

4 An Easter Egg in PCI Express

by Jacob Torrey

Dear Pastor Laphroaig,


Please consider the following submission to your church newsletter. I hope you think it worthy of your holy parishioners and readers.

Our friends at Intel are always providing Easter eggs for us to enjoy, and having stumbled across a new one for x86, the most neighborly option was naturally to share with all interested parties. This PoC is a weird quirk in which a newer x86 feature-set breaks invariants/security guarantees from older version. Specifically, the newer PCI Express configuration space access mechanism breaks virtual memory. Virtual memory is orchestrated by the CR3 register (storing the *physical address* of the page tables) and the page tables themselves. An issue with kernel shell-code and live memory forensics is that unless the *virtual address* of the page tables is known, it is impossible to map them (or any other physical address for that matter) into virtual memory, resulting in a chicken-and-egg problem. Luckily, most operating systems keep the page tables at a known virtual address (0xC0000000 on many Windows systems), but this Easter egg allows access to the page tables on *any* OS.

In kernel space, CR3 can be read, providing the physical address of the OS page tables; however, due to Intel's virtual memory protections, there is no way to create a recursive virtual mapping to that physical address. All that is needed to do so, is a way to write an arbitrary 32-bits (which will become a PDE mapping in the page tables) to a known physical location.

This is the crux of the issue, and the security of virtual memory depends on it. Luckily, with the advent of PCI Express, there is now the "Enhanced Configuration Access Mechanism" (ECAM), which shadows PCI configuration space registers into physical memory at an address kept in the PCIEXPBAR register (DO:F0 offset: 0x60). This is typically enabled on all the systems the author has come across, but your mileage may vary. With this ECAM, changes made to the configuration space via the legacy port I/O mechanism (0xCF8/0xCFC) will be reflected in physical memory. Now all that is needed is a register in configuration space that is at least 32-bits wide and can be changed to an arbitrary value without impacting the system. Again, Intel is looking out for our church, and through their grace, they provide a "Scratchpad Data" register (DO:F0 offset: 0xDC) that has no semantic meaning, just a location for software to store data. Now we have the function ModifyPM() for physical memory. (This is for Windows 32-bit without PAE, running as driver code.)

```
/**
2   Sets up the PDE to map in the real PDT using the MMIO ranges of PCI
   Configuration space
4   @return The PCIEXPBAR for comparison
*/
6 ULONG ModifyPM()
{
8     ULONG MMIORange = 0;
       asm
10    {
       pushad
```

| | |
|--|--|
| We Recommend | |
|  | |
| CHAMBARD'S TEA | |
| To All Persons Suffering from | |
| CHRONIC CONSTIPATION, | |
| Caused Either by their Temperament or by their Sedentary Occupations. | |
| Without necessitating any change in the habits, or in the regime, and without causing any fatigue, CHAMBARD'S TEA rapidly restores the functions of the digestive tract, and maintains them in their normal condition. The trade-mark, "THE CENTAUR," is on each genuine box. 30 cents; post-paid, 35 cents. Ask for free samples. Ask your druggist for it. He will get it for you. | |
| LEGOLL'S PHARMACY, 286 7th Avenue, New York. | |
| And Leading Druggists. | |
|  | |
| VIN URANÉ PESQUI | |
| (Pesqui's Uranated Wine) | |
| FOR THE CURE OF DIABETES. | |
| It has been shown by medical statistics that there are in France every year 10,000 deaths, or more, due to Diabetes through a deficient treatment, whilst they could have been cured by taking the VIN URANÉ PESQUI. This scientific preparation allays at once the unquenchable thirst, decreases rapidly the sugar. It strengthens, restores health and vigor, and prevents diabetic complications, such as gangrene, anthrax, etc. Pamphlet free. | |
| LEGOLL'S PHARMACY, 286 7th Avenue, New York. | |
| New Scientific Discovery! | |
| NO MORE BALD HEADS. | |
| Rational Treatment of | |
| Baldness, Alopecia, Diseases of the Scalp, Beard, Eyebrows, and Eyelashes, Scurf, Scald, Psoriasis, Pityriasis, Dandruff, Itching, Etc., | |
| By the Use of the | |
| DEQUÉANT LOTION | |
| Ask for Free Pamphlet. | |
| L. DEQUÉANT, Chemist, | |
| 38 Rue Clignancourt, - - PARIS. | |
| -DEPOT:- | |
| LEGOLL'S PHARMACY, | |
| 286 7th Ave., - - New York. | |
| Is Fatal to Health and Beauty. | |
| Numerous experiments in the hospitals of Paris and Europe in the treatment of obesity with | |
| Flourens' Thyroidine Pills and Tablets have been successful in all cases. They are perfectly harmless, and never fail. By mail, \$1.00. | |
| LEGOLL'S PHARMACY, | |
| 286 7th Ave., - - New York. | |
| ULCERATED LEGS | |
| Resulting from Varicose Veins, Ecremas, and other diseases of the skin, are surely and rapidly cured by the use of the | |
| Eau Précieuse, | |
| DEPENSIER, Chemist, ROUEN (France). | |
| LEGOLL'S PHARMACY, | |
| 286 7th Ave., - - New York. | |

```

12 // Utilize the scratch pad register as our mini-PDE
13 mov ebx, cr3
14 and ebx, 0xFFC00000 // This is going to hold our new PDE (The bits in
15 // CR3 with the least significant stuff removed)
16 or ebx, 0x83 // P | RW | PS
17
18 mov dx, 0x0cf8
19 mov eax, 0x800000DC // Offset 0x37 (0xDC / 4)
20 out dx, eax
21
22 mov dx, 0x0CFC
23 mov eax, ebx
24 out dx, eax // Write our PDE
25
26 // Determine where in physical memory we can find the PDE
27 mov dx, 0x0cf8
28 mov eax, 0x80000060
29 out dx, eax
30
31 mov dx, 0x0CFC
32 in eax, dx
33 mov MMIORange, eax // Save our value and BAM!
34
35 popad
36 }
37
38 if (VDEBUG)
39     DbgPrint("MMIO Base Address: %x", MMIORange);
40
41 return MMIORange;
42 }

