Robin Ward
COMP 7720/7726/4970
Summer 2018
7/24/2018
Auburn University

1.
```
.data:0040511B ; char a?dontuseme[]
.data:0040511B a?dontuseme     db '?dontuseme',0        ; DATA XREF: sub_403016:loc_40308B↑
```

   a. The dontuseme command is implemented at .text:0040308B. I posted an image below for clarification

```
.text:0040308B
.text:0040308B loc_40308B:                             ; CODE XREF: sub_403016+4F↑j
.text:0040308B                 push    offset a?dontuseme ; "?dontuseme"
.text:00403090                 lea     eax, [ebp+Dst]
.text:00403096                 push    eax             ; char *
.text:00403097                 call    strcmp
.text:0040309C                 add     esp, 8
.text:0040309F                 or      eax, eax
.text:004030A1                 jnz     short loc_4030B2
.text:004030A3                 call    sub_401AA0
.text:004030A8                 mov     eax, 3
.text:004030AD                 jmp     loc_403640
```

   b.
      i. The first thing that happens, the program has to look for calls to string comparison functions and determine the string being compared
      ii. The string is then compares against **text:00403090 lea eax, [ebp+Dst]**
      iii. If the call passes, it will jump to loc_4030b2 which is the quit command, otherwise is will jump to loc_403640 after calling the 00401AA0, which creates the bat file

   c. If !?dontuseme then execute function at 00401AA0

```
.text:00401AAC                 push    1
.text:00401AAE                 call    sub_4019CB
.text:00401AB3                 push    ds:hMutex       ; hMutex
.text:00401AB9                 call    ReleaseMutex
.text:00401ABE                 lea     eax, [ebp+File]
.text:00401AC4                 push    eax             ; lpBuffer
.text:00401AC5                 push    104h            ; nBufferLength
.text:00401ACA                 call    GetTempPathA
.text:00401ACF                 push    offset aRm_bat  ; "rm.bat"
.text:00401AD4                 lea     eax, [ebp+File]
.text:00401ADA                 push    eax             ; char *
.text:00401ADB                 call    strcat
.text:00401AE0                 lea     eax, [ebp+File]
.text:00401AE6                 push    eax
.text:00401AE7                 push    offset Filename
.text:00401AEC                 push    offset Filename
```

   Else execute quit command

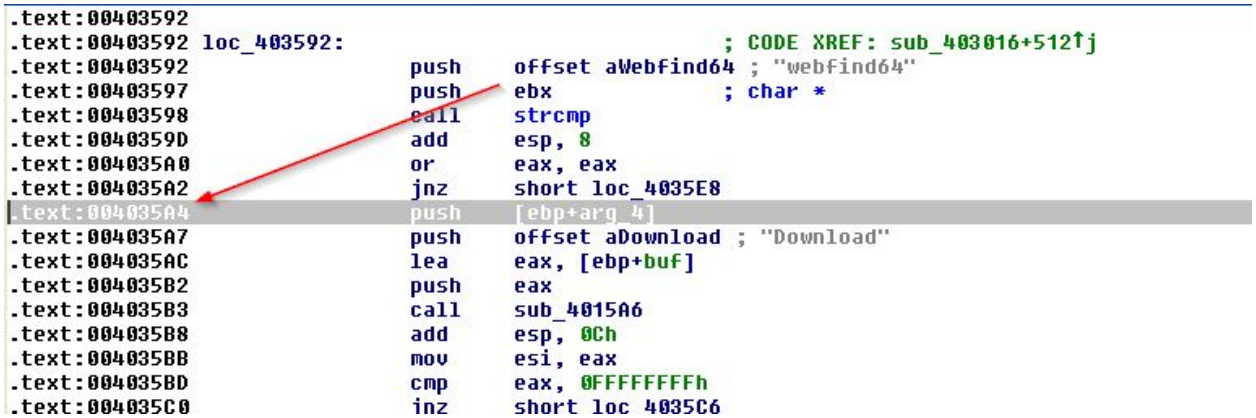
2. A. The webfind64 command is implemented at .text:00403592. I posted an image below for clarification

```
.text:00403592
.text:00403592 loc_403592:                             ; CODE XREF: sub_403016+512↑j
.text:00403592                 push    offset aWebfind64 ; "webfind64"
.text:00403597                 push    ebx               ; char *
.text:00403598                 call    strcmp
.text:0040359D                 add     esp, 8
.text:004035A0                 or      eax, eax
.text:004035A2                 jnz     short loc_4035E8
.text:004035A4                 push    [ebp+arg_4]
.text:004035A7                 push    offset aDownload ; "Download"
.text:004035AC                 lea     eax, [ebp+buf]
.text:004035B2                 push    eax
.text:004035B3                 call    sub_4015A6
.text:004035B8                 add     esp, 0Ch
.text:004035BB                 mov     esi, eax
.text:004035BD                 cmp     eax, 0FFFFFFFFh
.text:004035C0                 jnz     short loc_4035C6
.text:004035C2                 xor     eax, eax
.text:004035C4                 jmp     short loc_403640
```

