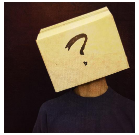
Getting Hooked on Profiling

Markerless Dynamic Analysis of C++ Executables

Pierric Gimmig - September 2017

Who am I 

Pierric Gimmig, Animation/Physics/System Programmer

kaydara

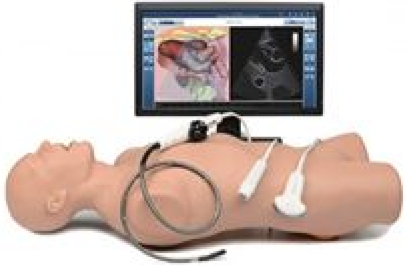
UBISOFT

TEAM BONDI
SYDNEY • AUSTRALIA

CAE Healthcare

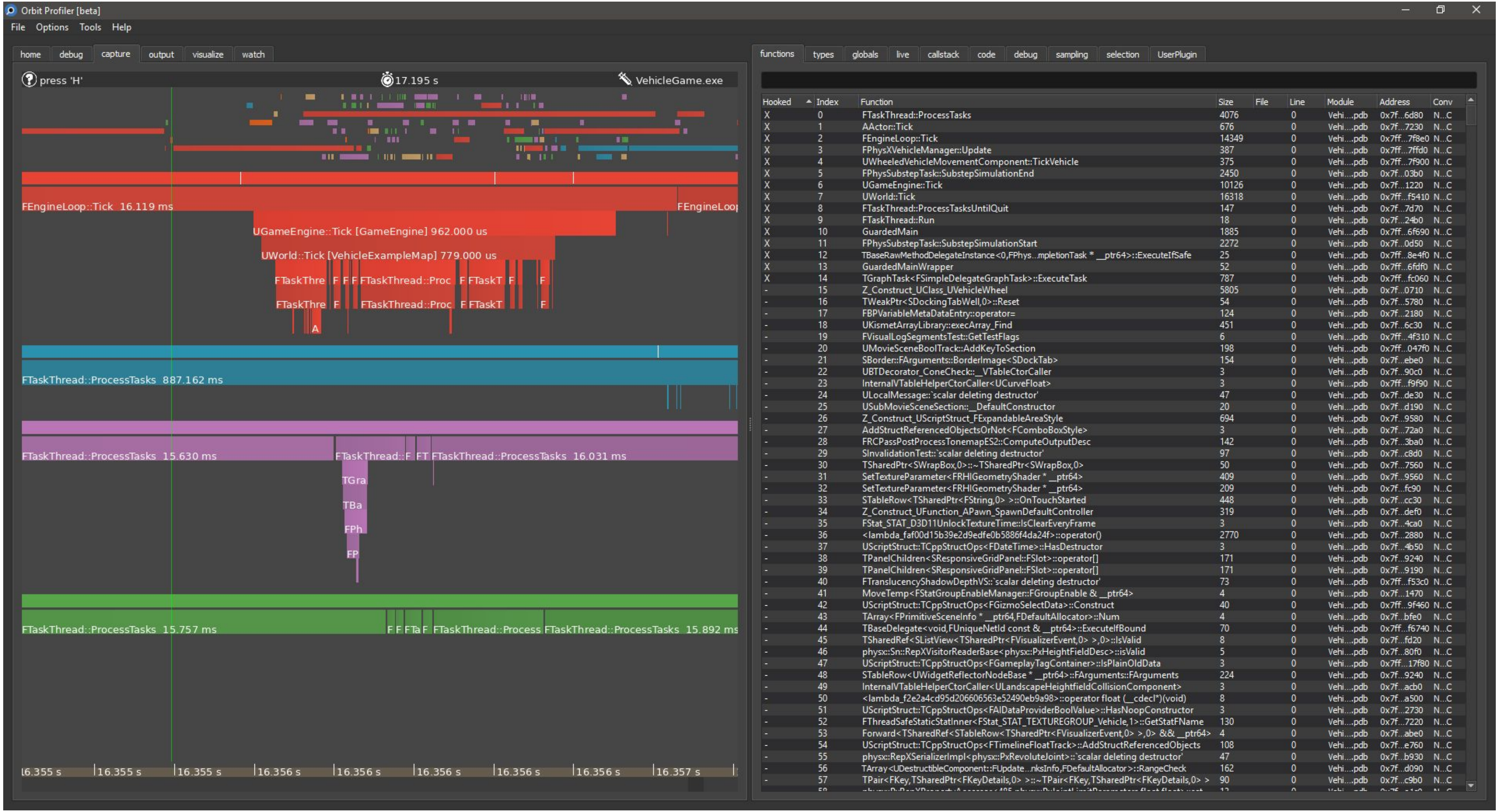
WB GAMES MONTRÉAL

EPIC GAMES



orbit

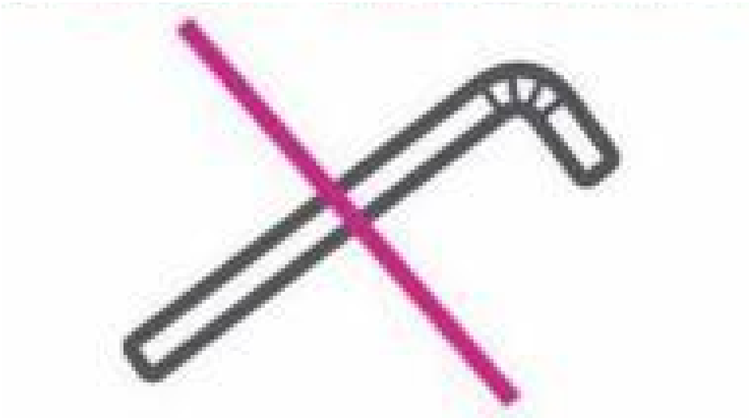
PROFILER



Hooked	Index	Function	Size	File	Line	Module	Address	Conv
X	0	FTaskThread::ProcessTasks	4076		0	Vehi....pdb	0x7f...6d80	N...C
X	1	AActor::Tick	676		0	Vehi....pdb	0x7f...7230	N...C
X	2	FEngineLoop::Tick	14349		0	Vehi....pdb	0x7ff...7f8e0	N...C
X	3	FPhysXVehicleManager::Update	387		0	Vehi....pdb	0x7ff...7ffd0	N...C
X	4	UWheeledVehicleMovementComponent::TickVehicle	375		0	Vehi....pdb	0x7ff...7f900	N...C
X	5	FPhysSubstepTask::SubstepSimulationEnd	2450		0	Vehi....pdb	0x7f...03b0	N...C
X	6	UGameEngine::Tick	10126		0	Vehi....pdb	0x7f...1220	N...C
X	7	UWorld::Tick	16318		0	Vehi....pdb	0x7ff...f5410	N...C
X	8	FTaskThread::ProcessTasksUntilQuit	147		0	Vehi....pdb	0x7f...7d70	N...C
X	9	FTaskThread::Run	18		0	Vehi....pdb	0x7f...24b0	N...C
X	10	GuardedMain	1885		0	Vehi....pdb	0x7ff...6f690	N...C
X	11	FPhysSubstepTask::SubstepSimulationStart	2272		0	Vehi....pdb	0x7f...0d50	N...C
X	12	TBaseRawMethodDelegateInstance<0,FPhys...mpletionTask * __ptr64>::ExecuteIfSafe	25		0	Vehi....pdb	0x7ff...8e4f0	N...C
X	13	GuardedMainWrapper	52		0	Vehi....pdb	0x7ff...6fdf0	N...C
X	14	TGraphTask<FSimpleDelegateGraphTask>::ExecuteTask	787		0	Vehi....pdb	0x7ff...fc060	N...C
-	15	Z_Construct_UClass_UVehicleWheel	5805		0	Vehi....pdb	0x7f...0710	N...C
-	16	TWeakPtr<SDockingTabWell,0>::Reset	54		0	Vehi....pdb	0x7f...5780	N...C
-	17	FBPVariableMetaDataEntry::operator=	124		0	Vehi....pdb	0x7f...2180	N...C
-	18	UKismetArrayLibrary::execArray_Find	451		0	Vehi....pdb	0x7f...6c30	N...C
-	19	FVisualLogSegmentsTest::GetTestFlags	6		0	Vehi....pdb	0x7ff...4f310	N...C
-	20	UMovieSceneBoolTrack::AddKeyToSection	198		0	Vehi....pdb	0x7ff...047f0	N...C
-	21	SBorder::FArguments::BorderImage<SDockTab>	154		0	Vehi....pdb	0x7f...ebe0	N...C
-	22	UBTDecorator_ConeCheck::__VTableCtorCaller	3		0	Vehi....pdb	0x7f...90c0	N...C
-	23	InternalVTableHelperCtorCaller<UCurveFloat>	3		0	Vehi....pdb	0x7ff...f9f90	N...C
-	24	ULocalMessage::`scalar deleting destructor'	47		0	Vehi....pdb	0x7f...de30	N...C
-	25	USubMovieSceneSection::__DefaultConstructor	20		0	Vehi....pdb	0x7f...d190	N...C
-	26	Z_Construct_UScriptStruct_FExpandableAreaStyle	694		0	Vehi....pdb	0x7f...9580	N...C
-	27	AddStructReferencedObjectsOrNot<FComboBoxStyle>	3		0	Vehi....pdb	0x7f...72a0	N...C
-	28	FRCPassPostProcessTonemapES2::ComputeOutputDesc	142		0	Vehi....pdb	0x7f...3ba0	N...C
-	29	SInvalidationTest::`scalar deleting destructor'	97		0	Vehi....pdb	0x7f...c8d0	N...C
-	30	TSharedPtr<SWrapBox,0>::~TSharedPtr<SWrapBox,0>	50		0	Vehi....pdb	0x7f...7560	N...C
-	31	SetTextureParameter<FRHIGeometryShader * __ptr64>	409		0	Vehi....pdb	0x7f...9560	N...C
-	32	SetTextureParameter<FRHIGeometryShader * __ptr64>	209		0	Vehi....pdb	0x7f...fc90	N...C
-	33	STableRow<TSharedPtr<FString,0> >::OnTouchStarted	448		0	Vehi....pdb	0x7f...cc30	N...C
-	34	Z_Construct_UFunction_APawn_SpawnDefaultController	319		0	Vehi....pdb	0x7f...def0	N...C
-	35	FStat_STAT_D3D11UnlockTextureTime::IsClearEveryFrame	3		0	Vehi....pdb	0x7f...4ca0	N...C
-	36	<lambda_faf00d15b39e2d9edfe0b5886f4da24f>::operator()	2770		0	Vehi....pdb	0x7f...2880	N...C
-	37	UScriptStruct::TCppStructOps<FDateTime>::HasDestructor	3		0	Vehi....pdb	0x7f...4b50	N...C
-	38	TPanelChildren<SResponsiveGridPanel::FSlot>::operator[]	171		0	Vehi....pdb	0x7f...9240	N...C
-	39	TPanelChildren<SResponsiveGridPanel::FSlot>::operator[]	171		0	Vehi....pdb	0x7f...9190	N...C
-	40	FTranslucencyShadowDepthVS::`scalar deleting destructor'	73		0	Vehi....pdb	0x7ff...f53c0	N...C
-	41	MoveTemp<FStatGroupEnableManager::FGroupEnable & __ptr64>	4		0	Vehi....pdb	0x7f...1470	N...C
-	42	UScriptStruct::TCppStructOps<FGizmoSelectData>::Construct	40		0	Vehi....pdb	0x7ff...9f460	N...C
-	43	TArray<FPrimitiveSceneInfo * __ptr64,FDefaultAllocator>::Num	4		0	Vehi....pdb	0x7f...bfe0	N...C
-	44	TBaseDelegate<void,FUniqueNetId const & __ptr64>::ExecuteIfBound	70		0	Vehi....pdb	0x7ff...f6740	N...C
-	45	TSharedRef<SListView<TSharedPtr<FVisualizerEvent,0> >,0>::IsValid	8		0	Vehi....pdb	0x7f...fd20	N...C
-	46	physx::Sn::RepXVisitorReaderBase<physx::PxHeightFieldDesc>::isValid	5		0	Vehi....pdb	0x7f...80f0	N...C
-	47	UScriptStruct::TCppStructOps<FGameplayTagContainer>::IsPlainOldData	3		0	Vehi....pdb	0x7ff...17f80	N...C
-	48	STableRow<UWidgetReflectorNodeBase * __ptr64>::FArguments::FArguments	224		0	Vehi....pdb	0x7f...9240	N...C
-	49	InternalVTableHelperCtorCaller<ULandscapeHeightfieldCollisionComponent>	3		0	Vehi....pdb	0x7f...acb0	N...C
-	50	<lambda_f2e2a4cd95d206606563e52490eb9a98>::operator float (__cdecl*)(void)	8		0	Vehi....pdb	0x7f...a500	N...C
-	51	UScriptStruct::TCppStructOps<FAIDataProviderBoolValue>::HasNoopConstructor	3		0	Vehi....pdb	0x7f...2730	N...C
-	52	FThreadSafeStaticStatInner<FStat_STAT_TEXTUREGROUP_Vehicle,1>::GetStatFName	130		0	Vehi....pdb	0x7f...7220	N...C
-	53	Forward<TSharedRef<STableRow<TSharedPtr<FVisualizerEvent,0> >,0> && __ptr64>	4		0	Vehi....pdb	0x7f...abe0	N...C
-	54	UScriptStruct::TCppStructOps<FTimelineFloatTrack>::AddStructReferencedObjects	108		0	Vehi....pdb	0x7f...e760	N...C
-	55	physx::RepXSerializerImpl<physx::PxRevoluteJoint>::`scalar deleting destructor'	47		0	Vehi....pdb	0x7f...b930	N...C
-	56	TArray<UDestructibleComponent::FUpdate...nksInfo,FDefaultAllocator>::RangeCheck	162		0	Vehi....pdb	0x7f...d090	N...C
-	57	TPair<FKey,TSharedPtr<FKeyDetails,0> >::~TPair<FKey,TSharedPtr<FKeyDetails,0> >	90		0	Vehi....pdb	0x7f...c9b0	N...C

STRAWS ARE
FOR SUCKERS!

REFUSE THE STRAW



(Because straws are for suckers)

What’s readily available?

