WRMSR MSR_DEBUGCTLA for Fun and Profit

Ryan MacArthur
라이언 맥아더
rpm@jhu.edu
Ekoparty 2011

!hardcore

- lex parsimoniae
- 1 year tinkering in the kernel

Model Specific Registers

- Model specific registers are used to provide access to features that are generally tied to implementation dependent aspects of a particular processor
- It's all about the Pentium
  - (amd k6 too)

Model Specific Registers

- Performance Monitoring
  - PEBS
- Debugging
  - program execution tracing
- Tweak Functionality
  - Hardware Prefetcher Disable

Debugging Artifacts

- VxWorks WDB Agent
- Peek/poke jtag
- Skype debug logs
- … so many more

Access to MSR's

- WRMSR
- RDMSR
- Real mode
  - CPL == 0
- Virtual 8086
  - #GP(0)
- VMM
  - Guest access?

Enumeration

- Check for support with CPUID
- CPUID.1:EDX[21] == 1
- IA32_MISC_ENABLE

Access in Windows (cpl 3)

- NtSystemDebugControl (xp)
  - SeDebugPrivilege
- KdSystemDebugControl (7)
  - Boot DEBUG
  - SeDebugPrivilege
  - \\.\Kldbgdrv
    - Send IOCTLs

Previous Abuse

- Drivers provide unrestricted interface to wrmsr/rdmsr
- Hijacking IA32_SYSENTER_EIP
- Using IA32_SYSENTER_EIP to leak base of ntoskrnl

Today

- Detect code running in emulated/virtualized environment
  - Yet another Redpill
- Fine grained tracing without binary instrumentation
  - No diasm though, kids

Your sandbox Sucks

- Detours?
  - Oh, we use push; ret;
    - lol

Yet another Redpill

```
u_int64_t matrix;
u_int msr = 0x1db9;

 __asm __volatile("rdmsr" : "=A" (matrix) : "c" (msr));
```

YARP

- Vmware Workstation
- Vmware ESXi
- VirtualBox
- HyperV
- KVM
- Bochs
- Qemu

Intel VMM

- MSR Access Control

Intel to the rescue

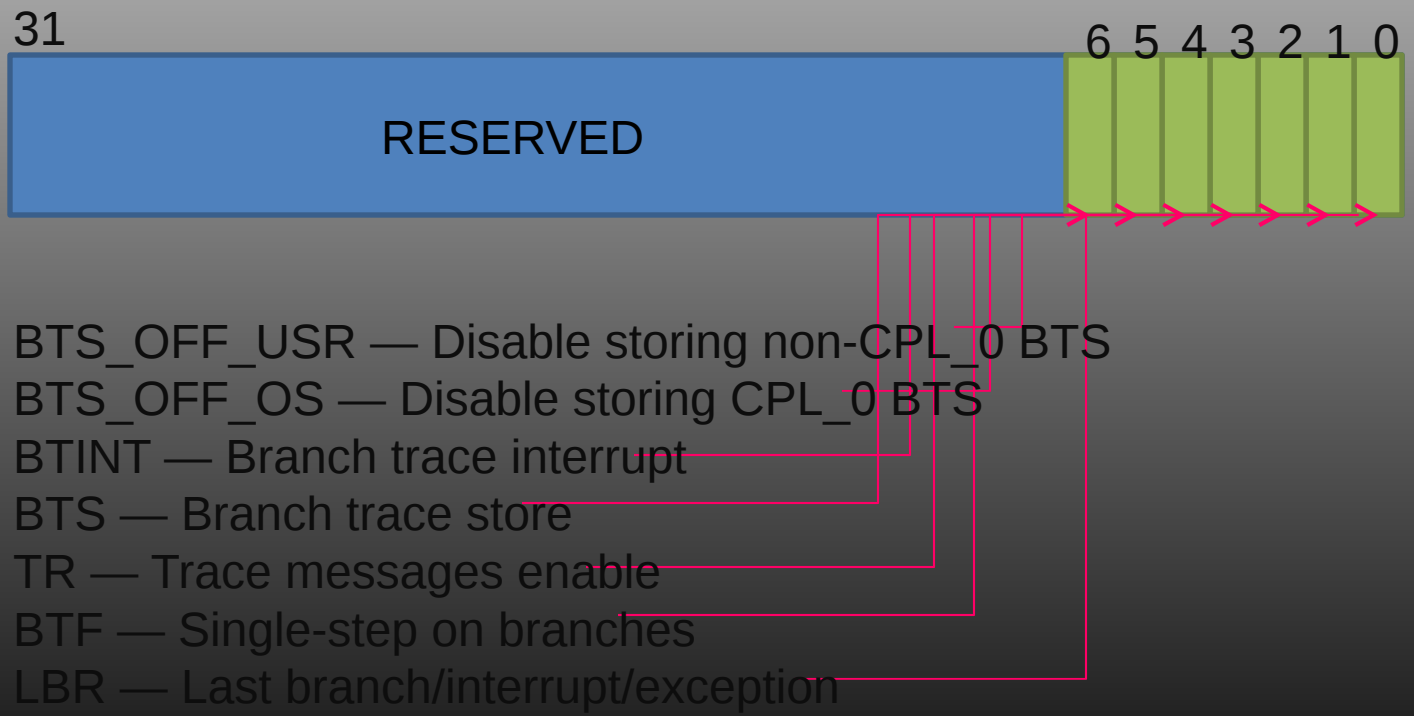
- Last Branch Recording (LBR)
- Lots of fun Debugging/Performance stuff to play with (Volume 3B)

