Hyara
Generator for Yara rules

선린인터넷고등학교 이현

https://github.com/hy00un/Hyara

INDEX

1

제작배경

제작배경

- 작년 비오비 수업시간에 Yara rule과 IDAPython을 배움
  - 다른 IDA Plugin들을 보면서 나도 한 번 만들고 싶어짐

- Yara rule을 제작할 때 유니크한 스트링이 수백 개가 존재 할 경우
  rule 제작에 번거로움이 존재함

- 타 도구를 사용해본 결과, 유의미한 rule을 제작해주지 못함
  분석가 입장에서 분석한 데이터를 기반으로 쉽게 제작하기 위한 도구 제작

- IDA 플러그인 형식으로 제작된 Yara 도구가 많지 않음

2

Yara의 정의 및 활용성

Yara의 정의 및 활용성

Yara 소개

- 구글이 인수한 virustotal에서 만든 유사도 탐지 도구
- Strings에서 제작한 rule을 기반으로 condition에서 bool형으로 True 또는 False 반환

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        thread_level = 3
        in_the_wild = true

    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

    condition:
        $a or $b or $c
}
```

출처 : virustotal.github.io

Yara의 정의 및 활용성

Yara 소개

- 구글이 인수한 virustotal에서 만든 유사도 탐지 도구
- Strings에서 제작한 rule을 기반으로 condition에서 bool형으로 True 또는 False 반환

rule에 대하여
설명하기 위한 정보

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        thread_level = 3
        in_the_wild = true

    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

    condition:
        $a or $b or $c
}
```

출처 : virustotal.github.io

Yara의 정의 및 활용성

Yara 소개

- 구글이 인수한 virustotal에서 만든 유사도 탐지 도구
- Strings에서 제작한 rule을 기반으로 condition에서 bool형으로 True 또는 False 반환

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        thread_level = 3
        in_the_wild = true

    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

    condition:
        $a or $b or $c
}
```

rule에서 탐지 할 데이터를 저장

출처 : virustotal.github.io

Yara의 정의 및 활용성

Yara 소개

- 구글이 인수한 virustotal에서 만든 유사도 탐지 도구
- Strings에서 제작한 rule을 기반으로 condition에서 bool형으로 True 또는 False 반환

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        thread_level = 3
        in_the_wild = true

    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

    condition:
        $a or $b or $c
}
```

데이터를 어떠한 조건으로 탐지할지 지정

출처 : virustotal.github.io

Yara의 정의 및 활용성

Yara 세부 option

Rule 옵션	설명
wide	한 글자를 2바이트로 읽는 문자열을 검색할 때 사용
ascii	UTF-16이 포함된 문자열이 있는 경우 해당 옵션을 사용
nocase	대소문자 구분 없이 탐지를 허용
fullword	탐지하려는 문자열 사이에 이상한 문자열이 포함되어 있는 경우 탐지 X

Ex) $a = “test” nocase ascii wide (대소문자 구분 X, utf-16 문자열 탐지, 2바이트 문자열 탐지)

바이너리는 동일한 기능을 하더라도 다른 어셈 코드를 사용 할 가능성이 있음
와일드 카드 기능을 이용하여 rule의 탐지 범위를 넓이는 작업을 해야 좋은 rule이 될 수 있음

Ex) push eax = 0x50, push edx = 0x52 push ebx = 0x53
-> $a = { 5? } (push eax, edx, ebx 모두 탐지 가능)

Yara의 정의 및 활용성

Yara For Loop 기능

```
BE 20 37 EF C6                          mov     esi, 0C6EF3720h
89 4D F8                                mov     [ebp+var_8], ecx
89 45 FC                                mov     [ebp+var_4], eax
C7 45 0C 20 00 00 00                    mov     [ebp+arg_4], 20h

                        loc_40141E:                             ; COD
6A 00                                   push    0               ; lpS
FF 15 4C 40 41 00                       call    AddAtomW
FF 15 20 40 41 00                       call    GetLastError
FF 15 34 40 41 00                       call    GetTickCount
8B CF                                   mov     ecx, edi
8B C7                                   mov     eax, edi
C1 E9 05                                shr     ecx, 5
03 4D FC                                add     ecx, [ebp+var_4]
C1 E0 04                                shl     eax, 4
03 45 F8                                add     eax, [ebp+var_8]
33 C8                                   xor     ecx, eax
8D 04 3E                                lea     eax, [esi+edi]
33 C8                                   xor     ecx, eax
2B D9                                   sub     ebx, ecx
8B CB                                   mov     ecx, ebx
8B C3                                   mov     eax, ebx
C1 E9 05                                shr     ecx, 5
03 4D F4                                add     ecx, [ebp+var_C]
C1 E0 04                                shl     eax, 4
03 45 F0                                add     eax, [ebp+var_10]
33 C8                                   xor     ecx, eax
8D 04 1E                                lea     eax, [esi+ebx]
33 C8                                   xor     ecx, eax
8D B6 47 86 C8 61                       lea     esi, [esi+61C88647h]
```

Assemble code

```
v7 = *a2;
v2 = a1[1];
v8 = a2[1];
v3 = *a1;
v4 = 0xC6EF3720;
v9 = a2[2];
v10 = a2[3];
v11 = 32;
do
{
  AddAtomW(0);
  GetLastError();
  GetTickCount();
  v2 -= (v4 + v3) ^ (v9 + 16 * v3) ^ (v10 + (v3 >> 5));
  result = v4 + v2;
  v6 = (v4 + v2) ^ (v7 + 16 * v2) ^ (v8 + (v2 >> 5));
  v4 += 0x61C88647;
  v3 -= v6;
  --v11;
}
while ( v11 );
*a1 = v3;
a1[1] = v2;
return result;
```

pseudocode

Yara의 정의 및 활용성

Yara For Loop 기능

```
BE 20 37 EF C6                  mov     esi, 0C6EF3720h
89 4D F8                        mov     [ebp+var_8], ecx
89 45 FC                        mov     [ebp+var_4], eax
C7 45 0C 20 00 00 00            mov     [ebp+arg_4], 20h

                loc_40141E:                             ; COD
6A 00                           push    0               ; lpS
FF 15 4C 40 41 00               call    AddAtomW
FF 15 20 40 41 00               call    GetLastError
FF 15 34 40 41 00               call    GetTickCount
8B CF                           mov     ecx, edi
8B C7                           mov     eax, edi
C1 E9 05                        shr     ecx, 5
03 4D FC                        add     ecx, [ebp+var_4]
C1 E                            shl     eax, 4
03 45 F8                        add     eax, [ebp+var_8]
33 C8                           xor     ecx, eax
8D 04 3E                        lea     eax, [esi+edi]
33 C8                           xor     ecx, eax
2B D9                           sub     ebx, ecx
8B CB                           mov     ecx, ebx
8B C3                           mov     eax, ebx
C1 E9 05                        shr     ecx, 5
03 4D F4                        add     ecx, [ebp+var_C]
C1 E0 04                        shl     eax, 4
03 45 F0                        add     eax, [ebp+var_10]
33 C8                           xor     ecx, eax
8D 04 1E                        lea     eax, [esi+ebx]
33 C8                           xor     ecx, eax
8D B6 47 86 C8 61               lea     esi, [esi+61C88647h]
```

92 Bytes 간격 존재

Assemble code

```
v7 = *a2;
v2 = a1[1];
v8 = a2[1];
v3 = *a1;
v4 = 0xC6EF3720;
v9 = a2[2];
v10 = a2[3];
v11 = 32;
do
{
  AddAtomW(0);
  GetLastError();
  GetTickCount();
  v2 -= (v4 + v3) ^ (v9 + 16 * v3) ^ (v10 + (v3 >> 5));
  result = v4 + v2;
  v6 = (v4 + v2) ^ (v7 + 16 * v2) ^ (v8 + (v2 >> 5));
  v4 += 0x61C88647;
  v3 -= v6;
  --v11;
}
while ( v11 );
*a1 = v3;
a1[1] = v2;
return result;
```

pseudocode

Yara의 정의 및 활용성

Yara For Loop 기능

```
BE 20 37 EF C6                  mov     esi, 0C6EF3720h
89 4D F8                        mov     [ebp+var_8], ecx
89 45 FC                        mov     [ebp+var_4], eax
C7 45 0C 20 00 00 00            mov     [ebp+arg_4], 20h

                        loc_40141E:                             ; COD
6A 00                           push    0                       ; lpS
FF 15 4C 40 41 00               call    AddAtomW
FF 15 20 40 41 00               call    GetLastError
FF 15 34 40 41 00               call    GetTickCount
8B CF                           mov     ecx, edi
8B C7                           mov     eax, edi
C1 E9 05                        shr     ecx, 5
03 4D FC                        add     ecx, [ebp+var_4]
C1 E                            shl     eax, 4
03 45 F8                        add     eax, [ebp+var_8]
33 C8                           xor     ecx, eax
8D 04 3E                        lea     eax, [esi+edi]
33 C8                           xor     ecx, eax
2B D9                           sub     ebx, ecx
8B CB                           mov     ecx, ebx
8B C3                           mov     eax, ebx
C1 E9 05                        shr     ecx, 5
03 4D F4                        add     ecx, [ebp+var_C]
C1 E0 04                        shl     eax, 4
03 45 F0                        add     eax, [ebp+var_10]
33 C8                           xor     ecx, eax
8D 04 1E                        lea     eax, [esi+ebx]
33 C8                           xor     ecx, eax
8D B6 47 86 C8 61               lea     esi, [esi+61C88647h]
```

