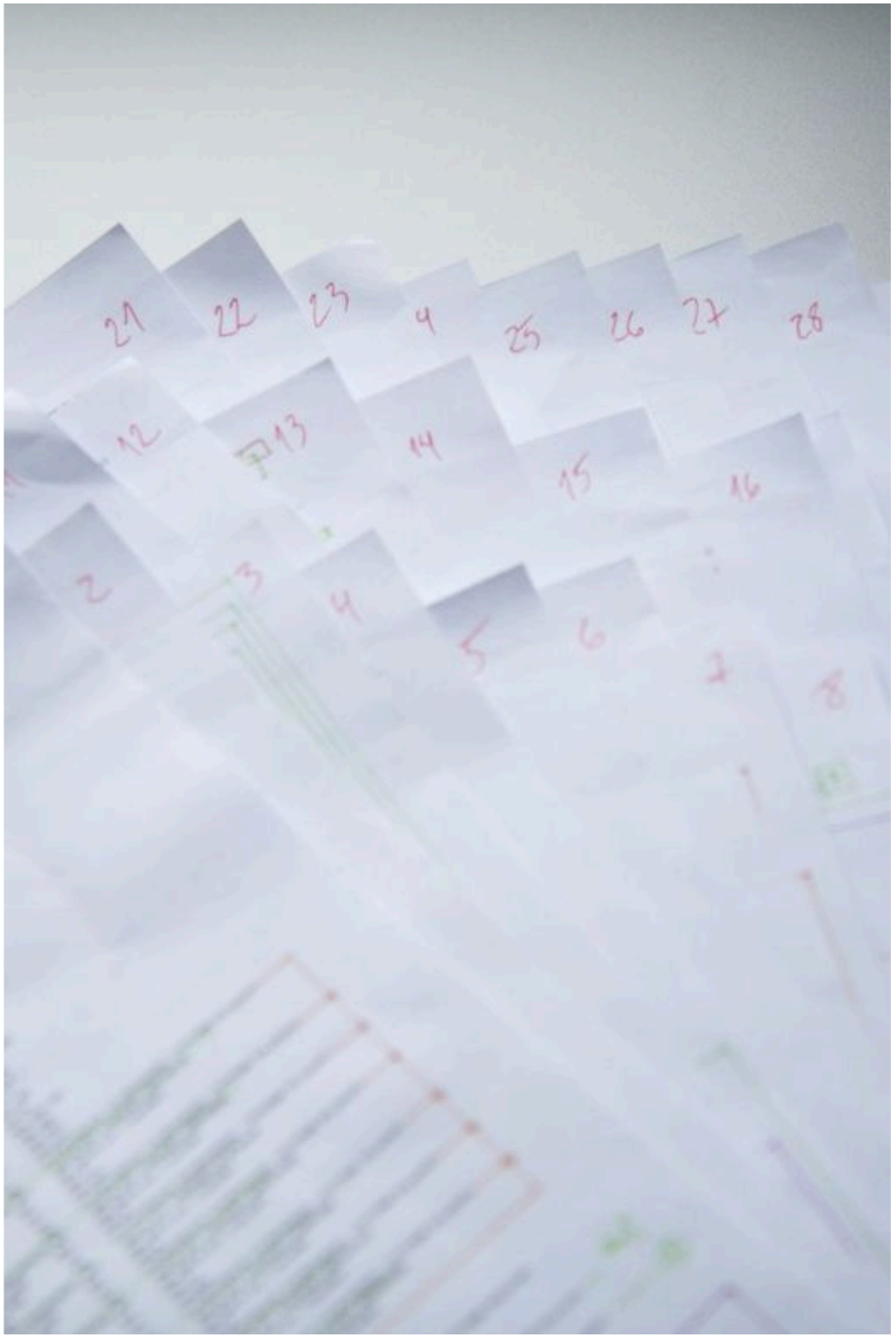
0-RING x64

Gustavo Scotti,
Immunity, Inc.

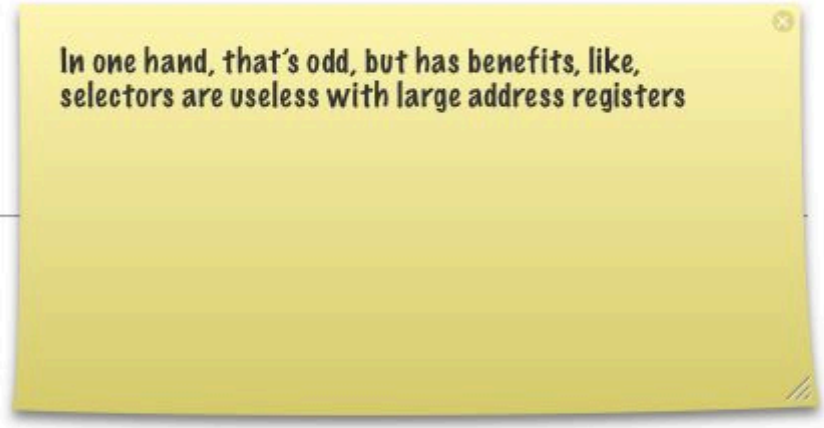
Agenda:

- Differences for x86 vs. x64
- The techniques
- PatchGuard
- Deactivating it
- 2-stages approach
- Lessons learnt



Differences for x86 vs x64

- No GDT / selectors (cs and ds only)
- 64-registers, address is 48-bits, bye bye selectors
- Still have 4k paging
- CR0 bit 10h trick still valid (?)
- No task switching (TSS), but Task Priority Level (CR8)
- The calling conventions



Code Signing

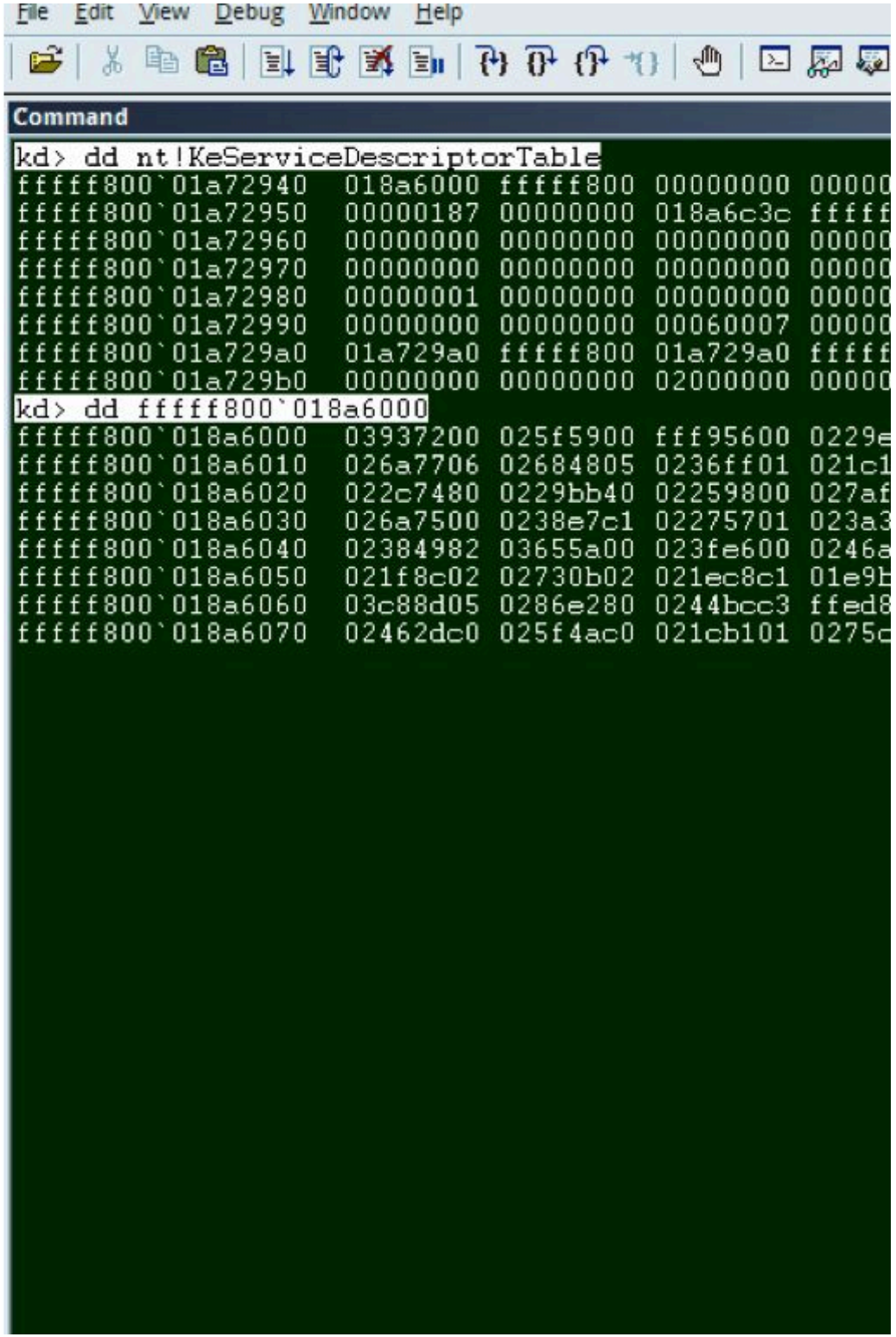
- In 64-bit, all drivers need to be signed
- PnP has a mandatory catalog (.cat) companion, which is also signed
- Non-PnP or boot drivers, embedded signature is enough
- Have you bought your ticket to the Ring-0?