LBR

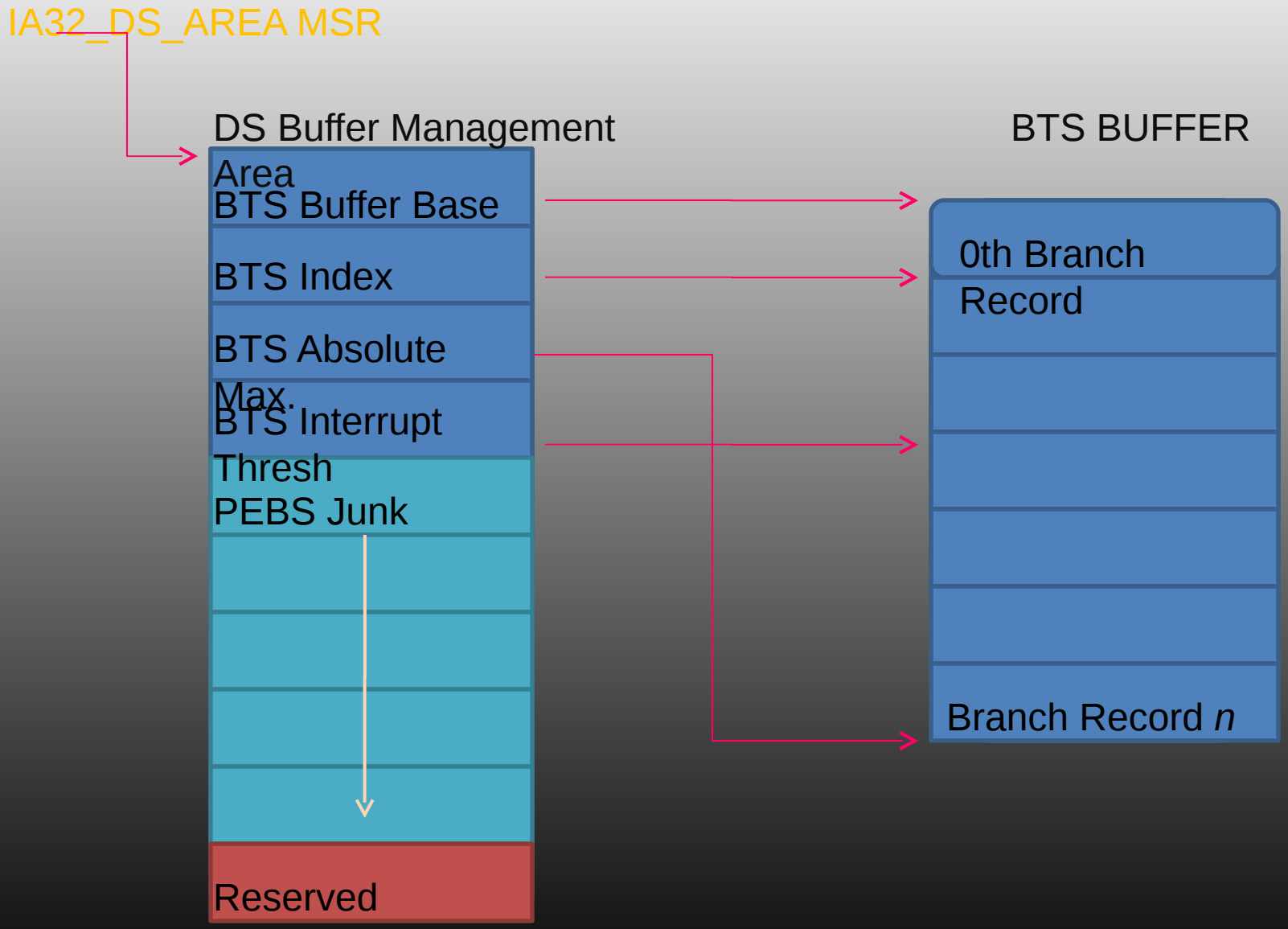
- LBR Stack
- Trap On Branch
- Branch Trace Store
  - Cyclical buffer or interrupt
    - Local APIC
  - Trace only userland or trace OS

