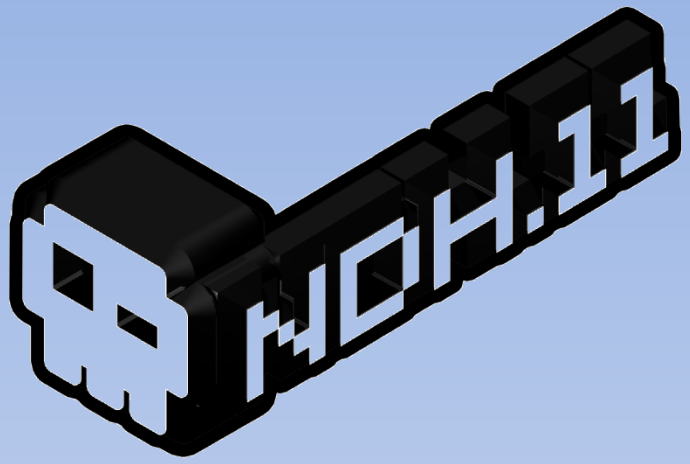
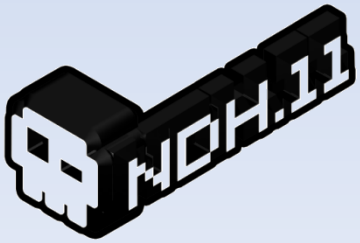


ZeroFuzz

Vulnerability research in Windows Kernel

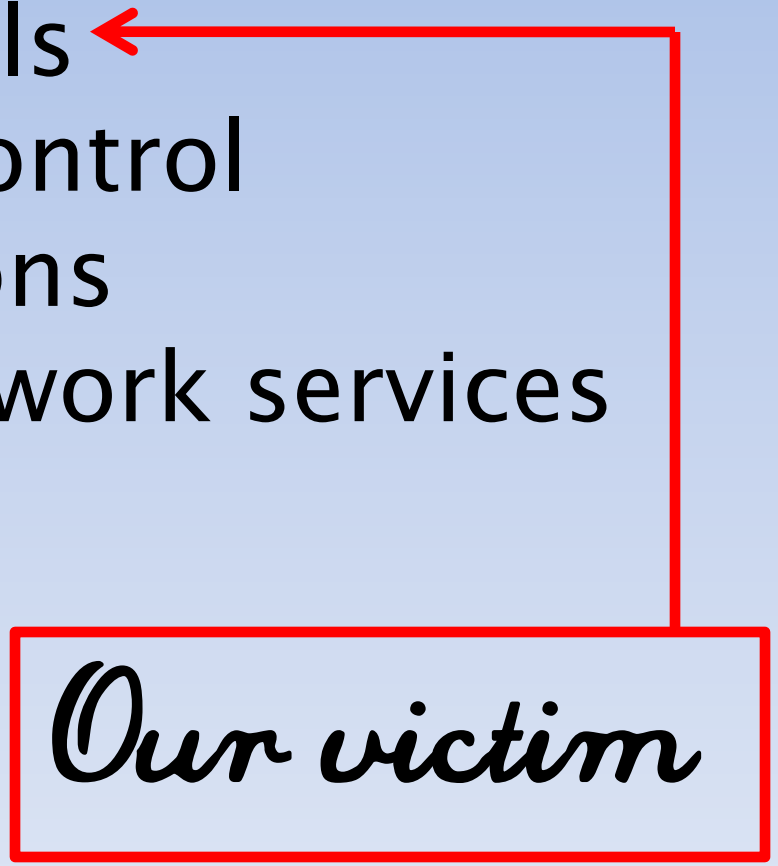
Summary

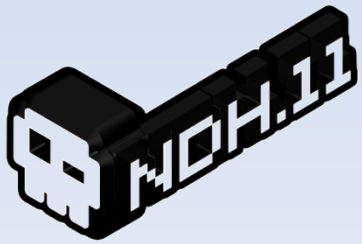
Windows Kernel interactions
Previous fuzzings
My own fuzzing method
ZeroFuzz concepts
Final version !
Demo
Your downloadable version



Windows Kernel interactions

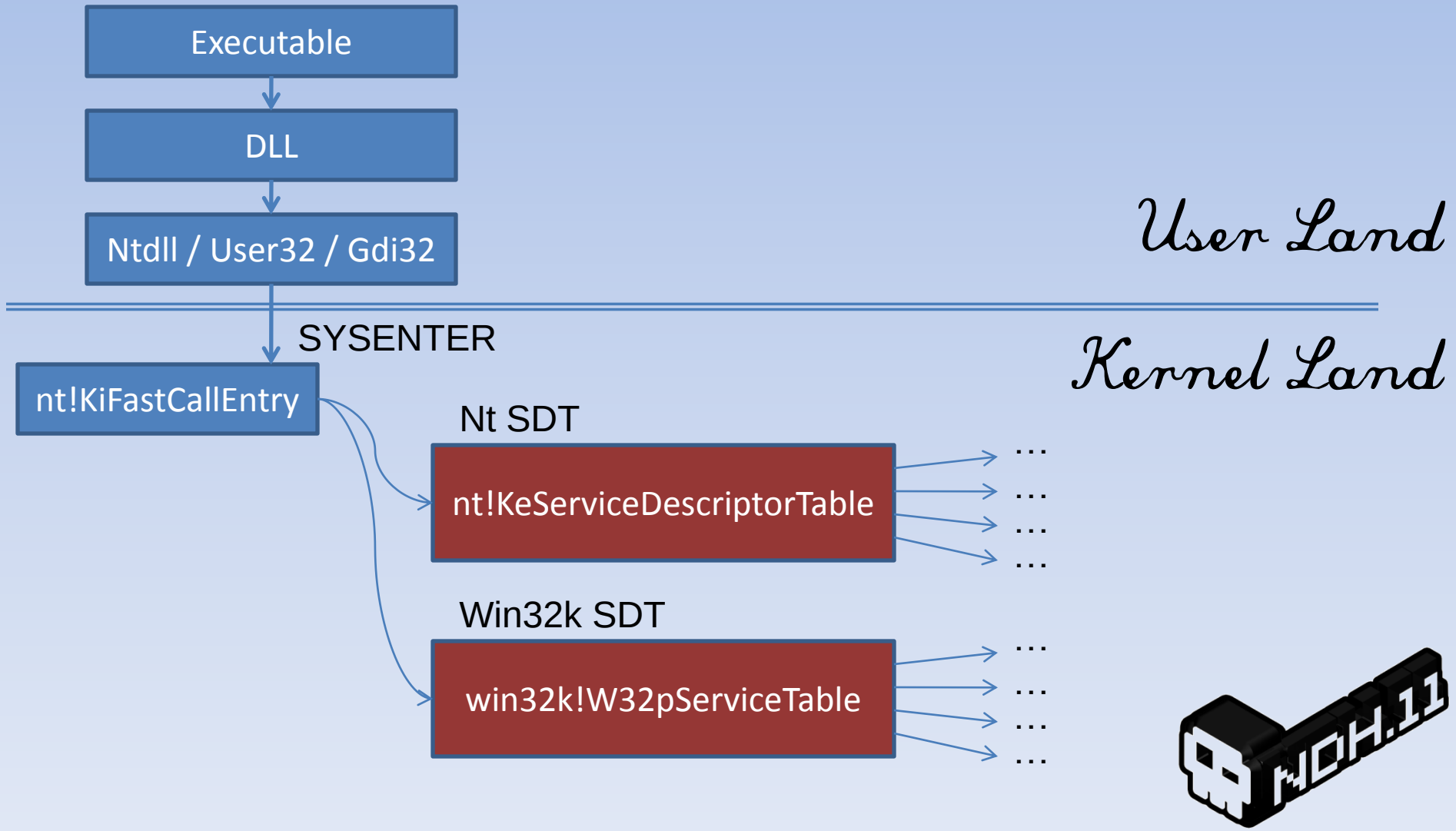
System calls
DeviceIoControl
Interruptions
Kernel network services





