***Injected DLL Analysis Method - Written by force(forceteam01@gmail.com)***

*** 선행조건**

***IDA*를 이용하여 악성 *DLL*의 정적 분석을 먼저 완료하고 동적 분석을 통하여**
**확인해야 할 코드블럭 또는 변수 값을 확인하고 동적분석 진행 프로세스를 확립한 후**
**동적분석을 시행한다**

***A. Dummy Program Inject***

***1.*** 악성코드로 부터 ***DLL***확보
***2.*** 더미 프로그램을 디버거를 이용해서 아래와 같은 옵션을 주고 실행 및 분석

***2- 1. Break on New Thread***

***2- 1- 01 Ollydbg***를 이용하여 ***Dummy*** 프로그램을 실행한다
***2- 1- 02 Option - > Debugging Options - > Event - > Break on New Thread Option***을 활성화 한다
***2- 1- 03 Dll Inject Program***을 실행한다.
***2- 1- 04*** 악성 ***Dll***을 더미 프로그램에 ***Inject*** 한다
***2- 1- 05 Ollydbg***의 ***"Executable module"*** 창에 악성 ***dll***이 나타날때까지 악성 프로그램을 실행시킨다
***2- 1- 06*** 악성 ***Dll***이 나타나면 ***dll***의 ***Base Address*** 와 ***Entry Point***를 확인한다
***2- 1- 07*** 악성 ***Dll*** 리스트를 마우스 우측 클릭하고 ***"Follow Entry"*** 메뉴를 선택한다
***2- 1- 08*** 화면이 ***CPU*** 화면으로 전환되며 ***Entry Point*** 값에 ***BreakPoint***를 걸어준다

***2- 1- 09 Option - > Debugging Options - > Event - > Break on New Thread Option***을 비활성화 한다
***2- 1- 10*** 악성프로그램을 실행시켜 설정된 악성 ***dll Entry Point bp***에서 실행이 멈추는것을 확인한다
***2- 1- 11*** 악성 ***dll***을 동적 분석한다

## *2- 2. Break on New module(DLL)*

***2- 2- 01 Ollydbg***를 이용하여 ***Dummy*** 프로그램을 실행한다
***2- 2- 02 Option - > Debugging Options - > Event - > Break on New Module Option***을 활성화 한다
***2- 2- 03 Dll Inject Program***을 실행한다.
***2- 2- 04*** 악성 ***Dll***을 더미 프로그램에 ***Inject*** 한다
***2- 2- 05 Ollydbg***의 ***"Executable module"*** 창에 악성 ***dll***이 나타날때까지 악성 프로그램을 실행시킨다
***2- 2- 06*** 악성 ***Dll***이 나타나면 ***dll***의 ***Base Address*** 와 ***Entry Point***를 확인한다
***2- 2- 07*** 악성 ***Dll*** 리스트를 마우스 우측 클릭하고 ***"Follow Entry"*** 메뉴를 선택한다
***2- 2- 08*** 화면이 ***CPU*** 화면으로 전환되며 ***Entry Point*** 값에 ***BreakPoint***를 걸어준다
***2- 2- 09 Option - > Debugging Options - > Event - > Break on New Module Option***을 비활성화 한다
***2- 2- 10*** 악성프로그램을 실행시켜 설정된 악성 ***dll Entry Point bp***에서 실행이 멈추는것을 확인한다
***2- 2- 11*** 악성 ***dll***을 동적 분석한다

***2- 3. DLL Entry Point first byte 0xCC Modify***

***2- 3- 1. Inject***되는 ***DLL***의 ***Entry Point***의 첫째 바이트를 ***CC***로 변경한다***.***
***CC***로 변경하게 되면 어셈블리어는 ***int 3***이 되어서 해당 ***DLL***이 실행될때
***exception***이 발생하게 되어서 프로그램이 멈추게 된다

***2- 3- 2.*** 이때 다시 정상값으로 수정해주면 된다 ***entry point***이기 때문에 대부분
***"55" "push ebp"*** 명령이 된다***.***
주의해야할 사항은 우리가 수정한 값은 원래 ***"push ebp"***이다 해당 명령이 실행되지 않는다
그러므로 분석할때는 다음 명령인 ***"mov ebp,esp"***를 ***push ebp, mov ebp,esp*** 로 변경한다***.***

***2- 3- 3.*** 악성 ***dll***을 동적 분석한다***.***

***B. Target Program Inject***

***1. Target Program*** 에 적용되어 있는 ***Anti- Debug, Anti- Attach*** 기능에 대하여 조사하고 우회한다
***2. Target Program Attach to Debugger***
***3.*** 아래 각 방법에 따른 분석

