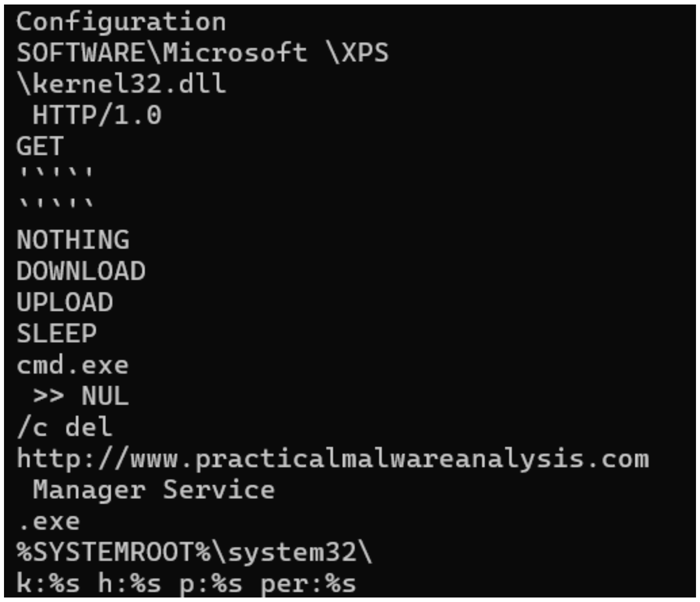
ADVAPI32.dll - Advanced API services related to Windows security, registry operations, and services management.

SHELL32.dll - Provides Windows Shell API functions, which allow interaction with the Windows user interface and file management.

WS2_32.dll - Provides networking capabilities using the Windows Sockets API, often called Winsock.

Looking at these import files, it seems this program can connect to the internet, manipulate files, manipulate the registry, and execute shell commands. This can be legitimate operations required by the program but can also be malware.

4) Answer -

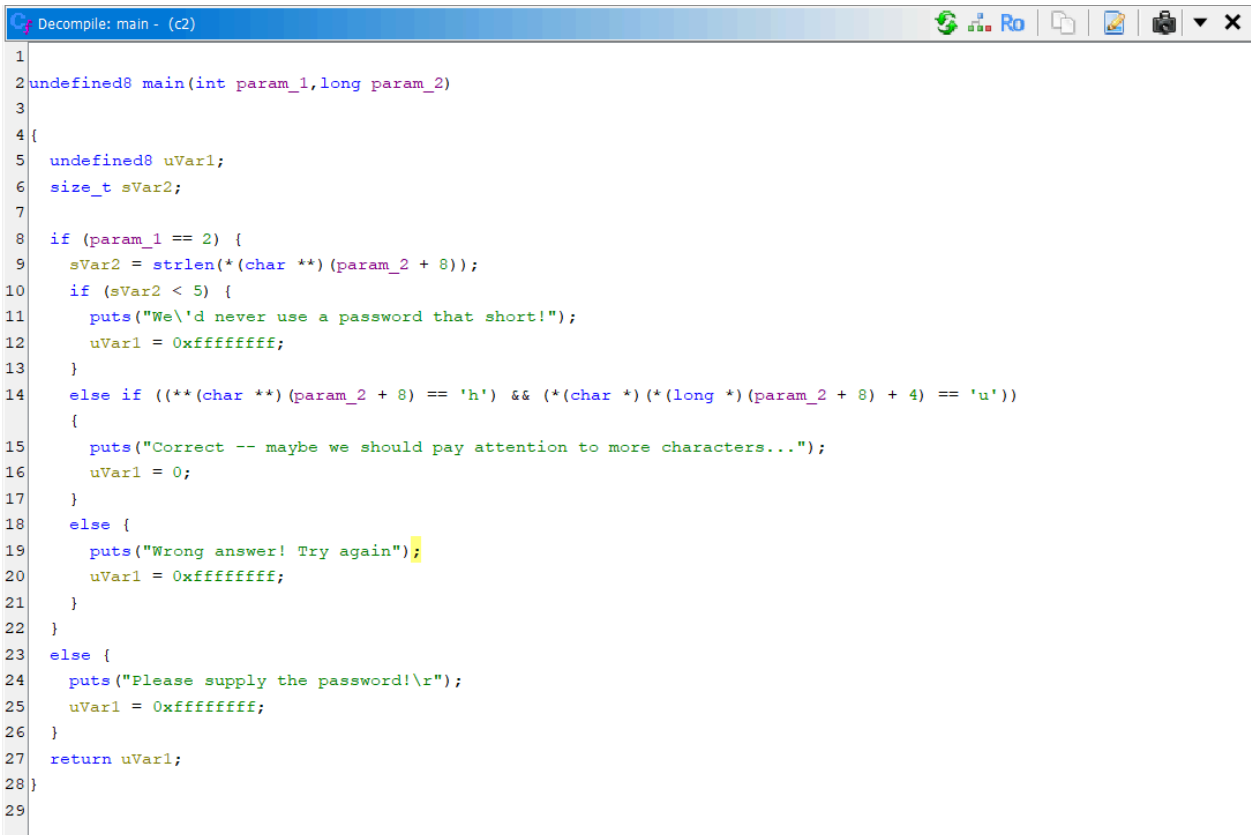


```
Configuration
SOFTWARE\Microsoft \XPS
\kernel32.dll
 HTTP/1.0
GET
'`'`'
`'`'`
NOTHING
DOWNLOAD
UPLOAD
SLEEP
cmd.exe
 >> NUL
/c del
http://www.practicalmalwareanalysis.com
 Manager Service
.exe
%SYSTEMROOT%\system32\
k:%s h:%s p:%s per:%s
```