Windows Kernel interactions

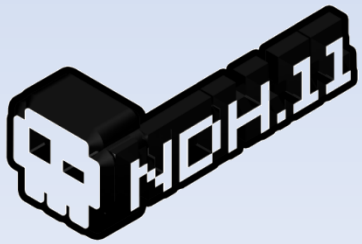
Windows Kernel interactions



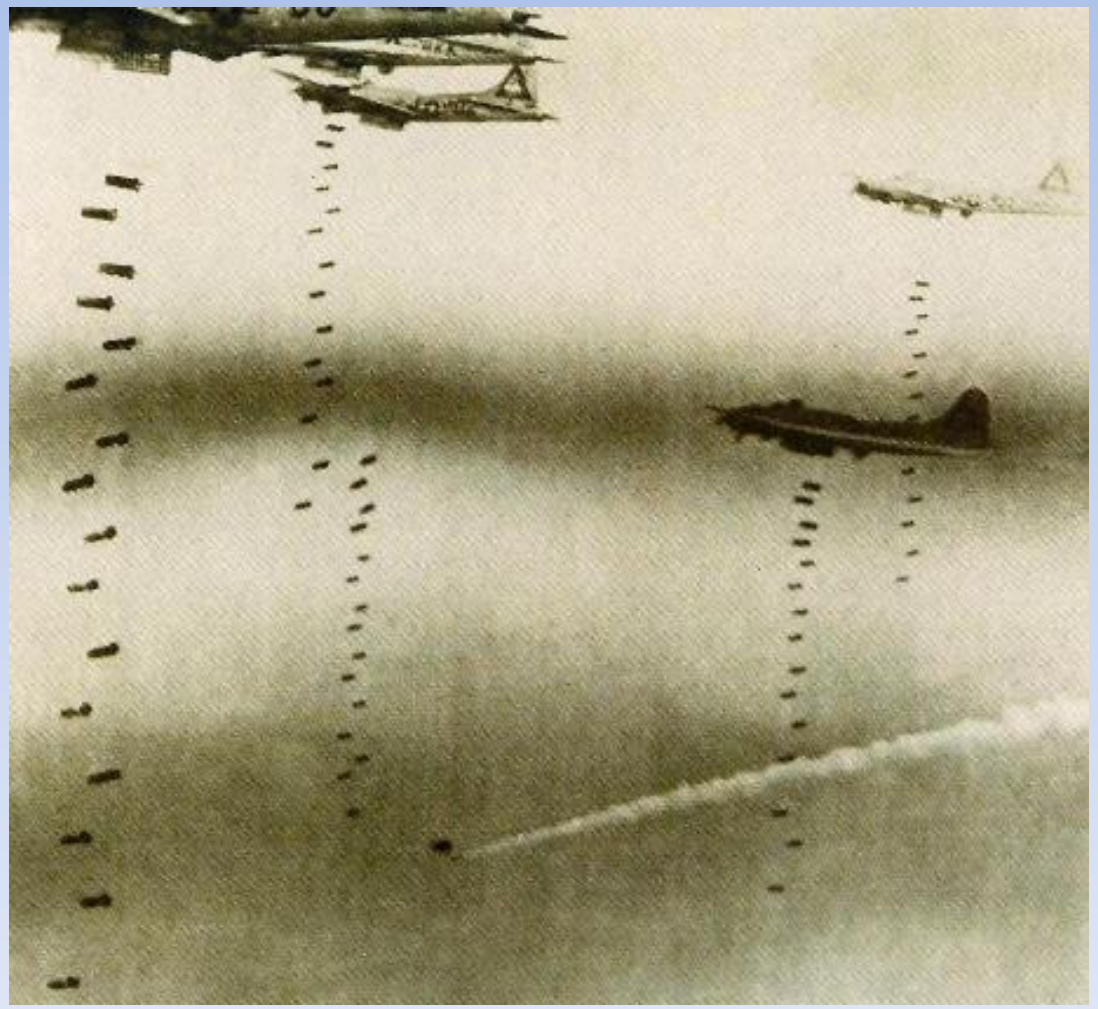
Previous fuzzing

IoCtlFuzzer (just one kernel function)
NtCrash2 (Mark Russinovich)

- Old fuzzer (Int 2E)
- Static fuzzing
- Low code covering
- Fuzz Nt kernel & Win32k
- Code is available on archive.org ;-)



