# Project Boomerang

*"Nostalgia is like a boomerang. It all just comes back ..."*

1.11.2019

Danda
Dandaware
Melbourne

## Story

Project Boomerang came to be due to me (Danda) searched through the GOG (Good Old Games) store, in search for a game to replace my desire to replay the original Mechwarrior 3 video game.
The concept and mechanics found within this game (Parkan: Iron Strategy) was a breath of fresh air, and I wanted to play more of it, if it were not for the crashes and incredibly low frame-rate. Thus, Project Boomerang was born.

## Goals

1. Investigate and comprehend the Iron3D.dll
2. Reverse Engineer Iron3D.dll and develop a DLL Injector for run-time to optimise frame-rate issues and screen properties/dimensions.

## Specifications

Employing multiple EXE and DLL softwares, to ensure consistency and to validate/confirm code found from the original resource.

### Ghidra

Ghidra will be the main debugging and reversing tool for this project, due to how well developed this program is (developed by the American Government branch - the NSA) we can utilize this software for our reconnaissance.

### X92DBG (X64DBG/X32DBG)

This will provide additional assistance in confirming the reversal or review of a code, to ensure that Assembly registers and values match and are not incorrect/false.

### OllyDBG

Useful for ensuring both Ghidra and X96DBG are both correct or to confirm biases.

### Radare2

Radare2 (version 3.3.0) will allow us to do furthermore debugging and reversal of encrypted/digested values.

## Visual Studio (IDE)

Develop the DLL here with the available DLL template and allow for shared solutions.

## Dependency Walker

Provides additional assistance in realizing the dependencies of a program.

## PEBrowser Pro

Works similarly to Dependency Walker, but this program can seek and peek inside the innards of a multiple file types (DLL/EXE ...)

## Milestones

### I. Iron3D.dll Review and Reversal

Review, comprehend and compose a report as we reverse Iron3D.dll, and hopefully achieve 10KB per weekend or so.

### II. DLL Injector

Develop a run-time DLL (Dynamic-link Library) Injector - optimising frame-rate and allow consistent updating of application window properties/dimensions.

### III. SYS.LIB file re-use

Be able to unpack the SYS.LIB found in “Parkan - Iron Strategy” and implement the possible header files accompanied inside the package.

### IV. Develop a re-purpose-able DTI

We are developing a run-time Dynamic-link Library, we need to be able to repurpose the project for different programming languages - although having at run-time may prove this to be ‘challenging’.

### V. Enable project for Open-Source

For the final milestone, we release the project for open-source under the GNU License (version 3) — enable other open-source teams to commercialise this project for a living and further development of the Direct Tool Help.

## Iron Direct Tool Injector

Employing C++ we need to first develop a program which manipulates a snapshot of the system. Once we reverse the IRON3D.DLL, the snapshot DLL program will allow us to manipulate virtual memory, hopefully reproducing something close to a stutter fixer/remover when it comes down to the operations of the Thread · Memory · Queue & Heap with the use types such as HANDLE and ProcessID · ProcessEntry ... etcetera.

Later in the project, the design of the DLL will require Assembly code for the Read/Write operations of the program as well as the inputs and outputs of binaries — we must keep in mind the incorporation and integration of Assembly in mind when developing the DTI. (Although if this scope is forgotten, we can repurpose the code for straight C++)

With the use of the Windows ToolHelp32 header file, we can create snapshot(s) of the system, including the ability to manipulate virtual memory with the DLL template found in the Visual Studio (2017) IDE.

Iron3d.dll and the goggame-.dll file utilized the "Kernel32.dll" found in Windows operating systems, as well as the "ProcessThreadApi.h".