By looking at the strings in Lab03-04.exe, I see "HTTP/1.0" and "GET" (maybe it is sending HTTP requests to malicious websites). Also, there is mention of the string "http://www.practicalmalwareanalysis.com" and as we have seen in part 3, this program has access to connect to internet maybe it is downloading or will download the files from internet or exchange the data with http://www.practicalmalwareanalysis.com this website. By looking at all the data, this program looks suspicious(have access to the internet, can change the registry, manipulate files, and execute shell commands)

GHIDRA Assignment:

1)For file c2 -

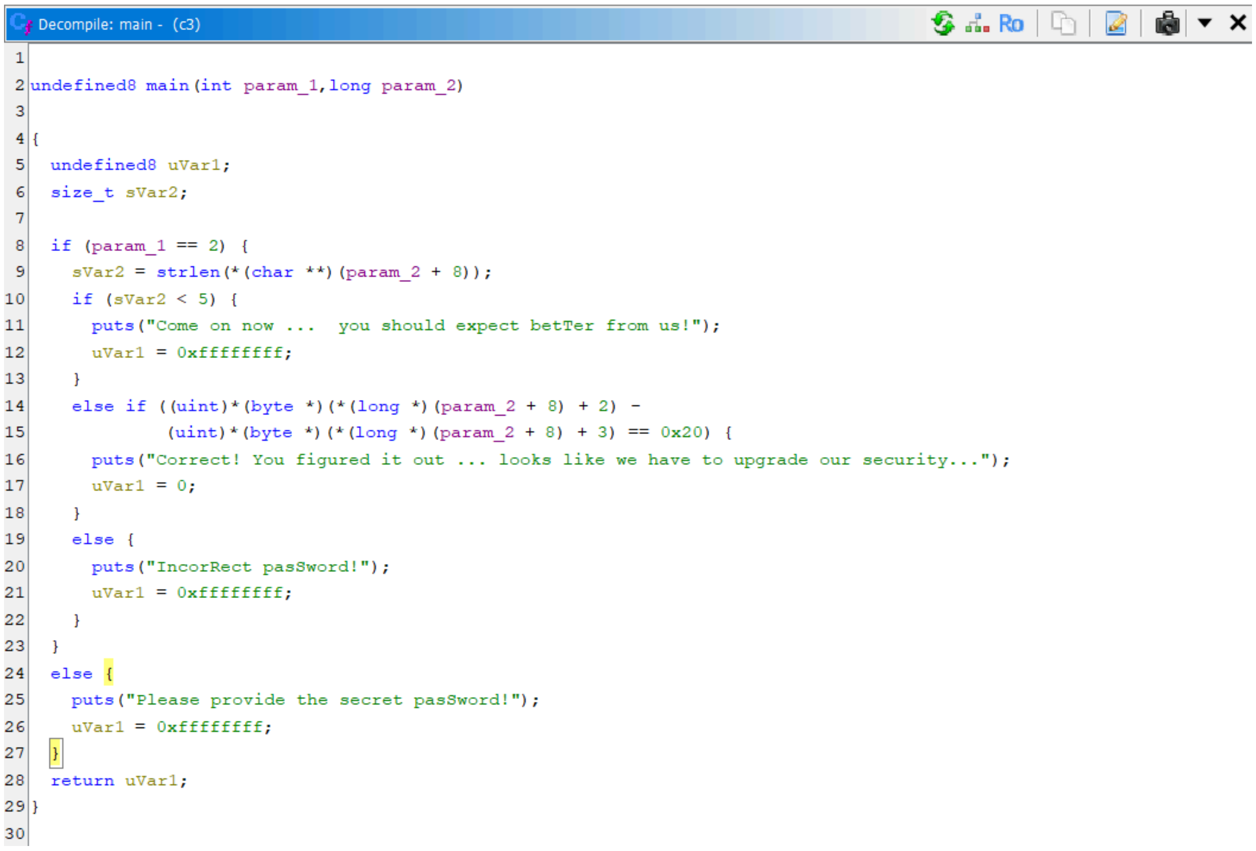


```
undefined8 main(int param_1,long param_2)

{
  undefined8 uVar1;
  size_t sVar2;

  if (param_1 == 2) {
    sVar2 = strlen(*(char **)(param_2 + 8));
    if (sVar2 < 5) {
      puts("We\'d never use a password that short!");
      uVar1 = 0xffffffff;
    }
    else if ((**(char **)(param_2 + 8) == 'h') && (*(char *)(*(long *)(param_2 + 8) + 4) == 'u'))
    {
      puts("Correct -- maybe we should pay attention to more characters...");
      uVar1 = 0;
    }
    else {
      puts("Wrong answer! Try again");
      uVar1 = 0xffffffff;
    }
  }
  else {
    puts("Please supply the password!\r");
    uVar1 = 0xffffffff;
  }
  return uVar1;
}
```