92 Bytes 간격 존재

Assemble code

```
v7 = *a2;
v2 = a1[1];
v8 = a2[1];
v3 = *a1;
```

```
rule TEA_algorithm : TEA
{
    strings:
        $a = {?? 20 37 EF C6} // v2 = 0xC6EF3720
        $b = {8D ?? 47 86 C8 61} // v2 += 0x61C88647
        /*
            shr ecx, 5
            add ecx, [ebp+var_10]
            shl eax, 4
        */
        $c = {C1 E? 05 03 4D ?? C1 E? 04}
    condition:
        for all i in (1..#a) : ($b in (@a[i]-128..@a[i]+128)) and $c
}
```

```
while ( v11 );
*a1 = v3;
a1[1] = v2;
return result;
```

pseudocode

3

제작 도구 소개 및 타 도구와의 차별성

제작 도구 소개 및 타 도구와의 차별성

yarGen – Florian Roth

- 973MB의 sqlite 데이터와 하나하나 비교하여 매칭시키는 방식
- Opcode 기능은 현재 미흡함
- 도구 제작자도 현재 string 기능만 사용하고 있음

```
strings:
   $x1 = "C:\\WINDOWS\\system32\\ntdll.dll" fullword ascii /* score: '32.00'*/
   $s2 = "%userappdata%\\RestartApp.exe" fullword ascii /* score: '26.42'*/
   $s3 = "%s\\system32\\drivers\\oreans32.sys" fullword ascii /* score: '24.00'*/
   $s4 = "C:\\Documents and Settings\\Administrator\\" fullword ascii /* score: '20.00'*/
   $s5 = "NTDLL.dll" fullword ascii /* score: '18.00'*/
   $s6 = "|C:\\Documents and Settings\\Administrator\\" fullword ascii /* score: '17.00'*/
   $s7 = "APIC error: Cannot find Processors Control Blocks. Please," fullword ascii /* score: '17.00'*/
   $s8 = "oreans32.sys" fullword ascii /* score: '17.00'*/
   $s9 = "oreansx64.sys" fullword ascii /* score: '17.00'*/
   $s10 = "GetEnvironmentVariable API Error while extraction the driver" fullword ascii /* score: '16.00'*/
   $s11 = "contact info@oreans.com for this error" fullword ascii /* score: '15.00'*/
   $s12 = "%s\\system32\\drivers\\%s" fullword ascii /* score: '14.50'*/
   $s13 = "Please, contact yoursite@yoursite.com. Thank you!" fullword ascii /* score: '14.00'*/
   $s14 = "CloseServiceHandle API Error while extraction the driver" fullword ascii /* score: '14.00'*/
   $s15 = "CreateService API Error while extraction the driver" fullword ascii /* score: '13.00'*/
   $s16 = "StartService API Error while extraction the driver" fullword ascii /* score: '13.00'*/
   $s17 = "OpenService API Error while extraction the driver" fullword ascii /* score: '13.00'*/
   $s18 = "3Cannot Update oreans.sys driver. Please, make sure that you have" fullword ascii /* score: '12.00'*/
   $s19 = "\\\\.\\Global\\oreansx64" fullword ascii /* score: '10.00'*/
   $s20 = "CreateEvent API Error while extraction the driver" fullword ascii /* score: '10.00'*/
condition:
   ( uint16(0) == 0x5a4d and filesize < 10000KB and ( 1 of ($x*) and 4 of them )
   ) or ( all of them )
```

제작 도구 소개 및 타 도구와의 차별성

YaraGenerator - Xen0ph0n

- 지정한 폴더의 모든 샘플을 탐지해주는 rule을 만들었으나,
  활용도가 높은 rule은 아님

```
strings:
    $string0 = "         h((((                    H" wide
    $string1 = "abcdefghijklmnopqrstuvwxyz{"
    $string2 = "UQPXY]Y["
    $string3 = "Monday"
    $string4 = "@abcdefghijklmnopqrstuvwxyz[\\]"
    $string5 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ{"
    $string6 = "         (((((                    H" wide
    $string7 = ",-./0123456789:;<"
    $string8 = "Sunday"
    $string9 = "@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]"
    $string10 = ";t$,v-"
    $string11 = "                          "
condition:
    11 of them
```

제작된 rule 결과

```
test.yar(41): warning: $string11 is slowing down scanning (critical!)
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩45EE81F48959FC50320AE3A950D13A08
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩06A778A1F55E15C880628F2C20DB930D6657DC8225A0527F0F044D88F8E9199D
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩0a8c4d07abc13e2c26193127e3c73926
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩AC3C5383432F8AA6A462F86B1EC00919
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩084F2FA731C32DA46BA08BA05EC0E62BA57EE000D05A96A67DF17618FCDD0754
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩6B0551C4912E098AFA0C72264FC5DF9A2B21995436E15ED4A3C1FFF06EF4CEE3
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩809aa788d87bfb4fcaa8d75e07c85abc7874f95169cd2ab7867b6421f8b65d58
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩7B6955A67AF385913D5AF4E8116A6B2A
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩EA38BDC05F3D357623A78E4A90613AE2
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩e777a78c907979591ae858a825b46d5e16754aa803cc7f284fd7709bccafadcc
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩00f850a82b366a2e4e0c312d1d7a1266
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩F066995689F57FF18CC51D48437D8AD7
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩1065EC65DE64FF441A4BAAAC2375E02A
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩ABFDA49440DB35DDA337646297E64701
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩357064b07399cd131e65f3d76b92fb16864692607b2db94adced827c1ad6875b
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩F90662273DB92AA8DE0ABED37767B911
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩F3D59F8D1ED96FCEB7C7C7D64235BB1A
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩36d968fee978d90089b47a489ada2ab65ed5696616a9d7716ede4a4ea0eda8d3
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩4D3D5EA52367C045767F23ECBA2DC01249E93E5AB202E9EB444847ACCE1C7B6A
test C:₩Users₩hyOOun₩Desktop₩new_rifle₩D42F486EBBD0AF5EF37B0A6609379554
```

제작된 rule 사용 결과

제작 도구 소개 및 타 도구와의 차별성

binsequencer - karttoon

- rule을 생성하기까지 긴 시간이 소요
- 해당 도구 또한 XOR Transform을 뽑아주지 못함

```
INFO:

binsequencer.py C:₩Users₩hyOOun₩Desktop₩new_rifle
Match SUCCESS for morphing

*/

rule rule0
    {
        meta:
            description = "Autogenerated by Binsequencer v.1.0.4 from C:₩Users₩hyOOun₩Desktop₩new_rifle₩084F2FA731C32DA4
6BA08BA05EC0E62BA57EE000D05A96A67DF17618FCDD0754"
            author      = ""
            date        = "2018-09-12"

        strings:
            $rule0_bytes = { 5356578B???????8B???????8B??????555250515168????????64????????????A1????????33??89??????64???
??????????8B???????8B?????8B??????33??8B?????83?????74??8B??????83?????74??3B???76??8D?????8D???????8B??89?????83????????75??68??????
????8B?????E8?????????B9?????????8B?????E8?????????EB??64?????????????83?????5F5E5BC38B???????F?????????????B8?????????74??8B??????
?8B?????33??E8?????????558B?????FF?????FF?????FF?????E8?????????83?????5D8B???????8B???????89??B8?????????C3558B???????8B??FF?????FF?
???FF?????E8?????????83?????5DC2?????555657538B??33??33??33??33??33??FF??5B5F5E5DC38B??8B??8B??6A??E8????????33??33??33??33?
?33??FF??558B??5356576A??(5?|6A|FF)[0-6]68????????51E8?????????5F5E5B5DC3558B??????5251FF???????E8????????83?????5DC2????? }


        condition:
            all of them
}
```

binsequencer 도구를 이용한 rule 생성 결과

제작 도구 소개 및 타 도구와의 차별성

binsequencer - karttoon

```
v8 = a3 - v4;
v10 = v3;
do
{
  *v4 = v6 ^ result ^ v5 ^ v4[v8];
  v6 = v6 & result ^ v5 & (v6 ^ result);
  v5 = (((v11 ^ (8 * v11)) & 0x7F8) << 20) | (v11 >> 8);
  result = (((result << 7) ^ (result ^ 16 * (result ^ 2 * result)) & 0xFFFFFF80) << 17) | (result
  ++v4;
  v9 = v10-- == 1;
  v11 = (((v11 ^ (8 * v11)) & 0x7F8) << 20) | (v11 >> 8);
}
while ( !v9 );
```