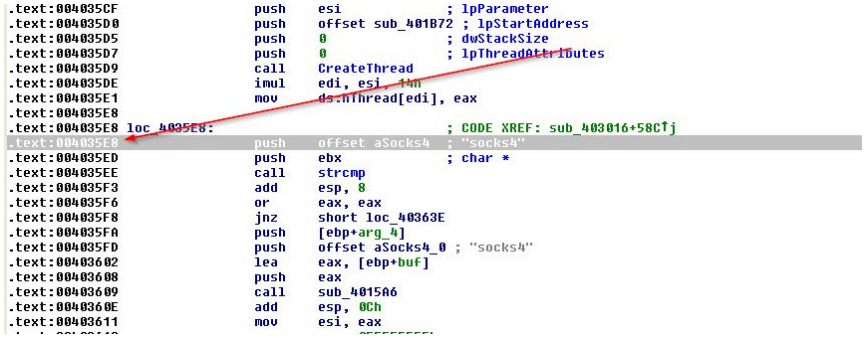
B.

i. this command instructs the host to download a file from a remote server. The webfind command is located at **.text:00403592 loc_403592:**

ii. If the command moves to the download section, it will jump to 4035A4, otherwise it will jump to 4035E8. This image below is the start of the download section



This image below is the start of the socks4 section



iii. from the download section it will move into the tread section, assuming it doesn't fail and go to address 403640

iv. From loc_4035C6, we are setting the thread id and then creating a new thread before moving to socks4

```
.text:004035C6 ; ---------------------------------------------------------------------------
.text:004035C6
.text:004035C6 loc_4035C6:                             ; CODE XREF: sub_403016+5AA↑j
.text:004035C6                 lea     eax, [ebp+ThreadId]
.text:004035CC                 push    eax             ; lpThreadId
.text:004035CD                 push    0               ; dwCreationFlags
.text:004035CF                 push    esi             ; lpParameter
.text:004035D0                 push    offset sub_401B72 ; lpStartAddress
.text:004035D5                 push    0               ; dwStackSize
.text:004035D7                 push    0               ; lpThreadAttributes
.text:004035D9                 call    CreateThread
.text:004035DE                 imul    edi, esi, 14h
.text:004035E1                 mov     ds:hThread[edi], eax
.text:004035E8
.text:004035E8 loc_4035E8:                             ; CODE XREF: sub_403016+58C↑j
.text:004035E8                 push    offset aSocks4  ; "socks4"
.text:004035ED                 push    ebx             ; char *
.text:004035EE                 call    strcmp
.text:004035F3                 add     esp, 8
.text:004035F6                 or      eax, eax
.text:004035F8                 jnz     short loc_40363E
.text:004035FA                 push    [ebp+arg_4]
```

C.

```
if (webfind64){
        if download{
            if (createthread){call asocks function}
            else return;
                }
        Else{
            call asocks4 function
                }
    }
else{return}
```

```
asocks function
if needanotherthread { jump to loc_40361C}
else return;
```