I used GHIDRA to import and analyze this file to extract the source code. And by looking at it, it looks like C code. main() takes two arguments as input, one integer and one long. The program checks if the password is provided or not using param_1; if not, then the program prints “Please supply the password” and returns 0xffffffff. If the password length is less than five, the program says, “never use a password that short!” and returns 0xffffffff. Else program checks if the first character of the string is ‘h’ and the fifth character is ‘u’ if both the conditions satisfy then the program prints “Correct – maybe we should pay attention to more characters…” and returns the value 0. Else if at least 1 of 2 conditions is not satisfied, then the program prints “Wrong answer! Try again” and returns 0xffffffff.

In brief, this program takes a password from a user whose length is greater than 5, and its first character is “h” and the fifth character is “u”.

2)For file c3 -

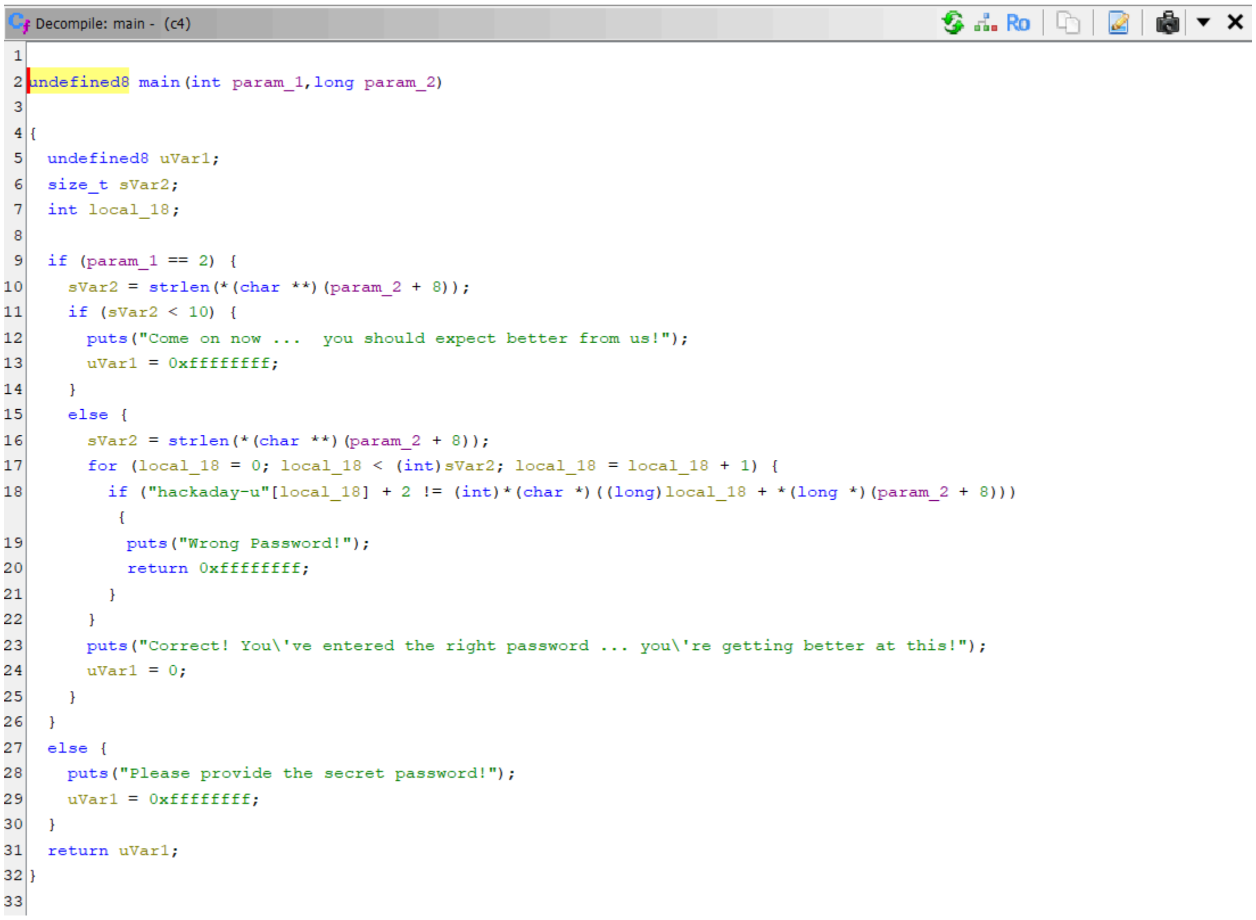


```

undefined8 main(int param_1,long param_2)

{
  undefined8 uVar1;
  size_t sVar2;

  if (param_1 == 2) {
    sVar2 = strlen(*(char **)(param_2 + 8));
    if (sVar2 < 5) {
      puts("Come on now ...  you should expect betTer from us!");
      uVar1 = 0xffffffff;
    }
    else if ((uint)*(byte *)(*(long *)(param_2 + 8) + 2) -
             (uint)*(byte *)(*(long *)(param_2 + 8) + 3) == 0x20) {
      puts("Correct! You figured it out ... looks like we have to upgrade our security...");
      uVar1 = 0;
    }
    else {
      puts("IncorRect pasSword!");
      uVar1 = 0xffffffff;
    }
  }
  else {
    puts("Please provide the secret pasSword!");
    uVar1 = 0xffffffff;
  }
  return uVar1;
}