Previous fuzzing

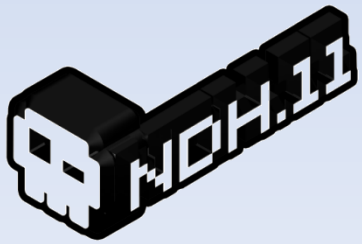


My own fuzzing method

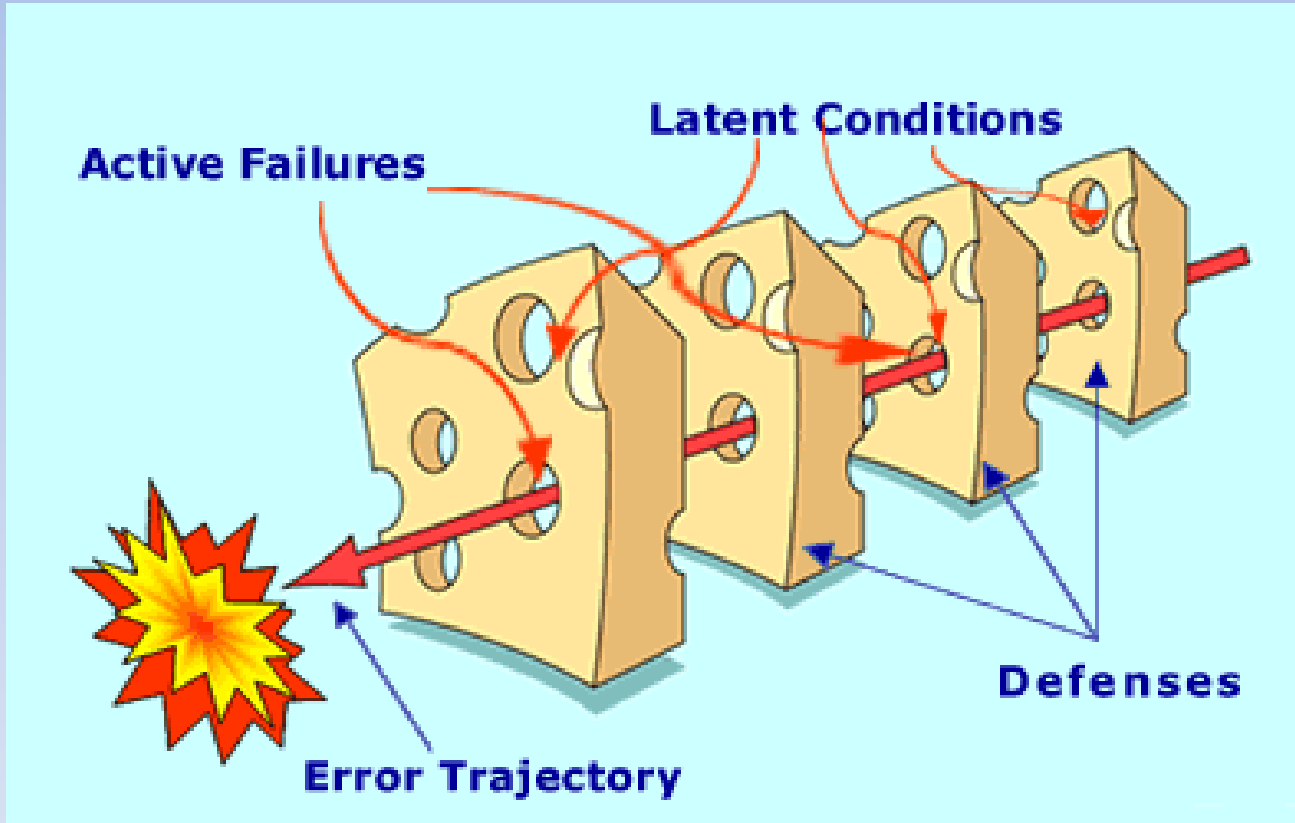
Problems of NtCrash2 :

