Computer Science 491/691
Malware Analysis
Homework 3
Assigned: March 4, 2019
Due: March 11, 2019

Submitted by
Name: Sai Sasaank Srivatsa Pallerla
ID: HG13015

How to turn this in for grading: You can edit your answers right into this file. Email it to the TAs as described in class. Make sure your name appears in the body of the document.

Hint: Chapters 4 and 6 of your Practical Malware Analysis textbook are very useful references!

Part 1

```
start:
      PUSH  EBP
      MOV   EBP, ESP
      MOV   EDI, [EBP+arg_0]
      XOR   EAX, EAX
      MOV   ECX, 0xFFFFFFFF // ECX = -1 || the register is filled with 1's -VE value
      REPNE SCASB // Search EDI for 0 || Finds the length of arg_0
      NEG   ECX // ECX = 1 || +VE value
      MOV   EAX, ECX
      MOV   ESI, [EBP+arg_0]
      MOV   EDI, [EBP+arg_4]
      REP MOVSB // Copy ESI to EDI
      MOV   ESP, EBP
      POP   EBP
      RETN // Returns the value stored in EAX – length of the string
```

1.1) In a few sentences, explain what this function does.

In short this is the assembly code that copies copy one string to another (arg_0 to arg_4) it also has the length of the string stored in EAX register.. The value that the function returns is copy of the string (arg_4).

1.2) Write a function in C that is equivalent to the assembly above.

```
int func1(char* arg_0, char* arg_4) {
        while (   ECX !=0  ) // REPNE SCASB;
        {
                if( EDI == 0 ) // byte in AL = 0
                ZF = 0;
                if( DF == 0 )
                        EDI++;
                else
                        EDI--;
                ECX--;
                if( ZF == 0 ) break; // if zero is encountered in EDI, we exit while loop, we get the length of the string because 0 typically indicates null terminator.
        }
        EDI = ESI; // copy arg_0 to arg_4
        EAX = ECX;
}

int func1(char* arg_0, char* arg_4) {
        arg_4 = arg_0;
}
```

1.3) Let arg_0 be a pointer to the null-terminated string "C:\Windows\System32\" and let arg_4 be a pointer to an empty buffer.

What is the value of the buffer pointed to by arg_4 when the function completes? What value does the function return?

arg_4 points to the null terminator. The function returns copy of the string, which is "C:\Windows\System32\"

Part 2

```
start:
      PUSH  EBP
      MOV   EBP, ESP
      MOV   ECX, [EBP+arg_0] // *arg_0 character pointer
      MOV   ESI, [EBP+arg_4] // arg_4 integer
      MOV   [EBP+var_1], 0 // var_1=0 possible integer
      JMP   loc_2
loc_1:
      MOV   EAX, [EBP+var_1] // defines the position
      ADD   EAX, ECX // moves to the required position
      MOV   EDX, byte ptr [EAX] // EDX acts as a pointer to char at EAX
      XOR   EDX, ESI
      MOV   [EAX], DL
      ADD   [EBP+var_1], 0x1 // var_1++;
loc_2:
      MOV   EAX, [EBP+var_1]
      CMP   byte ptr [ECX + EAX], 0 // while condition
      JNZ   loc_1
      MOV   ESP, EBP
      POP   EBP
      RETN
```

2.1) In a few sentences, explain what this function does.

The function simply XORs the given string (arg_0) with the given integer value (arg_4) and converts the string to something meaningful that the malware can use to perform specific functions.

2.2) Write a function in C that is equivalent to the assembly above.

```
int func2(char* arg_0, int arg_4) {
        int var_1;
        while( [ECX+EAX] != 0 )
                {
                EDX = arg_0[var_1];
                EDX = EDX^arg_4;
                var_1++;
                }
}
```

2.3) Let arg_0 be a pointer to the null-terminated string "\xa7\xa4\xe2\xaf\xcf\xd2\xc6\xf1\xe1\xe3\xf3\xcc\xaf\xd8\xef\xdb\xf1\xe1\xa3\xa7\xaf\xe6\xa1\xd7\xd7\xae\xff\xe5\xfc\xe4\xa0\xc5\xdb\xe1" and let arg_4 be the integer 0x96.

What is the value of the string pointed to by arg_0 when the function completes? What value does the function return?

The value pointed by arg_0 is a null pointer. The value that the function return is 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw.

Part 3

```
start:
      PUSH  EBP
      MOV   EBP, ESP
      MOV   EDX, [EBP+arg_0] // character pointer
      MOV   ESI, [EBP+arg_4] // character pointer
      MOV   EDI, 0x1A // EDI = 26
      XOR   EAX, EAX // EAX = 0
loc_1:
      MOV   ECX, byte ptr [EDX]
      MOV   EBX, byte ptr [ESI]
      SUB   ECX, EBX // difference between arg_0 and arg_4
      SUB   ECX, EAX
      MOV   [EDX], cx
      SHR   ECX, 0x10 // clear the left side of the register
      AND   ECX, 0x1
      MOV   EAX, ECX
      INC   EDX // EDX++
      INC   ESI // ESI++
      DEC   EDI // DEC--
      TEST  EDI, EDI // EDI!=0
      JNZ loc_1
loc_2:
      XOR   EAX, EAX
      MOV   ESP, EBP
      POP   EBP
      RETN
```

3.1) In a few sentences, explain what this function does.

The function take two strings as an input (arg_0 and arg_4) and find the difference between them and returns a meaningful string that the malware will use to preform/ connect to a function/URL. It is converting the hex string (arg_0) to a char string with another string (arg_4) and key to decipher it.

3.2) Write a function in C that is equivalent to the assembly above.

```
int func3(char* arg_0, char* arg_4) {
        char* result;
        int i = 26, j=0;
        while ( i !=0) {
                result[j] = arg_0[j] - arg_4[j];
                j++;
                i--;
        }
}
```

3.3) Let arg_0 be a pointer to the null-terminated string "\xe8\xee\xdc\xa0\xe1\xda\xe1\xcd\xde\xdd\xc2\xdc\xd2\x94\xca\xd7\xd7\x9a\xd0\xe9\xef\xd1\xe2\xd1\xcf\xd1\x0a" and let arg_4 be a pointer to the null-terminated string "qwertyuiopasdfghjklzxcvbnm".

What is the value of the string pointed to by arg_0 when the function completes?

The value pointed by arg_0 is "x0a". The function returns the value www.maldomain.com/download

** I made efforts in converting the assembly to C and the C program not the exact equivalent but it mimics what is happening in the assembly program.