The Techniques Repertoire

- Hook'n'Roll (Jump Around)
- Page Re-referencing
- Rewriting the Service Table
- IAT for ntdll
- Function Rewind Exception Hijack

KeServiceTable

- #1 choice of 32-bitters
- X64 uses a different approach
- Offset to the service is 32-bit
- Base address is the table
- 0x10 aligned, last 4 bits used to stack alignment



KeServiceTable

- The trick is to rebase the entire table
- JMP [0] will jump to the next “instruction” as it were the address (in 64-bits)
- Will show you how it’s done!

Next Slide is the video

KssCloneSsdt.for.else | else

```
yours_service_table_base_address = (PUCHAR )ExAllocatePool( N
if (yours_service_table_base_address)
{
    // we align it so the process will be happy -- so is wink
    yours_service_table_base_address = (PUCHAR) ALIGN_UP_POIN
}

__pseudo_service_descriptor_table__[ TableIdx ].ServiceTable
__pseudo_service_descriptor_table__[ TableIdx ].TableSize = o

// get original service table base address
original_service_table_base_address = (PTR_INTEGER) original_

// yours code start offset will point right after the service
yours_code_start_offset = ALIGN_UP( (PTR_INTEGER) original_se

// now we will fix every single service
for (n = 0; n < original_service_table->TableSize; n++)
{
    PTR_INTEGER     original_service_ptr;
    ULONG           yours_service_code_offset;
    ULONG           service_stack_reserve;
    ULONG           original_service_value;

    /* this is the trick system service table offset:
       as an ULONG (32 bits)

       first 4 bits will get us how much of the stack should
       remaining 28 bits will get us service table address of
     */

    original_service_value = (ULONG) original_service_table->
    service_stack_reserve = original_service_value & 0xf;

    // this is for vista and 7
    yours_service_code_offset = (ULONG) (yours_code_start_off

    // get original service function address
    // checking if this value is not negative
```

KeServiceTable - In Action

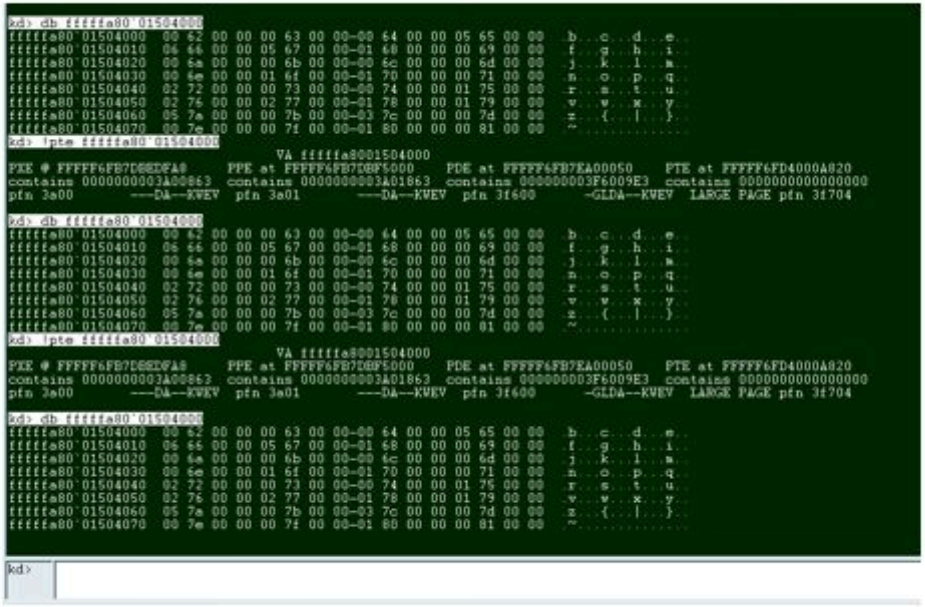
Hook'n'Roll

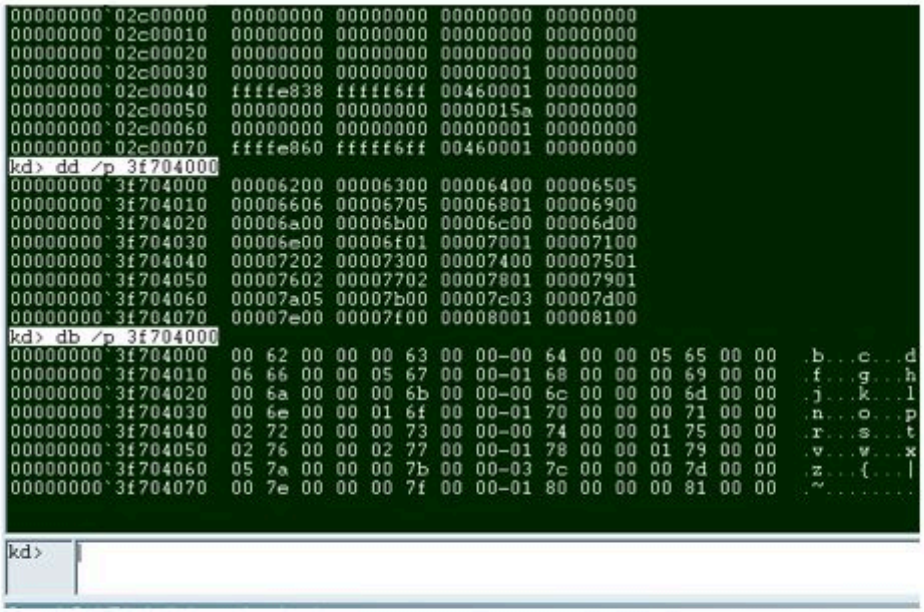
- Copy overwritten bytes to a temp buff
- Make code to jump to somewhere
- Original call back is the temp buff
- And a jump back to original code
- Voila

```
    {
        return 0;
    }

    return 1;
}


PGATE_HOOKER HookAt( void * TargetPtr, void * Hooke
{
    PGATE_HOOKER    hooker;
    int             i;
    unsigned char           *TargetBytePtr = (unsig

    if (!is_system_initialized)
    {
        HookInit();
    }


    for ( i = 0; i < MAX_HOOK_GATES; i++)
    {
        hooker = &g_hookers[ i ];
        if ( !hooker->TransientSize ) break;
    }

    if ( i == MAX_HOOK_GATES ) return NULL;


    // test if our transient pool size isn't too bi
    if (TransientSize > MAX_TRANSIENT_INSTRUCTION_S

    if (!AssemblyJump( hooker->CodeGate, HookedPtr
    {
        return NULL;
    }

    hooker->TransientSize = TransientSize;
```