- Just inject bad code
- Don't test any Handle
- No fuzzing of pointed datas

Solution : Hot fuzzing !
Tamper arguments based on a valid system call

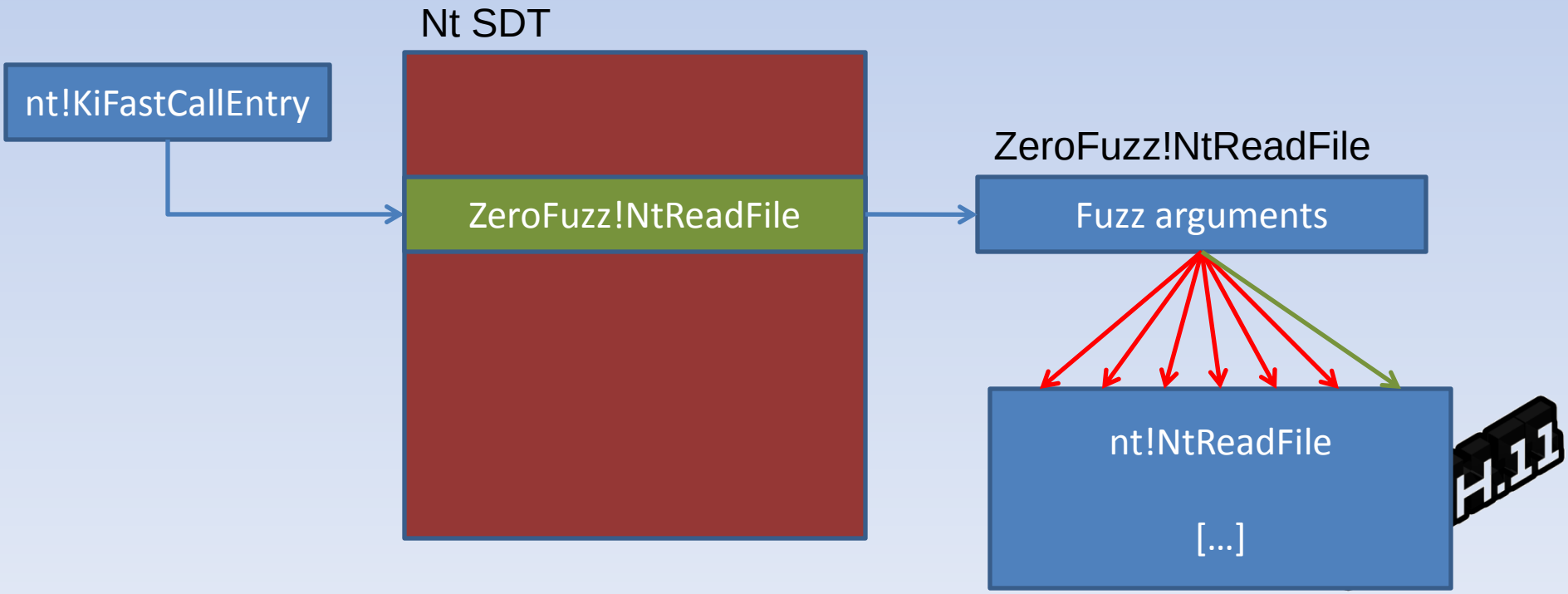


We can pass some checks…



ZeroFuzz concepts

Hook a SDT kernel function
Tamper its arguments

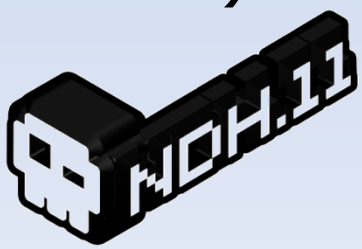


IeroFuzz concepts

FuzzedFunction(Argument1 , 0)
FuzzedFunction(0x00000000 , 0)
FuzzedFunction(0xFFFFFFFF , 0)
FuzzedFunction(0x80000000 , 0)
FuzzedFunction(0x7FFFF000 , 0)
FuzzedFunction(Argument1 , 0)

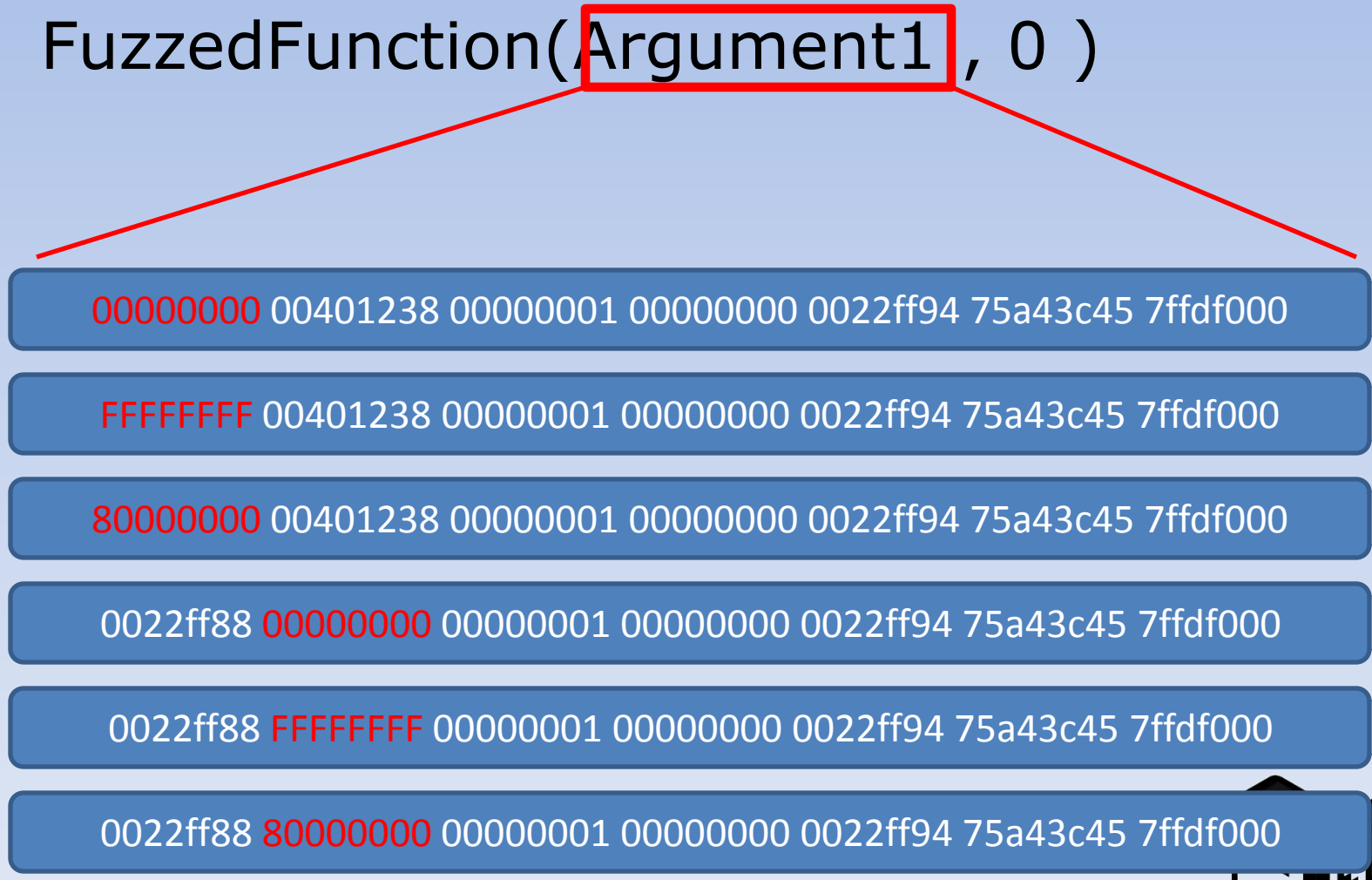
→ AAAAA[...]AAAAA

FuzzedFunction(Argument1 , 0x00000000)
FuzzedFunction(Argument1 , 0xFFFFFFFF)
[...]



TeroFuzz concepts

FuzzedFunction(Argument1 , 0)

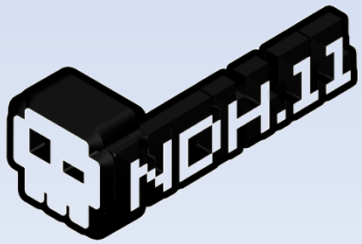


IeroFuzz Final version

We hook all entries of Nt table and Win32k table
Just one fuzzing module

Problems :

- Retrieve id of system call
- Retrieve number of arguments
- Fuzz just for a limited user
- Make sure we're called from the user land