***2- 1. Break on New Thread***

***2- 1- 01 Ollydbg***를 이용하여 ***Target*** 프로그램을 실행 또는 실행된 ***Target*** 프로그램을 ***Attach*** 한다
***2- 1- 02 Option - > Debugging Options - > Event - > Break on New Thread Option***을 활성화 한다
***2- 1- 03 Dll Inject Program***을 실행한다***.***
***2- 1- 04*** 악성 ***Dll***을 ***Target*** 프로그램에 ***Inject*** 한다

***2- 1- 05 Ollydbg*의 *"Executable module"* 창에 악성 *dll*이 나타날때까지 악성 프로그램을 실행시킨다**

***2- 1- 06* 악성 *Dll*이 나타나면 *dll*의 *Base Address* 와 *Entry Point*를 확인한다**

***2- 1- 07* 악성 *Dll* 리스트를 마우스 우측 클릭하고 *"Follow Entry"* 메뉴를 선택한다**

***2- 1- 08* 화면이 *CPU* 화면으로 전환되며 *Entry Point* 값에 *BreakPoint*를 걸어준다**

***2- 1- 09 Option - > Debugging Options - > Event - > Break on New Thread Option*을 비활성화 한다**

***2- 1- 10* 악성프로그램을 실행시켜 설정된 악성 *dll Entry Point bp*에서 실행이 멈추는것을 확인한다**

***2- 1- 11* 악성 *dll*을 동적 분석한다**

## *2- 2. Break on New module(DLL)*

***2- 2- 01 Ollydbg*를 이용하여 *Target* 프로그램을 실행 또는 실행된 *Target* 프로그램을 *Attach* 한다**

***2- 2- 02 Option - > Debugging Options - > Event - > Break on New Module Option*을 활성화 한다**

***2- 2- 03 Dll Inject Program*을 실행한다.**

***2- 2- 04* 악성 *Dll*을 *Target* 프로그램에 *Inject* 한다**

***2- 2- 05 Ollydbg*의 *"Executable module"* 창에 악성 *dll*이 나타날때까지 악성 프로그램을 실행시킨다**

***2- 2- 06* 악성 *Dll*이 나타나면 *dll*의 *Base Address* 와 *Entry Point*를 확인한다**

***2- 2- 07* 악성 *Dll* 리스트를 마우스 우측 클릭하고 *"Follow Entry"* 메뉴를 선택한다**

***2- 2- 08* 화면이 *CPU* 화면으로 전환되며 *Entry Point* 값에 *BreakPoint*를 걸어준다**

***2- 2- 09 Option - > Debugging Options - > Event - > Break on New Module Option*을 비활성화 한다**

***2- 2- 10* 악성프로그램을 실행시켜 설정된 악성 *dll Entry Point bp*에서 실행이 멈추는것을 확인한다**

***2- 2- 11* 악성 *dll*을 동적 분석한다**

***2- 3. DLL Entry Point first byte 0xCC Modify***

***2- 3- 1. Inject***되는 ***DLL***의 ***Entry Point*** 의 첫째 바이트를 ***CC***로 변경한다***.***
***CC***로 변경하게 되면 어셈블리어는 ***int 3***이 되어서 해당 ***DLL***이 실행될때
***exception***이 발생하게 되어서 프로그램이 멈추게 된다

***2- 3- 2.*** 이때 다시 정상값으로 수정해주면 된다 ***entry point***이기 때문에 대부분
***"55" "push ebp"*** 명령이 된다***.***
주의해야할 사항은 우리가 수정한 값은 원래 ***"push ebp"***이다 해당 명령이 실행되지 않는다
그러므로 분석할때는 다음 명령인 ***"mov ebp,esp"***를 ***push ebp, mov ebp,esp*** 로 변경한다***.***

***2- 3- 3.*** 악성 ***dll***을 동적 분석한다***.***

***#### TO DO ####***

***1.*** 악성 프로그램에서 해당 ***DLL EntryPoint***로 ***Instruction Flow Modify***

***1- 1.*** 전달되는 인자 확인 및 ***Stack*** 구조 확인
***1- 2. Instruction Flow Modify***
***1- 3. DLL Thread Execution***

***2. XP/S2K3/Vista/Win7 Bugs help malware to survive – Codegate 2009 Kris Kaspersky***
참고자료 ***: http://www.window31.com/entry/kasperskyCodegate2009***