Immunity Debugger:

Pattern Offset Script

Rajendra Umadas

D1AB1069

Devanand Singh

krypt0niC

Note: We’re not affiliated with Immunity Inc., we just like their slides

Overview

- What is Immunity Debugger?
- Simple Questions
- Plan of Attack
- Toolset: Immunity Debugger Python API
- Demo!!!
- Future Expansion

Immunity Debugger

- A debugger specifically designed for security professionals
- Shortens exploit development time
- Simple intuitive interfaces
- Fully integrated Python scripting engine
- Lightweight and fast debugging
- Connectivity to fuzzers and exploit development tools

A Few Simple Questions

- What are a few steps towards a shell? (Dino’s Six Steps)
    - Fuzz to trigger the vulnerability (Fuzzer/Debugger)
    - Enumerate bad characters (Metasploit/Debugger)
    - Find interesting offsets in attack vector (Metasploit/Debugger)
    - Load payload to test for executable memory space (Debugger)
    - Discover the size of our memory buffer (Debugger)
    - Generate payload and OWN!!! (Metasploit)

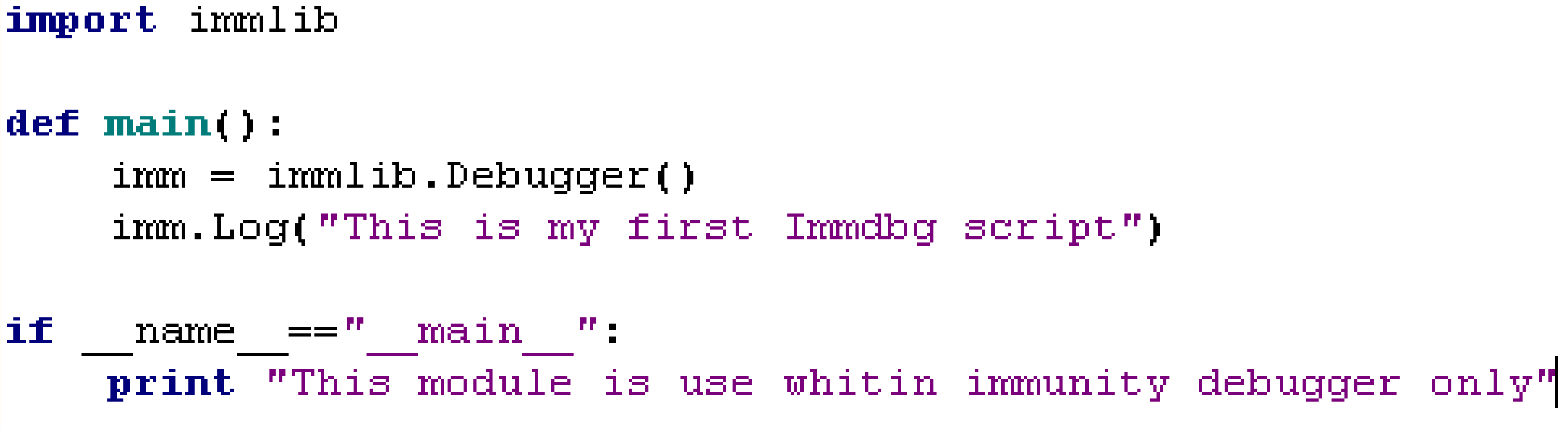
A Few Simple Questions

- How can we optimize this process?
  - ➢ Allow our tools to work together
    - Fuzzer
    - Debugger
    - Metasploit
- Where should we start?
  - ➢ Where ever you like…
    - Metasploit/Debugger interface

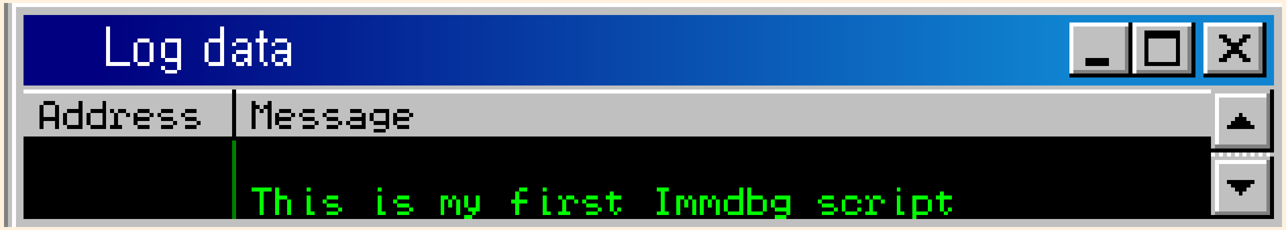
Plan of Attack

- Metasploit/Debugger interface
  - ➢ Find interesting offsets in attack vector
    - “PatternCreate” in Metasploit
    - Trigger Vulnerability while debugging
    - Capture bytes in all interesting registers and memory locations
    - “PatternOffset” to find the locations within our attack vector