```

Once the scratchpad register is primed and ready, and the physical address of the ECAM is known, the next step is to treat the register as a PDE mapping in the OS page tables to add a recursive mapping at a known location.

```

1 /**
2  Sets up a recursive mapping to the OS page directory
3  I commented it very thoroughly because it's quite complex.
4
5  Basically it:
6  -> Saves the current (real) CR3 value
7  -> Creates a new PDE to map in the (real) PDT
8  -> Creates a virtual address using the (fake) PDE we inserted in ModifyPM
9  -> Switches to the (fake) CR3 and utilizes the constructed virtual
10     address to insert the new recursive mapping into the (real) PDT
11  -> Switches the CR3 back and continues on smugly
12 */
13 ULONG recurMap()
14 {
15     ULONG MMIORange = 0;
16     ULONG PDEBase = 0;
17     ULONG PDEoffset = 0;
18
19     // Sets up the (fake) PDE and
20     MMIORange = ModifyPM();
21     MMIORange &= 0xF0000000;
22
23     if (VDEBUG)
24         DbgPrint("Mapping PDT to itself");
25
26     __asm {

```

```

27     cli
29
31     // Save the current CR3, seems like overkill, but it makes sense
33     mov ebx, cr3 // A copy to use to construct our virtual address
35     mov ecx, cr3 // Save a copy so we don't mess up things up too much
37
39     mov edx, MMIORange // Our new CR3 val
41
43     // Setup our virtual address
45     and ebx, 0x003FFFFFF // Gets us our offset into stuff
47     or ebx, 0x0DC00000 // Reference the PDE offset of (0x37 << 22)
49     // EBX should now have our virtual address :)
51
53     // Tests to see if the PDE is free for use
55     test_pde:
57
59     add ebx, 0x4 // Offset to unused PDE
61
63     // Keep the offset var up to date (but uint32 aligned, not uint8)
65     mov eax, PDEoffset
67     add eax, 0x1
69     mov PDEoffset, eax
71
73     //***** BEGIN CRITICAL SECTION
75     mov cr3, edx // Inject our new CR3
77
79     mov eax, [ebx] // Add our mirthful PDE entry which should map in the PD
81     invlpg [ebx] // Invalidates the virtual address we used just in
83     // case it could cause later problems.
85
87     mov cr3, ecx // Restore everything nicely
89     //***** END CRITICAL SECTION
91     cmp eax, 0 // Can we use this entry?
93     je inject_pde // Try the next one
95     jmp test_pde // Found an empty one, w00t!
97
99     // Injects our recursive PDE into the PDT
101    inject_pde:
103    // Setup our recursive PDE (again)
105    mov eax, cr3 // A copy to modify for our new recursive PDE
107    and eax, 0xFFC00000 // Only the most significant bits stay for 4M pages
109    or eax, 0x93 // P | RW | PS | PCD
111    // EAX now holds the same PDE to put into the 'real' PDT
113    //***** BEGIN CRITICAL SECTION
115    mov cr3, edx // Inject our new CR3
117
119    mov [ebx], eax // Add our mirthful PDE entry which should map in the PD
121    invlpg [ebx] // Invalidates the virtual address we used just in
123    // case it could cause later problems
125
127    mov cr3, ecx // Restore everything nicely
129    //***** END CRITICAL SECTION
131
133    // Determine the virtual address of the base of the PDT
135    // (remembering the differences in alignment)
137    mov eax, cr3 // A copy to modify for our new recursive PDE
139    and eax, 0x003FFFFFF // Only the most significant bits stay for 4M pages
141    mov ebx, PDEoffset
143    shl ebx, 22 // Offset into the PDT
145    or eax, ebx
147    mov PDEoffset, eax