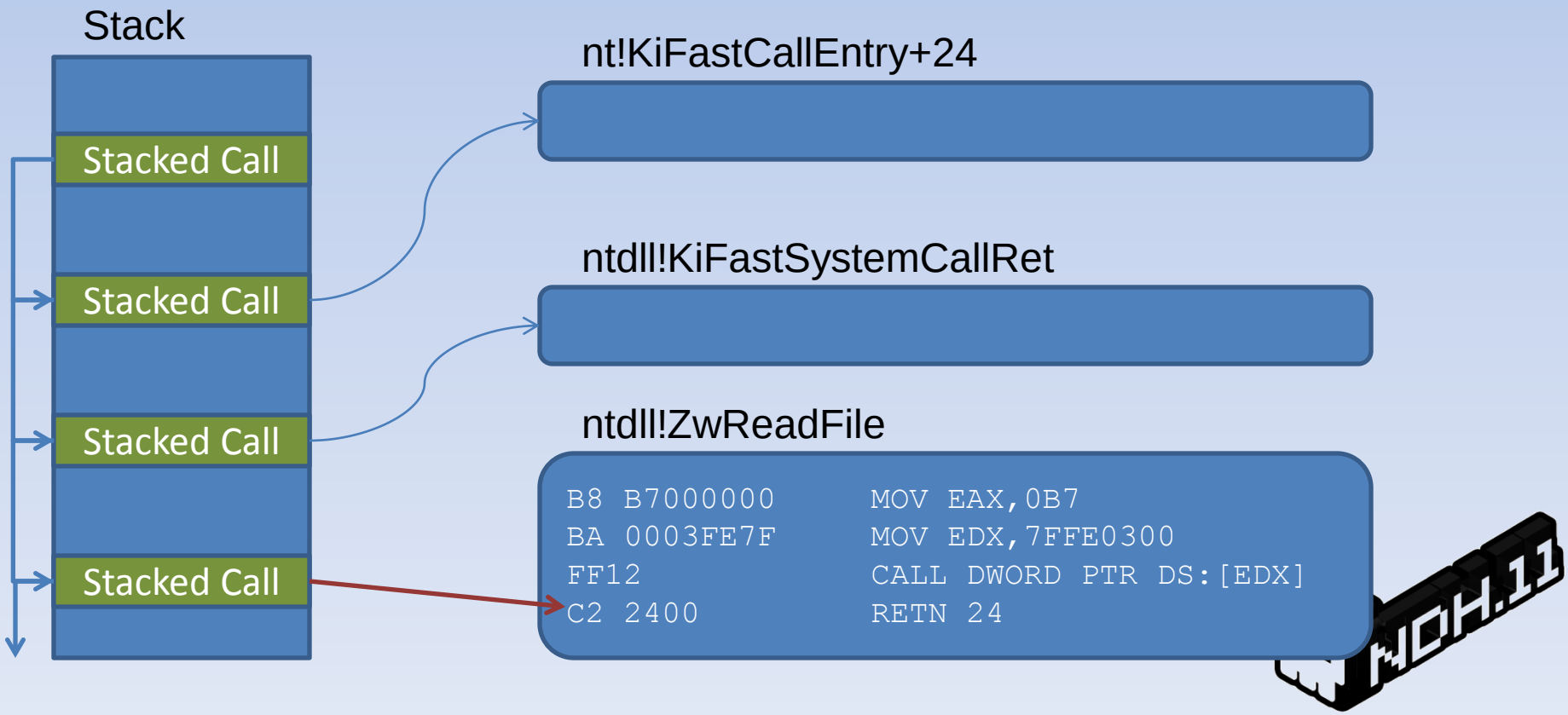


Walk up the stack

TeroFuzz Final version

Retrieve ID of system call

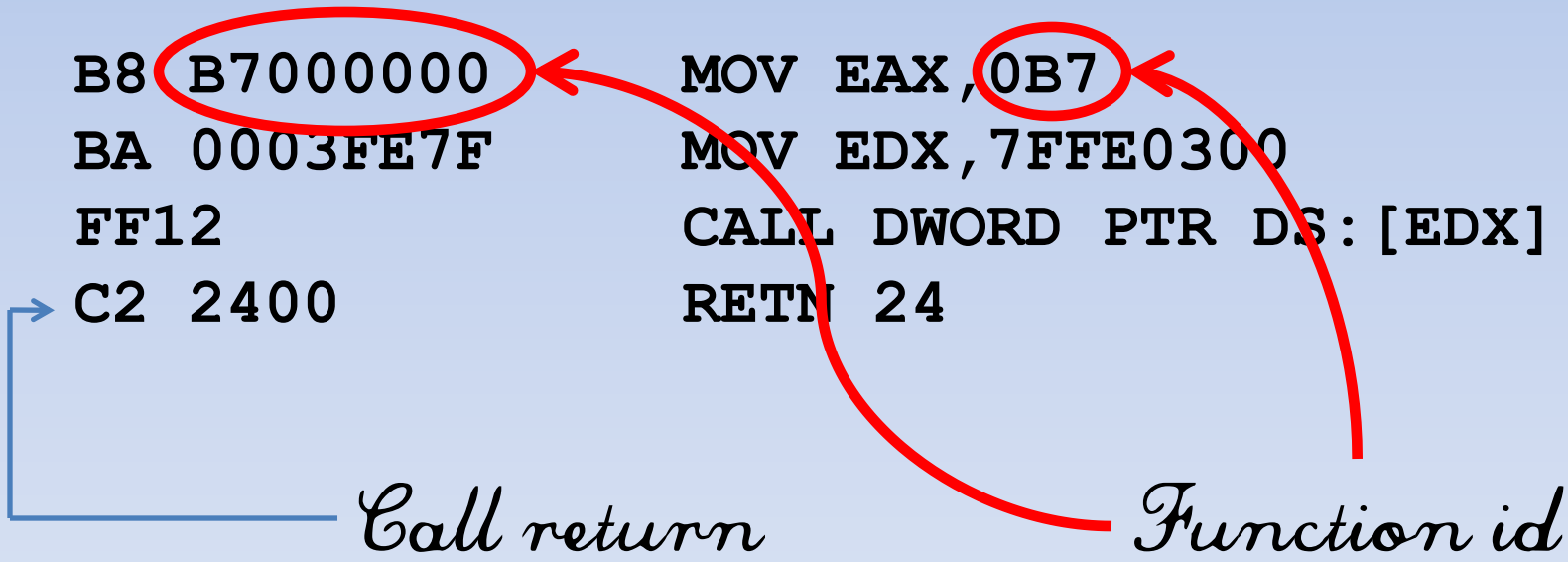
Parsing the call stack to found the Zw…



TeroFuzz Final version

Retrieve ID of system call

Sub 0xB to get the ID



IeroFuzz Final version

Retrieve number of arguments

Add 0x1 to get how much argument are present (you must div by 4)

```
B8 B7000000      MOV EAX,0B7
BA 0003FE7F      MOV EDX,7FFE0300
FF12             CALL DWORD PTR DS:[EDX]
C2 2400          RETN 24
```