Last branch recording facilities (netburst)

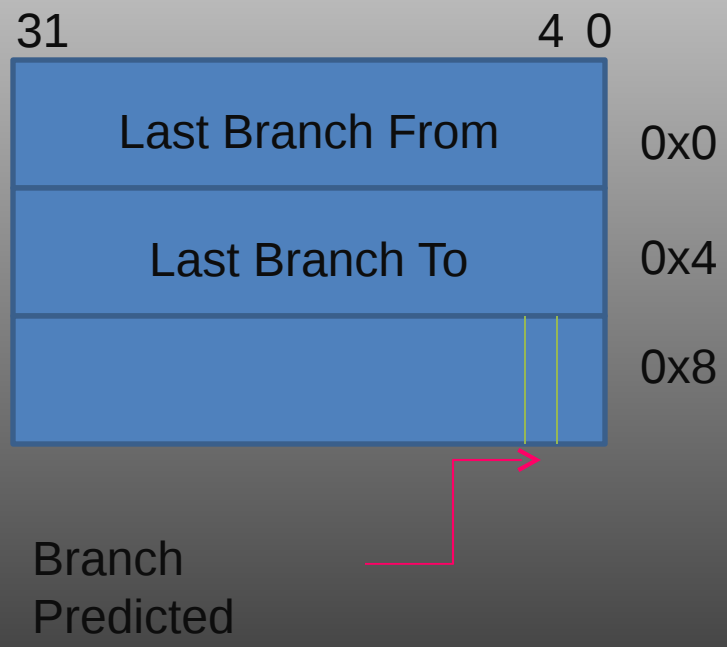
- MSR_DEBUGCTLA



Debug Store


IA32_DS_AREA MSR
DS Buffer Management Area
BTS Buffer Base
BTS Index
BTS Absolute Max.
BTS Interrupt Thresh
PEBS Junk
Reserved
BTS BUFFER
0th Branch Record
Branch Record n