Page Referencing

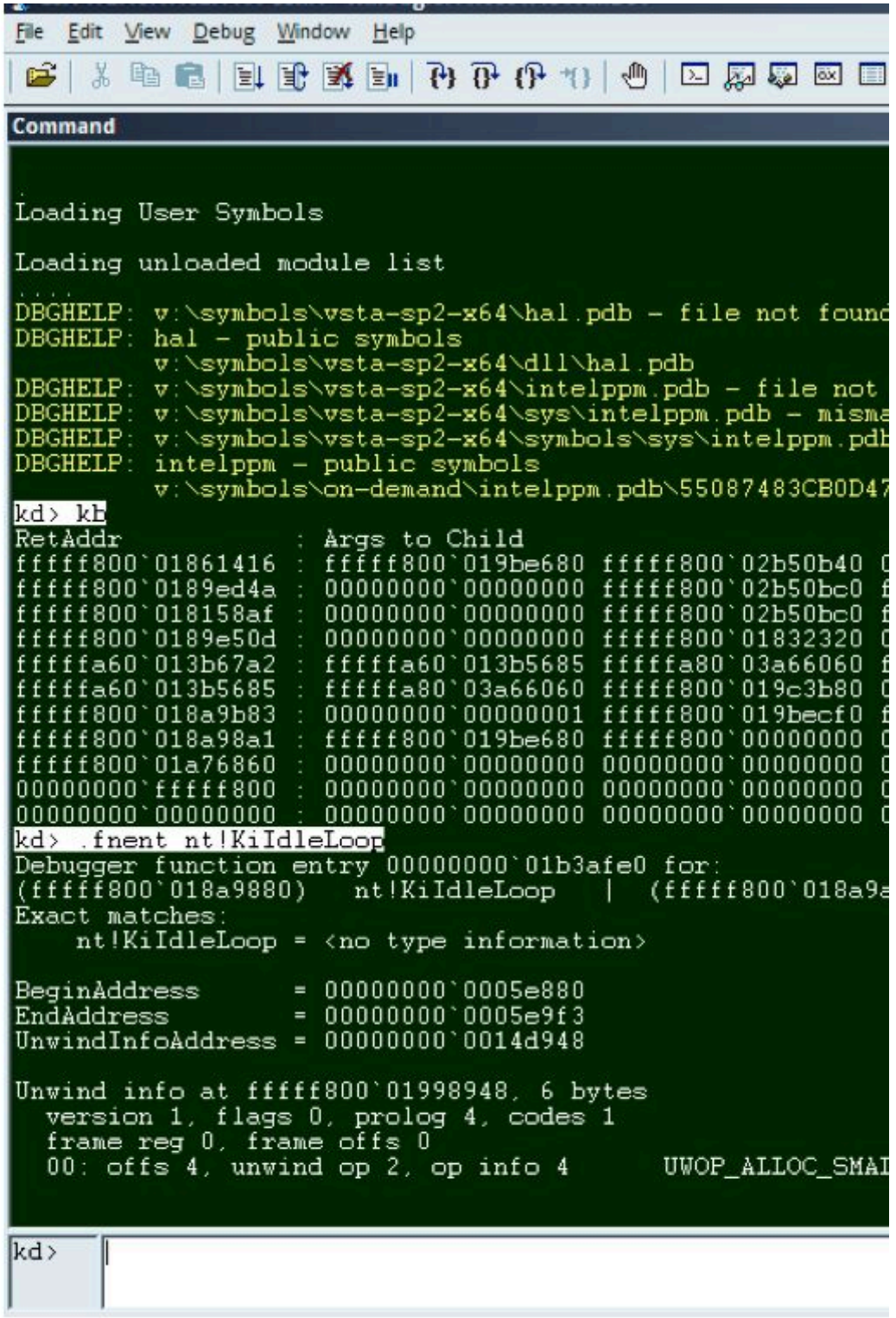
- Figure out where in physical memory it is
- You can remap the same physical address
- You can copy your content to another physical memory
- Reference a virtual address to another physical page
- That’s the PAGE fun!