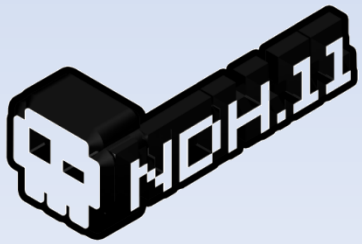
TeroFuzz Final version

Fuzz just for a limited user

IeroFuzz Final version

Fuzz just for a limited user

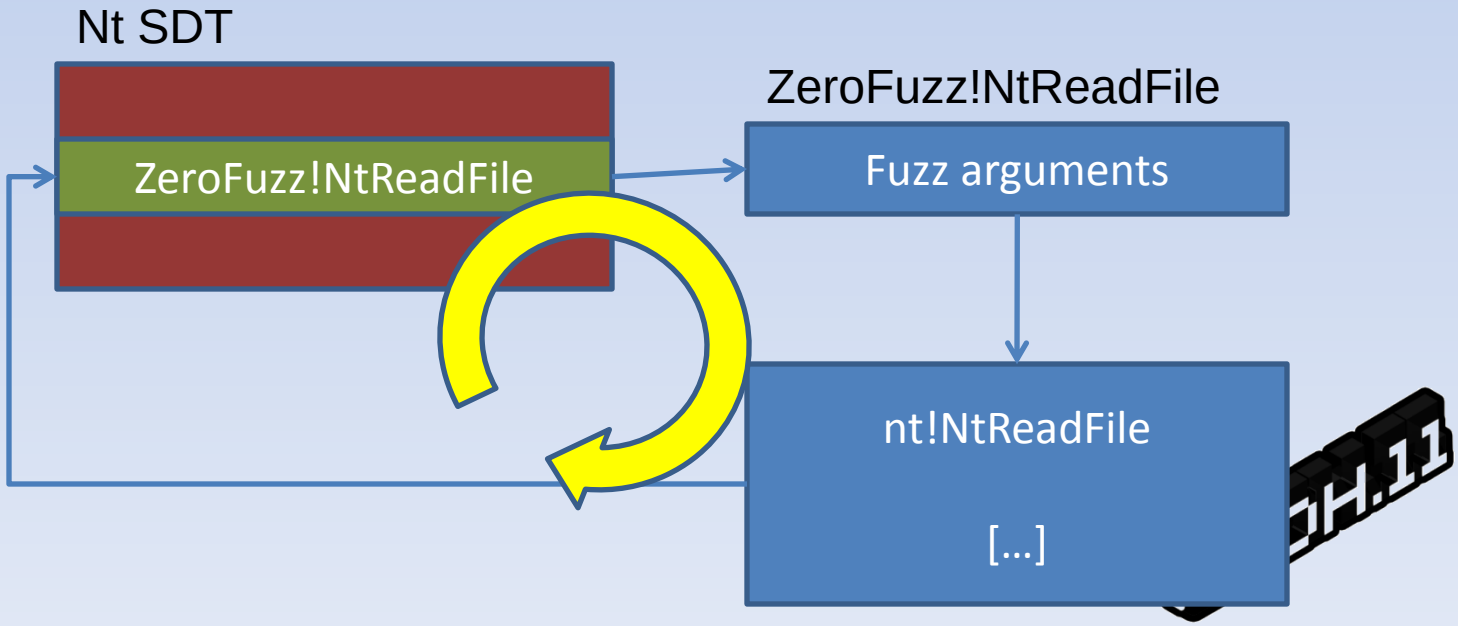
If we fuzz interactions between smss, csrss, lsass, etc and the kernel we'll have probably a no reproducible crash.
For each process a SID is used, it's the identifier of user rights.
We can get a user SID and when a function is called to check if user SID is same.



ZeroFuzz Final version

Make sure we're called of call from the user land

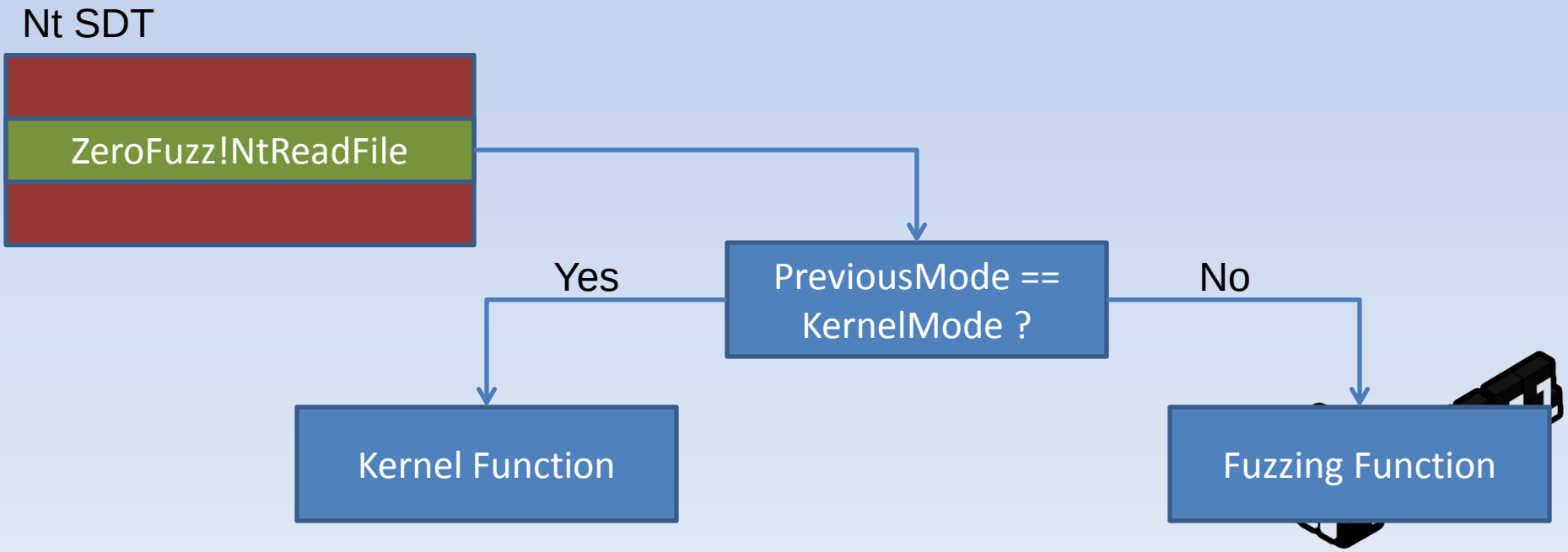
Kernel Land can to call SDT and at this moment we can enter in a loop :-(



ZeroFuzz Final version

Make sure we're called of call from the user land

Before call a SDT function the kernel change his "PreviousMode" to "KernelMode".