XOR Transform pseudocode

```
v14 = a1;
v13 = a2;
v12 = a3;
v10 = _unwind_handler4;
v11 = &v9 ^ __security_cookie;
while ( 1 )
{
  result = a2;
  v4 = *(a2 + 12);
  if ( v4 == -2 || a3 != -2 && v4 <= a3 )
    break;
  v5 = 3 * v4;
  v6 = (*a1 ^ *(a2 + 8)) + 4 * v5 + 16;
  *(a2 + 12) = *((*a1 ^ *(a2 + 8)) + 4 * v5 + 0x10);
  if ( !*(v6 + 4) )
  {
    v7 = *(v6 + 8);
    _NLG_Notify(257);
    v8 = *(v6 + 8);
    _NLG_Call(1);
  }
}
return result;
```

binsequencer에서 제작한 rule

제작 도구 소개 및 타 도구와의 차별성

yara_fn - williballenthin

- call api 주소만 와일드 카드 처리
- IDA 블록과 jmp 구문으로 나눠 rule을 각각 생성

  현재 IDA View에 켜져 있는 함수의 코드 전체를 rule로 제작하기 때문에 비효율적
- GUI 도구가 아니었고, 기능이 한 가지만 존재하였음

```
rule a_7CAA500B60A536D7501E7A6C02408538_sub_401510 {
  meta:
    sample_md5 = "7CAA500B60A536D7501E7A6C02408538"
    function_address = "0x401510"
    function_name = "sub_401510"
  strings:
    $0x401510 = { 83 EC 0C 53 8B 5C 24 18 56 6A 04 68 00 10 00 00 53 6A 00 FF 15 ?? ?? ?? ?? 8B F0 BA 82 94 6F
55 89 74 24 10 B1 05 89 54 24 08 B8 58 20 C1 AF 85 DB }
    $0x401543 = { 55 57 8B 7C 24 20 2B FE 89 5C 24 14 90 }
    $0x401550 = { 8A 1C 37 32 DA 32 D8 32 D9 88 1E 8A D8 32 D9 22 DA 8A D0 22 D1 32 DA 8B 54 24 10 8A CB 8D 1C
D5 00 00 00 00 33 DA 81 E3 F8 07 00 00 C1 E3 14 C1 EA 08 0B D3 8D 1C 00 33 D8 C1 E3 04 33 D8 8B E8 83 E3 80 C1
E5 07 33 DD C1 E3 11 C1 E8 08 0B C3 46 83 6C 24 14 01 89 54 24 10 }
    $0x4015ac = { 8B 74 24 18 8B 5C 24 24 5F 5D }
    $0x4015b6 = { 8B 44 24 18 53 56 50 ?? ?? ?? ?? ?? 83 C4 0C 68 00 80 00 00 6A 00 56 FF 15 ?? ?? ?? ?? 5E 5B
83 C4 0C C3 }
  condition:
    all of them
}
```

yara_fn 도구 사용 결과

제작 도구 소개 및 타 도구와의 차별성

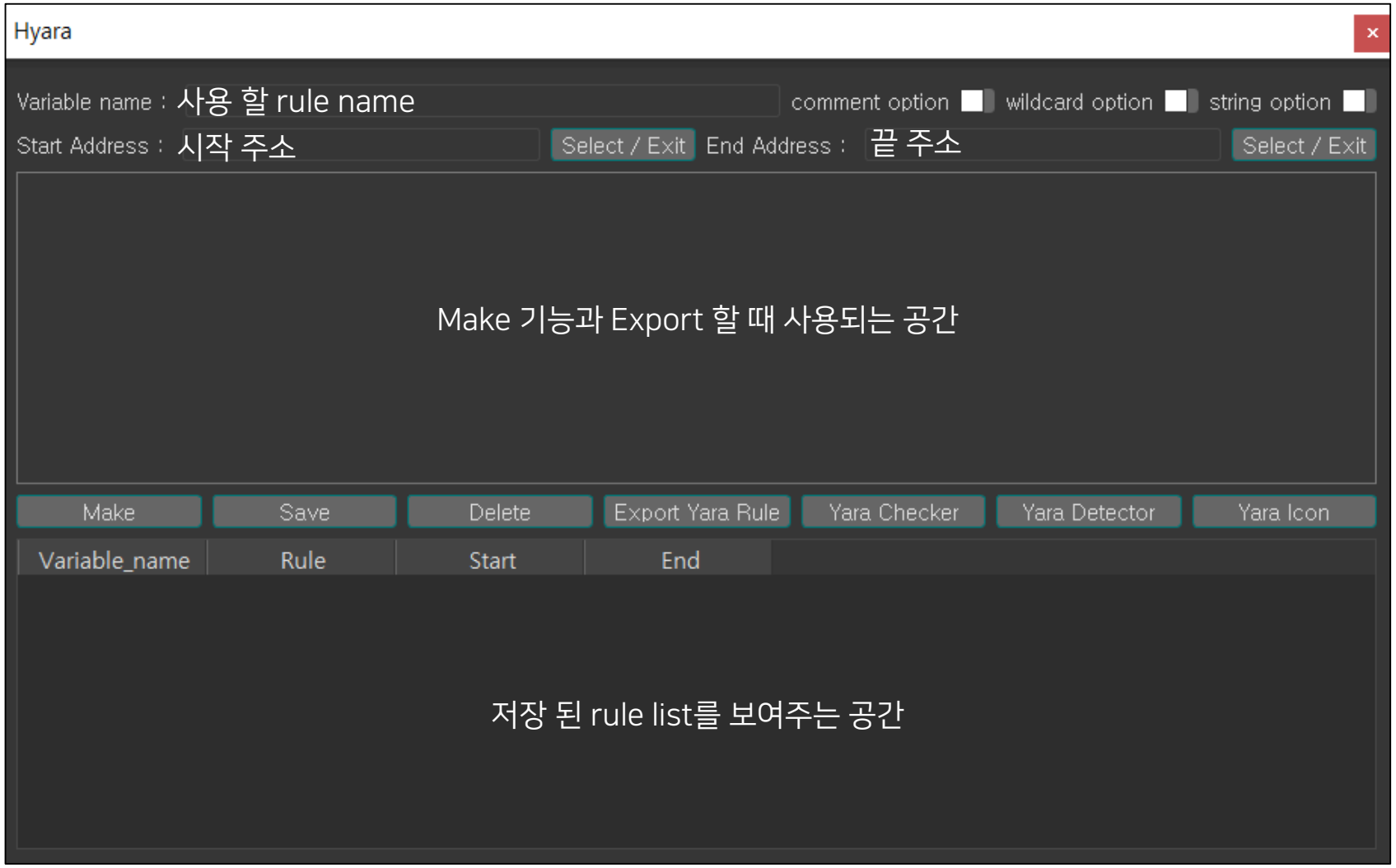
타 도구의 retrohunt 결과

도구 이름	Scanned Data	Scanning speed	Matches
YaraGenerator	417.1 GB	9.3 GB/s	10000
yarGen	67.6 TB	10.1 GB/s	10000
yara_fn	105.9 TB	9.4 GB/s	0
binsequencer	377.8 GB	12.6 GB/s	10000

retrohunt를 각각 돌린 결과 오탐 혹은 미탐인 rule을 만들었음을 의미함

제작 도구 소개 및 타 도구와의 차별성

Hyara



제작한 도구의 main 화면

제작 도구 소개 및 타 도구와의 차별성

Hyara

Select / Exit 버튼 동작 방식

```
def OnViewClick(self, px, py, state):
    widget = pycim_get_tcustom_control(self)
    from_mouse = False
    line = get_custom_viewer_curline(widget, from_mouse)
    print line
    print(binascii.hexlify(line))
```

```
┌‼.text:00403344 ┐‼                    ┌│push┐│     ┐)┌!edi┐!┐)
01132e746578743a3030343033333434200213202020202020202020202020202020200105707573680205202020200129012165646902210229
Python>binascii.unhexlify("01132e746578743a3030343033333434")
┌‼.text:00403344
┌‼.text:0040334A ┐‼                    ┌│xor┐│      ┐)┌!eax┐!┐)┌    ,┐         ┌*┌!ebp┐!┐*
01132e746578743a3030343033333441200213202020202020202020202020202020200105786f7202052020202020012901216561780221022
Python>binascii.unhexlify("01132e746578743a3030343033333441")
┌‼.text:0040334A
```

IDAViewWrapper 기능을 이용하여 클릭한 해당 라인 정보를 가져올 수 있음

제작 도구 소개 및 타 도구와의 차별성

Hyara

```
if self.CheckBox3.isChecked(): # Use String Option
    StringData = []
    ## https://reverseengineering.stackexchange.com/questions/3603/how-to-extract-all-the-rodata-data-ar
    text_section_endEA = idaapi.get_segm_by_name(".text").endEA
    blacklist = ["unk_", "loc_", "SEH_"]
    if text_section_endEA > start:
        while start <= end:
            if "offset" in GetOpnd(start, 0) and not any(i in GetOpnd(start, 0) for i in blacklist):
                variable = GetOpnd(start, 0).split(" ")[1]
                add = get_name_ea(start,variable)
                string, endEA = get_string(add)
                StringData.append(string)

            elif "offset" in GetOpnd(start, 1) and not any(i in GetOpnd(start, 1) for i in blacklist):
                variable = GetOpnd(start, 1).split(" ")[1]
                add = get_name_ea(start,variable)
                string, endEA = get_string(add)
                StringData.append(string)

            start = idc.NextHead(start)
```

string option 설정 + 지정된 주소가 .text section의 end 주소 이전인 경우
offset 변수에 있는 문자열을 가져옴

제작 도구 소개 및 타 도구와의 차별성

Hyara

```
if self.CheckBox3.isChecked(): # Use String Option
    StringData = []
    ## https://reverseengineering.stackexchange.com/questions/3603/how-to-extract-all-the-rodata-data-ar
    text_section_endEA = idaapi.get_segm_by_name(".text").endEA
    blacklist = ["unk_", "loc_", "SEH_"]
    if text_section_endEA > start:
        while start <= end:
            if "offset" in GetOpnd(start, 0) and not any(i in GetOpnd(start, 0) for i in blacklist):
                variable = GetOpnd(start, 0).split(" ")[1]
                add = get_name_ea(start,variable)
                string, endEA = get_string(add)
                StringData.append(string)

            elif "offset" in GetOpnd(start, 1) and not any(i in GetOpnd(start, 1) for i in blacklist):
                variable = GetOpnd(start, 1).split(" ")[1]
                add = get_name_ea(start,variable)
                string, endEA = get_string(add)
                StringData.append(string)

            start = idc.NextHead(start)
```