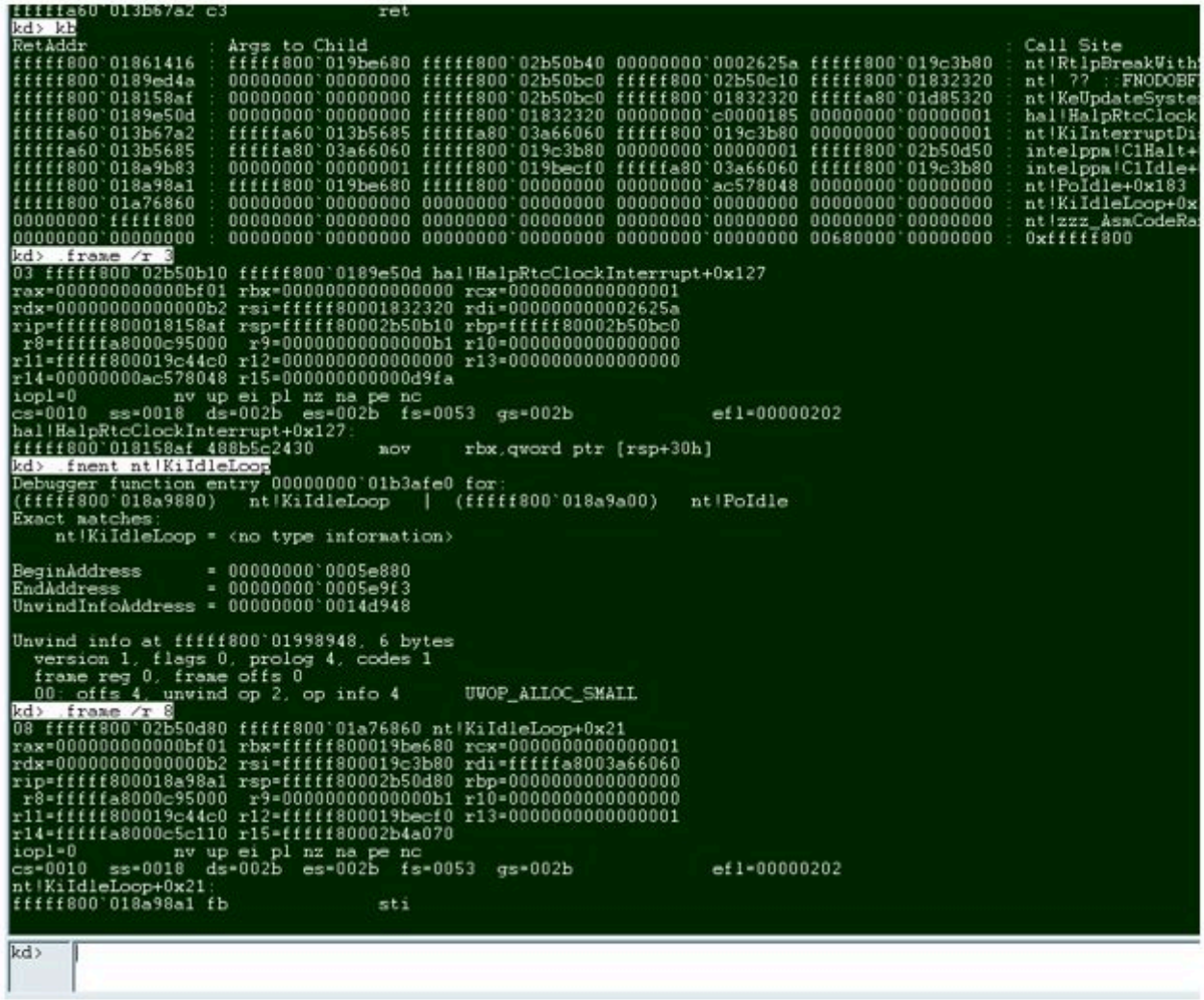
Exception Handler

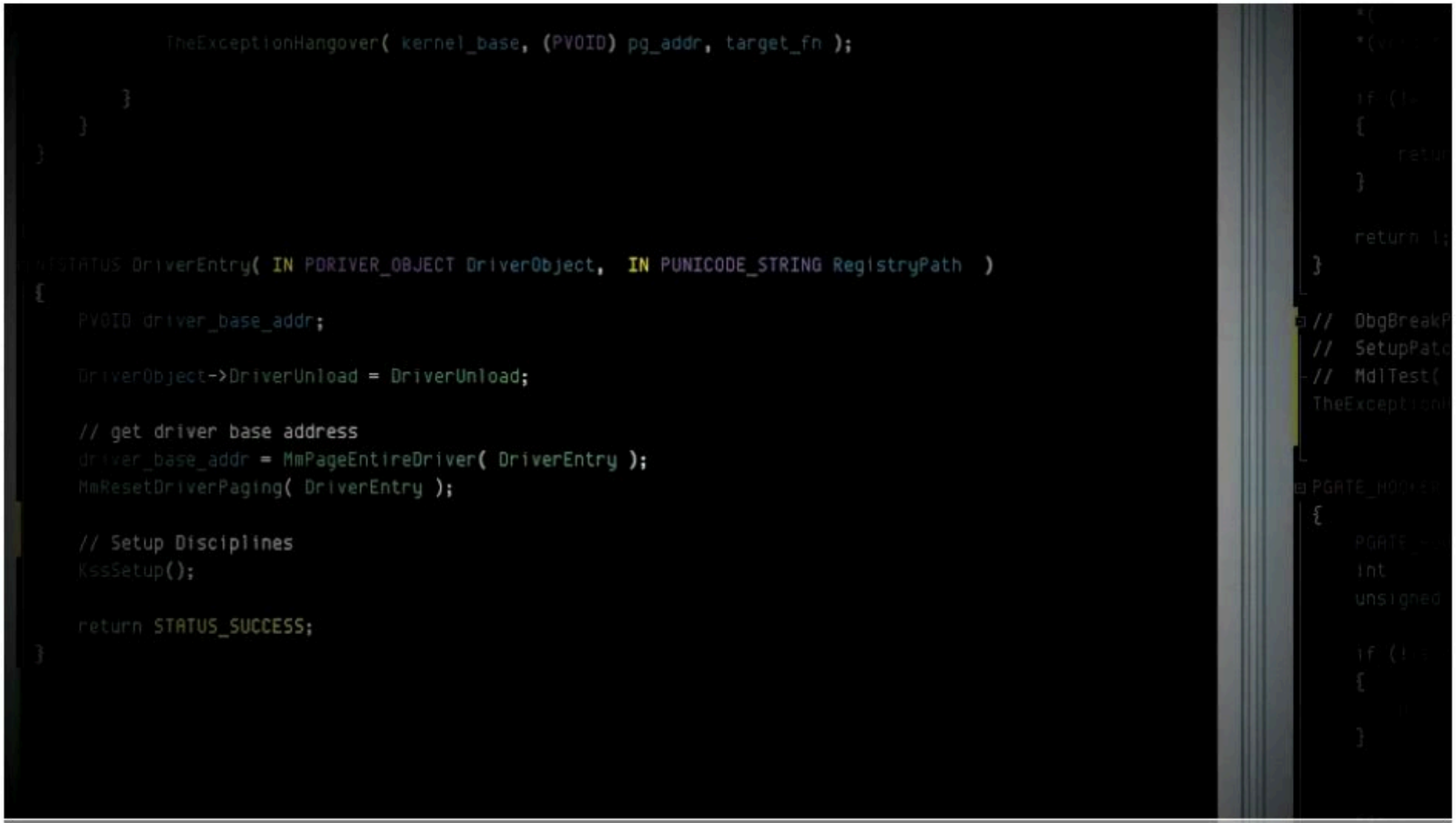
- Calling Convention for x64
- How to backtrace?
- Several general purpose regs
- The opposite happens: regs are stored in local stack



Exception Handler

- Unwind Info holds:
- Stacked saved parameters
- Internal stack changes at code flow
- Holds function Exception Handler for SEH
- Function information is inside PE's Directory





Using Exception - In Action

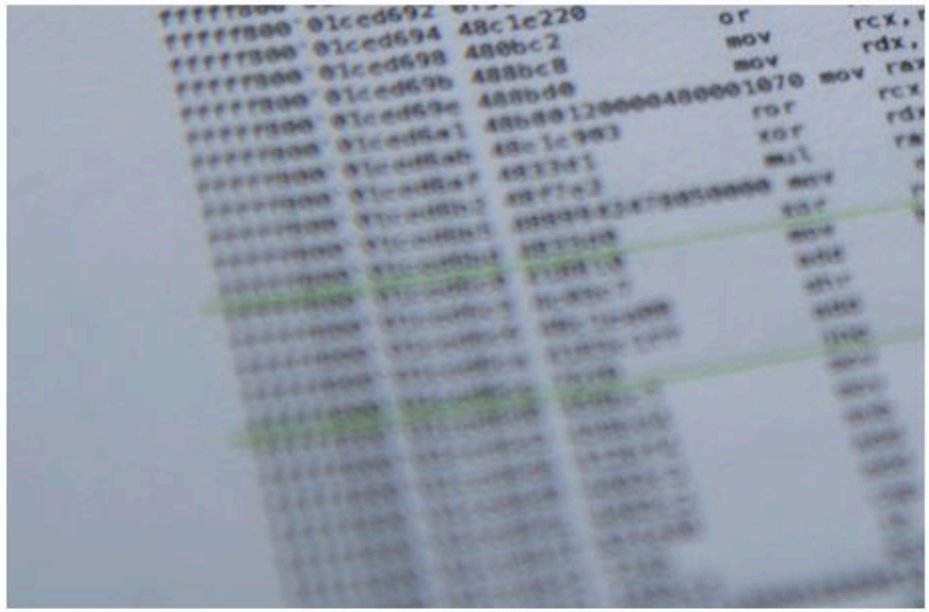
The PatchGuard

What is it?

- Protects the Operating System vital structures
- “Code our way, or die in our way too” - (failed to rhyme)
- Asserts that drivers use the right API
- Gives the kernel team flexibility to change internals w/o supporting the vendors
- Maybe security?

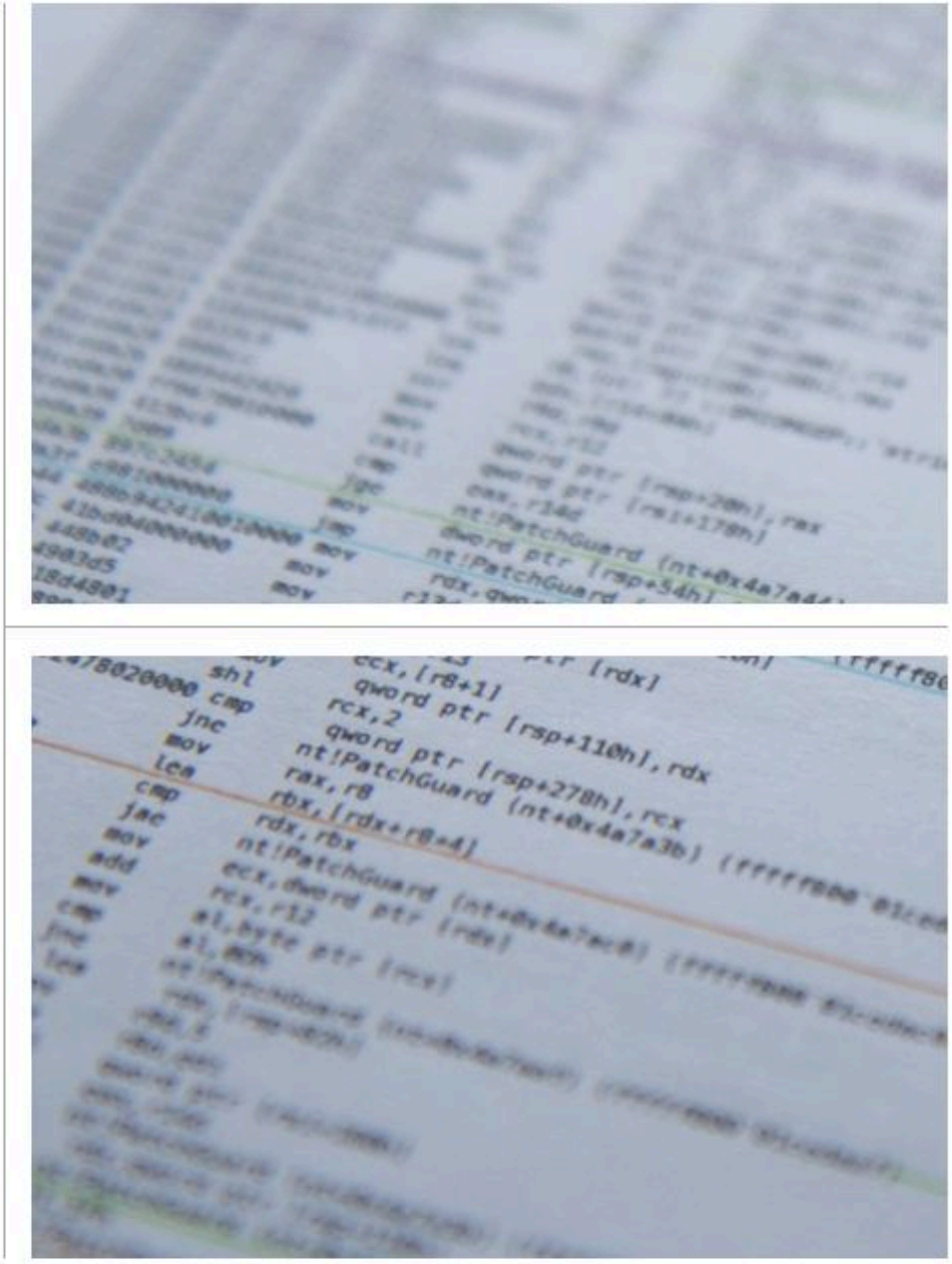
Obfuscation++

- PoolTag randomly chosen
- Random “Fat” allocation space
- Random split the Fat before and after real data
- Fat filled with garbage
- Uses Timer + DPC, using random valid DPC dispatchers, with invalid context data (which will throw an exception)