```

```

93     popad
95     }
97     if (VDEBUG)
99         DbgPrint("Mapping complete should be mapped in at 0x%x!", PDEoffset);
101 }

```

The above, on a 32-bit non-PAE system, will return the virtual address that maps in the page directory and allows you to map in arbitrary physical memory as a known location. It should be noted that kernel privileges are needed (to access CR3) and to operate on a kernel page marked as Global so as to persist through the CR3 changes. The author hopes you enjoyed this weird machine and remember to treat your input data as formally as code, for only you can prevent vulnerabilities!

Sincerely,
@JacobTorrey

New Produced and widely used in England and U.S.A. COMPLETE BUSINESS PACKAGE

**INCLUDES EVERYTHING FROM INVENTORY TO SALES SUMMARY
PROMPTS USER, VALIDATES EACH ENTRY, MENU DRIVEN**

Approximately 60-100 entries/Inputs require only 2-4 hours weekly and your entire business is under control.

PROGRAMS ARE INTEGRATED-

01 = ENTER NAMES/ADDRESS, ETC
02 = ENTER/PRINT INVOICES
03 = ENTER PURCHASES
04 = ENTER A/C RECEIVABLES
05 = ENTER A/C PAYABLES
06 = ENTER/UPDATE INVENTORY
07 = ENTER/UPDATE ORDERS
08 = ENTER/UPDATE BANKS
09 = EXAMINE/MONITOR SALES LEDGER
10 = EXAMINE/MONITOR PURCHASE LEDGER
11 = EXAMINE/MONITOR (INCOMPLETE RECORDS)
12 = EXAMINE PRODUCT SALES

SELECT FUNCTION BY NUMBER-

13 = PRINT CUSTOMER STATEMENTS
14 = PRINT SUPPLIER STATEMENTS
15 = PRINT AGENT STATEMENTS
16 = PRINT TAX STATEMENTS
17 = PRINT WEEK/MONTH SALES
18 = PRINT WEEK/MONTH PURCHASES
19 = PRINT YEAR AUDIT
20 = PRINT PROFIT/LOSS ACCOUNT
21 = UPDATE END MONTH FILES MAINTENANCE
22 = PRINT CASH FLOW FORECAST
23 = ENTER/UPDATE PAYROLL (NOT YET AVAILABLE)
24 = RETURN TO BASIC

WHICH ONE? (ENTER 1-24)

**01 SUB. MENU EXAMPLE: 01 = EXAMINE: 02 = INSERT: 03 = AMEND: 04 = DELETE
05 = PRINT (1,2,3): 06 = NUMERIC COMBINATIONS: 07 = SORT
VERY FLEXIBLE. ADD YOUR OWN FUNCTIONS. EASY TO INTEGRATE.**

All programs in BASIC for CP/M. PET. 8800

G. W. COMPUTERS LTD, the producers of this beautiful package in U.K.

**WE EXPORT TO ALL COUNTRIES:
BARCLAYCARD ACCEPTED
CBM APPROVED**

**CALLERS BY APPOINTMENT ONLY
89 Bedford Court Mansions
Bedford Avenue
London WC1, U.K.**

**CONTACT TONY WINTER 01-636-8210
BARCLAYCARD ACCEPTED
CBM APPROVED**

CP/M Ver. 9.00 is one 16 K core program
using random access releasing both drives for
data storage, and 250 word vocabulary is
translatable in any foreign language.

CP/M Ver. 9.00 is one 16 K core program
using random access releasing both drives for
data storage, and 250 word vocabulary is
translatable in any foreign language.

PRICES: Programs 1-23 EXC (19,20,22,23) £475

£575 Stock Integrated Option + £100 Bank Integrated Option + £100