string option 설정 + 지정된 주소가 .text section의 end 주소 이전인 경우
offset 변수에 있는 문자열을 가져옴

제작 도구 소개 및 타 도구와의 차별성

Hyara

```
if self.CheckBox3.isChecked(): # Use String Option
    StringData = []
    ## https://reverseengineering.stackexchange.com/questions/3603/how-to-extract-all-the-rodata-data-ar
    text_section_endEA = idaapi.get_segm_by_name(".text").endEA
    blacklist = ["unk_", "loc_", "SEH_"]
    if text_section_endEA > start:
        while start <= end:
            if "offset" in GetOpnd(start, 0) and not any(i in GetOpnd(start, 0) for i in blacklist):
                variable = GetOpnd(start, 0).split(" ")[1]
                add = get_name_ea(start,variable)
                string, endEA = get_string(add)
                StringData.append(string)

            elif "offset" in GetOpnd(start, 1) and not any(i in GetOpnd(start, 1) for i in blacklist):
                variable = GetOpnd(start, 1).split(" ")[1]
                add = get_name_ea(start,variable)
                string, endEA = get_string(add)
                StringData.append(string)

            start = idc.NextHead(start)
```

string option 설정 + 지정된 주소가 .text section의 end 주소 이전인 경우
offset 변수에 있는 문자열을 가져옴

제작 도구 소개 및 타 도구와의 차별성

Hyara

```
mov     ecx, offset aSCryptreleasec ; "S^CryptReleaseContext"
mov     dword_414DDC, eax
call    sub_403770
push    eax             ; lpProcName
push    edi             ; hModule
call    esi ; GetProcAddress
mov     ecx, offset aSCryptencrypt ; "S^CryptEncrypt"
```

```
push    ebp
mov     ebp, esp
push    0FFFFFFFFh
push    offset SEH_403330
mov     eax, large fs:0
push    eax
push    ecx
push    ebx
```

```
mov     ecx, dword_42509C
mov     eax, dword_425094
push    offset unk_4250B0
push    ecx
mov     ecx, dword_425090
```

```
push    offset sub_402E80
push    0
push    0
call    dword_414EC4
push    0FFFFFFFFh
push    eax
mov     dword_414D08, eax
call    dword_414EC0
```

SEH,_ unk_, sub_, loc_는 필터링을 거치고 그 이외에는 문자열로 취급하여 사용함

제작 도구 소개 및 타 도구와의 차별성

Hyara

```
else:
    while start <= end:
        string, endEA = get_string(start)
        StringData.append(string)
        start = endEA
    StringData = [x for x in StringData if x]
    self.TextEdit1.clear()
    for i in StringData:
        if i == "":
            continue

        self.TextEdit1.insertPlainText("\""+ i.replace("\\","\\\\") +"\"" + " nocase wide ascii" + "\n")
```

지정된 주소가 .text 섹션 이후인 경우 get_string 함수로 문자열을 가져옴

제작 도구 소개 및 타 도구와의 차별성

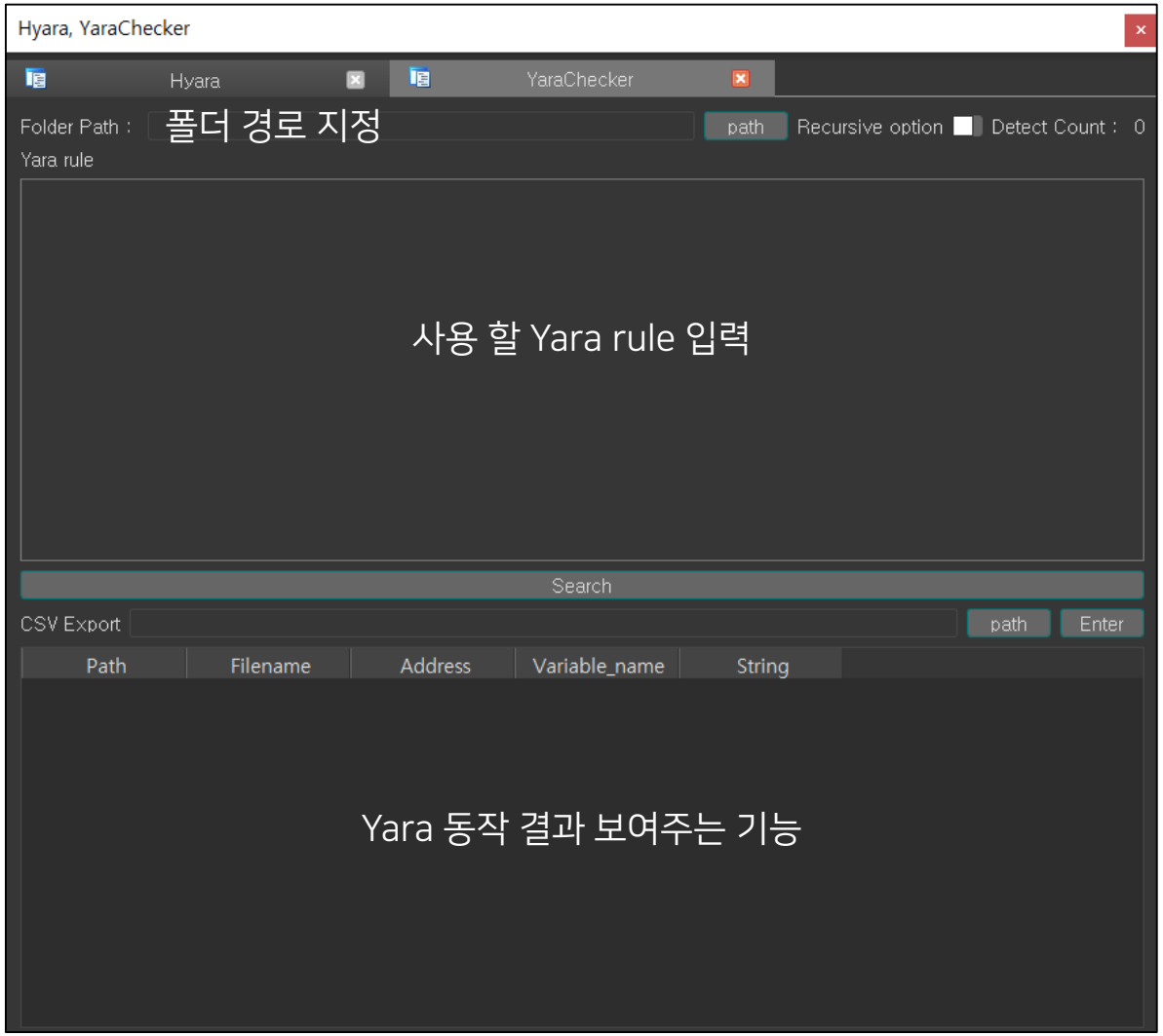
Hyara

```
def get_string(addr):
    out = ""
    assem_data = GetDisasm(addr)

    if "text \"UTF-16LE\"" in assem_data or "unicode 0," in assem_data:
        while True:
            if Byte(addr) == 0 and Byte(addr+1) == 0:
                addr += 2
                break
            else:
                out += chr(Byte(addr))
                out += chr(Byte(addr+1))
            addr += 2
        return out.decode("utf-16le"), addr

    else:
        while True:
            if Byte(addr) != 0:
                out += chr(Byte(addr))
            else:
                addr += 1
                break
            addr += 1

        return out, addr
```

