black hat ASIA 2024

Unveiling the Cracks in Virtualization, Mastering the Host System

— VMware Workstation Escape

Speaker: VictorV

About Me : VictorV (@vv474172261)

赛博昆仑 CYBER KUNLUN

VMware Workstation Escape TianfuCup 2018/2021/2023

Top 3 of MSRC 2023 Q3/Q4 Leaderboard

Zer0Con 2022
HITB 2020

Bugs in SQLServer, RDP, QEMU, DNS, DHCP, Samba, ESXi...

Hyper-V Escape CVE-2019-0887 In 2021

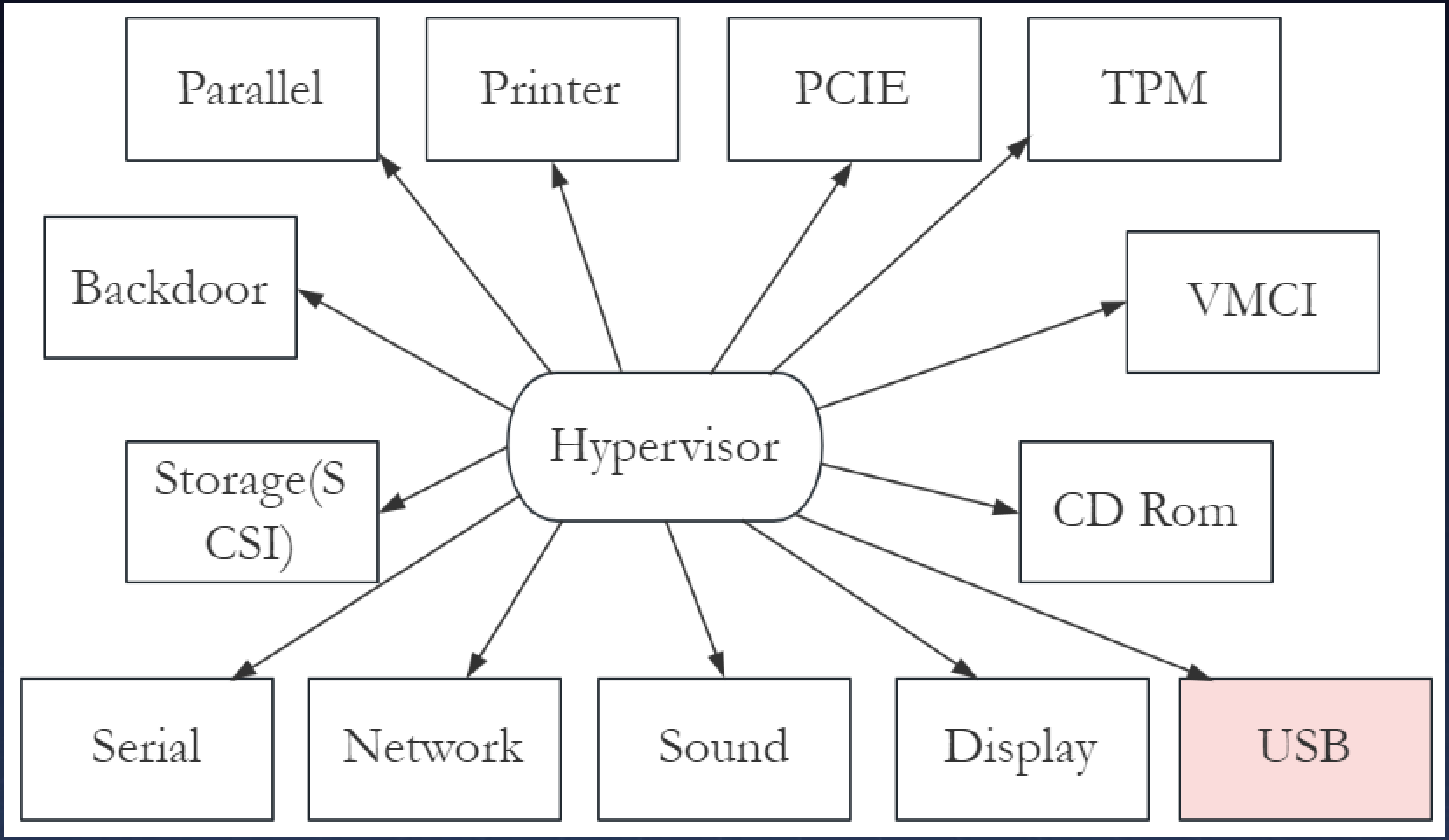
目录 CONTENTS

black hat ASIA 2024

PART ONE

Virtualization Basic Info

Virtualization Basic Info