CPU Sampling

- Very Sleepy
- Visual Studio
- XPerf (ETW)
- VTune

Manual Instrumentation (Code Markers)

- Telemetry ($$$)
- In-house visualizers (Ubisoft’s Pel Viewer)
- (PIX)

Existing Techniques

Pros

Sampling

- No code modification
- Fast iteration time

Manual Instrumentation

- Exact function stats
- Invaluable flame chart view

Cons

Sampling

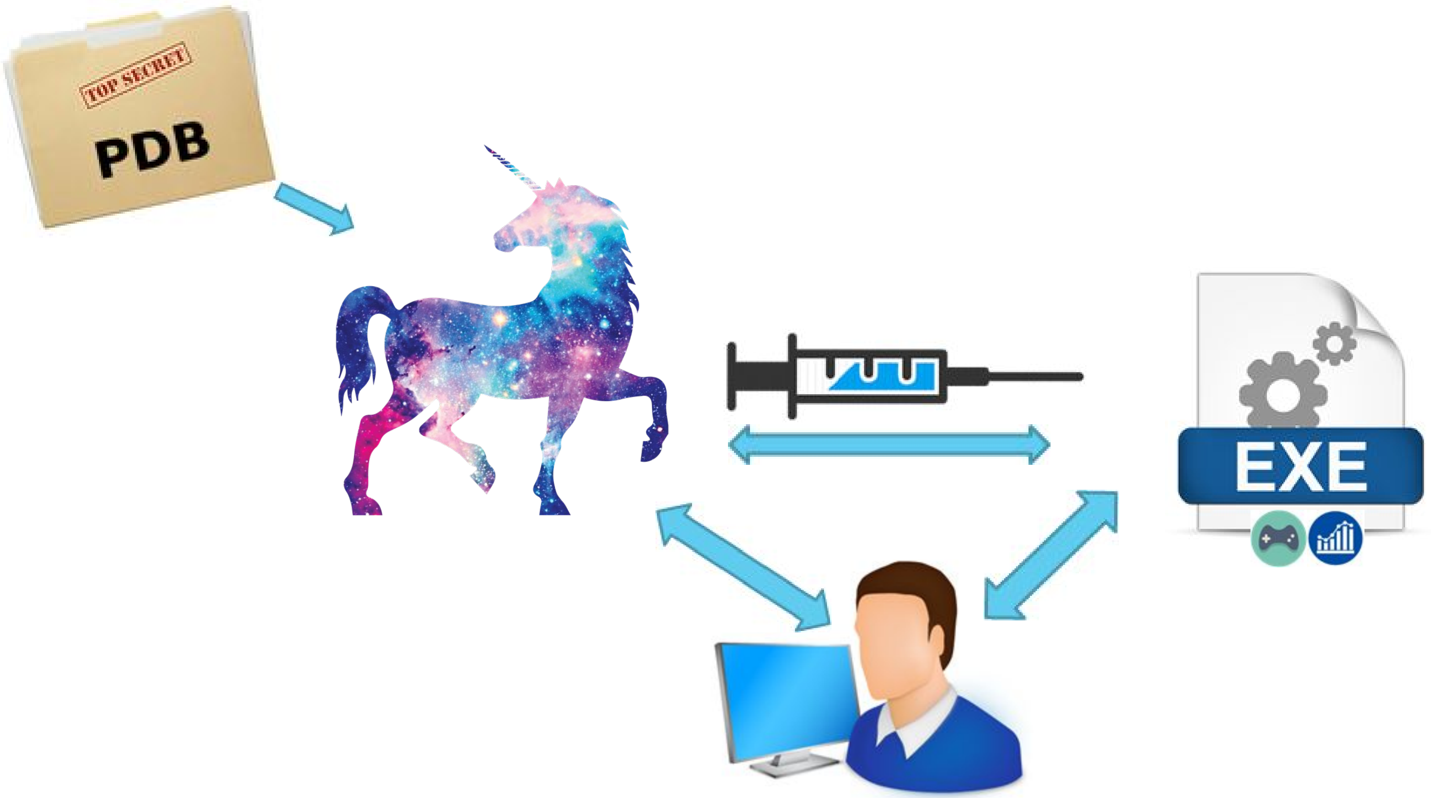
- Statistical, not exact
- Typically poor visualisation

Manual Instrumentation

- NEED TO CHANGE CODE
  - (you need access to code)
- ITERATION SUCKS

Unicorn Profiler™

- Works out of the box on any exe (**NO CODE CHANGE**)
- **Zero friction** to utilization
- Combines advantages of sampling and instrumenting
- Quickly gives the big picture
- Goes beyond measuring performance
- Helps acquire a deep understanding of a program


TOP SECRET
PDB
EXE

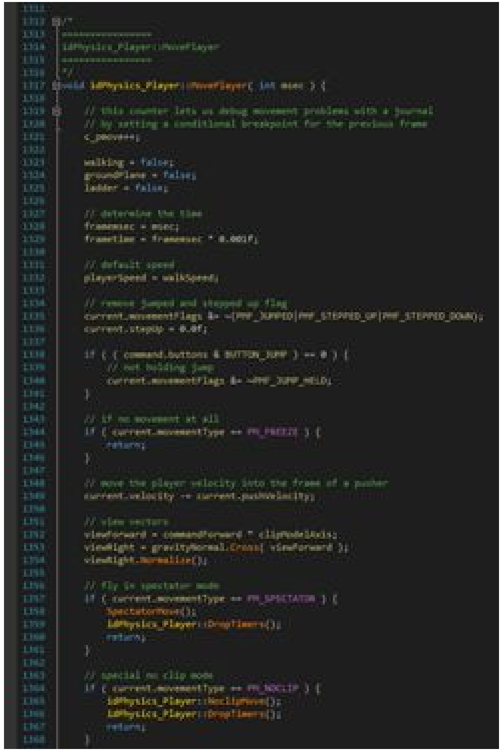
What would it involve?

1. Some information about the target program

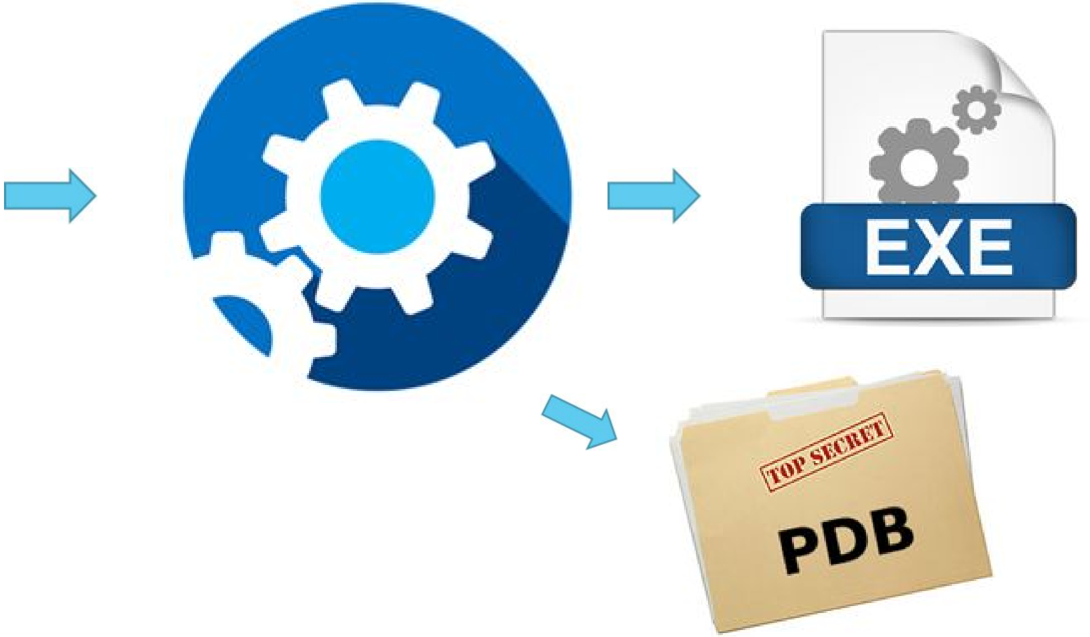
2. A way to inject our code and open up a communication channel

3. A way to dynamically instrument (live code modification)

PDB Parsing



```
/*
================
idPhysics_Player::MovePlayer
================
*/
void idPhysics_Player::MovePlayer( int msec ) {

    // this counter lets us debug movement problems with a journal
    // by setting a conditional breakpoint for the previous frame
    c_pmove++;

    walking = false;
    groundPlane = false;
    ladder = false;

    // determine the time
    framemsec = msec;
    frametime = framemsec * 0.001f;

    // default speed
    playerSpeed = walkSpeed;

    // remove jumped and stepped up flag
    current.movementFlags &= ~(PMF_JUMPED|PMF_STEPPED_UP|PMF_STEPPED_DOWN);
    current.stepUp = 0.0f;

    if ( ( command.buttons & BUTTON_JUMP ) == 0 ) {
        // not holding jump
        current.movementFlags &= ~PMF_JUMP_HELD;
    }

    // if no movement at all
    if ( current.movementType == PM_FREEZE ) {
        return;
    }

    // move the player velocity into the frame of a pusher
    current.velocity -= current.pushVelocity;

    // view vectors
    viewForward = commandForward * clipModelAxis;
    viewRight = gravityNormal.Cross( viewForward );
    viewRight.Normalize();

    // fly in spectator mode
    if ( current.movementType == PM_SPECTATOR ) {
        SpectatorMove();
        idPhysics_Player::DropTimers();
        return;
    }

    // special no clip mode
    if ( current.movementType == PM_NOCLIP ) {
        idPhysics_Player::NoclipMove();
        idPhysics_Player::DropTimers();
        return;
    }
```



Program Database

A native C++ PDB file contains quite a bit of information:

- Public, private, and static **function addresses**
- **Global variables** names and addresses
- Parameters and local variables names and offsets where to find them on the stack or in register
- **Type information**
- Frame Pointer Omission (FPO) data, which is the key to native stack walking on x86
- Source file names and their lines

https://www.wintellect.com/pdb-files-what-every-developer-must-know

PDB Parsing

Option 1: **dbghelp.dll**

- Somewhat easy to use
- Black box
- Can be uber slow (see Bruce Dawson’s post)
  https://randomascii.wordpress.com/2012/10/04/xperf-symbol-loading-pitfalls/

Option 2: **DIA SDK** (Debug Interface Access)

- Not as easy to use
- Much richer interface
- Much, much faster

PDB Parsing

Option 1: **dbghelp.dll** ☒

- Somewhat easy to use
- Black box
- Can be uber slow (see Bruce Dawson’s post)
  https://randomascii.wordpress.com/2012/10/04/xperf-symbol-loading-pitfalls/

Option 2: **DIA SDK** (Debug Interface Access) ☑

- Not as easy to use
- Much richer interface
- Much, much faster

After parsing PDBs, we get:

- List of all Functions and their Relative Virtual Addresses (RVA)
- Function signatures, calling convention, etc.
- List of all Types and their exact memory layout
- Detailed type hierarchy
- Global variables, their type and their RVA
- From IP or code address : line, file

RV-What?

- Modules can be loaded at arbitrary addresses in memory (**ModuleBase**)
- Pdb’s contain Relative Virtual Addresses (**RVA**)
- To compute the Virtual Address (**VA)**, we need to do some **complex mathematics**:

RV-What?

- Modules can be loaded at arbitrary addresses in memory (**ModuleBase**)
- Pdb’s contain Relative Virtual Addresses (**RVA**)
- To compute the Virtual Address (**VA)**, we need to do some **complex mathematics**:

$$VA = ModuleBase + RVA$$

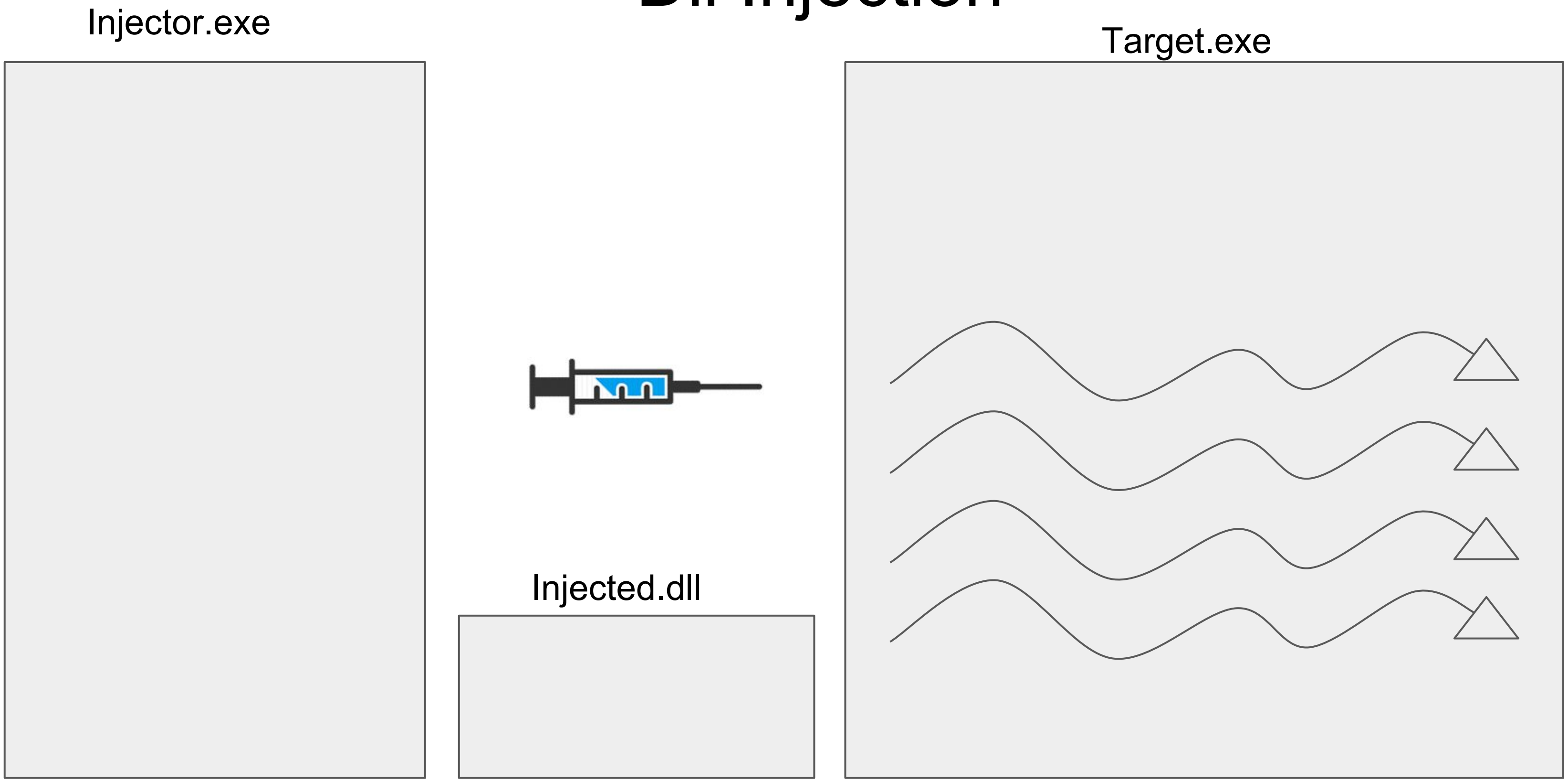
$$RVA = VA - ModuleBase$$

What would it involve?