get_string 함수는 utf-16인지 아닌지 체크하여 문자열을 저장해주는 기능

제작 도구 소개 및 타 도구와의 차별성

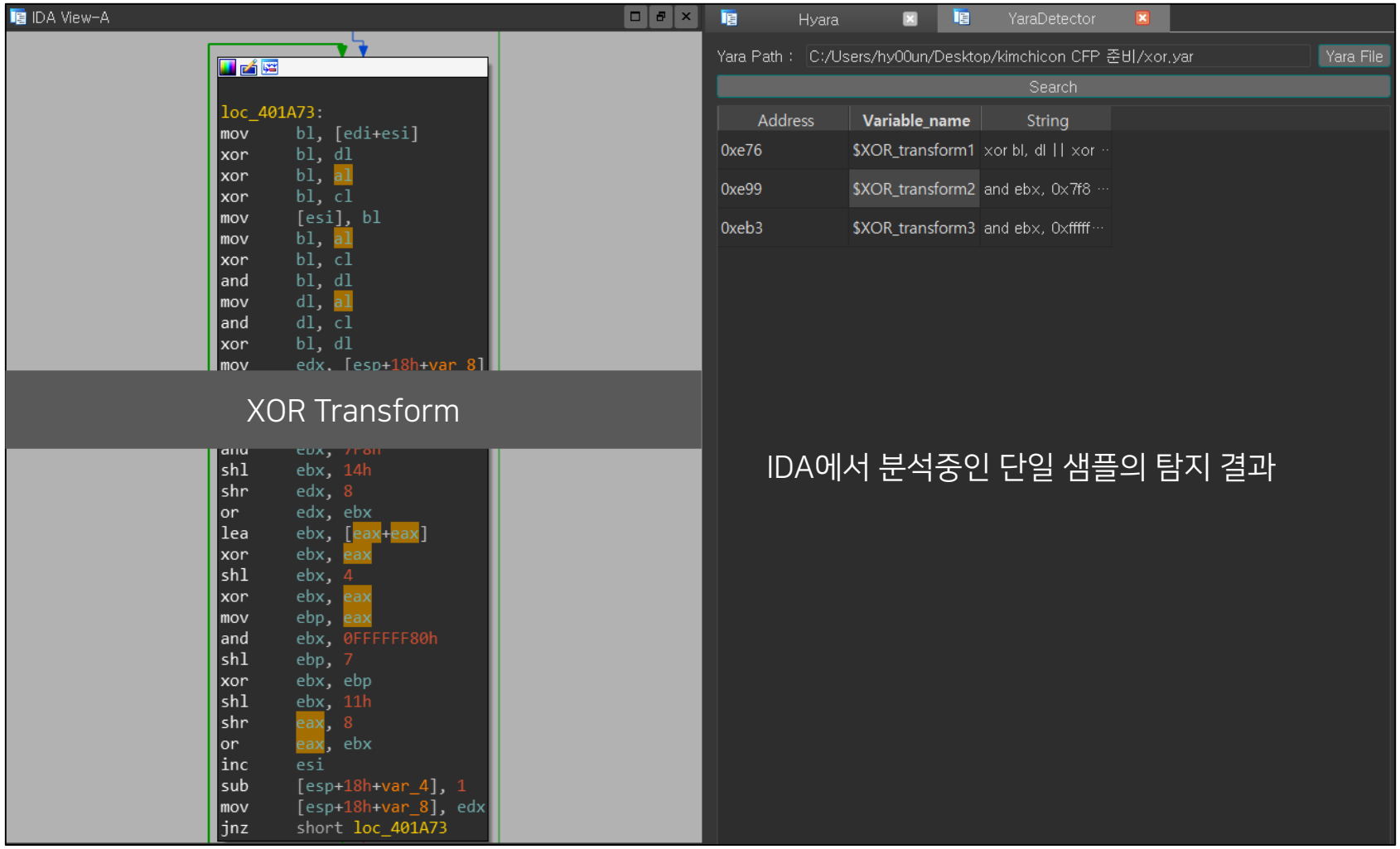
Hyara



YaraChecker 도구 사용 결과

제작 도구 소개 및 타 도구와의 차별성

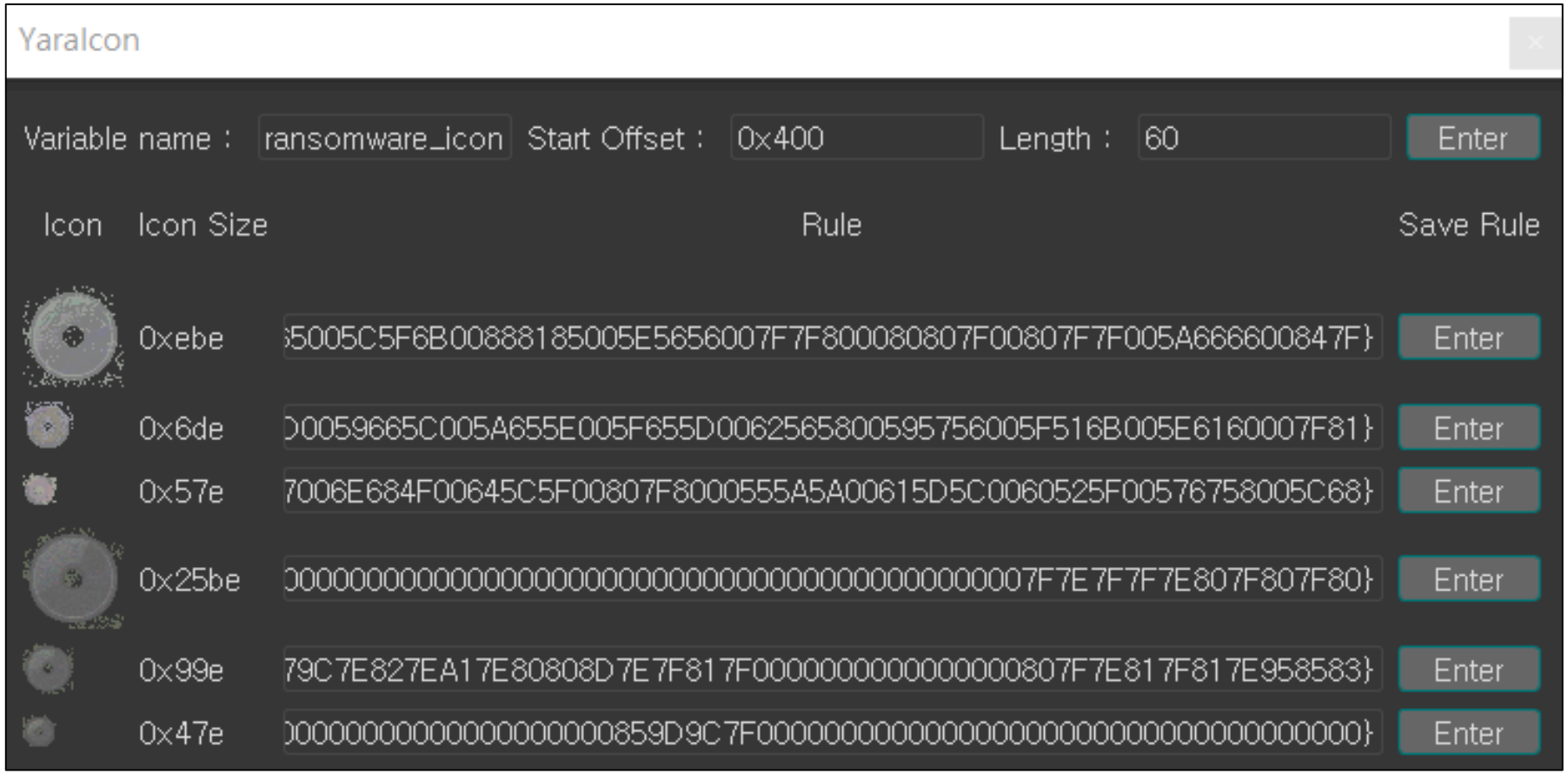
Hyara



YaraDetector 도구 사용 결과

제작 도구 소개 및 타 도구와의 차별성

Hyara



YaraIcon 도구 사용 결과

제작 도구 소개 및 타 도구와의 차별성

Hyara

```
for entry in self.pe.DIRECTORY_ENTRY_RESOURCE.entries:
    resource_type = entry.name
    if resource_type is None:
        resource_type = pefile.RESOURCE_TYPE.get(entry.struct.Id)

    for directory in entry.directory.entries:
        for resource in directory.directory.entries:
            name = str(resource_type)
            if name in "RT_ICON":
                name = str(resource_type)
                offset = resource.data.struct.OffsetToData
                size = resource.data.struct.Size
                RVA_ = int(self.section_list['.rsrc'][0],16) - int(self.section_list['.rsrc'][2],16) # VirtualAddress - PointerToRawData
                real_offset = offset - RVA_
                img_size = hex(size)[2:]
                if len(img_size) % 2 == 1:
                    img_size = "0"+img_size

                img_ = "\x00\x00\x01\x00\x01\x00\x30\x30\x00\x00\x01\x00\x08\x00" + bytearray.fromhex(img_size)[::-1] + "\x00\x00\x16\x00
                f = open(GetInputFilePath(),"rb")
                f.seek(real_offset)
                img_ += f.read(size)
                f.close()
                self.img.append(img_)
```

RT_ICON 데이터를 뽑은 결과, ico 헤더가 없어 헤더 양식을 맞춘 후에 size만큼 저장하고,
Qt에서 GUI로 보여주는 방식

4

활용 사례

활용 사례

XOR transform

```
v3 = a1;
LOBYTE(v4) = 0x49;
v5 = a2;
v6 = 0x92u;
v10 = 0x1ABC0949;
result = 0x18430647;
if ( v3 > 0 )
{
  v8 = a3 - v5;
  v11 = v3;
  do
  {
    *v5 = v6 ^ result ^ v4 ^ v5[v8];
    v6 = v6 & result ^ v4 & (v6 ^ result);
    v4 = (((v10 ^ (8 * v10)) & 0x7F8) << 20) | (v10 >> 8);
    result = (((result << 7) ^ (result ^ 16 * (result ^ 2 * result)) & 0xFFFFFF80) << 17) | (result >> 8);
    ++v5;
    v9 = v11-- == 1;
    v10 = (((v10 ^ (8 * v10)) & 0x7F8) << 20) | (v10 >> 8);
  }
  while ( !v9 );
}
return result;
```

Andariel 그룹에서 사용되는 유니크한 코드인 XOR Transform

활용 사례

XOR transform

```
v3 = a1;
LOBYTE(v4) = 0x49;
v5 = a2;
v6 = 0x92u;
v10 = 0x1ABC0949;
result = 0x18430647;
if ( v3 > 0 )
{
  v8 = a3 - v5;
  v11 = v3;
  do
  {
    *v5 = v6 ^ result ^ v4 ^ v5[v8];
    v6 = v6 & result ^ v4 & (v6 ^ result);
    v4 = (((v10 ^ (8 * v10)) & 0x7F8) << 20) | (v10 >> 8);
    result = (((result << 7) ^ (result ^ 16 * (result ^ 2 * result)) & 0xFFFFFF80) << 17) | (result >> 8);
    ++v5;
    v9 = v11-- == 1;
    v10 = (((v10 ^ (8 * v10)) & 0x7F8) << 20) | (v10 >> 8);
  }
  while ( !v9 );
}
return result;
```

0x7F8, 0xFFFFFF80과 같은 고유한 값을 이용하여 rule을 생성

활용 사례

XOR transform

IDA View-A | Pseudocode-C | Pseudocode-B | Pseudocode-A | Hex View-1 | Structures | Enums | Imports | Exports

```
int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
{
  LPSTR v4; // eax
  int v5; // ecx
  int *v7; // esi
  _DWORD *v8; // eax
  HWND v9; // eax
  HWND v10; // edi
  void **v11; // esi
  HWND v12; // [esp+10h] [ebp-1E8h]
  WNDCLASSEXA v13; // [esp+14h] [ebp-1E4h]
  struct tagMSG Msg; // [esp+44h] [ebp-1B4h]
  struct WSAData WSAData; // [esp+60h] [ebp-198h]

  v4 = GetCommandLineA();
  v5 = 0;
  if ( *v4 == 34 )
  {
    if ( *++v4 )
    {
      do
      {
        if ( v4[v5] == 34 )
          break;
        ++v5;
      }
      while ( v4[v5] );
    }
    v4[v5] = 0;
  }
  dword_40C700 = v4;
  if ( hPrevInstance )
    return 1;
  dword_40C704 = 1698183009;
  dword_40C710 = 1684431212;
  dword_40C708 = 875784297;
  dword_40C70C = 1715740963;
  byte_40C714 = 0;
```