Toolset: Immunity Debugger Python API



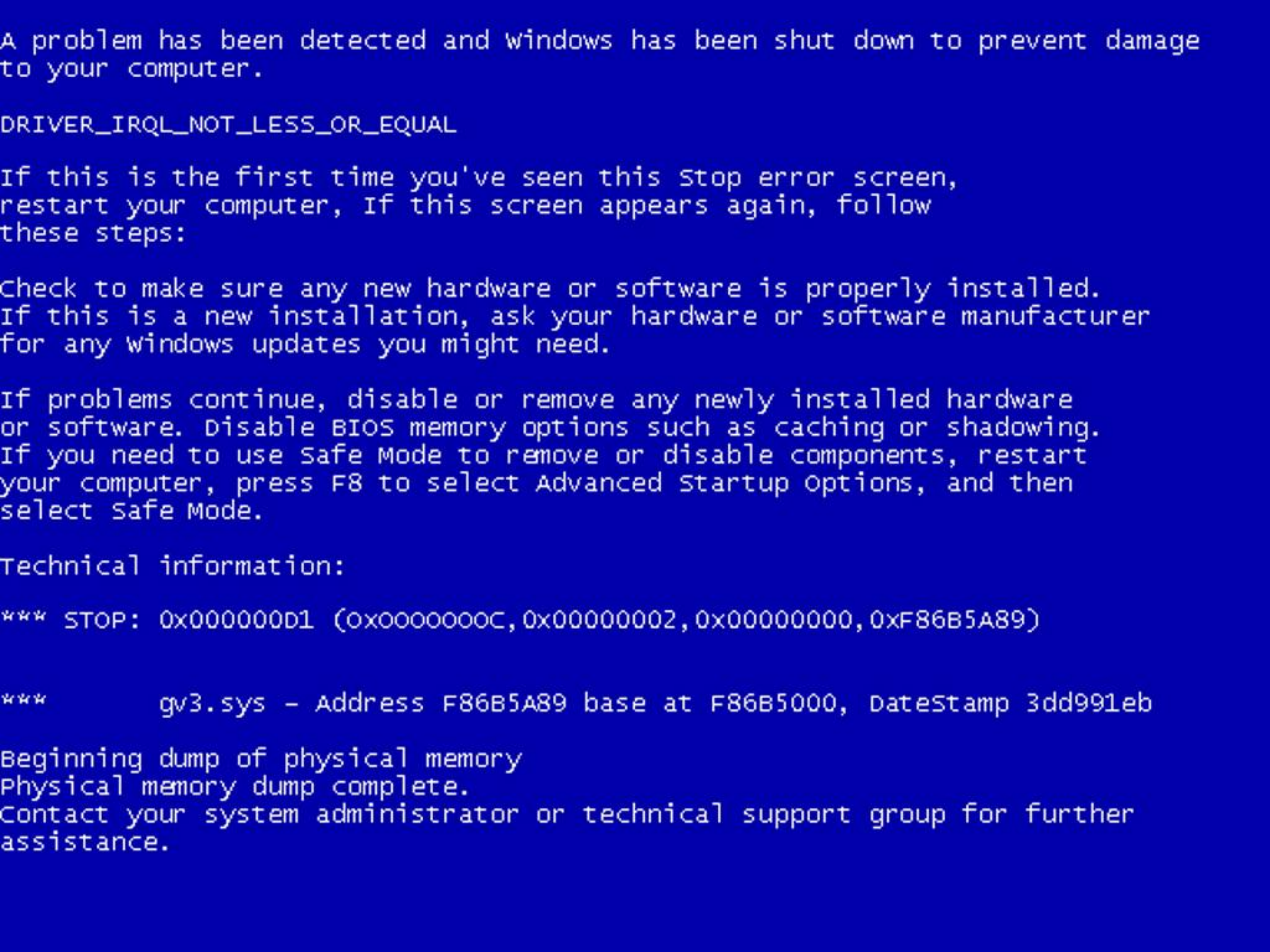
```
import immlib

def main():
    imm = immlib.Debugger()
    imm.Log("This is my first Immdbg script")

if __name__=="__main__":
    print "This module is use whitin immunity debugger only"
```



```
This is my first Immdbg script
```

Toolset: Immunity Debugger Python API

- Register Access
  - getRegs()
    - Returns an associative array of registers and the values stored within them
- Memory Access
  - readMemory(address, size)
    - Returns value pointed to by *address* (*size* # of bytes)
  - readLong(address)
    - Returns value pointed to by *address* (*4* bytes)



```
A problem has been detected and windows has been shut down to prevent damage
to your computer.

DRIVER_IRQL_NOT_LESS_OR_EQUAL

If this is the first time you've seen this Stop error screen,
restart your computer, If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x000000D1 (0x0000000C,0x00000002,0x00000000,0xF86B5A89)


***        gv3.sys - Address F86B5A89 base at F86B5000, DateStamp 3dd991eb

Beginning dump of physical memory
Physical memory dump complete.
Contact your system administrator or technical support group for further
assistance.
```

Future Expansion

- How can we FURTHER optimize this process?
  - ➢ Allow our tools to work together
    - Fuzzer
    - Debugger
    - Metasploit

- Where should we END?
  - ➢ …………..

AUTOSPLOIT!!!

Special Thanks To

Dan Guido
Dino Dai Zovi
Dean De Beer
Erik Cabetas
Mike Zusman
Stephen A. Ridely

Questions?

Rajendra Umadas
http://twitter.com/D1AB1069
http://rajweb.net

Devanand Singh
http://twitter.com/krypt0nic

IMMUNITY
DEBUGGER