1. Some information about the target program ☑

2. A way to inject our code and open up a communication channel
3. A way to dynamically instrument (live code modification)

Dll Injection

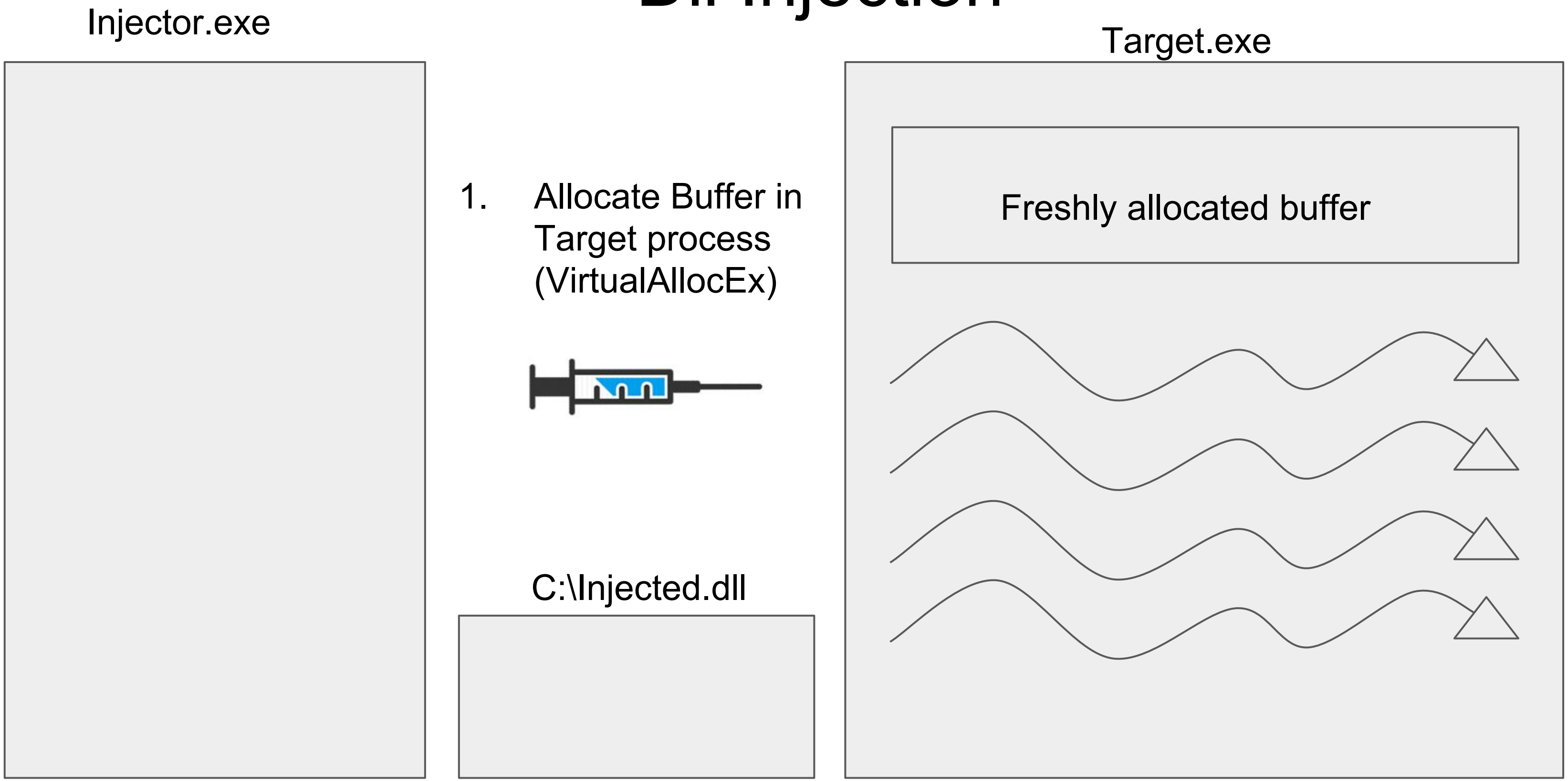


Injector.exe

Target.exe

Injected.dll

Dll Injection



Injector.exe

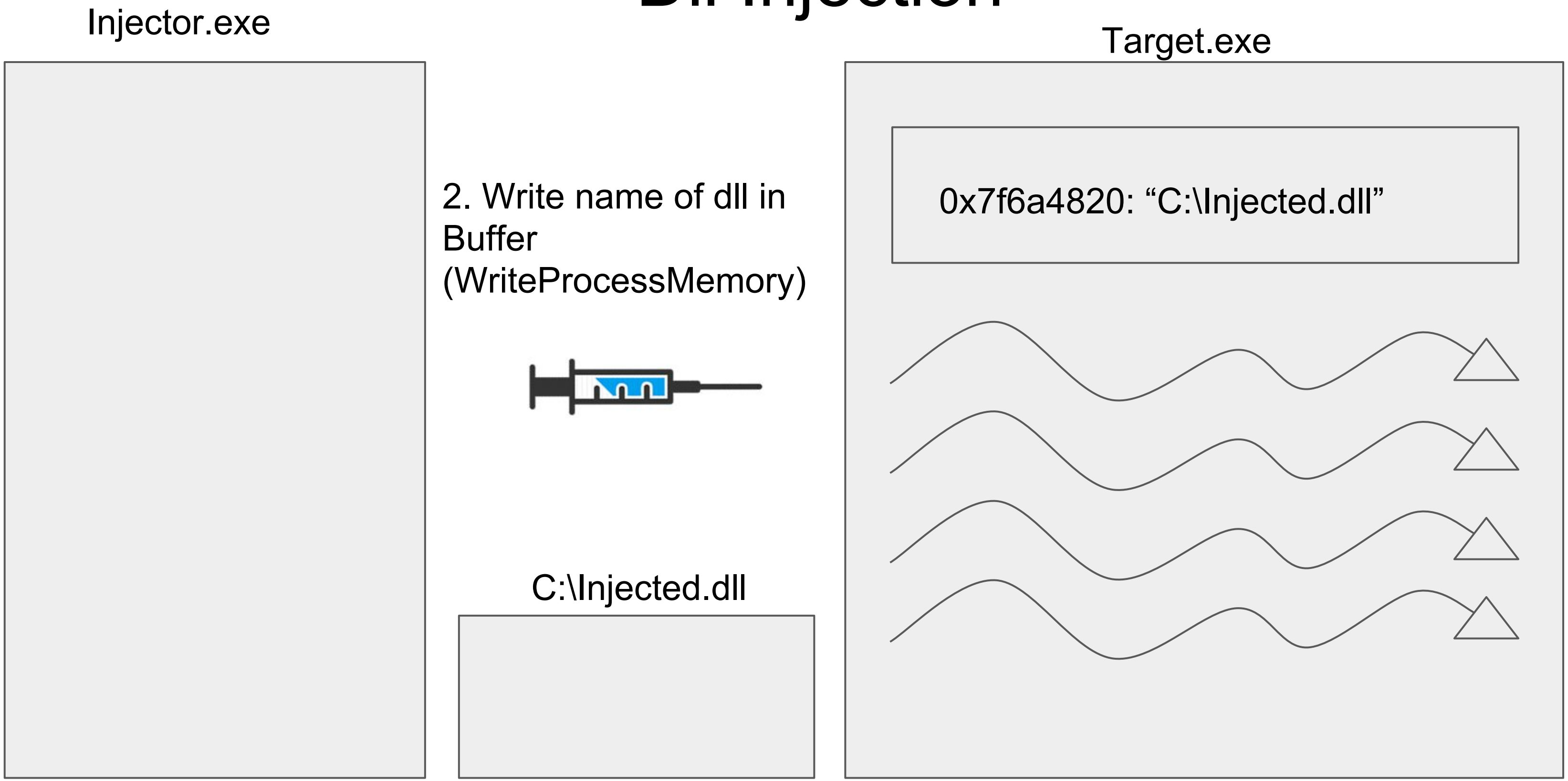
1. Allocate Buffer in Target process (VirtualAllocEx)

C:\Injected.dll

Target.exe

Freshly allocated buffer

Dll Injection



Injector.exe

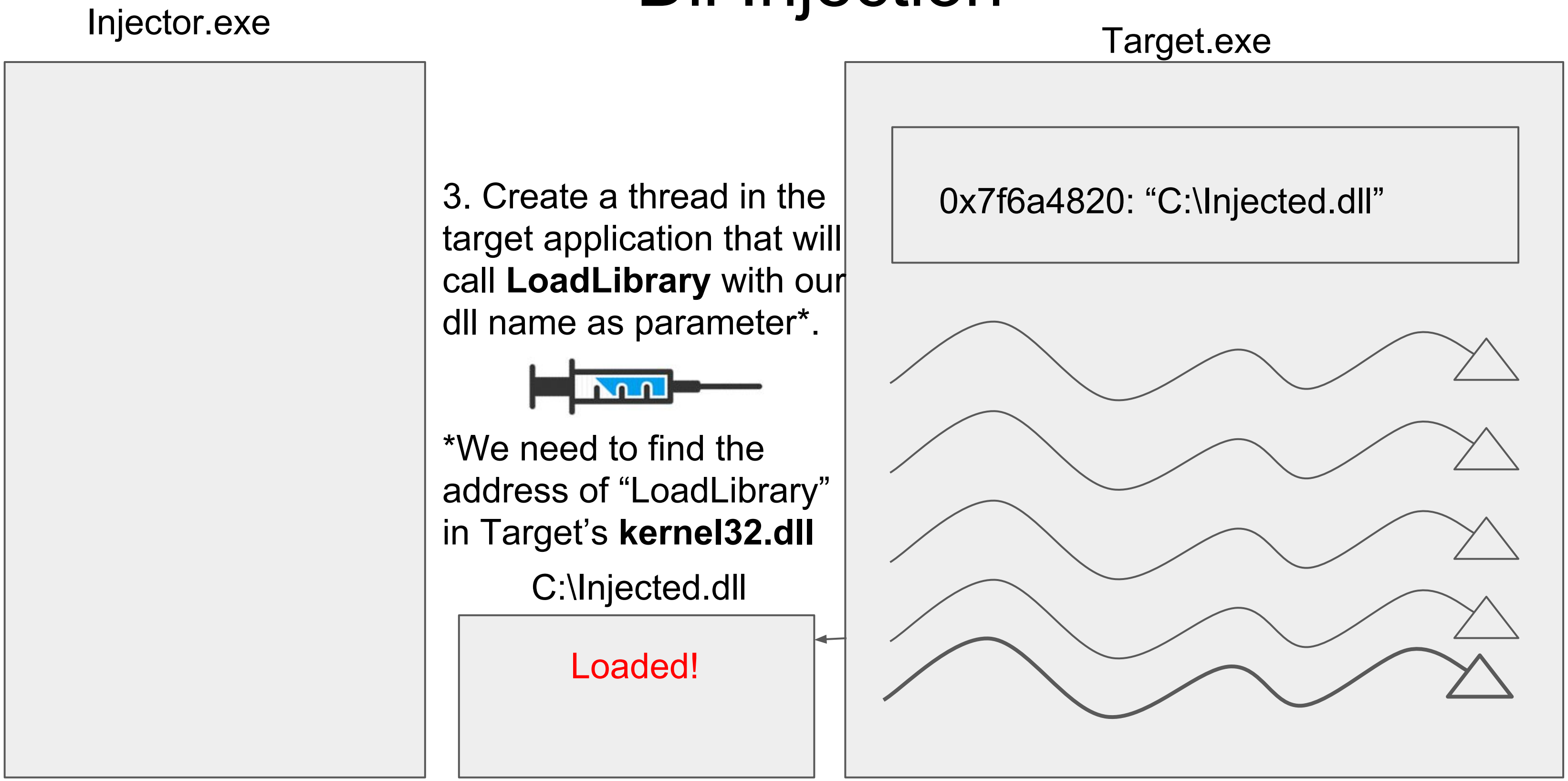
2. Write name of dll in Buffer (WriteProcessMemory)

C:\Injected.dll

Target.exe

0x7f6a4820: “C:\Injected.dll”

Dll Injection



Injector.exe

Target.exe

0x7f6a4820: “C:\Injected.dll”

3. Create a thread in the target application that will call **LoadLibrary** with our dll name as parameter*.

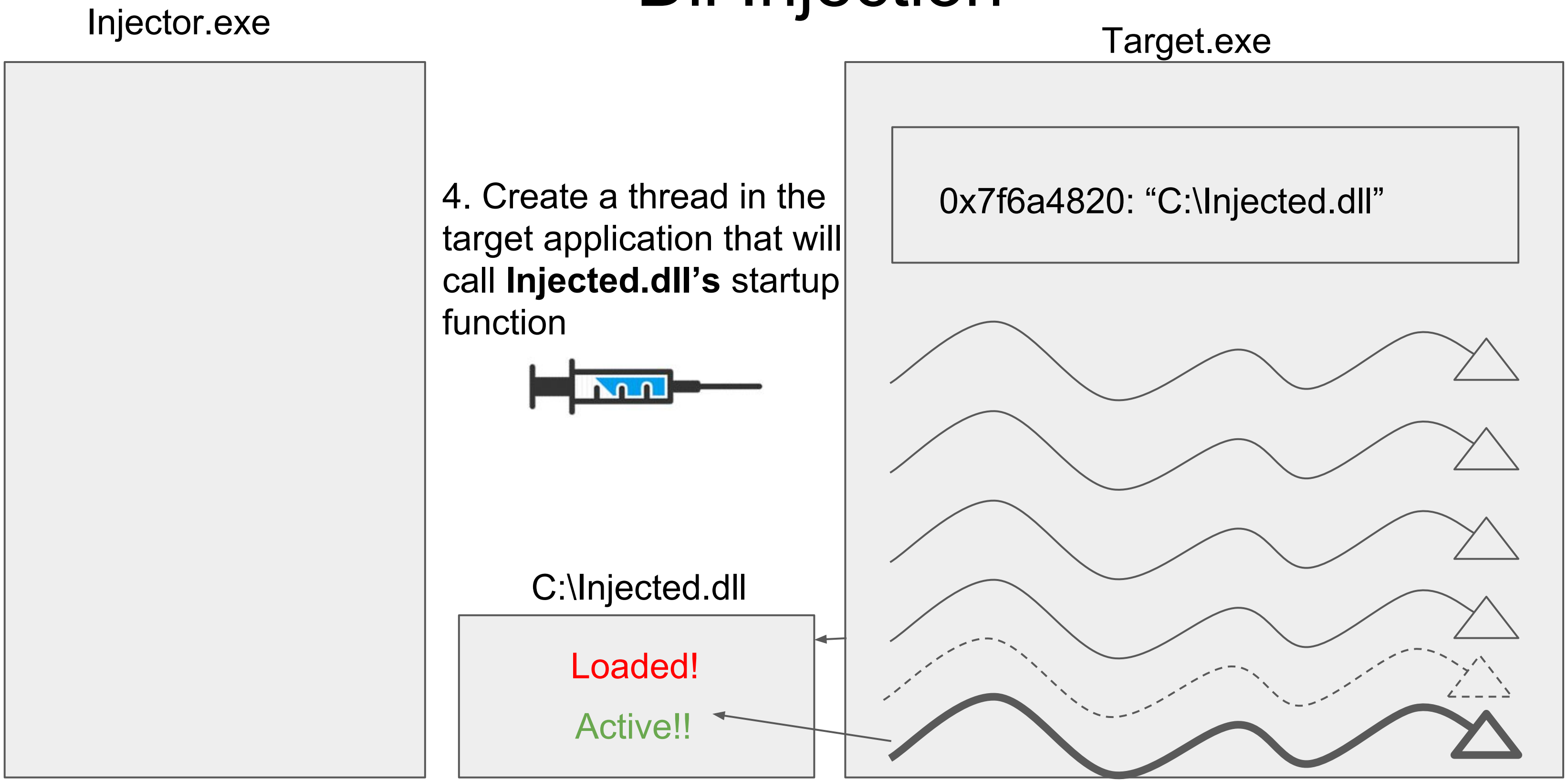
*We need to find the address of “LoadLibrary” in Target’s **kernel32.dll**

C:\Injected.dll

Loaded!