활용 사례

XOR transform

```
xor     ebx, edx
and     ebx, 7F8h
shl     ebx, 14h
shr     edx, 8
or      edx, ebx
lea     ebx, [eax+eax]
xor     ebx, eax
shl     ebx, 4
xor     ebx, eax
mov     ebp, eax
and     ebx, 0FFFFFF80h
shl     ebp, 7
xor     ebx, ebp
shl     ebx, 11h
shr     eax, 8
or      eax, ebx
```

F90662273DB92AA8DE0ABED37767B911

```
xor     edi, edx
and     edi, 7F8h
shl     edi, 14h
shr     edx, 8
or      edx, edi
lea     edi, [eax+eax]
xor     edi, eax
and     cl, al
shl     edi, 4
xor     edi, eax
xor     cl, bl
mov     ebx, eax
and     edi, 0FFFFFF80h
shl     ebx, 7
xor     edi, ebx
shl     edi, 11h
shr     eax, 8
or      eax, edi
```

EE778BE503FDA770EE2F40E51EDFD595

```
xor     ebx, edx
and     ebx, 7F8h
shl     ebx, 14h
shr     edx, 8
or      edx, ebx
lea     ebx, [eax+eax]
xor     ebx, eax
shl     ebx, 4
xor     ebx, eax
mov     ebp, eax
and     ebx, 0FFFFFF80h
shl     ebp, 7
xor     ebx, ebp
shl     ebx, 11h
shr     eax, 8
inc     esi
or      eax, ebx
```

AC3C5383432F8AA6A462F86B1EC00919

레지스터 값이 다른 샘플 존재, 어셈 코드가 추가 되어있는 샘플도 존재
이러한 변종까지 잡기 위해서는 와일드 카드 기능 필요

활용 사례

XOR transform

```
xor     ebx, edx
and     ebx, 7F8h
shl     ebx, 14h
shr     edx, 8
or      edx, ebx
lea     ebx, [eax+eax]
xor     ebx, eax
shl     ebx, 4
xor     ebx, eax
mov     ebp, eax
and     ebx, 0FFFFFF80h
shl     ebp, 7
xor     ebx, ebp
shl     ebx, 11h
shr     eax, 8
or      eax, ebx
```

F90662273DB92AA8DE0ABED37767B911

```
xor     edi, edx
and     edi, 7F8h
shl     edi, 14h
shr     edx, 8
or      edx, edi
lea     edi, [eax+eax]
xor     edi, eax
and     cl, al
shl     edi, 4
xor     edi, eax
xor     cl, bl
mov     ebx, eax
and     edi, 0FFFFFF80h
shl     ebx, 7
xor     edi, ebx
shl     edi, 11h
shr     eax, 8
or      eax, edi
```

EE778BE503FDA770EE2F40E51EDFD595

```
xor     ebx, edx
and     ebx, 7F8h
shl     ebx, 14h
shr     edx, 8
or      edx, ebx
lea     ebx, [eax+eax]
xor     ebx, eax
shl     ebx, 4
xor     ebx, eax
mov     ebp, eax
and     ebx, 0FFFFFF80h
shl     ebp, 7
xor     ebx, ebp
shl     ebx, 11h
shr     eax, 8
inc     esi
or      eax, ebx
```

AC3C5383432F8AA6A462F86B1EC00919

레지스터 값이 다른 샘플 존재, 어셈 코드가 추가 되어있는 샘플도 존재
이러한 변종까지 잡기 위해서는 와일드 카드 기능 필요

활용 사례

XOR transform

```
xor     ebx, edx
and     ebx, 7F8h
shl     ebx, 14h
shr     edx, 8
or      edx, ebx
lea     ebx, [eax+eax]
xor     ebx, eax
shl     ebx, 4
xor     ebx, eax
mov     ebp, eax
and     ebx, 0FFFFFF80h
shl     ebp, 7
xor     ebx, ebp
shl     ebx, 11h
shr     eax, 8
or      eax, ebx
```

F90662273DB92AA8DE0ABED37767B911

```
xor     edi, edx
and     edi, 7F8h
shl     edi, 14h
shr     edx, 8
or      edx, edi
lea     edi, [eax+eax]
xor     edi, eax
and     cl, al
shl     edi, 4
xor     edi, eax
xor     cl, bl
mov     ebx, eax
and     edi, 0FFFFFF80h
shl     ebx, 7
xor     edi, ebx
shl     edi, 11h
shr     eax, 8
or      eax, edi
```

EE778BE503FDA770EE2F40E51EDFD595

```
xor     ebx, edx
and     ebx, 7F8h
shl     ebx, 14h
shr     edx, 8
or      edx, ebx
lea     ebx, [eax+eax]
xor     ebx, eax
shl     ebx, 4
xor     ebx, eax
mov     ebp, eax
and     ebx, 0FFFFFF80h
shl     ebp, 7
xor     ebx, ebp
shl     ebx, 11h
shr     eax, 8
inc     esi
or      eax, ebx
```

AC3C5383432F8AA6A462F86B1EC00919

레지스터 값이 다른 샘플 존재, 어셈 코드가 추가 되어있는 샘플도 존재
이러한 변종까지 잡기 위해서는 와일드 카드 기능 필요

활용 사례

XOR transform

32 DA
32 D8
32 D9

```
xor     bl, dl
xor     bl, al
xor     bl, cl
```

81 E7 F8 07 00 00
C1 E7 14
C1 EA 08

```
and     edi, 7F8h
shl     edi, 14h
shr     edx, 8
```

83 E7 80
C1 E3 07

```
and     edi, 0FFFFFF80h
shl     ebx, 7
```

F90662273DB92AA8DE0ABED37767B911

32 DA
32 D8
32 D9

```
xor     bl, dl
xor     bl, al
xor     bl, cl
```

81 E3 F8 07 00 00
C1 E3 14
C1 EA 08

```
and     ebx, 7F8h
shl     ebx, 14h
shr     edx, 8
```

83 E3 80
C1 E5 07

```
and     ebx, 0FFFFFF80h
shl     ebp, 7
```

EE778BE503FDA770EE2F40E51EDFD595

활용 사례

XOR transform

32 DA
32 D8
32 D9

```
xor     bl, dl
xor     bl, al
xor     bl, cl
```

81 E7 F8 07 00 00
C1 E7 14
C1 EA 08

```
and     edi, 7F8h
shl     edi, 14h
shr     edx, 8
```

83 E7 80
C1 E3 07

```
and     edi, 0FFFFFF80h
shl     ebx, 7
```

F90662273DB92AA8DE0ABED37767B911

32 DA
32 D8
32 D9

```
xor     bl, dl
xor     bl, al
xor     bl, cl
```

81 E3 F8 07 00 00
C1 E3 14
C1 EA 08

```
and     ebx, 7F8h
shl     ebx, 14h
shr     edx, 8
```

83 E3 80
C1 E5 07

```
and     ebx, 0FFFFFF80h
shl     ebp, 7
```

EE778BE503FDA770EE2F40E51EDFD595

활용 사례

XOR transform

Job status	Finished
Rules	rule XOR_transform : XOR { meta: tool = "https://github.com/hy00un/Hyara" version = "1.4" date = "2018-09-07" MD5 = "F3D5...
Creation time	9 7, 2018, 9:25 오후
Start time	9 7, 2018, 11:58 오후
Finish time	9 8, 2018, 3:22 오전
Scanned data	107.7 TB
Scanning speed	9.0 GB/s
Matches	21 Download hashes

Job status	Finished
Rules	rule XOR_transform_wildcard : XOR { meta: tool = "https://github.com/hy00un/Hyara" version = "1.4" date = "2018-09-07" MD...
Creation time	9 7, 2018, 9:24 오후
Start time	9 7, 2018, 11:58 오후
Finish time	9 8, 2018, 3:22 오전
Scanned data	107.7 TB
Scanning speed	9.0 GB/s
Matches	58 Download hashes

Start new job

와일드카드를
사용하지 않은 rule

와일드카드를
사용한 rule

활용 사례

S^ Transform

```
v0 = (const CHAR *)sub_403770("S^Kernel32.dll");
v1 = LoadLibraryA(v0);
if ( v1 )
{
  v2 = (const CHAR *)sub_403770("S^HeapCreate");
  dword_414E94 = (int)GetProcAddress(v1, v2);
  v3 = (const CHAR *)sub_403770("S^GetProcessHeap");
  dword_414E38 = (int)GetProcAddress(v1, v3);
  v4 = (const CHAR *)sub_403770("S^HeapDestroy");
  dword_414E88 = (int)GetProcAddress(v1, v4);
  v5 = (const CHAR *)sub_403770("S^HeapAlloc");
  dword_414F38 = (int)GetProcAddress(v1, v5);
  v6 = (const CHAR *)sub_403770("S^HeapReAlloc");
  dword_414F50 = (int)GetProcAddress(v1, v6);
  v7 = (const CHAR *)sub_403770("S^HeapFree");
  dword_414EAC = (int)GetProcAddress(v1, v7);
  v8 = (const CHAR *)sub_403770("S^GetModuleFileNameA");
  dword_414F28 = (int)GetProcAddress(v1, v8);
  v9 = (const CHAR *)sub_403770("S^DeleteFileA");
  dword_414E40 = (int)GetProcAddress(v1, v9);
  v10 = (const CHAR *)sub_403770("S^CreateMutexA");
  dword_414DE0 = (int)GetProcAddress(v1, v10);
  v11 = (const CHAR *)sub_403770("S^CreateThread");
  dword_414EC4 = (int)GetProcAddress(v1, v11);
  v12 = (const CHAR *)sub_403770("S^CreateFileA");
  dword_414E6C = (int)GetProcAddress(v1, v12);
  v13 = (const CHAR *)sub_403770("S^GetFileSize");
  dword_414E28 = (int)GetProcAddress(v1, v13);
  v14 = (const CHAR *)sub_403770("S^LockFile");
  dword_414E24 = (int)GetProcAddress(v1, v14);
  v15 = (const CHAR *)sub_403770("S^WaitForSingleObject");
  dword_414EC0 = (int)GetProcAddress(v1, v15);
  v16 = (const CHAR *)sub_403770("S^ReleaseMutex");
```