Wikipedia webpage on "Hooking" with the Windows API
https://en.wikipedia.org/wiki/Hooking#Windows

Faking Kernel32.DLL by Alexey Lyashkov
https://syprog.blogspot.com/2012/03/faking-kernel32dll-amateur-sandbox.htmlhttps://syprog.blogspot.com/2012/03/faking-kernel32dll-amateur-sandbox.html

## Resources

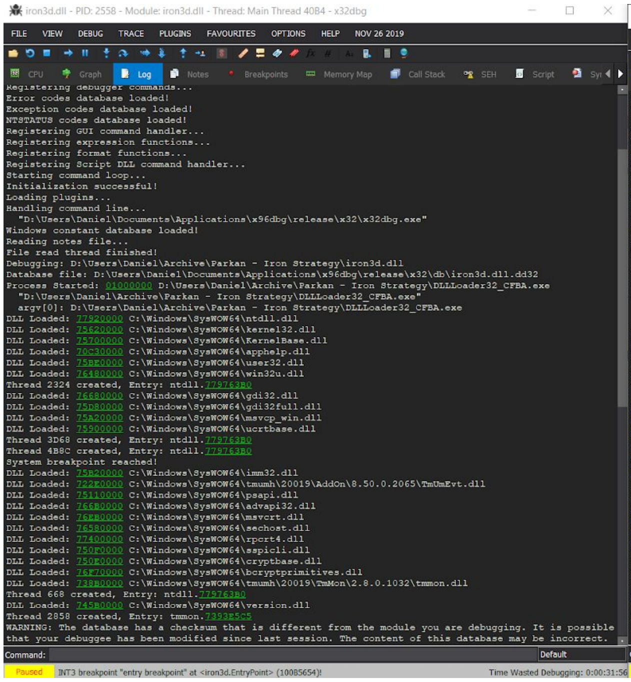
Tool Help 32 - Taking a Snapshot and Viewing Processes

https://docs.microsoft.com/en-us/windows/win32/toolhelp/taking-a-snapshot-and-viewing-processes

Creating a Simple Dynamic-link Library

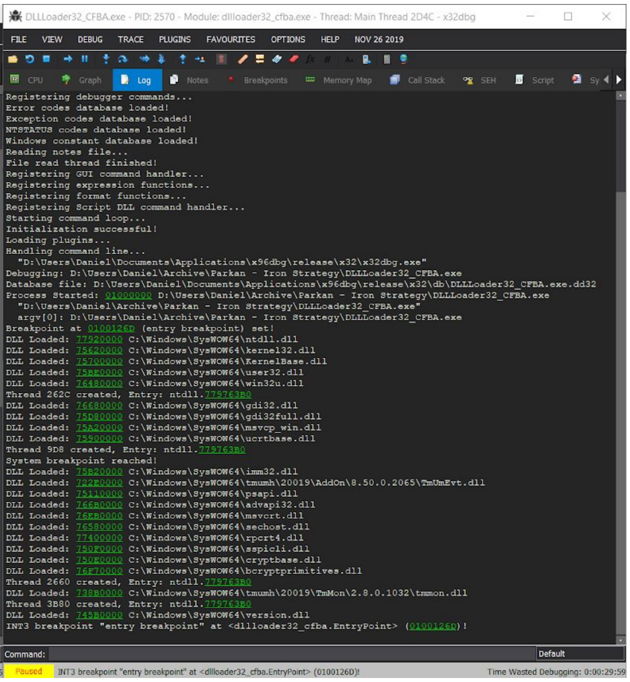
https://docs.microsoft.com/en-us/windows/win32/dlls/creating-a-simple-dynamic-link-library