Branch Record

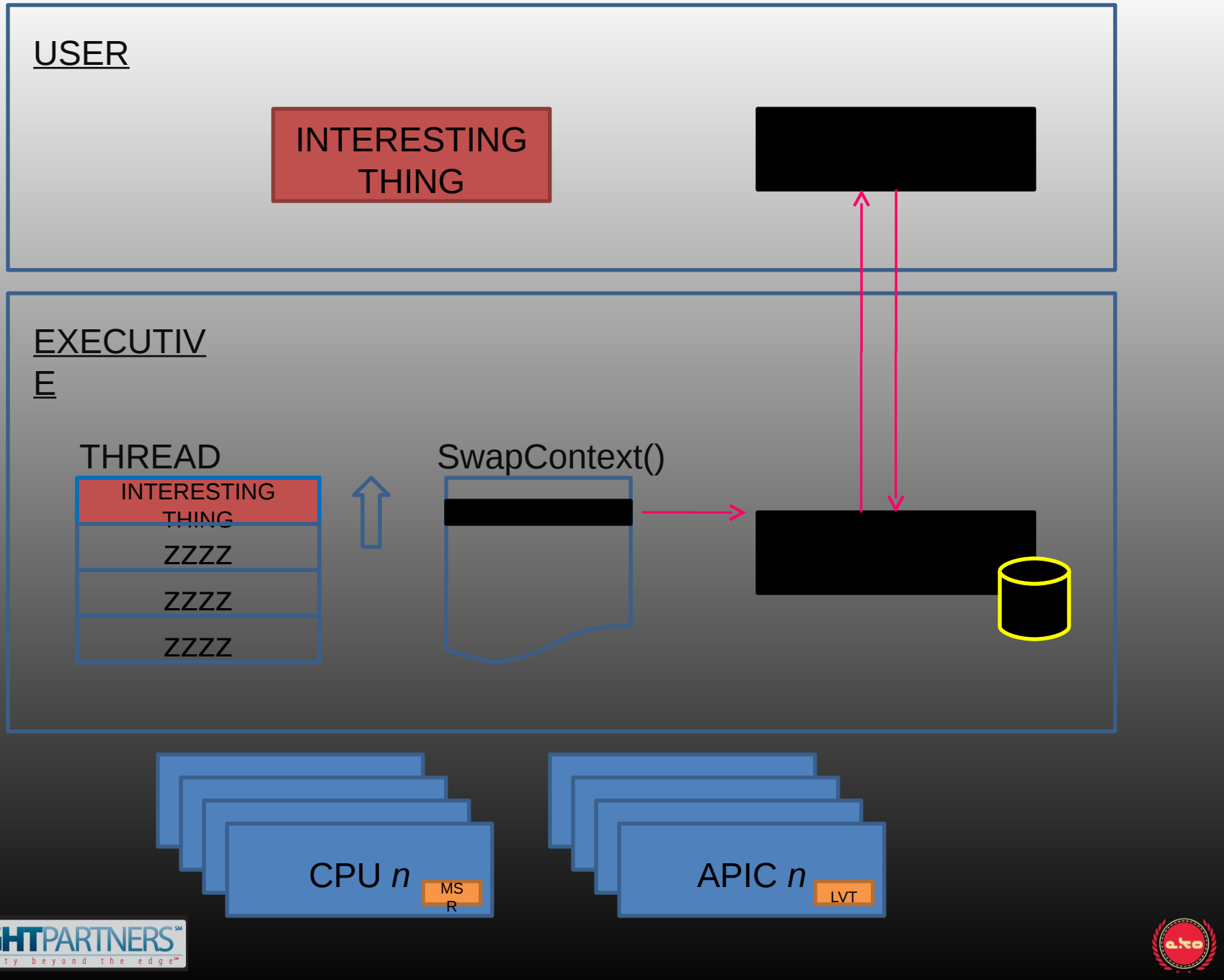


My Windows Implementation

- Mimics malware
  - Userland exe drops kernel driver, patches it with address of swapcontext() and installs it
  - Driver hooks swapcontext() and the fun begins
    - Walk kernel objects and turn on tracing for threads of interest
      - KTHREAD -> ETHREAD
- MiSeR

Userland Component

- Registers processes of interest
- Consumes trace data
  - Event Driven


USER
INTERESTING THING
EXECUTIV E
THREAD
INTERESTING THING
ZZZZ
ZZZZ
ZZZZ
SwapContext()
CPU n
MS R
APIC n
LVT

Code up on Github

- Real shitty

Performance

- I should have used PEBS to measure, but ran out of time
- Used jump heavy code as sample
- Timing resolution is millisecond :/
- Miser, Coseinc Trace Suite, Pin inscount2
  - Could not get saffron from author :(

perf_test.c

```
#include <windows.h>


void main(int argc, char **argv)
{
 unsigned int ticks,i,j;


 ticks = GetTickCount();
```

Perf. Results

- $x=$ 10 runs of perf_test, argv[1] = 1
  - Native: $\bar{x}=0ms$
  - Miser: $\bar{x}=0ms$
  - Coseinc: $\bar{x}=2{,}391ms$
  - Inscount1: $\bar{x}=38{,}406ms$
  - Process Stalker (BTF) $\bar{x}=43{,}672ms$
  - Single Step $\bar{x}=84{,}656ms$

Perf. Results

- $x=$ 10 runs of perf_test, argv[1] = 10,000
  - Native: $\bar{x}=15{,}724ms$
  - Miser: $\bar{x}=15{,}811ms$
  - Coseinc: $\bar{x}=17{,}359ms$
  - Inscount1: $\bar{x}=75{,}250ms$

Post Processing

- What to do with this data?
- Working on visualization techniques for quick ID by analysts(?)
- Malware classification
- Code coverage
- Pretty Pictures

Heatmaps

- Ours are low resolution

Demo

- P4 (netburst box)

Related

- ptrace BTS implementation
  - Done by Markus Metzger of Intel
- XTRec: Secure Real-Time Execution Trace Recording on Commodity Platforms
  - Amit Vasudevan
  - on AMD

Related

- Process Stalker
  - Trap On Branch (BTF)
- Saffron
  - ring0 dynamic instrumentation with pin
  - Driver source not released
  - Page faults
- Azure
  - Virtual machine introspection

Extensions

- Full fledged kernel debugger leveraging Branch Tracing/PEBS
- Huffman compression of in-kernel branch records
  - Fucking loops
- Using compressibility to classify traces
- IDA scripts to enrich IDB's (Julio Auto de Medeiros)

Extensions

- Measuring distance between basic blocks?
- Clobbering memory strategically with debug store