Andariel 그룹에서 분석 지연 목적으로 추정되는 s^ transform 문자열

활용 사례

S^ Transform

IDA View-A | Pseudocode-C | Pseudocode-B | Pseudocode-A | Strings window | Hex View-1 | Structures | Enums | Imports | Exports

```
131  v121 = 0;
132  v120 = 0;
133  v0 = (const CHAR *)sub_403770("S^Kernel32.dll");
134  v1 = LoadLibraryA(v0);
135  if ( v1 )
136  {
137    v2 = (const CHAR *)sub_403770("S^HeapCreate");
138    dword_414E94 = (int)GetProcAddress(v1, v2);
139    v3 = (const CHAR *)sub_403770("S^GetProcessHeap");
140    dword_414E38 = (int)GetProcAddress(v1, v3);
141    v4 = (const CHAR *)sub_403770("S^HeapDestroy");
142    dword_414E88 = (int)GetProcAddress(v1, v4);
143    v5 = (const CHAR *)sub_403770("S^HeapAlloc");
144    dword_414F38 = (int)GetProcAddress(v1, v5);
145    v6 = (const CHAR *)sub_403770("S^HeapReAlloc");
146    dword_414F50 = (int)GetProcAddress(v1, v6);
147    v7 = (const CHAR *)sub_403770("S^HeapFree");
148    dword_414EAC = (int)GetProcAddress(v1, v7);
149    v8 = (const CHAR *)sub_403770("S^GetModuleFileNameA");
150    dword_414F28 = (int)GetProcAddress(v1, v8);
151    v9 = (const CHAR *)sub_403770("S^DeleteFileA");
152    dword_414E40 = (int)GetProcAddress(v1, v9);
153    v10 = (const CHAR *)sub_403770("S^CreateMutexA");
154    dword_414DE0 = (int)GetProcAddress(v1, v10);
155    v11 = (const CHAR *)sub_403770("S^CreateThread");
156    dword_414EC4 = (int)GetProcAddress(v1, v11);
157    v12 = (const CHAR *)sub_403770("S^CreateFileA");
158    dword_414E6C = (int)GetProcAddress(v1, v12);
159    v13 = (const CHAR *)sub_403770("S^GetFileSize");
160    dword_414E28 = (int)GetProcAddress(v1, v13);
161    v14 = (const CHAR *)sub_403770("S^LockFile");
162    dword_414E24 = (int)GetProcAddress(v1, v14);
163    v15 = (const CHAR *)sub_403770("S^WaitForSingleObject");
164    dword_414EC0 = (int)GetProcAddress(v1, v15);
165    v16 = (const CHAR *)sub_403770("S^ReleaseMutex");
166    dword_414E8C = (int)GetProcAddress(v1, v16);
167    v17 = (const CHAR *)sub_403770("S^UnlockFile");
168    dword_414F74 = (int)GetProcAddress(v1, v17);
```

활용 사례

S^ Transform

Job status	Finished
Rules	rule S_transform { meta: tool = "https://github.com/hy00un/Hyara" version = "1.4" date = "2018-09-07" MD5 = "052596A8380E...
Creation time	9 6, 2018, 11:59 오후
Start time	9 7, 2018, 3 오전
Finish time	9 7, 2018, 6:14 오전
Scanned data	107.7 TB
Scanning speed	9.5 GB/s
Matches	11 Download hashes

Start new job

S^(apiname) 형식은 유니크한 스트링이 될 수 있기 때문에
rule 생성 후 retrohunt를 돌린 결과, 11개를 탐지하였음

활용 사례

Joanap

```
v5 = sub_401757(aEmcfgv7xc8itav, aIamsorry123456);
v6 = LoadLibraryA(v5);
if ( !v6 )
  return 0;
v7 = sub_401757(aUra9t1tcdes197, aIamsorry123456);
dword_418B18 = GetProcAddress(v6, v7);
v8 = sub_401757(aVwbebxyx1nzrck, aIamsorry123456);
dword_418B10 = GetProcAddress(v6, v8);
v9 = sub_401757(a2fachi224AQ8gs, aIamsorry123456);
dword_418B34 = GetProcAddress(v6, v9);
v10 = sub_401757(aGawd1uiqi6w8kj, aIamsorry123456);
dword_418B44 = GetProcAddress(v6, v10);
v11 = sub_401757(a6ro0eykriqfmph, aIamsorry123456);
dword_418B78 = GetProcAddress(v6, v11);
v12 = sub_401757(aM2mbhjehq7ik6u, aIamsorry123456);
dword_418B2C = GetProcAddress(v6, v12);
v13 = sub_401757(aCtrhfex5m9jnzd, aIamsorry123456);
dword_418B14 = GetProcAddress(v6, v13);
v14 = sub_401757(aTlpco4ikblt6jn, aIamsorry123456);
dword_418B0C = GetProcAddress(v6, v14);
v15 = sub_401757(a0e1mfduuanes8y, aIamsorry123456);
dword_418B3C = GetProcAddress(v6, v15);
v16 = sub_401757(aXjkuiwonzthbmw, aIamsorry123456);
dword_418B6C = GetProcAddress(v6, v16);
v17 = sub_401757(aN0u76ngone2y03, aIamsorry123456);
dword_418B58 = GetProcAddress(v6, v17);
v18 = sub_401757(a6y8iuawgbu7Tk, aIamsorry123456);
v19 = LoadLibraryA(v18);
```

```
.data:0040B264 aCtrhfex5m9jnzd db '!ctRHFEX5m9JnZdDfpK',0
.data:0040B264                                          ; DATA XREF: sub_401E91+155↑o
.data:0040B278 ; char aM2mbhjehq7ik6u[]
.data:0040B278 aM2mbhjehq7ik6u db '!m2MBHjehQ7IK6uqIsejT',0
.data:0040B278                                          ; DATA XREF: sub_401E91+13F↑o
.data:0040B28E                 align 10h
.data:0040B290 ; char a6ro0eykriqfmph[]
.data:0040B290 a6ro0eykriqfmph db '!6ro0EYkRiqFMphgymbcTsfJ60K',0
.data:0040B290                                          ; DATA XREF: sub_401E91+129↑o
.data:0040B2AC ; char aGawd1uiqi6w8kj[]
.data:0040B2AC aGawd1uiqi6w8kj db '!GawD1UIQi6w8kjUgleSNGrXVwcY',0
.data:0040B2AC                                          ; DATA XREF: sub_401E91+113↑o
.data:0040B2C9                 align 4
.data:0040B2CC ; char a2fachi224AQ8gs[]
.data:0040B2CC a2fachi224AQ8gs db '!_2FAcHI224$A_q8gS0dK',0
.data:0040B2CC                                          ; DATA XREF: sub_401E91+FD↑o
.data:0040B2E2                 align 4
.data:0040B2E4 ; char aVwbebxyx1nzrck[]
.data:0040B2E4 aVwbebxyx1nzrck db '!VWBeBxYx1nzrCkBLGQO',0
.data:0040B2E4                                          ; DATA XREF: sub_401E91+E7↑o
.data:0040B2F9                 align 4
.data:0040B2FC ; char aUra9t1tcdes197[]
.data:0040B2FC aUra9t1tcdes197 db '!uRa9t1tCDeS197CPt7I',0
.data:0040B2FC                                          ; DATA XREF: sub_401E91+D6↑o
.data:0040B311                 align 4
.data:0040B314 ; char aEmcfgv7xc8itav[]
.data:0040B314 aEmcfgv7xc8itav db '!emCFgv7Xc8ItaVGN0bMf',0
.data:0040B314                                          ; DATA XREF: sub_401E91+BC↑o
.data:0040B32A                 align 4
.data:0040B32C ; char aIamsorry123456[]
.data:0040B32C aIamsorry123456 db 'iamsorry!@1234567',0
.data:0040B32C                                          ; DATA XREF: sub_401E91+B1↑o
```

sub_401757에 의해 암호화된 API를 디코딩하여 사용하는 형식
iamsorry!@1234567이라는 문구를 동일하게 사용

활용 사례

Joanap

```
v23 = 0;
strcpy(v21, "1A2z3B4y5C6x7D8w9E0v$F_uGtHsIrJqKpLoMnNmOlPkQjRiShTgUfVeWdXcYbZa");
strcpy(&v22, "9025jhdho39ehe2");
v36 = 0;
v24 = 1;
v25 = 3;
v26 = 7;
v27 = 15;
v28 = 31;
v29 = 63;
if ( !Str )
  return 0;
if ( *Str && *Str == 33 )
{
  v2 = Source;
  Size = strlen(Str);
  if ( !Source )
    v2 = &v22;
  v32 = strlen(v2);
  if ( !v32 )
    return 0;
  v3 = strlen(v2);
  v4 = malloc(v3 + 1);
  v5 = v4;
  v30 = v4;
  strcpy(v4, v2);
  v6 = v32;
  for ( i = 1; i < v6; ++i )
    *(i + v5) += *(i + v5 - 1);
  for ( j = v6 - 1; j >= 1; --j )
    *(v5 + j - 1) += *(v5 + j);
  v9 = malloc(Size);
  v35 = 8;
  v34 = 1;
```