VMware Worksation Architecture



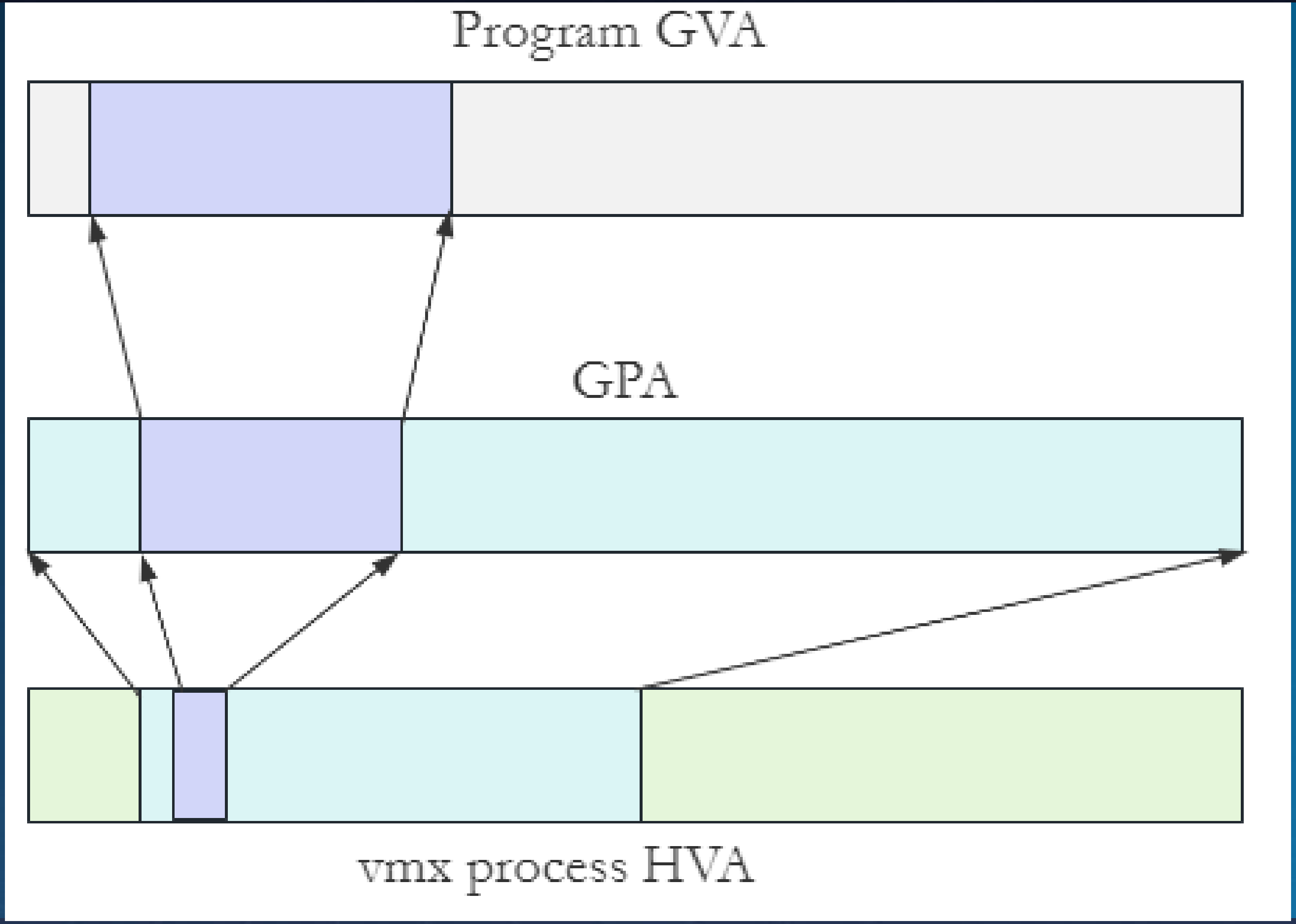
Virtualization Basic Info

Virtual Process Address and Guest Physical Address

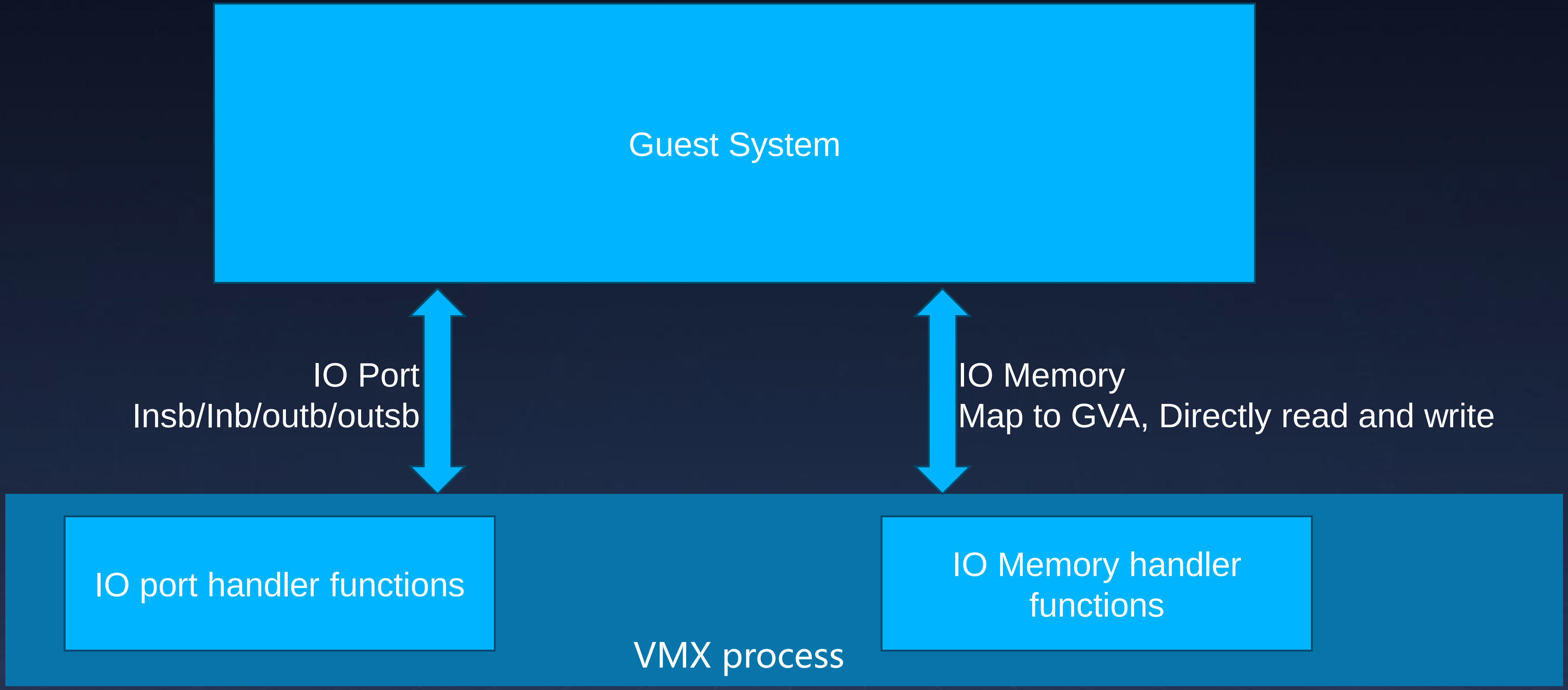
Guest Virtual Address(GVA)
Guest Physical Address(GPA)
Host process Virtual Address(HVA)