CreateRemoteThread!

Dll Injection



Injector.exe

Target.exe

0x7f6a4820: “C:\Injected.dll”

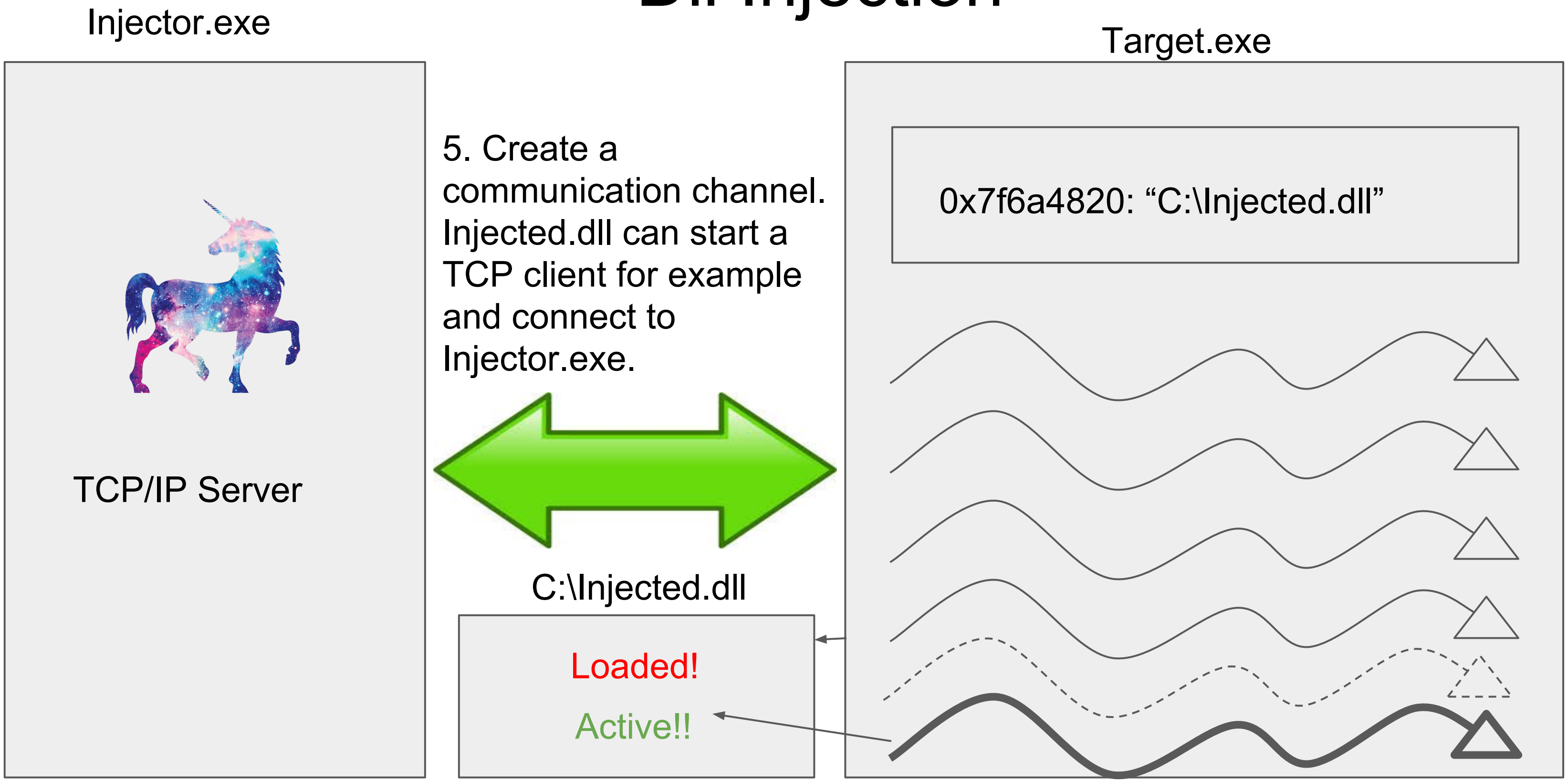
4. Create a thread in the target application that will call **Injected.dll’s** startup function

C:\Injected.dll

Loaded!

Active!!

Dll Injection



Injector.exe

TCP/IP Server

5. Create a communication channel. Injected.dll can start a TCP client for example and connect to Injector.exe.

C:\Injected.dll

Loaded!

Active!!

Target.exe

0x7f6a4820: “C:\Injected.dll”

What would it involve?

1. Some information about the target program ☑

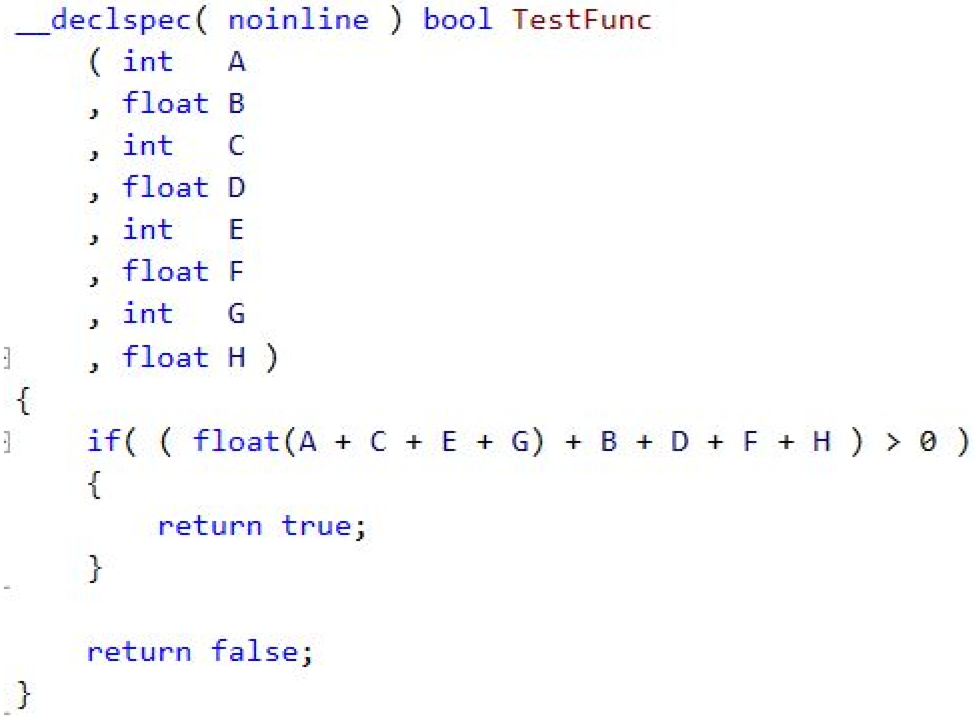
2. A way to inject our code and open up a communication channel ☑

3. A way to dynamically instrument (live code modification)

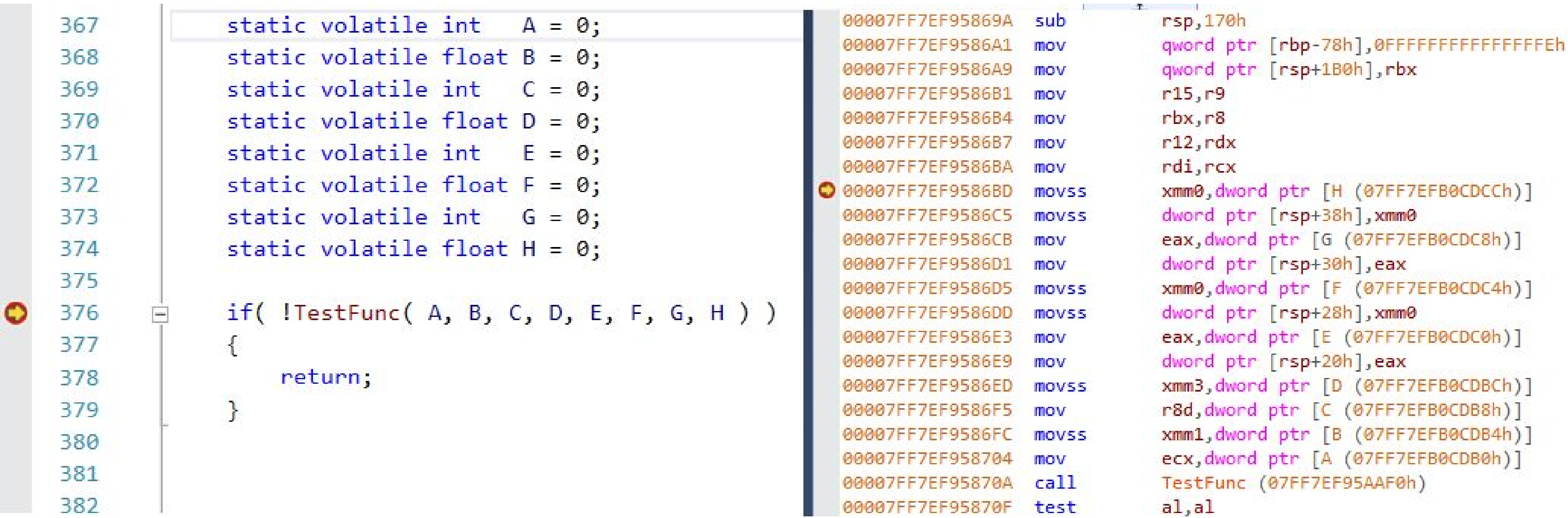
x64 Calling Convention

- The x64 Application Binary Interface (ABI) uses a four register **fast-call** calling convention by default.
- *Integer* arguments are passed in registers **RCX**, **RDX**, **R8**, and **R9**.
- *Floating point* arguments are passed in **XMM0**, **XMM1**, **XMM2**, and **XMM3**
- Return value in **RAX** or **XMM0**

Overview of x64 Calling Conventions
https://msdn.microsoft.com/en-us/library/ms235286.aspx

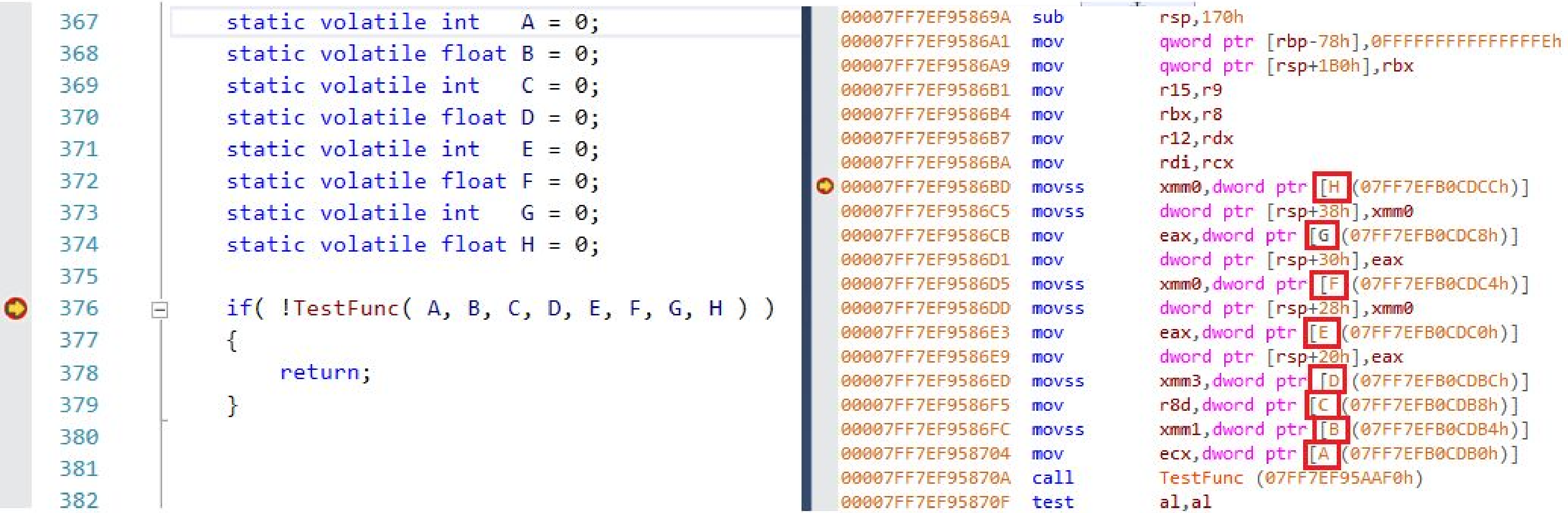


```
__declspec( noinline ) bool TestFunc
    ( int   A
    , float B
    , int   C
    , float D
    , int   E
    , float F
    , int   G
    , float H )
{
    if( ( float(A + C + E + G) + B + D + F + H ) > 0 )
    {
        return true;
    }

    return false;
}
```