1A2z3B4y5C6x7D8w9E0v$F_uGtHsIrJqKpLoMnNmOlPkQjRiSh

전체 지도 동영상 이미지 뉴스 더보기 설정 도구

검색결과 8개 (0.46초)

Botnet/Cry.cpp at master · malwares/Botnet · GitHub
https://github.com/malwares/Botnet/blob/master/.../Cry.cpp ▾ 이 페이지 번역하기
"**1A2z3B4y5C6x7D8w9E0v$F_uGtHsIrJqKpLoMnNmOlPkQjRiShTgUfVeWdXcYbZa**";. static unsigned char Mask[] = {0x00, 0x01, 0x03, 0x07, 0x0F, 0x1F, 0x3F};.

```
#ifdef PLAIN_CRYPT

static unsigned char SixbitToChar[] =
        "1A2z3B4y5C6x7D8w9E0v$F_uGtHsIrJqKpLoMnNmOlPkQjRiShTgUfVeWdXcYbZa";
static unsigned char Mask[] = {0x00, 0x01, 0x03, 0x07, 0x0F, 0x1F, 0x3F};
char DecryptKey[]="9024jhdho39ehe2";
```

sub_401757 함수를 직접 확인한 결과, key 값으로 사용되는 데이터가 있음
특정 고유한 값을 사용하기 때문에 유니크한 문자열이 될 수 있음

활용 사례

Joanap

IDA Vi… | Pseudoco… | Pseudoco… | Pseudoco… | Pseudoco… | Strings wi… | Hex Vi… | Struct… | En… | Im… | Ex…

```
39  dword_418B48 = (int)GetProcAddress(v2, aUrldownloadtof);
40  v3 = LoadLibraryA(aWininetDll);
41  v4 = v3;
42  if ( !v3 )
43    return 0;
44  dword_418B60 = (int)GetProcAddress(v3, aDeleteurlcache);
45  dword_418B4C = (int)GetProcAddress(v4, aInternetopena);
46  dword_418B24 = (int)GetProcAddress(v4, aInternetopenur);
47  dword_418B1C = (int)GetProcAddress(v4, aInternetcracku);
48  dword_418B28 = (int)GetProcAddress(v4, aInternetreadfi);
49  dword_418B68 = (int)GetProcAddress(v4, aInternetcloseh);
50  v5 = (const CHAR *)sub_401757(aEmcfgv7xc8itav, aIamsorry123456);
51  v6 = LoadLibraryA(v5);
52  if ( !v6 )
53    return 0;
54  v7 = (const CHAR *)sub_401757(aUra9t1tcdes197, aIamsorry123456);
55  dword_418B18 = (int)GetProcAddress(v6, v7);
56  v8 = (const CHAR *)sub_401757(aVwbebxyx1nzrck, aIamsorry123456);
57  dword_418B10 = (int)GetProcAddress(v6, v8);
58  v9 = (const CHAR *)sub_401757(a2fachi224AQ8gs, aIamsorry123456);
59  dword_418B34 = (int)GetProcAddress(v6, v9);
60  v10 = (const CHAR *)sub_401757(aGawd1uiqi6w8kj, aIamsorry123456);
61  dword_418B44 = (int)GetProcAddress(v6, v10);
62  v11 = (const CHAR *)sub_401757(a6ro0eykriqfmph, aIamsorry123456);
63  dword_418B78 = (int)GetProcAddress(v6, v11);
64  v12 = (const CHAR *)sub_401757(aM2mbhjehq7ik6u, aIamsorry123456);
65  dword_418B2C = (int)GetProcAddress(v6, v12);
66  v13 = (const CHAR *)sub_401757(aCrhfex5m9jnzd, aIamsorry123456);
67  dword_418B14 = (int)GetProcAddress(v6, v13);
68  v14 = (const CHAR *)sub_401757(aTlpco4ikblt6jn, aIamsorry123456);
69  dword_418B0C = (int)GetProcAddress(v6, v14);
70  v15 = (const CHAR *)sub_401757(a0e1mfduuanes8y, aIamsorry123456);
71  dword_418B3C = (int)GetProcAddress(v6, v15);
72  v16 = (const CHAR *)sub_401757(aXjkuiwonzthbmw, aIamsorry123456);
73  dword_418B6C = (int)GetProcAddress(v6, v16);
74  v17 = (const CHAR *)sub_401757(aN0u76ngone2y03, aIamsorry123456);
75  dword_418B58 = (int)GetProcAddress(v6, v17);
76  v18 = (const CHAR *)sub_401757(a6y8iuawgbu7Tk, aIamsorry123456);
```

활용 사례

Joanap

Job status	Finished
Rules	rule joanap : joa { meta: tool = "https://github.com/hy00un/Hyara" version = "1.4" date = "2018-09-07" MD5 = "7FE80CEE040...
Creation time	9 8, 2018, 3:28 오전
Start time	9 8, 2018, 6:14 오전
Finish time	9 8, 2018, 9:24 오전
Scanned data	104.1 TB
Scanning speed	9.4 GB/s
Matches	22 Download hashes

Start new job

22개 모두 정상적인 탐지를 하였음

활용 사례

Fallchill

```
dword_41B71C = Decode_sub_401890(v47, aRvawpilxvhhmvn);
dword_41B6E8 = Decode_sub_401890(v47, aSvgfrovplrmgvi);
dword_41B6BC = Decode_sub_401890(v47, aCivagvtllosvok);
dword_41B688 = Decode_sub_401890(v47, aGvgtvnkpagsw);
dword_41B680 = Decode_sub_401890(v47, aCivagvpilxvhhw);
dword_41B6B0 = Decode_sub_401890(v47, aGvgfrovaggiryf);
dword_41B708 = Decode_sub_401890(v47, aGvgllxaotrnv);
dword_41B6FC = Decode_sub_401890(v47, aGvgsbhgvndrivx);
dword_41B60C = Decode_sub_401890(v47, aGvgvlofnvimuli);
dword_41B6DC = Decode_sub_401890(v47, aGvgcfiivmgpilx);
dword_41B72C = Decode_sub_401890(v47, aUmnakvrvdoufro);
dword_41B624 = Decode_sub_401890(v47, aGvgvvihrlmecw);
dword_41B5B0 = Decode_sub_401890(v47, aSvgfrovtrnv);
dword_41B640 = Decode_sub_401890(v47, aGvglltrxaodire);
dword_41B5A8 = Decode_sub_401890(v47, aGvgcfiivmgdriv);
dword_41B590 = Decode_sub_401890(v47, aSvgcfiivmgdriv);
dword_41B5AC = Decode_sub_401890(v47, aOkvmpilxvhh);
dword_41B6A8 = Decode_sub_401890(v47, aCivagvfrovw);
dword_41B6A0 = Decode_sub_401890(v47, aTvinrmagvpilxv);
dword_41B6C0 = Decode_sub_401890(v47, aFivvlryiaib);
```

```
.data:004183C0     aRvawpilxvhhmvn db 'RvawPilxvhhMvnlib',0
.data:004183C0                                             ; D
.data:004183D2                     align 4
.data:004183D4     aMakvrvdoufrov  db 'MakVrvdOuFrov',0    ; D
.data:004183E2                     align 4
.data:004183E4     aSovvk          db 'Sovvk',0            ; D
.data:004183EA                     align 4
.data:004183EC     aPilxvhh32nvcgw db 'Pilxvhh32NvcgW',0   ; D
.data:004183FB                     align 4
.data:004183FC     aWirgvfrov      db 'WirgvFrov',0        ; D
.data:00418406                     align 4
.data:00418408     aGvgmlwfovfrovn db 'GvgMlwfovFrovNanvW',0
.data:00418408                                             ; D
.data:0041841B                     align 4
.data:0041841C     aWargflimfogrko db 'WargFliMfogrkovOyqvxgh'
.data:0041841C                                             ; D
.data:00418433                     align 4
.data:00418434     aWargflisrmtovo db 'WargFliSrmtovOyqvxg',0
.data:00418434                                             ; D
.data:00418448     aLlxaofivv      db 'LlxaoFivv',0        ; D
.data:00418452                     align 4
.data:00418454     aTvinrmagvtsiva db 'TvinrmagvTsivaw',0  ; D
.data:00418464     aGvgfrovtrnv    db 'GvgFrovTrnv',0      ; D
.data:00418470     aGvgecrgclwvtsi db 'GvgEcrgClwvTsivaw',0
.data:00418470                                             ; D
.data:00418482                     align 4
.data:00418484     aLlawlryiaibw   db 'LlawLryiaibW',0     ; D
```

Joanap 케이스와 동일하게 암호화된 API name을 사용

활용 사례

Fallchill

Job status	Finished
Rules	rule fallchill { meta: tool = "https://github.com/hy00un/Hyara" version = "1.4" date = "2018-09-07" MD5 = "A119AE22A15B32...
Creation time	9 8, 2018, 3:30 오전
Start time	9 8, 2018, 6:14 오전
Finish time	9 8, 2018, 9:24 오전
Scanned data	104.1 TB
Scanning speed	9.4 GB/s
Matches	32 Download hashes

Start new job

5

결론

결론

소제목

- 제작한 도구를 이용하여 Yara rule을 간편하고 빠르게 제작할 수 있음
- IDAPython은 참고 문서가 많이 없었지만, 끝내 도구를 완성할 수 있었음
- 와일드 카드 처리에 대한 추가적인 연구가 필요하다고 느꼈음

THANK YOU

Thank you for JH