Obfuscation++

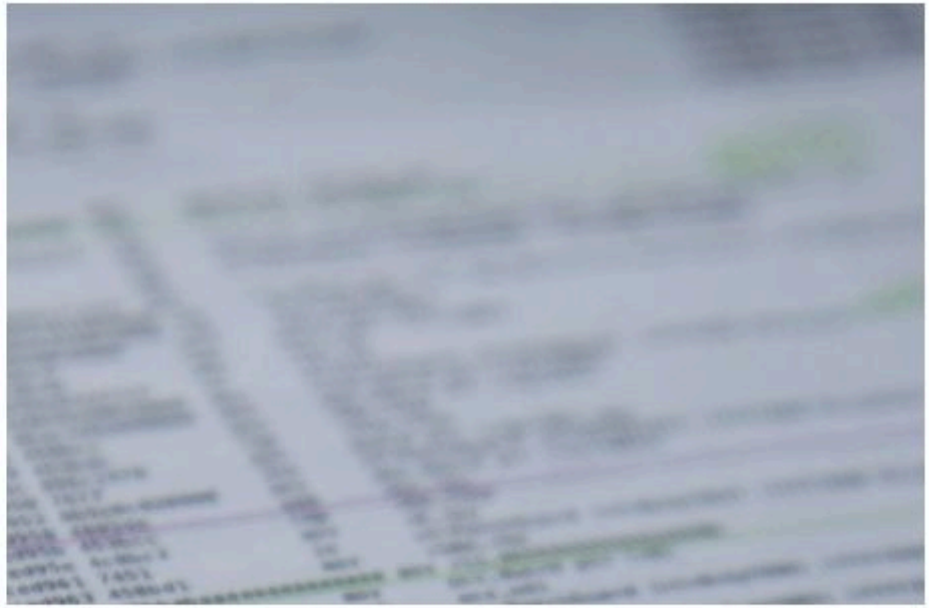
- 2 PgContexts are injected into the Kernel Memory
- One is close to the processor context
- 3/13 chances to have custom DPC dispatcher, to prevent public PatchGuard deactivators
- PgContext is encrypted
- Checks performed inside trap interruption



Obfuscation++

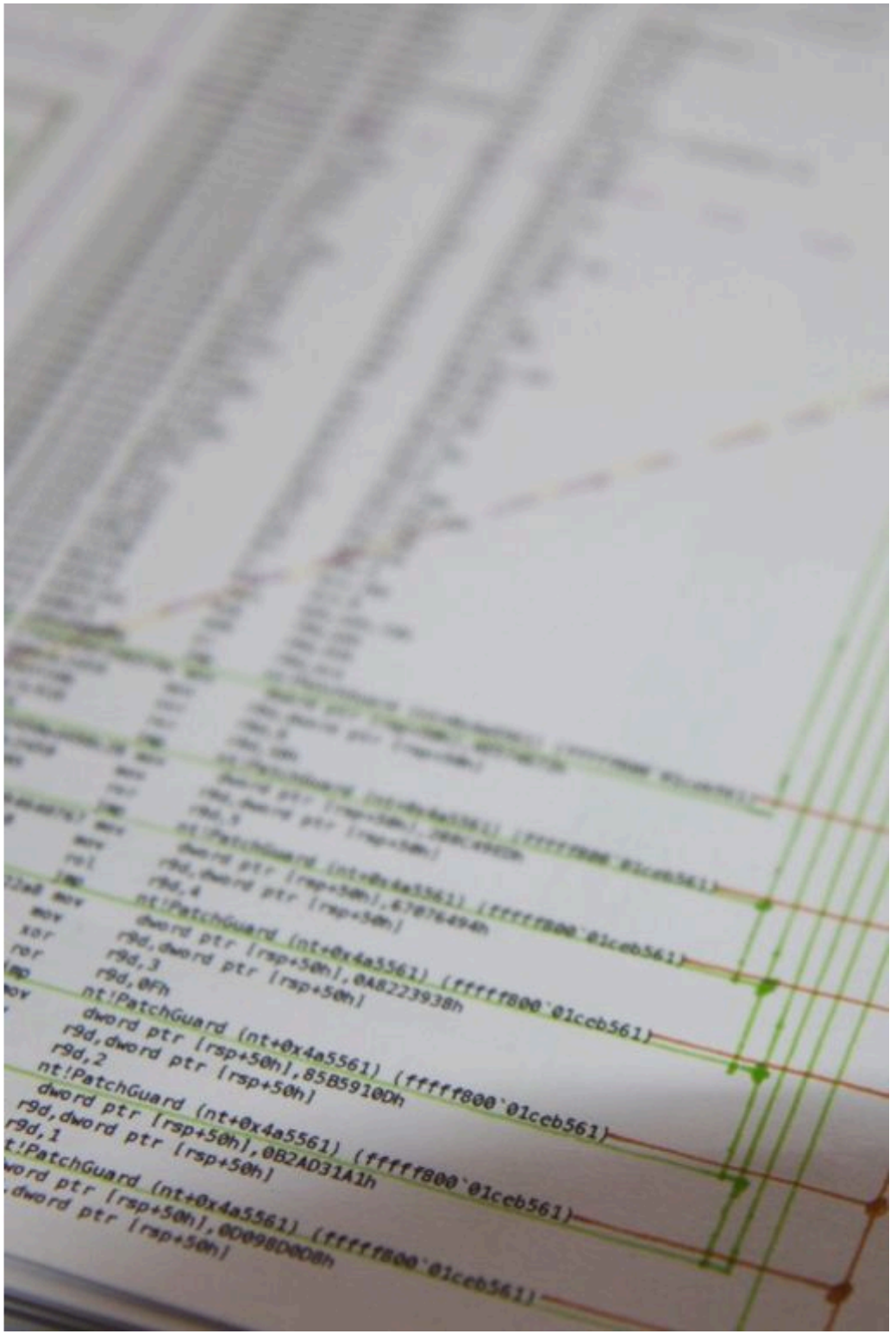
- Copy kernel vital functions like KeBugCheckEx, KeBugCheck2, KiBugCheckDebugBreak, etc...