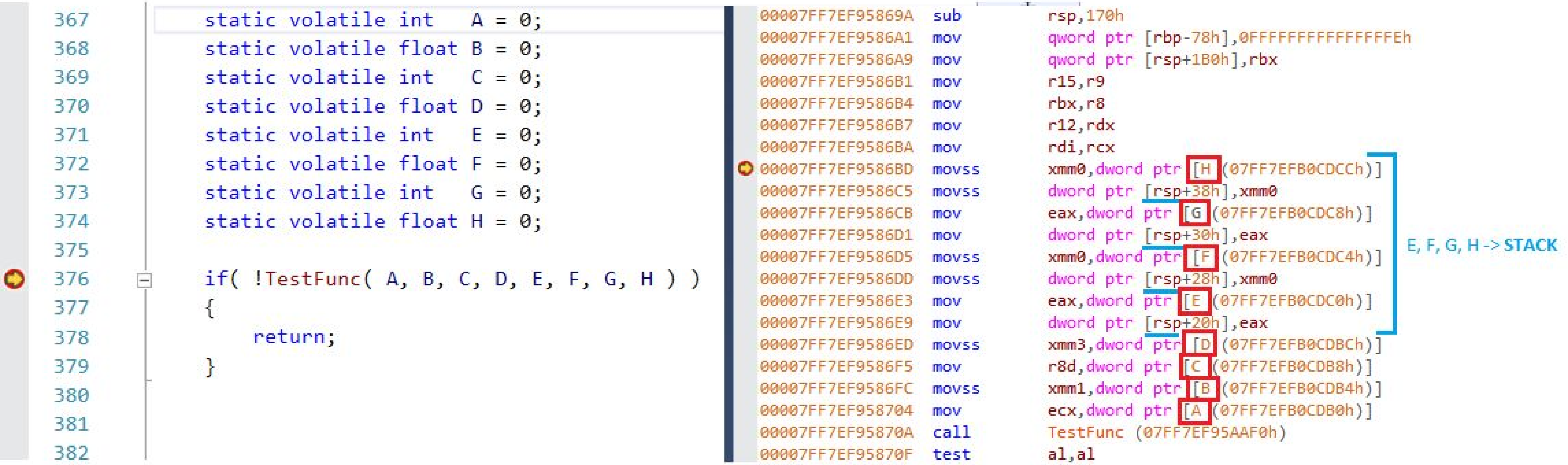
```
    static volatile int   A = 0;
    static volatile float B = 0;
    static volatile int   C = 0;
    static volatile float D = 0;
    static volatile int   E = 0;
    static volatile float F = 0;
    static volatile int   G = 0;
    static volatile float H = 0;

    if( !TestFunc( A, B, C, D, E, F, G, H ) )
    {
        return;
    }
```

```
00007FF7EF95869A  sub         rsp,170h
00007FF7EF9586A1  mov         qword ptr [rbp-78h],0FFFFFFFFFFFFFFFEh
00007FF7EF9586A9  mov         qword ptr [rsp+1B0h],rbx
00007FF7EF9586B1  mov         r15,r9
00007FF7EF9586B4  mov         rbx,r8
00007FF7EF9586B7  mov         r12,rdx
00007FF7EF9586BA  mov         rdi,rcx
00007FF7EF9586BD  movss       xmm0,dword ptr [H (07FF7EFB0CDCCh)]
00007FF7EF9586C5  movss       dword ptr [rsp+38h],xmm0
00007FF7EF9586CB  mov         eax,dword ptr [G (07FF7EFB0CDC8h)]
00007FF7EF9586D1  mov         dword ptr [rsp+30h],eax
00007FF7EF9586D5  movss       xmm0,dword ptr [F (07FF7EFB0CDC4h)]
00007FF7EF9586DD  movss       dword ptr [rsp+28h],xmm0
00007FF7EF9586E3  mov         eax,dword ptr [E (07FF7EFB0CDC0h)]
00007FF7EF9586E9  mov         dword ptr [rsp+20h],eax
00007FF7EF9586ED  movss       xmm3,dword ptr [D (07FF7EFB0CDBCh)]
00007FF7EF9586F5  mov         r8d,dword ptr [C (07FF7EFB0CDB8h)]
00007FF7EF9586FC  movss       xmm1,dword ptr [B (07FF7EFB0CDB4h)]
00007FF7EF958704  mov         ecx,dword ptr [A (07FF7EFB0CDB0h)]
00007FF7EF95870A  call        TestFunc (07FF7EF95AAF0h)
00007FF7EF95870F  test        al,al
```



```
    static volatile int   A = 0;
    static volatile float B = 0;
    static volatile int   C = 0;
    static volatile float D = 0;
    static volatile int   E = 0;
    static volatile float F = 0;
    static volatile int   G = 0;
    static volatile float H = 0;

    if( !TestFunc( A, B, C, D, E, F, G, H ) )
    {
        return;
    }
```

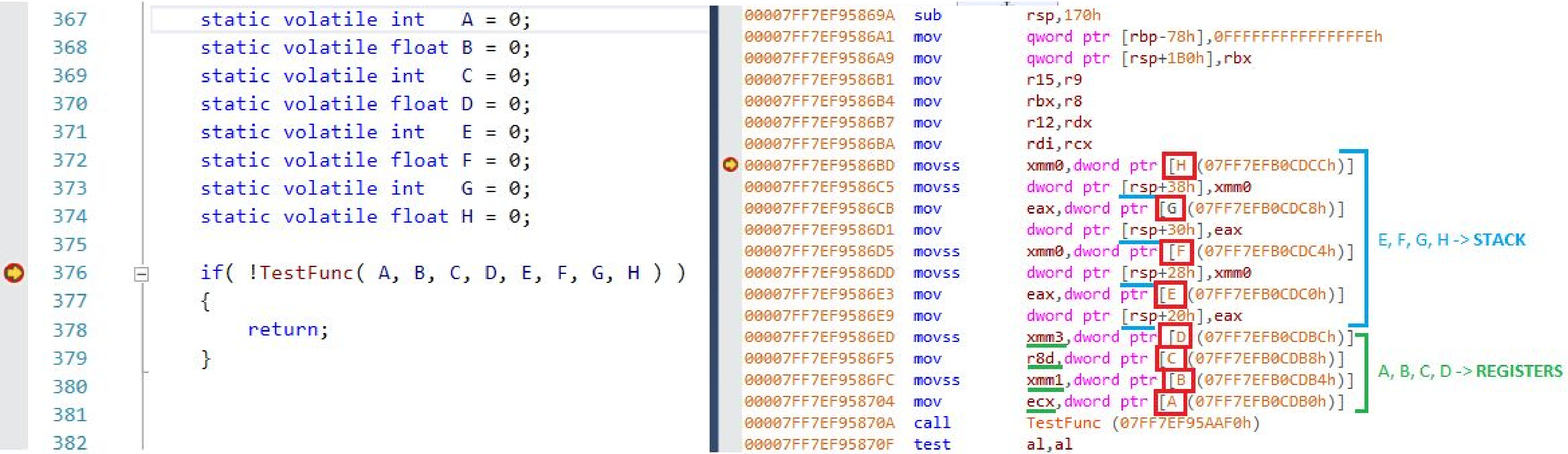
```
00007FF7EF95869A  sub         rsp,170h
00007FF7EF9586A1  mov         qword ptr [rbp-78h],0FFFFFFFFFFFFFFFEh
00007FF7EF9586A9  mov         qword ptr [rsp+1B0h],rbx
00007FF7EF9586B1  mov         r15,r9
00007FF7EF9586B4  mov         rbx,r8
00007FF7EF9586B7  mov         r12,rdx
00007FF7EF9586BA  mov         rdi,rcx
00007FF7EF9586BD  movss       xmm0,dword ptr [H (07FF7EFB0CDCCh)]
00007FF7EF9586C5  movss       dword ptr [rsp+38h],xmm0
00007FF7EF9586CB  mov         eax,dword ptr [G (07FF7EFB0CDC8h)]
00007FF7EF9586D1  mov         dword ptr [rsp+30h],eax
00007FF7EF9586D5  movss       xmm0,dword ptr [F (07FF7EFB0CDC4h)]
00007FF7EF9586DD  movss       dword ptr [rsp+28h],xmm0
00007FF7EF9586E3  mov         eax,dword ptr [E (07FF7EFB0CDC0h)]
00007FF7EF9586E9  mov         dword ptr [rsp+20h],eax
00007FF7EF9586ED  movss       xmm3,dword ptr [D (07FF7EFB0CDBCh)]
00007FF7EF9586F5  mov         r8d,dword ptr [C (07FF7EFB0CDB8h)]
00007FF7EF9586FC  movss       xmm1,dword ptr [B (07FF7EFB0CDB4h)]
00007FF7EF958704  mov         ecx,dword ptr [A (07FF7EFB0CDB0h)]
00007FF7EF95870A  call        TestFunc (07FF7EF95AAF0h)
00007FF7EF95870F  test        al,al
```



```
static volatile int   A = 0;
static volatile float B = 0;
static volatile int   C = 0;
static volatile float D = 0;
static volatile int   E = 0;
static volatile float F = 0;
static volatile int   G = 0;
static volatile float H = 0;

if( !TestFunc( A, B, C, D, E, F, G, H ) )
{
    return;
}
```

```
00007FF7EF95869A  sub      rsp,170h
00007FF7EF9586A1  mov      qword ptr [rbp-78h],0FFFFFFFFFFFFFFFEh
00007FF7EF9586A9  mov      qword ptr [rsp+1B0h],rbx
00007FF7EF9586B1  mov      r15,r9
00007FF7EF9586B4  mov      rbx,r8
00007FF7EF9586B7  mov      r12,rdx
00007FF7EF9586BA  mov      rdi,rcx
00007FF7EF9586BD  movss    xmm0,dword ptr [H (07FF7EFB0CDCCh)]
00007FF7EF9586C5  movss    dword ptr [rsp+38h],xmm0
00007FF7EF9586CB  mov      eax,dword ptr [G (07FF7EFB0CDC8h)]
00007FF7EF9586D1  mov      dword ptr [rsp+30h],eax
00007FF7EF9586D5  movss    xmm0,dword ptr [F (07FF7EFB0CDC4h)]
00007FF7EF9586DD  movss    dword ptr [rsp+28h],xmm0
00007FF7EF9586E3  mov      eax,dword ptr [E (07FF7EFB0CDC0h)]
00007FF7EF9586E9  mov      dword ptr [rsp+20h],eax
00007FF7EF9586ED  movss    xmm3,dword ptr [D (07FF7EFB0CDBCh)]
00007FF7EF9586F5  mov      r8d,dword ptr [C (07FF7EFB0CDB8h)]
00007FF7EF9586FC  movss    xmm1,dword ptr [B (07FF7EFB0CDB4h)]
00007FF7EF958704  mov      ecx,dword ptr [A (07FF7EFB0CDB0h)]
00007FF7EF95870A  call     TestFunc (07FF7EF95AAF0h)
00007FF7EF95870F  test     al,al
```

E, F, G, H -> **STACK**

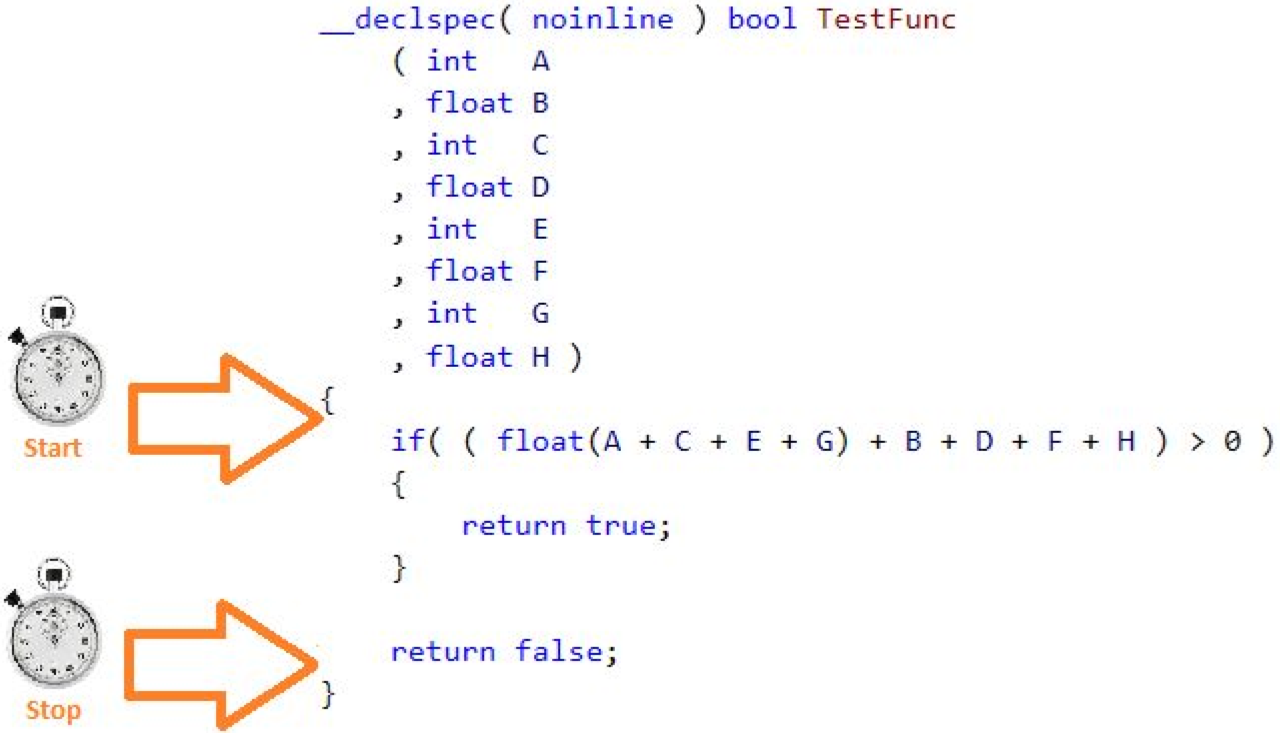


```
    static volatile int   A = 0;
    static volatile float B = 0;
    static volatile int   C = 0;
    static volatile float D = 0;
    static volatile int   E = 0;
    static volatile float F = 0;
    static volatile int   G = 0;
    static volatile float H = 0;

    if( !TestFunc( A, B, C, D, E, F, G, H ) )
    {
        return;
    }
```

```
00007FF7EF95869A  sub      rsp,170h
00007FF7EF9586A1  mov      qword ptr [rbp-78h],0FFFFFFFFFFFFFFFEh
00007FF7EF9586A9  mov      qword ptr [rsp+1B0h],rbx
00007FF7EF9586B1  mov      r15,r9
00007FF7EF9586B4  mov      rbx,r8
00007FF7EF9586B7  mov      r12,rdx
00007FF7EF9586BA  mov      rdi,rcx
00007FF7EF9586BD  movss    xmm0,dword ptr [H (07FF7EFB0CDCCh)]
00007FF7EF9586C5  movss    dword ptr [rsp+38h],xmm0
00007FF7EF9586CB  mov      eax,dword ptr [G (07FF7EFB0CDC8h)]
00007FF7EF9586D1  mov      dword ptr [rsp+30h],eax
00007FF7EF9586D5  movss    xmm0,dword ptr [F (07FF7EFB0CDC4h)]
00007FF7EF9586DD  movss    dword ptr [rsp+28h],xmm0
00007FF7EF9586E3  mov      eax,dword ptr [E (07FF7EFB0CDC0h)]
00007FF7EF9586E9  mov      dword ptr [rsp+20h],eax
00007FF7EF9586ED  movss    xmm3,dword ptr [D (07FF7EFB0CDBCh)]
00007FF7EF9586F5  mov      r8d,dword ptr [C (07FF7EFB0CDB8h)]
00007FF7EF9586FC  movss    xmm1,dword ptr [B (07FF7EFB0CDB4h)]
00007FF7EF958704  mov      ecx,dword ptr [A (07FF7EFB0CDB0h)]
00007FF7EF95870A  call     TestFunc (07FF7EF95AAF0h)
00007FF7EF95870F  test     al,al
```