## Step-through


iron3d.dll - PID: 2558 - Module: iron3d.dll - Thread: Main Thread 40B4 - x32dbg
FILE VIEW DEBUG TRACE PLUGINS FAVOURITES OPTIONS HELP NOV 26 2019
CPU Graph Log Notes Breakpoints Memory Map Call Stack SEH Script Syı
Command:
Default
Paused

```
Registering debugger commands...
Error codes database loaded!
Exception codes database loaded!
NTSTATUS codes database loaded!
Registering GUI command handler...
Registering expression functions...
Registering format functions...
Registering Script DLL command handler...
Starting command loop...
Initialization successful!
Loading plugins...
Handling command line...
  "D:\Users\Daniel\Documents\Applications\x96dbg\release\x32\x32dbg.exe"
Windows constant database loaded!
Reading notes file...
File read thread finished!
Debugging: D:\Users\Daniel\Archive\Parkan - Iron Strategy\iron3d.dll
Database file: D:\Users\Daniel\Documents\Applications\x96dbg\release\x32\db\iron3d.dll.dd32
Process Started: 01000000 D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe
  "D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe"
  argv[0]: D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe
DLL Loaded: 77920000 C:\Windows\SysWOW64\ntdll.dll
DLL Loaded: 75620000 C:\Windows\SysWOW64\kernel32.dll
DLL Loaded: 75700000 C:\Windows\SysWOW64\KernelBase.dll
DLL Loaded: 70C30000 C:\Windows\SysWOW64\apphelp.dll
DLL Loaded: 75BE0000 C:\Windows\SysWOW64\user32.dll
DLL Loaded: 76480000 C:\Windows\SysWOW64\win32u.dll
Thread 2324 created, Entry: ntdll.779763B0
DLL Loaded: 76680000 C:\Windows\SysWOW64\gdi32.dll
DLL Loaded: 75D80000 C:\Windows\SysWOW64\gdi32full.dll
DLL Loaded: 75A20000 C:\Windows\SysWOW64\msvcp_win.dll
DLL Loaded: 75900000 C:\Windows\SysWOW64\ucrtbase.dll
Thread 3D68 created, Entry: ntdll.779763B0
Thread 4B8C created, Entry: ntdll.779763B0
System breakpoint reached!
DLL Loaded: 75B20000 C:\Windows\SysWOW64\imm32.dll
DLL Loaded: 722E0000 C:\Windows\SysWOW64\tmumh\20019\AddOn\8.50.0.2065\TmUmEvt.dll
DLL Loaded: 75110000 C:\Windows\SysWOW64\psapi.dll
DLL Loaded: 766B0000 C:\Windows\SysWOW64\advapi32.dll
DLL Loaded: 76EB0000 C:\Windows\SysWOW64\msvcrt.dll
DLL Loaded: 76580000 C:\Windows\SysWOW64\sechost.dll
DLL Loaded: 77400000 C:\Windows\SysWOW64\rpcrt4.dll
DLL Loaded: 750F0000 C:\Windows\SysWOW64\sspicli.dll
DLL Loaded: 750E0000 C:\Windows\SysWOW64\cryptbase.dll
DLL Loaded: 76F70000 C:\Windows\SysWOW64\bcryptprimitives.dll
DLL Loaded: 738B0000 C:\Windows\SysWOW64\tmumh\20019\TmMon\2.8.0.1032\tmmon.dll
Thread 668 created, Entry: ntdll.779763B0
DLL Loaded: 745B0000 C:\Windows\SysWOW64\version.dll
Thread 2858 created, Entry: tmmon.7393E5C5
WARNING: The database has a checksum that is different from the module you are debugging. It is possible
that your debuggee has been modified since last session. The content of this database may be incorrect.

INT3 breakpoint "entry breakpoint" at <iron3d.EntryPoint> (10085654)!    Time Wasted Debugging: 0:00:31:56
```