- About 20 debugger_is_attached checks, leading to infinite loop with interrupts disabled



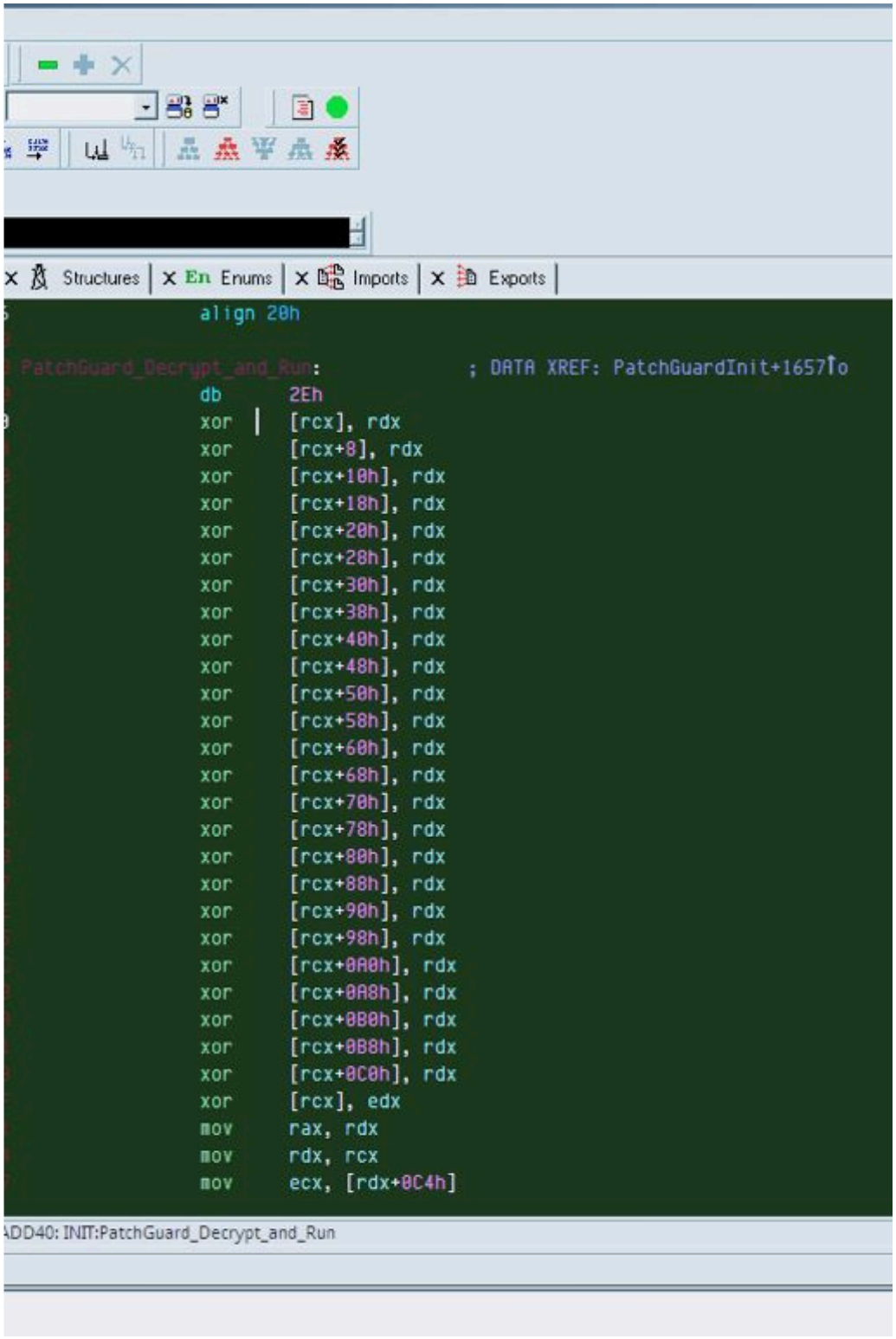
PatchGuard - review

- uninformed.org has published several papers on how to deactivate
- Some proposed paths to block is already patched by Microsoft in latest builds.
- Windows 7 follows the same PG code from Vista, even encryption constants are the same
- X86 can run PatchGuard



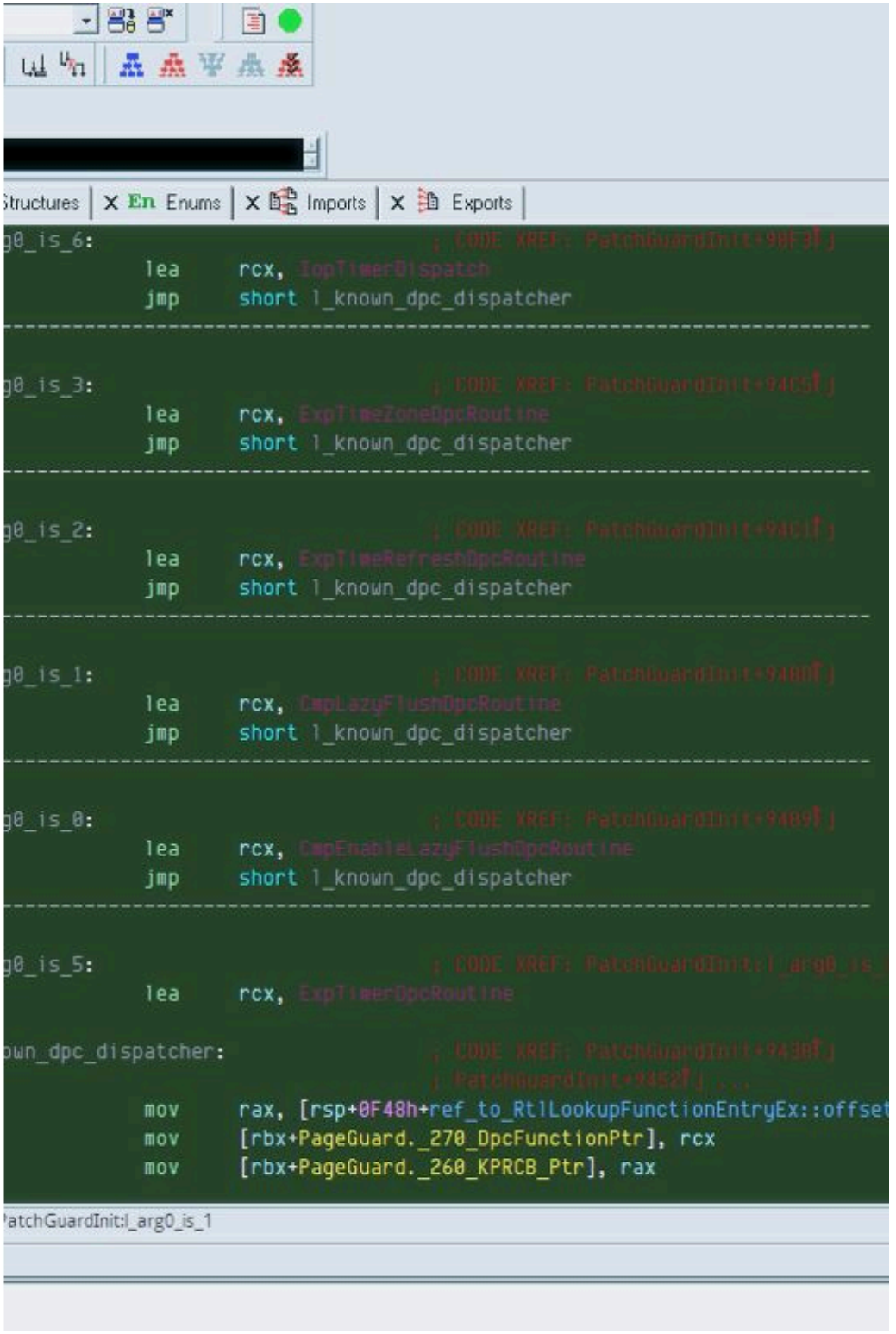
Deactivating it

- All encryption are based on RDTSC instruction, which can be deactivated by CR4.2
- DPC for timers are encrypted, but decryption is trivial
- Seek and destroy timers