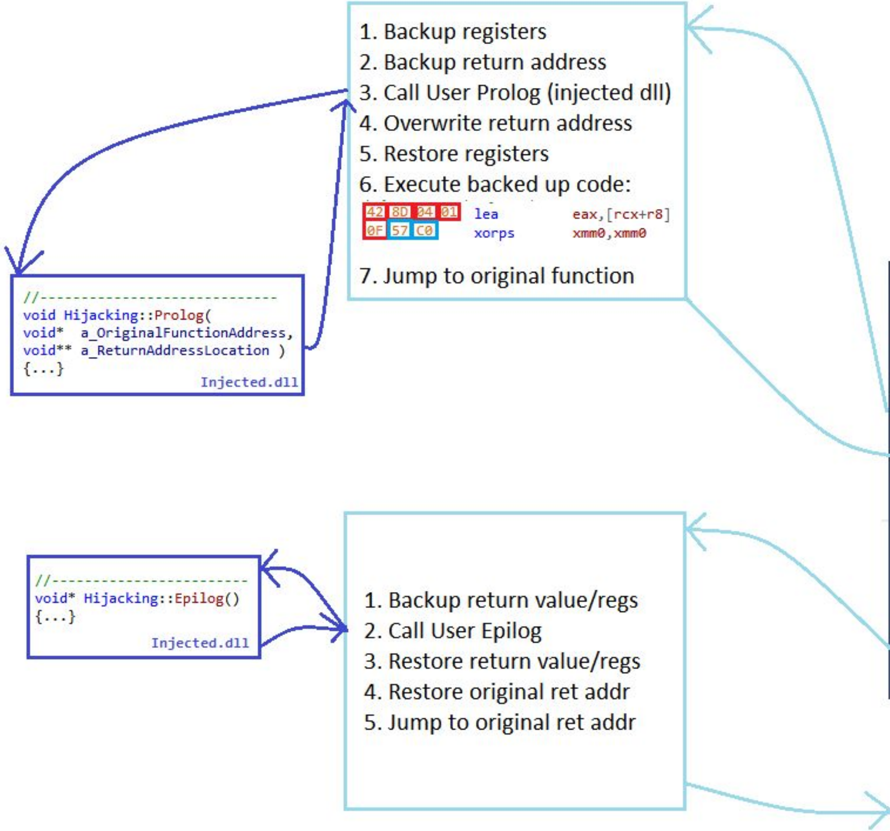
E, F, G, H -> **STACK**

A, B, C, D -> **REGISTERS**



```
__declspec( noinline ) bool TestFunc
    ( int   A
    , float B
    , int   C
    , float D
    , int   E
    , float F
    , int   G
    , float H )
{
    if( ( float(A + C + E + G) + B + D + F + H ) > 0 )
    {
        return true;
    }

    return false;
}
```

Trampolines



1. Backup registers
2. Backup return address
3. Call User Prolog (injected dll)
4. Overwrite return address
5. Restore registers
6. Execute backed up code:

```
42 8D 04 01    lea        eax,[rcx+r8]
0F 57 C0       xorps      xmm0,xmm0
```

7. Jump to original function

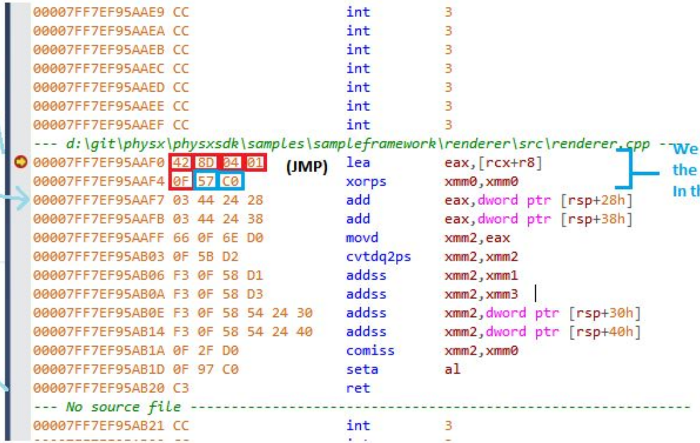
```
//-----------------------------
void Hijacking::Prolog(
void*  a_OriginalFunctionAddress,
void** a_ReturnAddressLocation )
{...}
```

Injected.dll

```
//-----------------------
void* Hijacking::Epilog()
{...}
```

Injected.dll

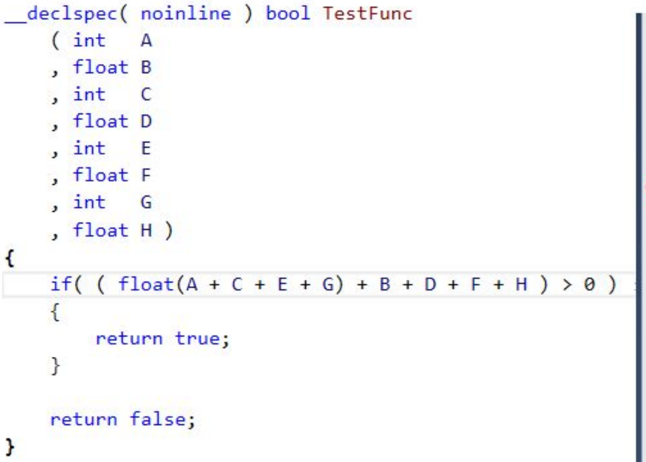
1. Backup return value/regs
2. Call User Epilog
3. Restore return value/regs
4. Restore original ret addr
5. Jump to original ret addr



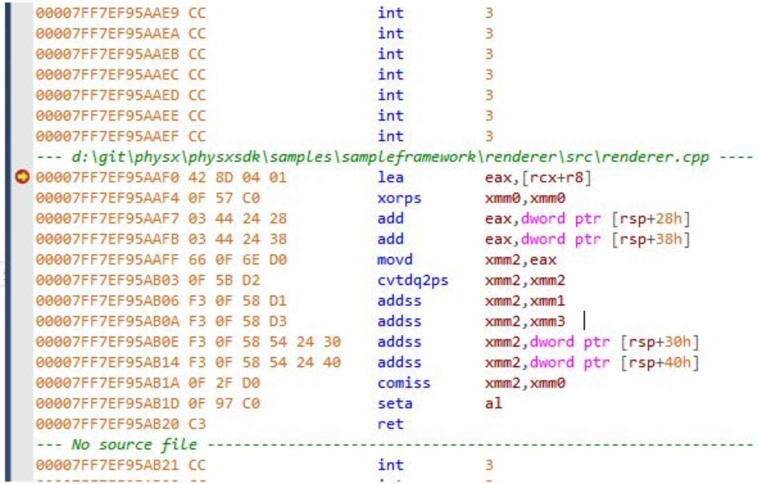
```
00007FF7EF95AAE9 CC                    int         3
00007FF7EF95AAEA CC                    int         3
00007FF7EF95AAEB CC                    int         3
00007FF7EF95AAEC CC                    int         3
00007FF7EF95AAED CC                    int         3
00007FF7EF95AAEE CC                    int         3
00007FF7EF95AAEF CC                    int         3
--- d:\git\physx\physxsdk\samples\sampleframework\renderer\src\renderer.cpp --
00007FF7EF95AAF0 42 8D 04 01   (JMP)   lea         eax,[rcx+r8]
00007FF7EF95AAF4 0F 57 C0              xorps       xmm0,xmm0
00007FF7EF95AAF7 03 44 24 28           add         eax,dword ptr [rsp+28h]
00007FF7EF95AAFB 03 44 24 38           add         eax,dword ptr [rsp+38h]
00007FF7EF95AAFF 66 0F 6E D0           movd        xmm2,eax
00007FF7EF95AB03 0F 5B D2              cvtdq2ps    xmm2,xmm2
00007FF7EF95AB06 F3 0F 58 D1           addss       xmm2,xmm1
00007FF7EF95AB0A F3 0F 58 D3           addss       xmm2,xmm3
00007FF7EF95AB0E F3 0F 58 54 24 30     addss       xmm2,dword ptr [rsp+30h]
00007FF7EF95AB14 F3 0F 58 54 24 40     addss       xmm2,dword ptr [rsp+40h]
00007FF7EF95AB1A 0F 2F D0              comiss      xmm2,xmm0
00007FF7EF95AB1D 0F 97 C0              seta        al
00007FF7EF95AB20 C3                    ret
--- No source file -------------------------------------------------------------
00007FF7EF95AB21 CC                    int         3
```

We have to backup remaining bytes of the instruction touched by the JMP!! In this case, backup 7 bytes, not 5.

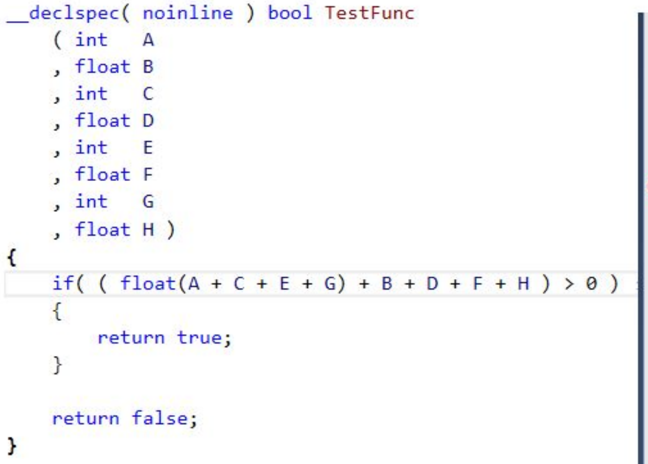
Return to original caller



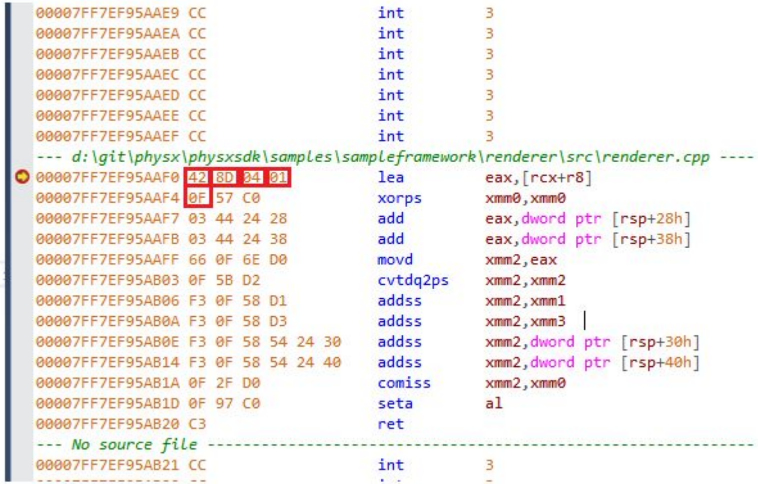
```
__declspec( noinline ) bool TestFunc
    ( int   A
    , float B
    , int   C
    , float D
    , int   E
    , float F
    , int   G
    , float H )
{
    if( ( float(A + C + E + G) + B + D + F + H ) > 0 )
    {
        return true;
    }

    return false;
}
```



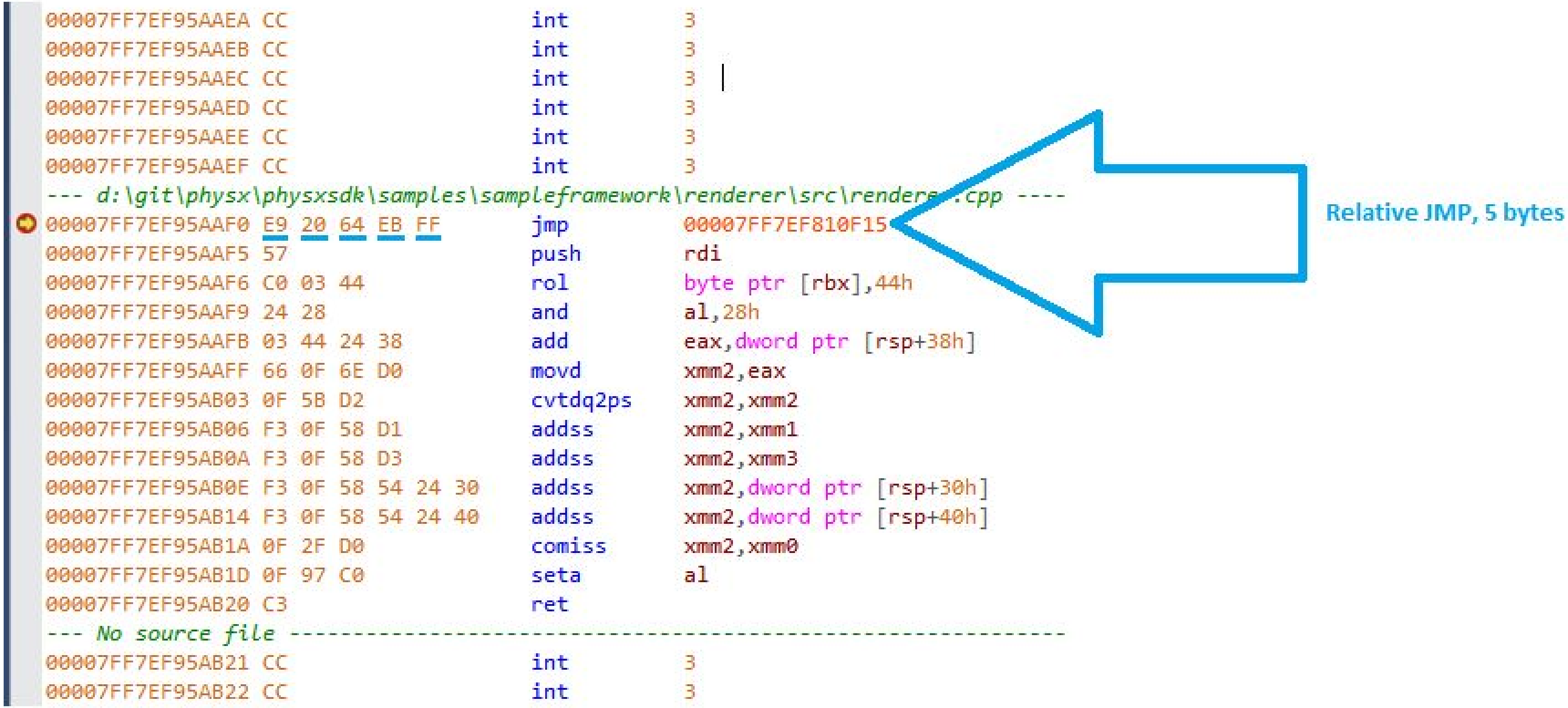
```
00007FF7EF95AAE9 CC                  int         3
00007FF7EF95AAEA CC                  int         3
00007FF7EF95AAEB CC                  int         3
00007FF7EF95AAEC CC                  int         3
00007FF7EF95AAED CC                  int         3
00007FF7EF95AAEE CC                  int         3
00007FF7EF95AAEF CC                  int         3
--- d:\git\physx\physxsdk\samples\sampleframework\renderer\src\renderer.cpp ----
00007FF7EF95AAF0 42 8D 04 01         lea         eax,[rcx+r8]
00007FF7EF95AAF4 0F 57 C0            xorps       xmm0,xmm0
00007FF7EF95AAF7 03 44 24 28         add         eax,dword ptr [rsp+28h]
00007FF7EF95AAFB 03 44 24 38         add         eax,dword ptr [rsp+38h]
00007FF7EF95AAFF 66 0F 6E D0         movd        xmm2,eax
00007FF7EF95AB03 0F 5B D2            cvtdq2ps    xmm2,xmm2
00007FF7EF95AB06 F3 0F 58 D1         addss       xmm2,xmm1
00007FF7EF95AB0A F3 0F 58 D3         addss       xmm2,xmm3
00007FF7EF95AB0E F3 0F 58 54 24 30   addss       xmm2,dword ptr [rsp+30h]
00007FF7EF95AB14 F3 0F 58 54 24 40   addss       xmm2,dword ptr [rsp+40h]
00007FF7EF95AB1A 0F 2F D0            comiss      xmm2,xmm0
00007FF7EF95AB1D 0F 97 C0            seta        al
00007FF7EF95AB20 C3                  ret
--- No source file -------------------------------------------------------------
00007FF7EF95AB21 CC                  int         3
```



```
__declspec( noinline ) bool TestFunc
    ( int   A
    , float B
    , int   C
    , float D
    , int   E
    , float F
    , int   G
    , float H )
{
    if( ( float(A + C + E + G) + B + D + F + H ) > 0 )
    {
        return true;
    }

    return false;
}
```