DLLLoader32_CFBA.exe - PID: 2570 - Module: dllloader32_cfba.exe - Thread: Main Thread 2D4C - x32dbg
FILE VIEW DEBUG TRACE PLUGINS FAVOURITES OPTIONS HELP NOV 26 2019
CPU Graph Log Notes Breakpoints Memory Map Call Stack SEH Script Sy
Command:
Default
Paused

```
Registering debugger commands...
Error codes database loaded!
Exception codes database loaded!
NTSTATUS codes database loaded!
Windows constant database loaded!
Reading notes file...
File read thread finished!
Registering GUI command handler...
Registering expression functions...
Registering format functions...
Registering Script DLL command handler...
Starting command loop...
Initialization successful!
Loading plugins...
Handling command line...
  "D:\Users\Daniel\Documents\Applications\x96dbg\release\x32\x32dbg.exe"
Debugging: D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe
Database file: D:\Users\Daniel\Documents\Applications\x96dbg\release\x32\db\DLLLoader32_CFBA.exe.dd32
Process Started: 01000000 D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe
  "D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe"
  argv[0]: D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe
Breakpoint at 0100126D (entry breakpoint) set!
DLL Loaded: 77920000 C:\Windows\SysWOW64\ntdll.dll
DLL Loaded: 75620000 C:\Windows\SysWOW64\kernel32.dll
DLL Loaded: 75700000 C:\Windows\SysWOW64\KernelBase.dll
DLL Loaded: 75BE0000 C:\Windows\SysWOW64\user32.dll
DLL Loaded: 76480000 C:\Windows\SysWOW64\win32u.dll
Thread 262C created, Entry: ntdll.779763B0
DLL Loaded: 76680000 C:\Windows\SysWOW64\gdi32.dll
DLL Loaded: 75D80000 C:\Windows\SysWOW64\gdi32full.dll
DLL Loaded: 75A20000 C:\Windows\SysWOW64\msvcp_win.dll
DLL Loaded: 75900000 C:\Windows\SysWOW64\ucrtbase.dll
Thread 9D8 created, Entry: ntdll.779763B0
System breakpoint reached!
DLL Loaded: 75B20000 C:\Windows\SysWOW64\imm32.dll
DLL Loaded: 722E0000 C:\Windows\SysWOW64\tmumh\20019\AddOn\8.50.0.2065\TmUmEvt.dll
DLL Loaded: 75110000 C:\Windows\SysWOW64\psapi.dll
DLL Loaded: 766B0000 C:\Windows\SysWOW64\advapi32.dll
DLL Loaded: 76EB0000 C:\Windows\SysWOW64\msvcrt.dll
DLL Loaded: 76580000 C:\Windows\SysWOW64\sechost.dll
DLL Loaded: 77400000 C:\Windows\SysWOW64\rpcrt4.dll
DLL Loaded: 750F0000 C:\Windows\SysWOW64\sspicli.dll
DLL Loaded: 750E0000 C:\Windows\SysWOW64\cryptbase.dll
DLL Loaded: 76F70000 C:\Windows\SysWOW64\bcryptprimitives.dll
Thread 2660 created, Entry: ntdll.779763B0
DLL Loaded: 738B0000 C:\Windows\SysWOW64\tmumh\20019\TmMon\2.8.0.1032\tmmon.dll
Thread 3B80 created, Entry: ntdll.779763B0
DLL Loaded: 745B0000 C:\Windows\SysWOW64\version.dll
INT3 breakpoint "entry breakpoint" at <dllloader32_cfba.EntryPoint> (0100126D)!

INT3 breakpoint "entry breakpoint" at <dllloader32_cfba.EntryPoint> (0100126D)!    Time Wasted Debugging: 0:00:29:59
```

Debugging: D:\Users\Daniel\Archive\Parkan - Iron Strategy\iron3d.dll

Database file: D:\Users\Daniel\Documents\Applications\x96dbg\release\x32\db\iron3d.dll.dd32

Process Started: 01000000 D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe