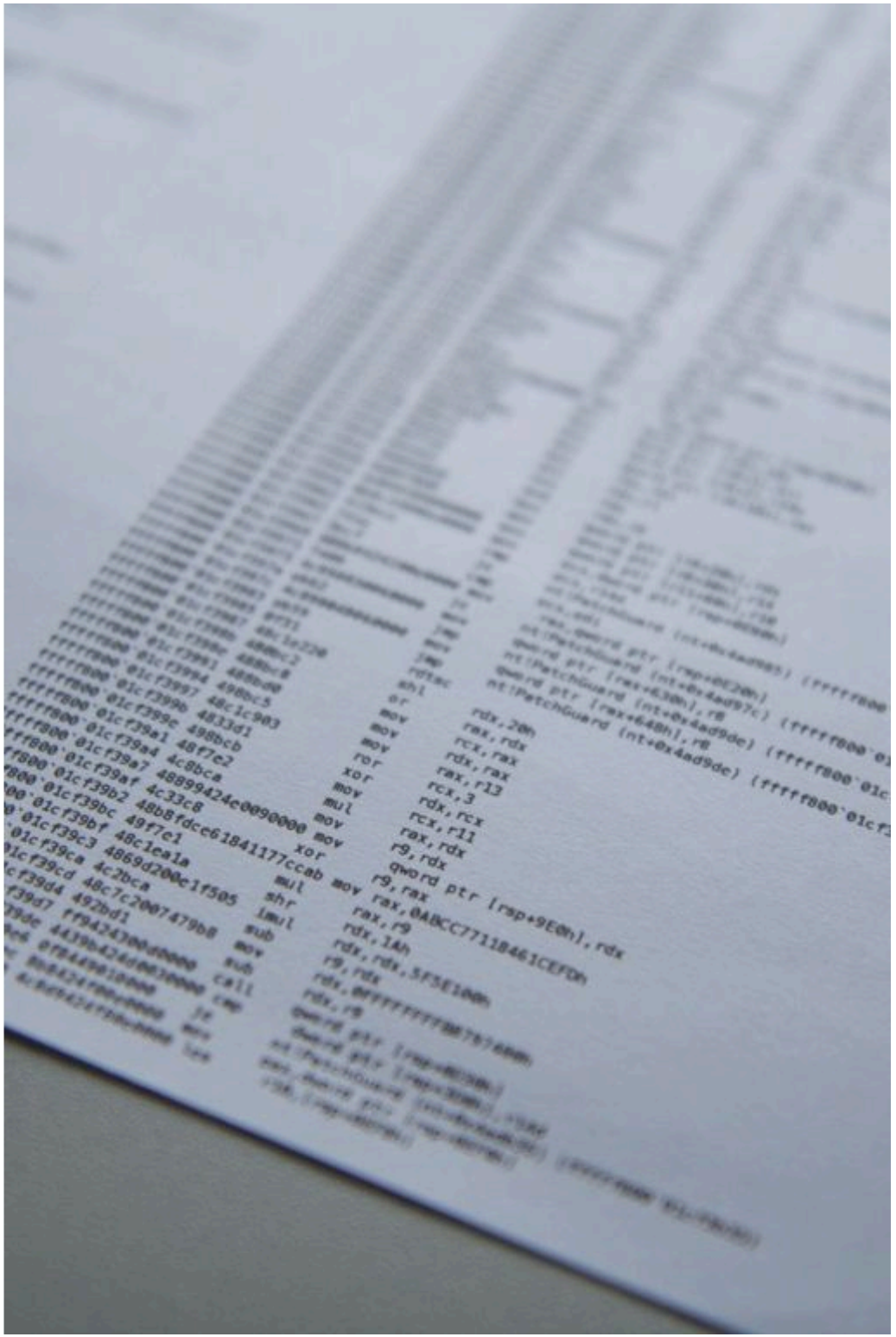
Deactivating it

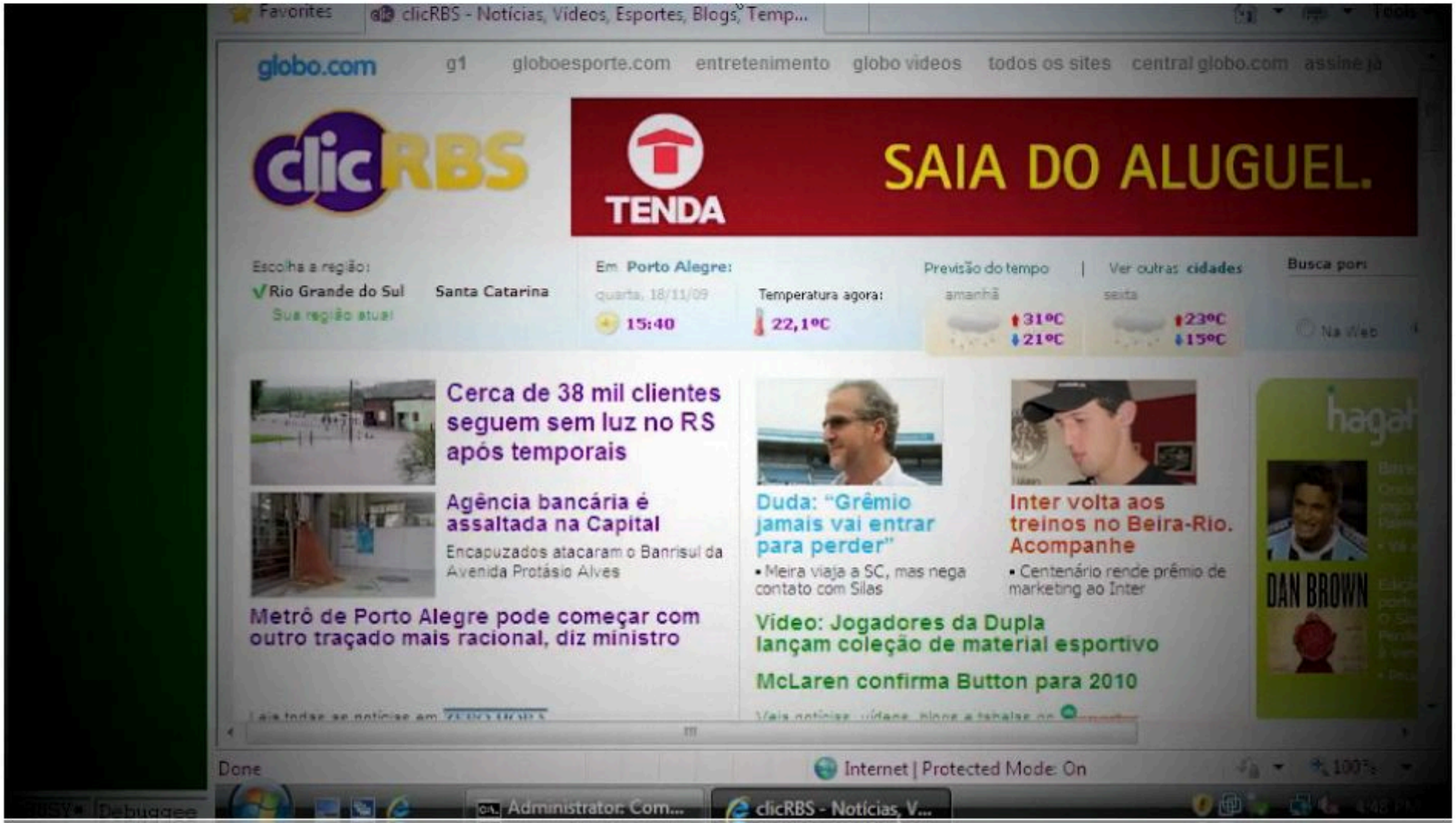
- You can change the IDT, but get ready for behind the scenes dirty job for exception handling
- Use the rewind info to construct the call backtrace
- Find if this is a Dpc in the list of PatchGuard borrowed Dispatcher routines (9/13 chances).
- If custom Dpc, figure the structures, go, go & go!