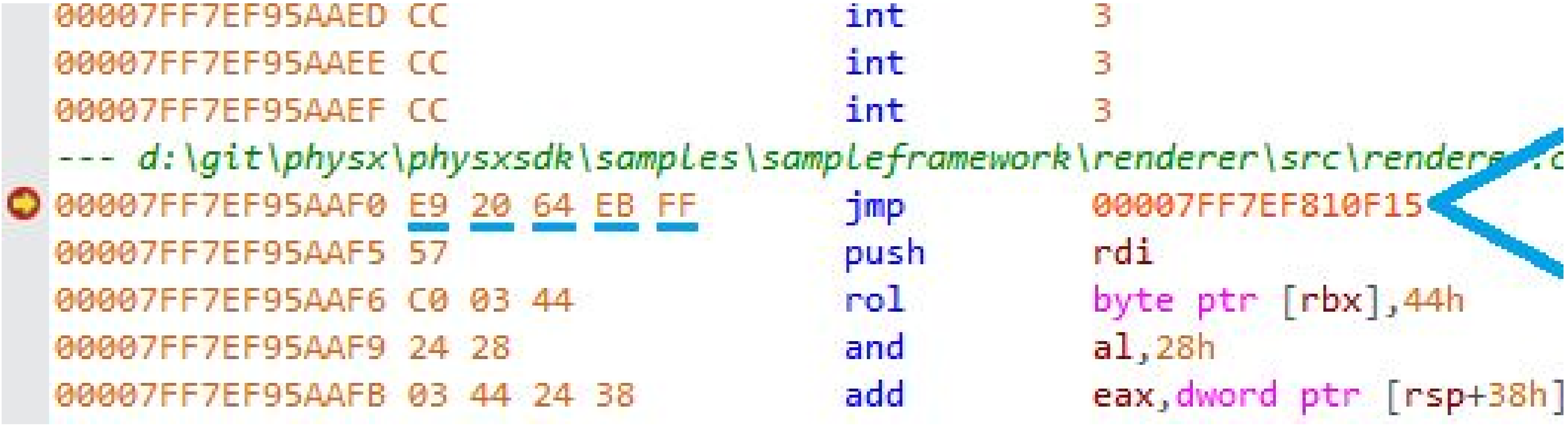


```
00007FF7EF95AAE9 CC                      int         3
00007FF7EF95AAEA CC                      int         3
00007FF7EF95AAEB CC                      int         3
00007FF7EF95AAEC CC                      int         3
00007FF7EF95AAED CC                      int         3
00007FF7EF95AAEE CC                      int         3
00007FF7EF95AAEF CC                      int         3
--- d:\git\physx\physxsdk\samples\sampleframework\renderer\src\renderer.cpp ----
00007FF7EF95AAF0 42 8D 04 01             lea         eax,[rcx+r8]
00007FF7EF95AAF4 0F 57 C0                xorps       xmm0,xmm0
00007FF7EF95AAF7 03 44 24 28             add         eax,dword ptr [rsp+28h]
00007FF7EF95AAFB 03 44 24 38             add         eax,dword ptr [rsp+38h]
00007FF7EF95AAFF 66 0F 6E D0             movd        xmm2,eax
00007FF7EF95AB03 0F 5B D2                cvtdq2ps    xmm2,xmm2
00007FF7EF95AB06 F3 0F 58 D1             addss       xmm2,xmm1
00007FF7EF95AB0A F3 0F 58 D3             addss       xmm2,xmm3
00007FF7EF95AB0E F3 0F 58 54 24 30       addss       xmm2,dword ptr [rsp+30h]
00007FF7EF95AB14 F3 0F 58 54 24 40       addss       xmm2,dword ptr [rsp+40h]
00007FF7EF95AB1A 0F 2F D0                comiss      xmm2,xmm0
00007FF7EF95AB1D 0F 97 C0                seta        al
00007FF7EF95AB20 C3                      ret
--- No source file -------------------------------------------------------------
00007FF7EF95AB21 CC                      int         3
```

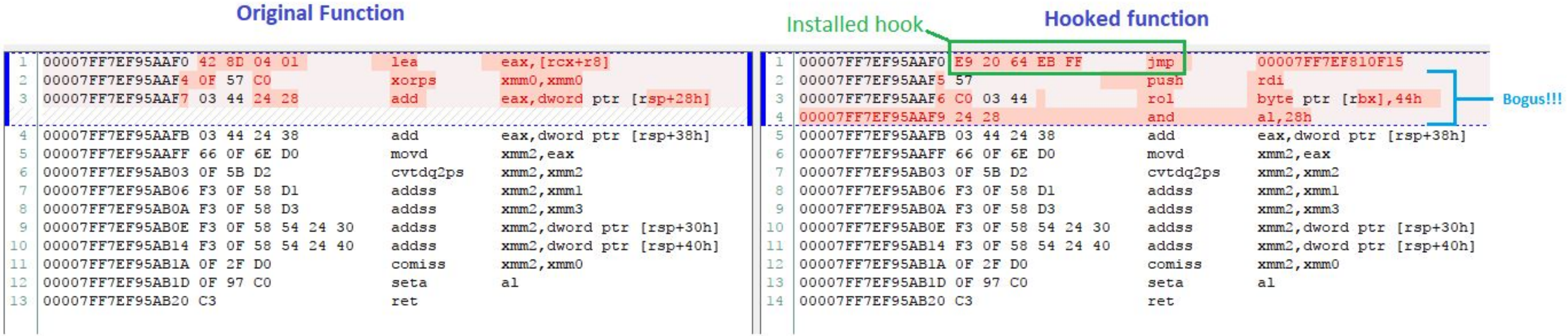


```
00007FF7EF95AAEA CC                   int         3
00007FF7EF95AAEB CC                   int         3
00007FF7EF95AAEC CC                   int         3
00007FF7EF95AAED CC                   int         3
00007FF7EF95AAEE CC                   int         3
00007FF7EF95AAEF CC                   int         3
--- d:\git\physx\physxsdk\samples\sampleframework\renderer\src\renderer.cpp ----
00007FF7EF95AAF0 E9 20 64 EB FF       jmp         00007FF7EF810F15
00007FF7EF95AAF5 57                   push        rdi
00007FF7EF95AAF6 C0 03 44             rol         byte ptr [rbx],44h
00007FF7EF95AAF9 24 28                and         al,28h
00007FF7EF95AAFB 03 44 24 38          add         eax,dword ptr [rsp+38h]
00007FF7EF95AAFF 66 0F 6E D0          movd        xmm2,eax
00007FF7EF95AB03 0F 5B D2             cvtdq2ps    xmm2,xmm2
00007FF7EF95AB06 F3 0F 58 D1          addss       xmm2,xmm1
00007FF7EF95AB0A F3 0F 58 D3          addss       xmm2,xmm3
00007FF7EF95AB0E F3 0F 58 54 24 30    addss       xmm2,dword ptr [rsp+30h]
00007FF7EF95AB14 F3 0F 58 54 24 40    addss       xmm2,dword ptr [rsp+40h]
00007FF7EF95AB1A 0F 2F D0             comiss      xmm2,xmm0
00007FF7EF95AB1D 0F 97 C0             seta        al
00007FF7EF95AB20 C3                   ret
--- No source file -------------------------------------------------------------
00007FF7EF95AB21 CC                   int         3
00007FF7EF95AB22 CC                   int         3
```

Relative JMP, 5 bytes



```
00007FF7EF95AAED CC                    int         3
00007FF7EF95AAEE CC                    int         3
00007FF7EF95AAEF CC                    int         3
--- d:\git\physx\physxsdk\samples\sampleframework\renderer\src\rendere
00007FF7EF95AAF0 E9 20 64 EB FF        jmp         00007FF7EF810F15
00007FF7EF95AAF5 57                    push        rdi
00007FF7EF95AAF6 C0 03 44              rol         byte ptr [rbx],44h
00007FF7EF95AAF9 24 28                 and         al,28h
00007FF7EF95AAFB 03 44 24 38           add         eax,dword ptr [rsp+38h]
```



Original Function

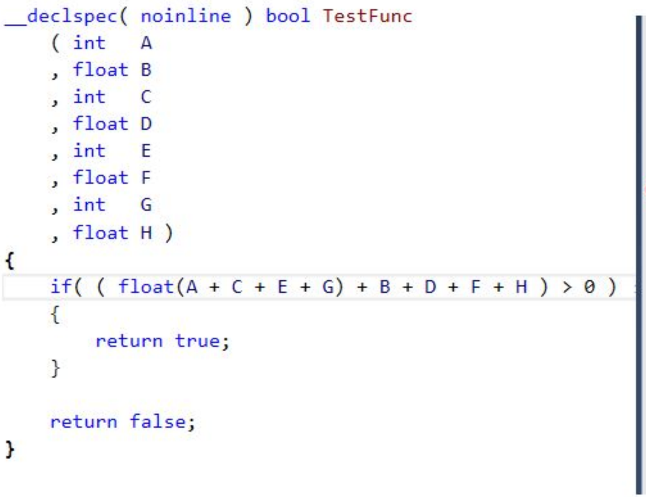
```
00007FF7EF95AAF0 42 8D 04 01           lea       eax,[rcx+r8]
00007FF7EF95AAF4 0F 57 C0              xorps     xmm0,xmm0
00007FF7EF95AAF7 03 44 24 28           add       eax,dword ptr [rsp+28h]
00007FF7EF95AAFB 03 44 24 38           add       eax,dword ptr [rsp+38h]
00007FF7EF95AAFF 66 0F 6E D0           movd      xmm2,eax
00007FF7EF95AB03 0F 5B D2              cvtdq2ps  xmm2,xmm2
00007FF7EF95AB06 F3 0F 58 D1           addss     xmm2,xmm1
00007FF7EF95AB0A F3 0F 58 D3           addss     xmm2,xmm3
00007FF7EF95AB0E F3 0F 58 54 24 30     addss     xmm2,dword ptr [rsp+30h]
00007FF7EF95AB14 F3 0F 58 54 24 40     addss     xmm2,dword ptr [rsp+40h]
00007FF7EF95AB1A 0F 2F D0              comiss    xmm2,xmm0
00007FF7EF95AB1D 0F 97 C0              seta      al
00007FF7EF95AB20 C3                    ret
```

Installed hook

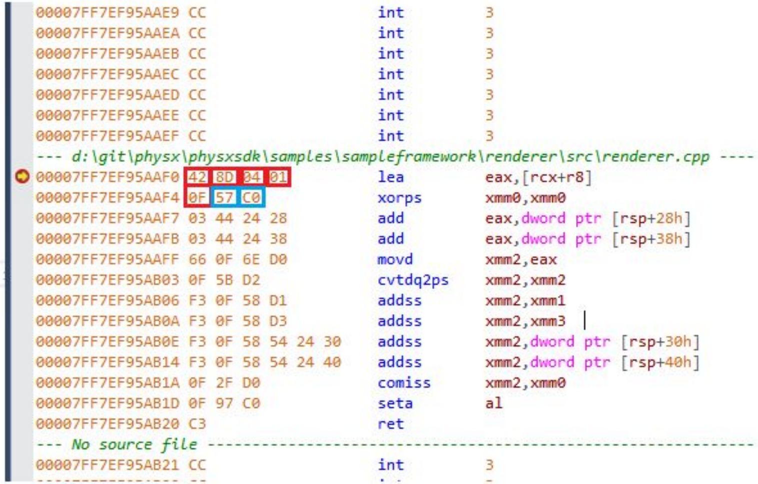
Hooked function

```
00007FF7EF95AAF0 E9 20 64 EB FF        jmp       00007FF7EF810F15
00007FF7EF95AAF5 57                    push      rdi
00007FF7EF95AAF6 C0 03 44              rol       byte ptr [rbx],44h
00007FF7EF95AAF9 24 28                 and       al,28h
00007FF7EF95AAFB 03 44 24 38           add       eax,dword ptr [rsp+38h]
00007FF7EF95AAFF 66 0F 6E D0           movd      xmm2,eax
00007FF7EF95AB03 0F 5B D2              cvtdq2ps  xmm2,xmm2
00007FF7EF95AB06 F3 0F 58 D1           addss     xmm2,xmm1
00007FF7EF95AB0A F3 0F 58 D3           addss     xmm2,xmm3
00007FF7EF95AB0E F3 0F 58 54 24 30     addss     xmm2,dword ptr [rsp+30h]
00007FF7EF95AB14 F3 0F 58 54 24 40     addss     xmm2,dword ptr [rsp+40h]
00007FF7EF95AB1A 0F 2F D0              comiss    xmm2,xmm0
00007FF7EF95AB1D 0F 97 C0              seta      al
00007FF7EF95AB20 C3                    ret
```

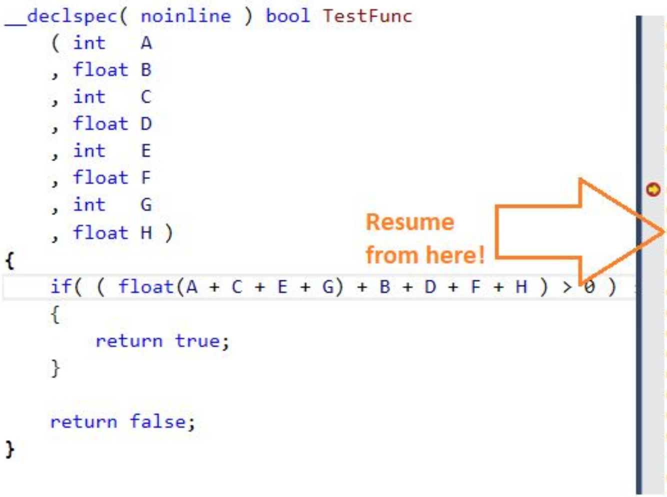
Bogus!!!



```
__declspec( noinline ) bool TestFunc
    ( int   A
    , float B
    , int   C
    , float D
    , int   E
    , float F
    , int   G
    , float H )
{
    if( ( float(A + C + E + G) + B + D + F + H ) > 0 )
    {
        return true;
    }

    return false;
}
```