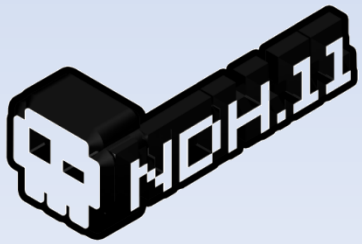
TeroFuzz Final version

Be sure of call from the user land

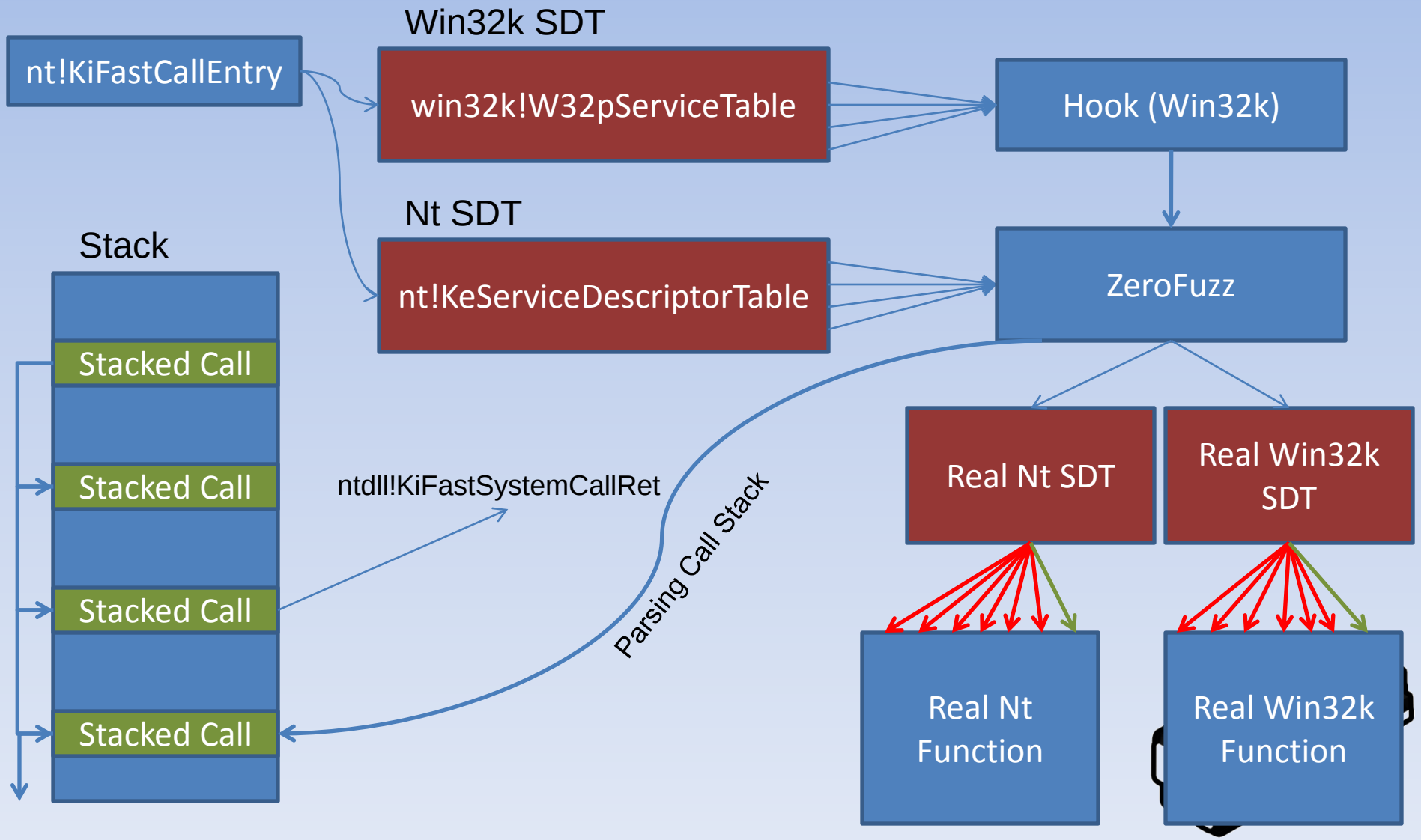
Parsing call stack to found :

- ntdll!KiFastSystemCallRet
- A user land call stack
- ID of the function
- Not KernelMode access

If one check isn't valid we pass the execution to the real function.



ZeroFuzz


Win32k SDT
nt!KiFastCallEntry
win32k!W32pServiceTable
Hook (Win32k)
Nt SDT
nt!KeServiceDescriptorTable
ZeroFuzz
Stack
Stacked Call
Stacked Call
Stacked Call
Stacked Call
ntdll!KiFastSystemCallRet
Parsing Call Stack
Real Nt SDT
Real Win32k SDT
Real Nt Function
Real Win32k Function

Catch vulnerabilities !

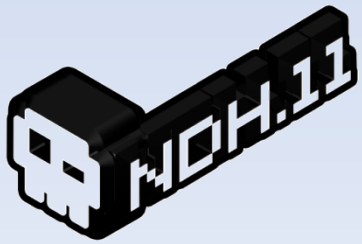
TeroFuzz

Advantages :

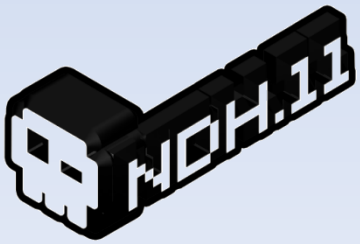
- Handles are respected
- Code coverage is optimized
- Few false positives with a limited user

Disadvantages:

- Lot of side effects
- Hard tracing
- Difficult replay



Demo !



