|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Storage value | Storage address | Storage address | Storage address | Storage address |  |
| [XXXXXXXX] =address of reg.  yyyyyyyyy = value of reg. | EAX = 8 | ECX  = 0019F770  =[**EF8B8477**] | ESP  = 0019F4BC  = [**0019FE38**] | EBP  = 0019F79C  =[**0019F4A4**] | ESI 16bit  = 0000000A  =[] |
| MOVE EAX, [EBP – 4] | 0 | // | // | // | // |
| PUSH EAX | 0 |  | 0019F744  =[00000000] |  |  |
| LEA ESI, [EBP + EAX – 2C]   * Load value from reg. with address [] to reg. ESI | 0 |  |  |  | 0019F770  = [**42384645**]  [xxxx xxxx]  hex  = [**EF8B||8477**] asc |
| MOVSX EAX, BYTE PTR [EBP + EAX – 2C]   * Load last 2 byte of value(hex) from address [] to reg. EAX | 45 |  |  |  |  |
| PUSH EAX   * Push value from EAX to top of stack | 0 | 0019F770 | 0019F744  = [45] | 0019F79C | 0019F770 |
| CALL 1\_2.00401247 |  |  | 0019F740  =[0040142E]  Storage address to RET |  |  |
| **MOV EAX, [ESP + 4]** | 0 | 0019F770 |  | 0019F79C | 0019F770 |
| **MOV ECX, EAX** | 0 | 0 |  |  |  |
| **SHL ECX, 2** |  | 0 |  |  |  |
| **MOV EAX, [ECX + 4032E8]**   * **4032E8 = [D1310BA6]** | D1310BA6 |  |  |  |  |
| **SUB EAX, [ECX + 404020]**   * **404020 = [0]** | D1310BA6 |  |  |  |  |
| **XOR EAX, [ESP + 4]**   * **ESP + 4 = 0019F774 = [45] = EAX before** | D1310BE3 |  |  |  |  |
| **RET** |  |  | 0019F744 = [45] |  |  |
| INC DWORD PTR [EBP – 4]   * EBD – 4 = 0019F798   = [0]  Inc -> [1] |  |  |  |  |  |
| MOV BYTE PTR [ESI], AL   * AL: last 8bit of EAX = [E3] * Put after last 8bit of value storage in reg. ESI | D1310BE3 |  |  |  | 0019F770  =[423846E3]  From **42384645**  = **[EF8B||8477]**  To **423846E3**  **=[F8B8||8477]** |
| LEA EAX, [EBP – 2C]   * EAX = EBP -2C | 0019770 |  |  |  |  |
| PUSH EAX | 0019F770 | 0 | 0019F740  =[0019F770] | 0019F79C | 0019F770 |
| CALL <JMP…..> | 8 | 0019F770 |  |  |  |
| ADD ESP, 0C | 8 |  | 0019F74C |  |  |
| CMP [EBP-4], EAX   * EBP – 4 = 0019F798   = [00000001] != [EAX] | 8 |  |  |  |  |
| JB SHORT 1\_2.0040141B |  |  |  |  |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Storage value | Storage address | Storage address | Storage address | Storage address |  |
| [XXXXXXXX] =address of reg.  yyyyyyyyy = value of reg. | EAX = 8 | ECX  = 0019F770  =[**EF8B8477**] | ESP  = 0019F74C  = [**0019FE38**] | EBP  = 0019F79C  =[**0019F7A4**] | ESI 16bit  = 0019F770  = |
| MOVE EAX, [EBP – 4] | 1 | // | // | // | // |
| PUSH EAX | 0 |  | 0019F748  =[00000001] |  |  |
| LEA ESI, [EBP + EAX – 2C]   * Load value from reg. with address [] to reg. ESI | 0 |  |  |  | 0019F771  = [**423846xx**] hex  xx vì chỉ lấy từ byte thứ 1, bỏ byte 0 nên E3 sẽ được cất đi  = [**F8B||8477**] asc |
| MOVSX EAX, BYTE PTR [EBP + EAX – 2C]   * Load last 2 byte of value(hex) from address [] to reg. EAX | 46 |  |  |  |  |
| PUSH EAX   * Push value from EAX to top of stack | 0 | 0019F770 | 0019F744  = [45] | 0019F79C | 0019F771   * Với lần chạy đầu tiên thi ta có string F8B8 8477 * Lần sau là F8B 8447 |
| CALL 1\_2.00401247 |  |  | 0019F740  =[0040142E]  Storage address to RET |  |  |
| **MOV EAX, [ESP + 8]** | 0 | 0019F770 | 0019F740 | 0019F79C | 0019F771 |
| **MOV ECX, EAX** | 1 | 0 |  |  |  |
| **SHL ECX, 2** | 1 | 1 |  |  |  |
| **MOV EAX, [ECX + 4032E8]**   * **4032E8 = [98DFB5AC]** | 98DFB5AC | 4 |  |  |  |
| **SUB EAX [ECX + 404020]**   * **404020 = [77073096]** | 21D88516 |  |  |  |  |
| **XOR EAX, [ESP + 4]**   * **ESP + 4 = 0019F774 = [46] = EAX before** | 21D88550 |  |  |  |  |
| **RET** |  |  | 0019F744 = [45] |  |  |
| INC DWORD PTR [EBP – 4]   * EBD – 4 = 0019F798   = [0]  Inc -> [1] |  |  |  |  |  |
| MOV BYTE PTR [ESI], AL   * AL: last 8bit of EAX = [E3] * Put after last 8bit of value storage in reg. ESI | D1310BE3 |  |  |  | 0019F770  =[423846E3]  From **42385045**  = **[EF8B||8477]**  To **423846E3**  **=[F8B8||8477]** |
| LEA EAX, [EBP – 2C]   * EAX = EBP -2C | 0019770 |  |  |  |  |
| PUSH EAX | 0019F770 | 0 | 0019F740  =[0019F770] | 0019F79C | 0019F770 |
| CALL <JMP…..> | 8 | 0019F770 |  |  |  |
| ADD ESP, 0C | 8 |  | 0019F74C |  |  |
| CMP [EBP-4], EAX   * EBP – 4 = 0019F798   = [00000001] != [EAX] | 8 |  |  |  |  |
| JB SHORT 1\_2.0040141B |  |  |  |  |  |