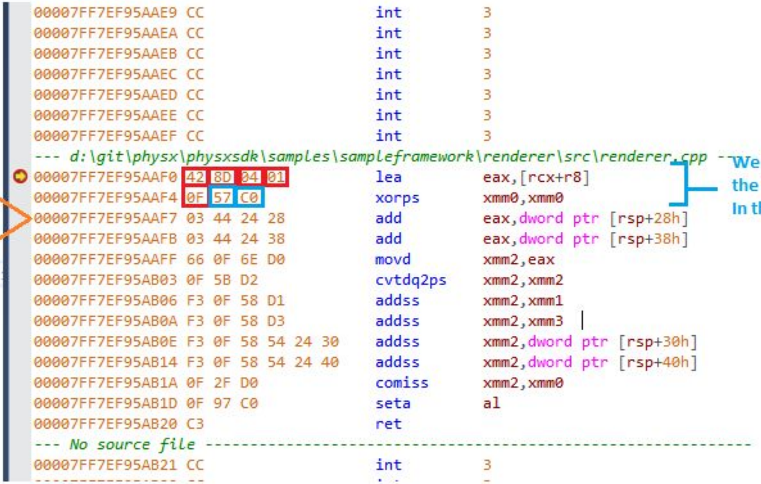


```
00007FF7EF95AAE9 CC                    int         3
00007FF7EF95AAEA CC                    int         3
00007FF7EF95AAEB CC                    int         3
00007FF7EF95AAEC CC                    int         3
00007FF7EF95AAED CC                    int         3
00007FF7EF95AAEE CC                    int         3
00007FF7EF95AAEF CC                    int         3
--- d:\git\physx\physxsdk\samples\sampleframework\renderer\src\renderer.cpp ----
00007FF7EF95AAF0 42 8D 04 01           lea         eax,[rcx+r8]
00007FF7EF95AAF4 0F 57 C0              xorps       xmm0,xmm0
00007FF7EF95AAF7 03 44 24 28           add         eax,dword ptr [rsp+28h]
00007FF7EF95AAFB 03 44 24 38           add         eax,dword ptr [rsp+38h]
00007FF7EF95AAFF 66 0F 6E D0           movd        xmm2,eax
00007FF7EF95AB03 0F 5B D2              cvtdq2ps    xmm2,xmm2
00007FF7EF95AB06 F3 0F 58 D1           addss       xmm2,xmm1
00007FF7EF95AB0A F3 0F 58 D3           addss       xmm2,xmm3
00007FF7EF95AB0E F3 0F 58 54 24 30     addss       xmm2,dword ptr [rsp+30h]
00007FF7EF95AB14 F3 0F 58 54 24 40     addss       xmm2,dword ptr [rsp+40h]
00007FF7EF95AB1A 0F 2F D0              comiss      xmm2,xmm0
00007FF7EF95AB1D 0F 97 C0              seta        al
00007FF7EF95AB20 C3                    ret
--- No source file -------------------------------------------------------------
00007FF7EF95AB21 CC                    int         3
```



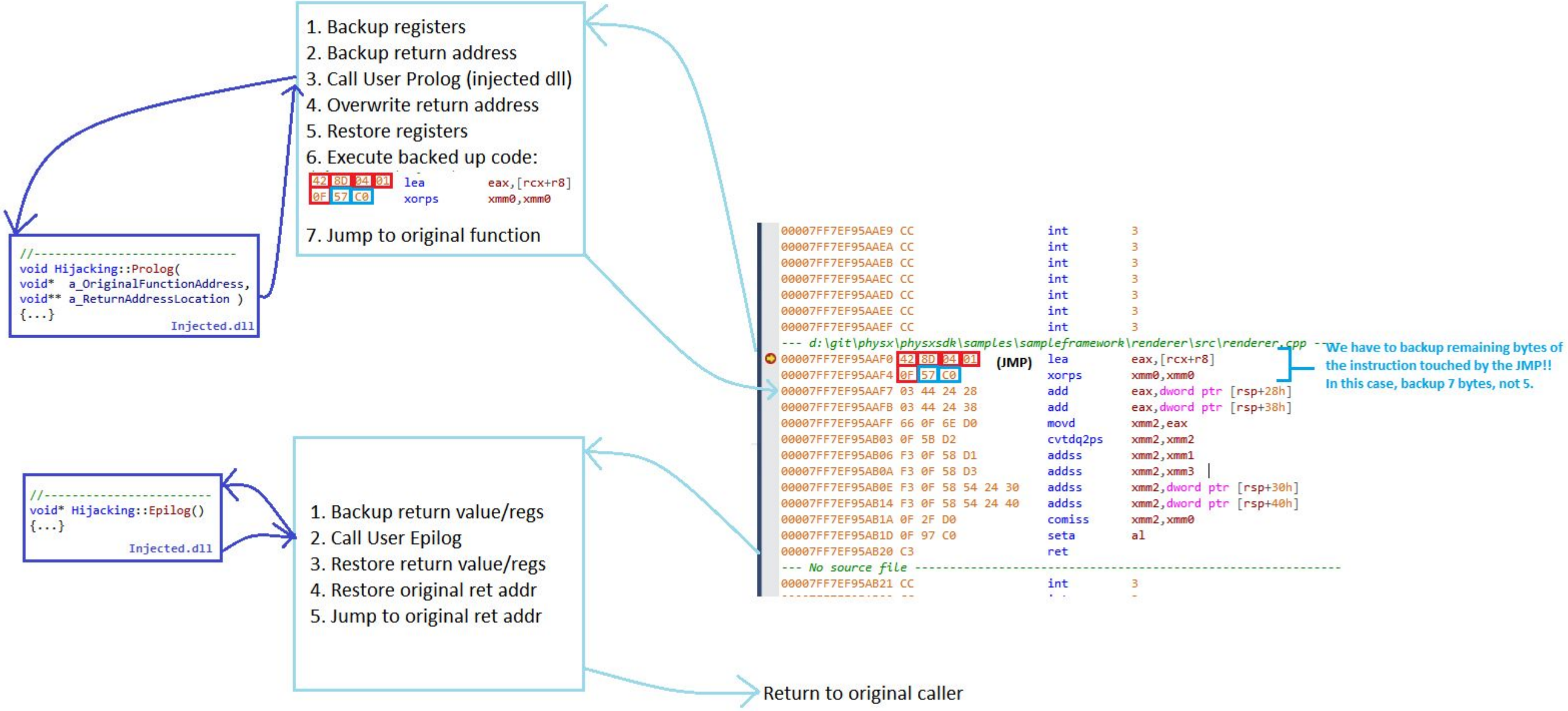
```
__declspec( noinline ) bool TestFunc
    ( int   A
    , float B
    , int   C
    , float D
    , int   E
    , float F
    , int   G
    , float H )
{
    if( ( float(A + C + E + G) + B + D + F + H ) > 0 )
    {
        return true;
    }

    return false;
}
```

Resume from here!



```
00007FF7EF95AAE9 CC                   int         3
00007FF7EF95AAEA CC                   int         3
00007FF7EF95AAEB CC                   int         3
00007FF7EF95AAEC CC                   int         3
00007FF7EF95AAED CC                   int         3
00007FF7EF95AAEE CC                   int         3
00007FF7EF95AAEF CC                   int         3
--- d:\git\physx\physxsdk\samples\sampleframework\renderer\src\renderer.cpp --
00007FF7EF95AAF0 42 8D 04 01          lea         eax,[rcx+r8]
00007FF7EF95AAF4 0F 57 C0             xorps       xmm0,xmm0
00007FF7EF95AAF7 03 44 24 28          add         eax,dword ptr [rsp+28h]
00007FF7EF95AAFB 03 44 24 38          add         eax,dword ptr [rsp+38h]
00007FF7EF95AAFF 66 0F 6E D0          movd        xmm2,eax
00007FF7EF95AB03 0F 5B D2             cvtdq2ps    xmm2,xmm2
00007FF7EF95AB06 F3 0F 58 D1          addss       xmm2,xmm1
00007FF7EF95AB0A F3 0F 58 D3          addss       xmm2,xmm3
00007FF7EF95AB0E F3 0F 58 54 24 30    addss       xmm2,dword ptr [rsp+30h]
00007FF7EF95AB14 F3 0F 58 54 24 40    addss       xmm2,dword ptr [rsp+40h]
00007FF7EF95AB1A 0F 2F D0             comiss      xmm2,xmm0
00007FF7EF95AB1D 0F 97 C0             seta        al
00007FF7EF95AB20 C3                   ret
--- No source file ---------------------------------------------------------------
00007FF7EF95AB21 CC                   int         3
```

We have to backup remaining bytes of the instruction touched by the JMP!! In this case, backup 7 bytes, not 5.



1. Backup registers
2. Backup return address
3. Call User Prolog (injected dll)
4. Overwrite return address
5. Restore registers
6. Execute backed up code:

```
42 8D 04 01    lea      eax,[rcx+r8]
0F 57 C0       xorps    xmm0,xmm0
```

7. Jump to original function

```
//----------------------------
void Hijacking::Prolog(
void*  a_OriginalFunctionAddress,
void** a_ReturnAddressLocation )
{...}
```

Injected.dll

```
00007FF7EF95AAE9 CC                          int         3
00007FF7EF95AAEA CC                          int         3
00007FF7EF95AAEB CC                          int         3
00007FF7EF95AAEC CC                          int         3
00007FF7EF95AAED CC                          int         3
00007FF7EF95AAEE CC                          int         3
00007FF7EF95AAEF CC                          int         3
--- d:\git\physx\physxsdk\samples\sampleframework\renderer\src\renderer.cpp --
00007FF7EF95AAF0 42 8D 04 01      (JMP)      lea         eax,[rcx+r8]
00007FF7EF95AAF4 0F 57 C0                    xorps       xmm0,xmm0
00007FF7EF95AAF7 03 44 24 28                 add         eax,dword ptr [rsp+28h]
00007FF7EF95AAFB 03 44 24 38                 add         eax,dword ptr [rsp+38h]
00007FF7EF95AAFF 66 0F 6E D0                 movd        xmm2,eax
00007FF7EF95AB03 0F 5B D2                    cvtdq2ps    xmm2,xmm2
00007FF7EF95AB06 F3 0F 58 D1                 addss       xmm2,xmm1
00007FF7EF95AB0A F3 0F 58 D3                 addss       xmm2,xmm3
00007FF7EF95AB0E F3 0F 58 54 24 30           addss       xmm2,dword ptr [rsp+30h]
00007FF7EF95AB14 F3 0F 58 54 24 40           addss       xmm2,dword ptr [rsp+40h]
00007FF7EF95AB1A 0F 2F D0                    comiss      xmm2,xmm0
00007FF7EF95AB1D 0F 97 C0                    seta        al
00007FF7EF95AB20 C3                          ret
--- No source file -------------------------------------------------------------
00007FF7EF95AB21 CC                          int         3
```

We have to backup remaining bytes of the instruction touched by the JMP!! In this case, backup 7 bytes, not 5.

```
//------------------------
void* Hijacking::Epilog()
{...}
```

Injected.dll

1. Backup return value/regs
2. Call User Epilog
3. Restore return value/regs
4. Restore original ret addr
5. Jump to original ret addr

Return to original caller

What would it involve?

1. Some information about the target program ☑

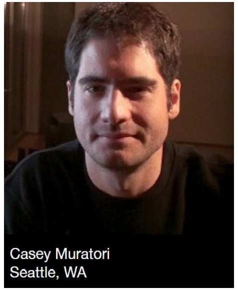
2. A way to inject our code and open up a communication channel ☑

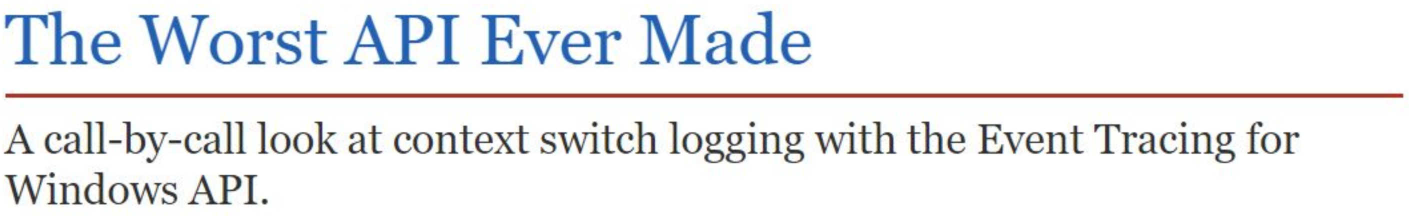
3. A way to dynamically instrument (live code modification) ☑

What about Sampling?

Turns out it’s fairly easy to do...

- Once you go through the hell of setting up ETW





The Worst API Ever Made

A call-by-call look at context switch logging with the Event Tracing for Windows API.

https://mollyrocket.com/casey/stream_0029.html

Thank You Casey!

(

Check out his blog:

https://mollyrocket.com/

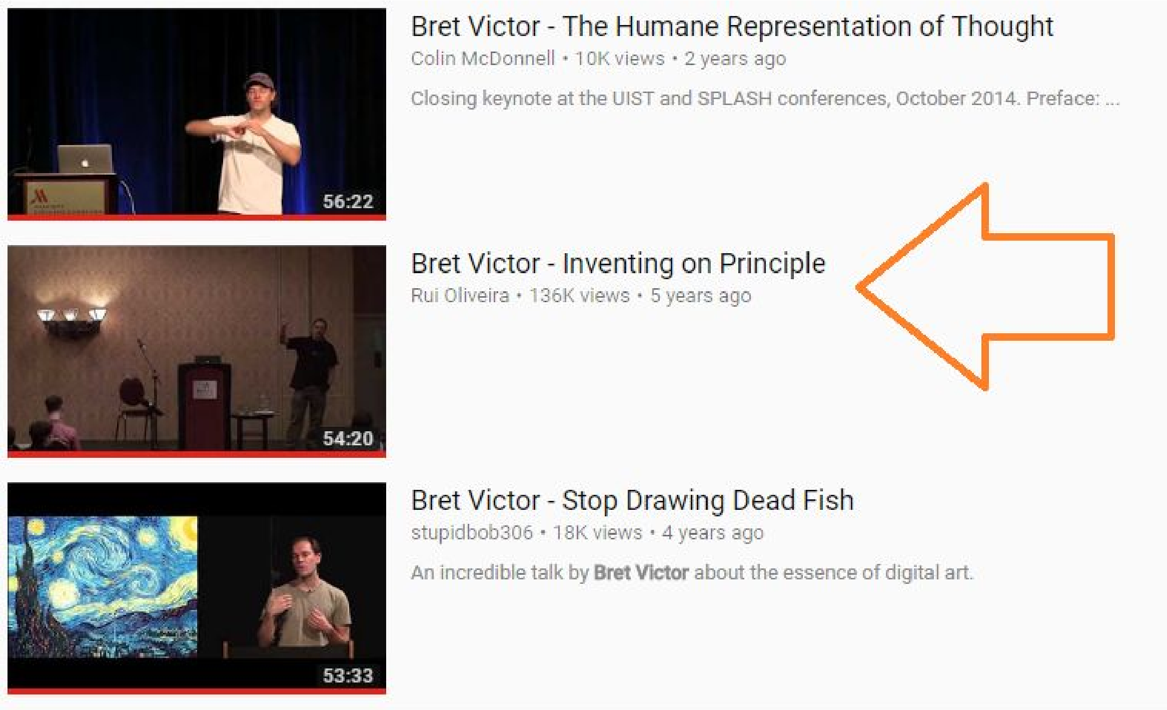
Awesome GJK video (Personal favorite):

https://mollyrocket.com/849

https://www.youtube.com/watch?v=SDS5gLSiLg0

)

Bret Victor



Bret Victor - The Humane Representation of Thought

Colin McDonnell • 10K views • 2 years ago

Closing keynote at the UIST and SPLASH conferences, October 2014. Preface: ...

Bret Victor - Inventing on Principle

Rui Oliveira • 136K views • 5 years ago

Bret Victor - Stop Drawing Dead Fish

stupidbob306 • 18K views • 4 years ago

An incredible talk by **Bret Victor** about the essence of digital art.

orbit

PROFILER

One more thing...

Setting it Free...

www.orbitprofiler.com

Get your licence here (temporary, until open source…)

orbitprofiler.com/cpp.html

orbit

PROFILER