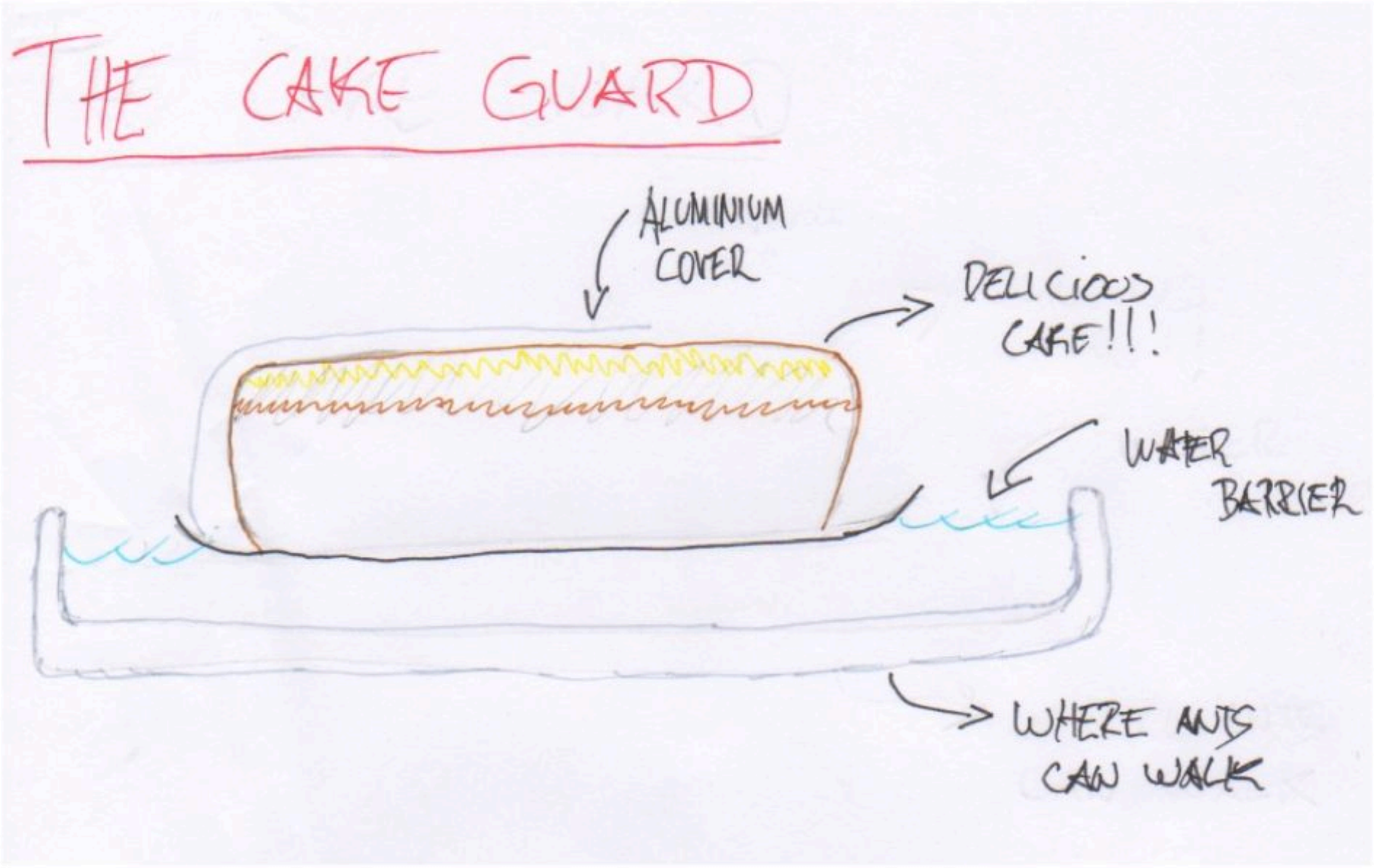
Deactivating it

- Get ready to the cat and mouse race! This obfuscation and Patchguard techniques WILL change, and eventually a KeBugCheck issued





Let's get it all

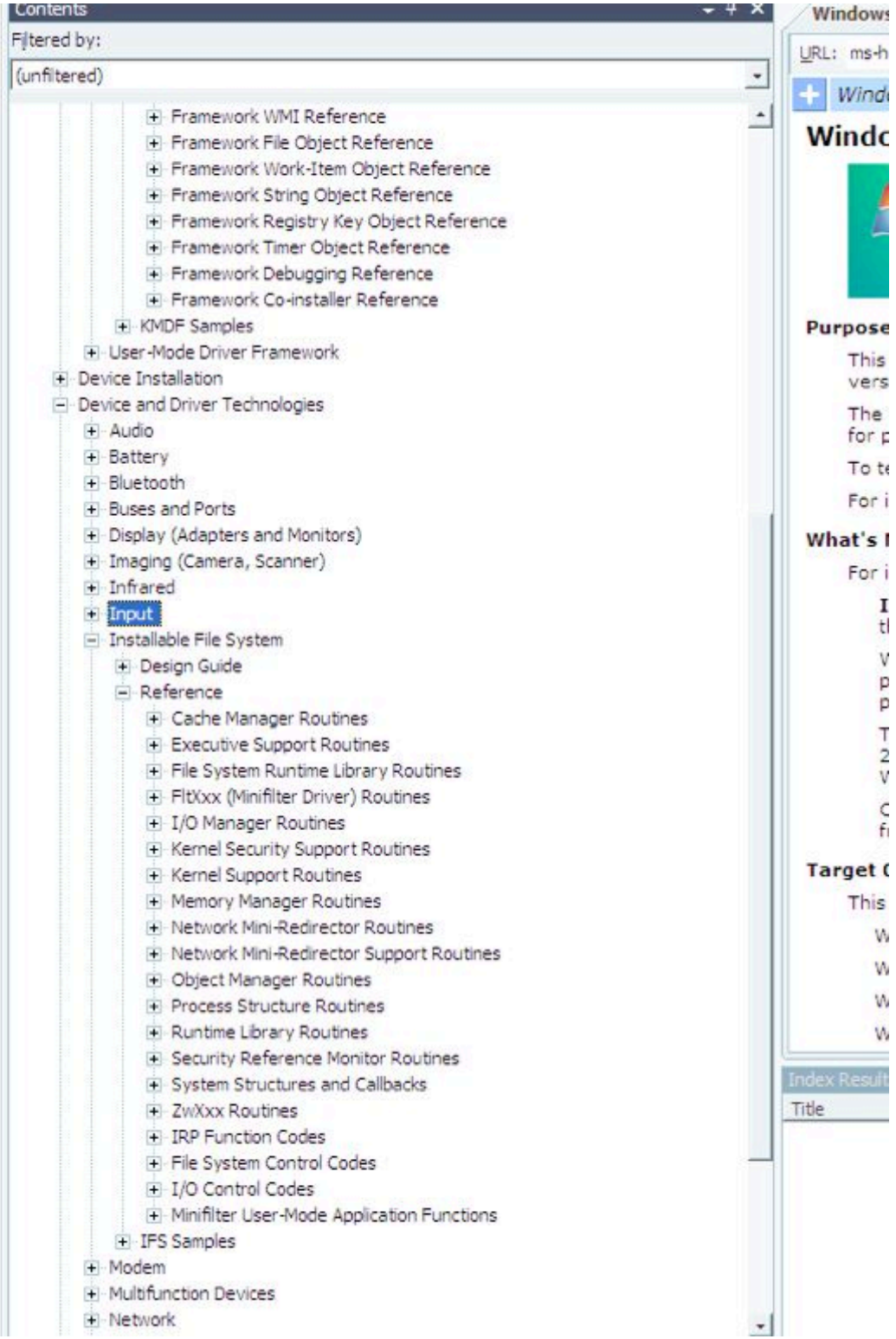


The Two-Stages

The Two-Stages

Natural root kits

- You can hide files and folders
- Hide registry information
- Change process information
- Open network connections
- A new TCP/IP stack can be built, and raw packets sent through ndis.sys
- A key logger still can be coded



Dialing The Patcher

- Is this kernel PG-removable?
- If not, tcp connect to TP server.
- Send me your kernel details
- If no patching code available, just leave
- Will operate in stage-1
- Once patched, go to stage-2

```
    __out PULONG BufferOutputLength
    )
{
    PIRP irp;
    PIO_STACK_LOCATION irp_stack;
    KEVENT event;
    IO_STATUS_BLOCK iostatus;
    NTSTATUS status;

    PAGED_CODE();


    irp = ObnlCreateSynchronousIrp( DeviceObject->StackSize, &iostatus
    if (irp == NULL)
    {
        status = STATUS_INSUFFICIENT_RESOURCES;
        goto on_exit;
    }

    //
    // associante info
    irp->AssociatedIrp.SystemBuffer = NULL;
    irp->UserBuffer = Buffer;
    irp_stack = IoGetNextIrpStackLocation( irp );


    irp_stack->MajorFunction = IRP_MJ_DIRECTORY_CONTROL;
    irp_stack->MinorFunction = IRP_MN_QUERY_DIRECTORY;
    irp_stack->FileObject = FileObject;
    if (SingleEntry)
    {
        irp_stack->Flags |= SL_RETURN_SINGLE_ENTRY;
    }

    irp_stack->Parameters.QueryDirectory.FileIndex = FileIndex;
    irp_stack->Parameters.QueryDirectory.FileInformationClass = FileInf
    irp_stack->Parameters.QueryDirectory.FileName = FileName;
    irp_stack->Parameters.QueryDirectory.Length = Length;

    // make synchronous event
    KeInitializeEvent( &event, NotificationEvent, FALSE );
    IoSetCompletionRoutine( irp,
```

➡“The BIOS is eternally the weakest spot.”

➡“Can we load your ntoskrnl.exe?”

➡“I noticed in your CR4 that VMX is not running.”

➡“The user mode is yet the blue ocean for the Ring0.”

➡“Do you want the real system safety? - get out of the virtual.”

➡“2-Stage approach for the sustainable ownage!”

Gustavo Scotti,
Immunity, Inc.