"D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe"

argv[0]: D:\Users\Daniel\Archive\Parkan - Iron Strategy\DLLLoader32_CFBA.exe

```
DLL Loaded: 77920000 C:\Windows\SysWOW64\ntdll.dll
DLL Loaded: 75620000 C:\Windows\SysWOW64\kernel32.dll
DLL Loaded: 75700000 C:\Windows\SysWOW64\KernelBase.dll
DLL Loaded: 70C30000 C:\Windows\SysWOW64\apphelp.dll
DLL Loaded: 75BE0000 C:\Windows\SysWOW64\user32.dll
DLL Loaded: 76480000 C:\Windows\SysWOW64\win32u.dll
Thread 2324 created, Entry: ntdll.779763B0
DLL Loaded: 76680000 C:\Windows\SysWOW64\gdi32.dll
DLL Loaded: 75D80000 C:\Windows\SysWOW64\gdi32full.dll
DLL Loaded: 75A20000 C:\Windows\SysWOW64\msvcp_win.dll
DLL Loaded: 75900000 C:\Windows\SysWOW64\ucrtbase.dll
Thread 3D68 created, Entry: ntdll.779763B0
System breakpoint reached!
```

“DLLLoader32_CFBA.exe” is made by X92DBG which proves to be an interesting executable the entry point is reached.

Process Started: 01000000 The process starting at this address is interesting, this is an address which is a real low address.

Thread 3D68 created, Entry: ntdll.779763B0 With Thread 3D68 having it’s Entry in ntdll.779763B0 shows that the engine is reliant on accessing the NT Layer of the Windows architecture, thus creating processes to interact with the Hardware Abstraction Layer (HAL).

Oddly enough “DLLLOader32_CFBA.exe” is just an executable which interacts with the Windows NT architecture to allow it go beneath the surface and work effectively within Windows. Although this lets us understand as to what modules/DLLs are needed to be employed for a Windows architecture.

The real icing on the cake is the “Ngi32.dll” which has the same DLL initialization as the DLL Loader executable before a system breakpoint is reached. But what interests me is the fact in which it asks for numerous more DLLs in order to function.

### Iron3d.dll -PID: 2558 - Module: iron3d.dll - Thread: Main Thread 40B4

### Ngi32.dll -PID: 384C - Module: ngi32.dll - Thread: Main Thread 440

The difference between the “iron3d.dll” and the “ngi32.dll” is that “ngi32.dll” contains the “imports” and “exports” of the Nikita 3D engine.

Ngi32.dll

- Project File Name: ngi32.dll
- Last Modified: Mon Dec 09 20:47:08 AEDT 2019
- Readonly: false
- Program Name: ngi32.dll
- Language ID: x86:LE:32:default (2.8)
- Compiler ID: windows
- Processor: x86
- Endian: Little
- Address Size: 32
- Minimum Address: 10000000
- Maximum Address: 10042101
- # of Bytes: 262727
- # of Memory Blocks: 7
- # of Instructions: 0
- # of Defined Data: 5040
- # of Functions: 32
- # of Symbols: 399
- # of Data Types: 45
- # of Data Type Categories: 3
- Comments: This product was designed & written by Alexander I. Okroug
- CompanyName: Nikita Ltd.
- Compiler: visualstudio:unknown
- Created With Ghidra Version: 9.0.4
- Date Created: Mon Dec 09 20:47:07 AEDT 2019
- Executable Format: Portable Executable (PE)
- Executable Location: D:/Users/Daniel/Archive/Parkan - Iron Strategy/Ngi32.dll
- Executable MD5: 09d87bb7c8d2b5eca253cb729afa62c7
- FSRL: file://D:/Users/Daniel/Archive/Parkan - Iron Strategy/Ngi32.dll?MD5=09d87bb7c8d2b5eca253cb729afa62c7
- FileDescription: Nikita Game Interface DLL
- FileVersion: 3.00
- InternalName: Nikita Game Interface Library
- LegalCopyright: Copyright © Nikita Ltd. 1993-2001
- LegalTrademarks: Nikita
- OriginalFilename: NGI32.DLL
- PrivateBuild: 003
- ProductName: Nikita Game Interface Library
- ProductVersion: 1, 10, 0, 141
- Relocatable: true
- SectionAlignment: 4096
- SpecialBuild: 003
- Translation: 4e30409