```

The program checks if the password is provided in the terminal while executing the program; if not program prints “Please provide the secret pasSword!” and returns 0xffffffff; else, it checks the length of the provided password if the length is less than five then it prints “Come on now … you should expect betTer from us!” and returns 0xffffffff if the length is greater than five then it checks the difference between ASCII codes of third and fourth characters of the password and if it is equal to 0x20(32 in decimal) then it prints “Correct! You figured it out … looks like we have to upgrade our security…” and returns 0. If it is not equal, then the program prints “IncorRect pasSword” and returns 0xffffffff.

In brief, this program takes a password from a user whose length is greater than 5 and whose ASCII difference between the third and fourth characters is 0x20(32 in decimal).

2)For file c4 -



```

undefined8 main(int param_1,long param_2)

{
  undefined8 uVar1;
  size_t sVar2;
  int local_18;

  if (param_1 == 2) {
    sVar2 = strlen(*(char **)(param_2 + 8));
    if (sVar2 < 10) {
      puts("Come on now ...  you should expect better from us!");
      uVar1 = 0xffffffff;
    }
    else {
      sVar2 = strlen(*(char **)(param_2 + 8));
      for (local_18 = 0; local_18 < (int)sVar2; local_18 = local_18 + 1) {
        if ("hackaday-u"[local_18] + 2 != (int)*(char *)((long)local_18 + *(long *)(param_2 + 8)))
         {
          puts("Wrong Password!");
          return 0xffffffff;
        }
      }
      puts("Correct! You\'ve entered the right password ... you\'re getting better at this!");
      uVar1 = 0;
    }
  }
  else {
    puts("Please provide the secret password!");
    uVar1 = 0xffffffff;
  }
  return uVar1;
}

```

The program checks if the password is provided in the terminal while executing the program; if not program prints "Please provide the secret password!" and returns 0xffffffff; else, it checks the length of the provided password if the length is less than 10, then it prints "Come on now … you should expect better from us!" and returns 0xffffffff else if the length is greater than 10 then it compares the user entered password with the string "hackaday-u"(by looking at code it seems like it is adding 2 in ASCII code of each character of string "hackaday-u" and then comparing it with user inputted password's ASCII values so it is comparing user-inputted password with the string "jcemcfc{/w" but I am not sure). If equal, then the program prints "Correct! You've entered the right password … you're getting better at this!" and returns 0; if not equal, it prints "Wrong Password!" and returns 0xffffffff.

In brief, this program takes a password as input from the user whose length is greater than 10. It compares it with the string “hackaday-u” with the string “jcemcfc{/w” and accepts the password if the comparison is successful.