# VMP Mutation Fix

shhoya.github.io/vmp_vmpmk.html

Feb 17, 2022 / Windows, Reversing, Dev

VMP Mutation 함수 복구

## [0x00] Overview

지속적으로 개발중인 분석 도구( `Shh0ya Rootkit` )에서 사용 중인 기능을 포터블 형식으로 개발하였습니다.

매우 단순한 형식이지만 분석에 매우 강력한 도구가 될 수 있습니다.

## [0x01] Requirements

파일 덤프를 필요로 합니다. 이는 메모리 상에서 덤프를 생성하고 파일 오프셋이 변환된 덤프를 의미합니다.

예로, `Scylla Dump` 와 같습니다. 상세한 내용은 코드를 참조 바랍니다.

## [0x02] Features

`Shh0ya Rootkit` 에서 간략한 기능만 추출하였으며 상세 기능은 제공하지 않습니다.

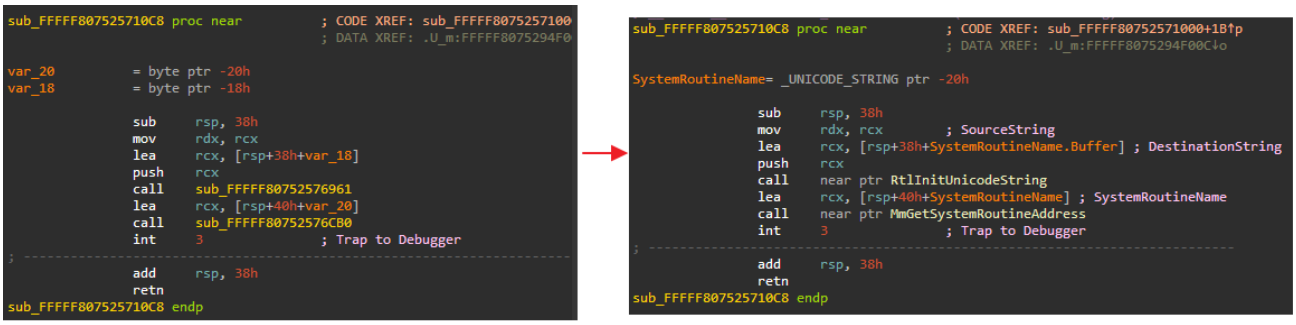
VMP Mutation 이 적용된 API를 복구합니다. 참조

- 32bit 실행 파일 지원
- 64bit 실행 파일 지원
- 64bit 커널 드라이버 지원 (ntoskrnl 한정)

## [0x03] How

`Zydis` 디스어셈블리 엔진을 이용하여 코드 구문을 분석하고 이를 토대로 Mutation이 적용된 API를 찾아 새로운 IAT를 할당하고 계산된 값으로 채워 넣습니다.

```
MutantKiller32.exe <32bit vmp dump> <process id>
MutantKiller64.exe <64bit vmp dump> <process id>
MutantKiller64.exe <64bit vmp driver dump> 4
```



```
sub_FFFFF807525710C8 proc near          ; CODE XREF: sub_FFFFF8075257100
                                        ; DATA XREF: .U_m:FFFFF8075294F0

var_20          = byte ptr -20h
var_18          = byte ptr -18h

                sub     rsp, 38h
                mov     rdx, rcx
                lea     rcx, [rsp+38h+var_18]
                push    rcx
                call    sub_FFFFF80752576961
                lea     rcx, [rsp+40h+var_20]
                call    sub_FFFFF80752576CB0
                int     3               ; Trap to Debugger
; ---------------------------------------------------------------------------
                add     rsp, 38h
                retn
sub_FFFFF807525710C8 endp
```

```
sub_FFFFF807525710C8 proc near          ; CODE XREF: sub_FFFFF80752571000+1B↑p
                                        ; DATA XREF: .U_m:FFFFF8075294F00C↓o

SystemRoutineName= _UNICODE_STRING ptr -20h

                sub     rsp, 38h
                mov     rdx, rcx        ; SourceString
                lea     rcx, [rsp+38h+SystemRoutineName.Buffer] ; DestinationString
                push    rcx
                call    near ptr RtlInitUnicodeString
                lea     rcx, [rsp+40h+SystemRoutineName] ; SystemRoutineName
                call    near ptr MmGetSystemRoutineAddress
                int     3               ; Trap to Debugger
; ---------------------------------------------------------------------------
                add     rsp, 38h
                retn
sub_FFFFF807525710C8 endp
```

## [0x04] Conclusion

소스코드는 아래에서 확인할 수 있습니다.

https://github.com/Shhoya/MutantKiller

**Tags:** Windows Reversing Dev