Boom !

```
Fuzz Datas pointed (value : 0x0553f938) of argument : 1, Value : 0x0, function : 0x188
Fuzz Datas pointed (value : 0x061df66c) of argument : 2, Value : 0x61, function : 0xea
Fuzz argument 1, Value : 0x00000000, function : 0x27 (39) Nt (PID : 2552, TID : 2296)
Fuzz DWORDs (value : 0x0bdbc180) pointed of argument : 4 Nt (PID : 39, TID : 39)
Fuzz Datas pointed (value : 0x061df66c) of argument : 2, Value : 0xffffffff, function
Fuzz Datas pointed (value : 0x061df66c) of argument : 2, Value : 0x0, function : 0xea
Fuzz DWORDs (value : 0x061df354) pointed of argument : 2 Nt (PID : 234, TID : 234)
Fuzz argument 2, Value : 0x05cddc5c, function : 0x149 (329) Nt (PID : 2436, TID : 3432
Fuzz argument 1, Value : 0x00000000, function : [redacted] ([redacted]) Nt (PID : 2436, TID : 3432)

*** Fatal System Error: 0x0000000a
                       (0xFFFFFFFC,0x000000FF,0x00000001,0x[redacted])

Break instruction exception - code 80000003 (first chance)

A fatal system error has occurred.
Debugger entered on first try; Bugcheck callbacks have not been invoked.

A fatal system error has occurred.

Connected to Windows 7 7600 x86 compatible target at (Tue May 24 14:59:53.624 2011 (GM
Loading Kernel Symbols
...............................................................
...............................................................
....................
Loading User Symbols
```

```
Que faire ?
 Installer (I)
 Desinstaller (D)
 Charger (L)
 Decharger (U)
 Call Driver (C)
>Le driver vas etre charge... [OK]


Programme termine avec succes.
Que faire ?
 Installer (I)
 Desinstaller (D)
 Charger (L)
 Decharger (U)
 Call Driver (C)
Fuzz to a special user ? (Y/N) [Y]
User name : toto

RuLlz !!!! Mouahahahahaha
```

Small analyze…

```
kd> !analyze -v
[…]
WRITE_ADDRESS:  fffffffc

CURRENT_IRQL:  2

FAULTING_IP:
nt!*****************+8c
******** 897308          mov     dword ptr [ebx+8],esi
eax=c0000005 ebx=fffffff4 ecx=853101c0 edx=00000400 esi=05000000 edi=844c3a78
eip= ******** esp=844c3a80 ebp=844c3a80 iopl=0         nv up di ng nz ac po cy
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00010093
nt!*****************+8c:
******** 897308          mov     dword ptr [ebx+8],esi
ds:0023:fffffffc=????????
kd> !peb
PEB at 7ffde000
[…]
 Base TimeStamp                     Module
          ec0000 4d65d5c3 Feb 24 04:51:31 2011 C:\Program Files\Internet
Explorer\iexplore.exe
[…]
 USERNAME=toto
[…]
```

Kernel pointer dereferencement

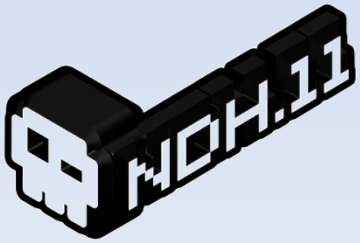
Toto is a limited user

Your downloadable version

You can:

- Fuzz functions one by one
- Select a limited user to fuzz
- Have a kernel tracing

Demo !



Your downloadable version

```
Programme termine avec succes.
Que faire ?
 Installer (I)
 Desinstaller (D)
 Charger (L)
 Decharger (U)
 Call Driver (C)
 Cmd.exe (S)

[0] Ntoskrnl SDT
[1] Win32k SDT
[R] Random Fuzz :-P
Select SDT to Fuzz : [R]
Fuzz to a special user ? (Y/N) [Y]
User name : toto
SID of the user is : S-1-5-21-1386401058-323789324-313879924-1001


There are 388 Fuzzable functions in Nt Kernel
There are 765 Fuzzable functions in Win32k Kernel
So, there are 1153 Fuzzable functions
Randomize....
Win32k Table is selected
Function 660 is selected
There are 3 args to fuzzRuLlz !!!!
Mouahahahahaha
```

```
Que faire ?
 Installer (I)
 Desinstaller (D)
 Charger (L)
 Decharger (U)
 Call Driver (C)
 Cmd.exe (S)

[0] Ntoskrnl SDT
[1] Win32k SDT
[R] Random Fuzz :-P
Select SDT to Fuzz : [0]


List of SSDT ? (Y/N) [N]

Select your function ID to FuZz : 1
  This Function have 8 arguments
  Argument n.0 (Y/N) [Y]
  Argument n.1 (Y/N) [N]
  Argument n.2 (Y/N) [Y]
  Argument n.3 (Y/N) [Y]
  Argument n.4 (Y/N) [N]
  Argument n.5 (Y/N) [N]
  Argument n.6 (Y/N) [Y]
  Argument n.7 (Y/N) [N]
Fuzz to a special user ? (Y/N) [Y]
User name : toto

RuLlz !!!! Mouahahahahaha
```

http://www.sysdream.com/sites/default/files/ZeroFuzz.v0.1.zip

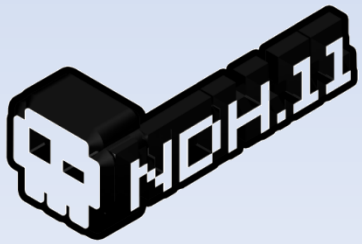
s.leberre@sysdream.com
heurs@ghostsinthestack.org

Credits

SYSDREAM : http://www.sysdream.com

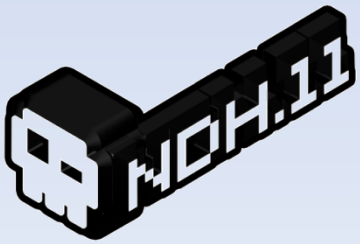
GhostsInTheStack : http://ghostsinthestack.org

Stéfan LE BERRE (Heurs)



Greetz

Trance
Virtualabs



Thanks for your attention

Any question ?

It's time to drink !