In Guest, use GVA access its physical memory

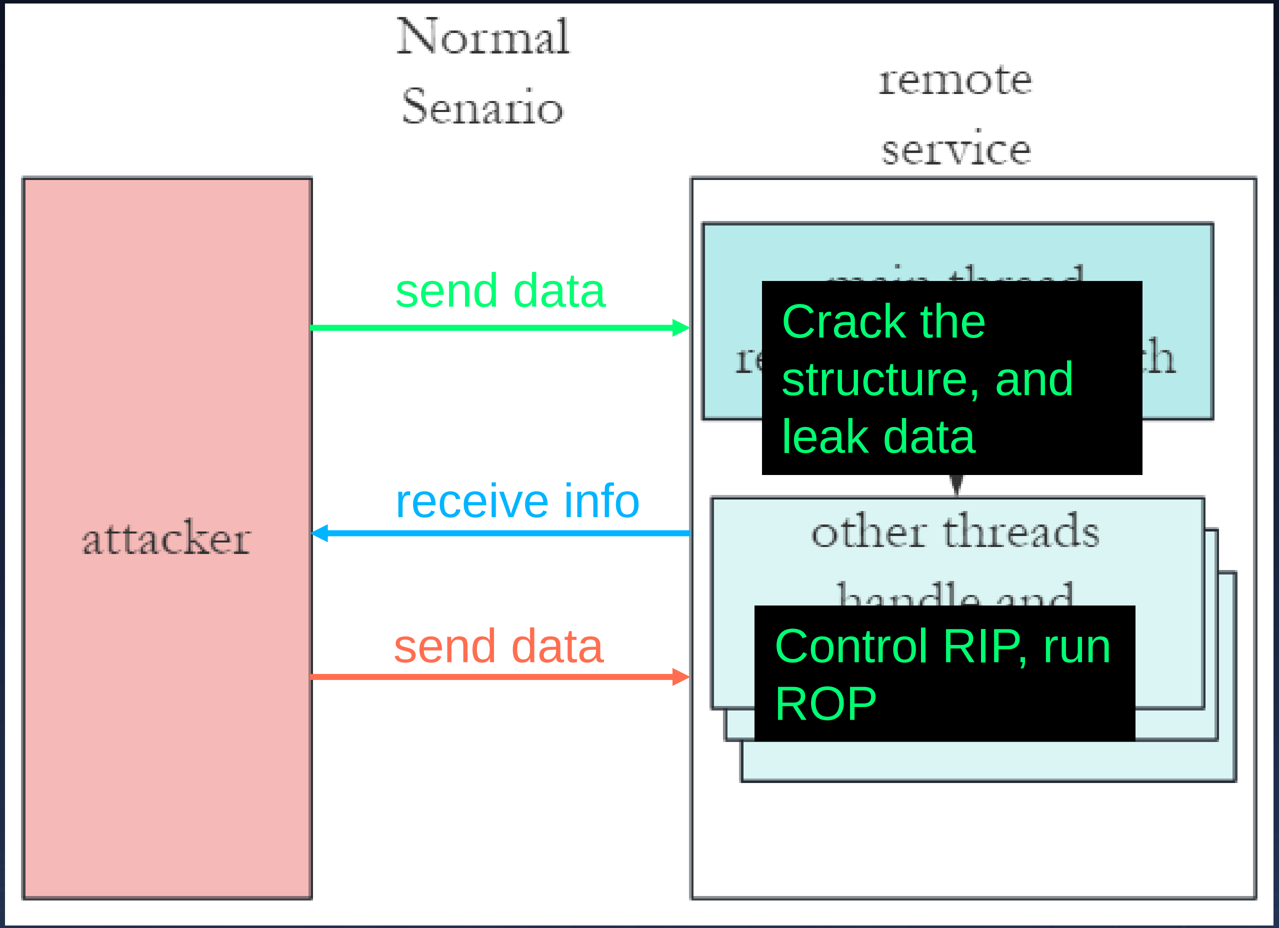
In Host vmx, use HVA of GPA access Guest memory

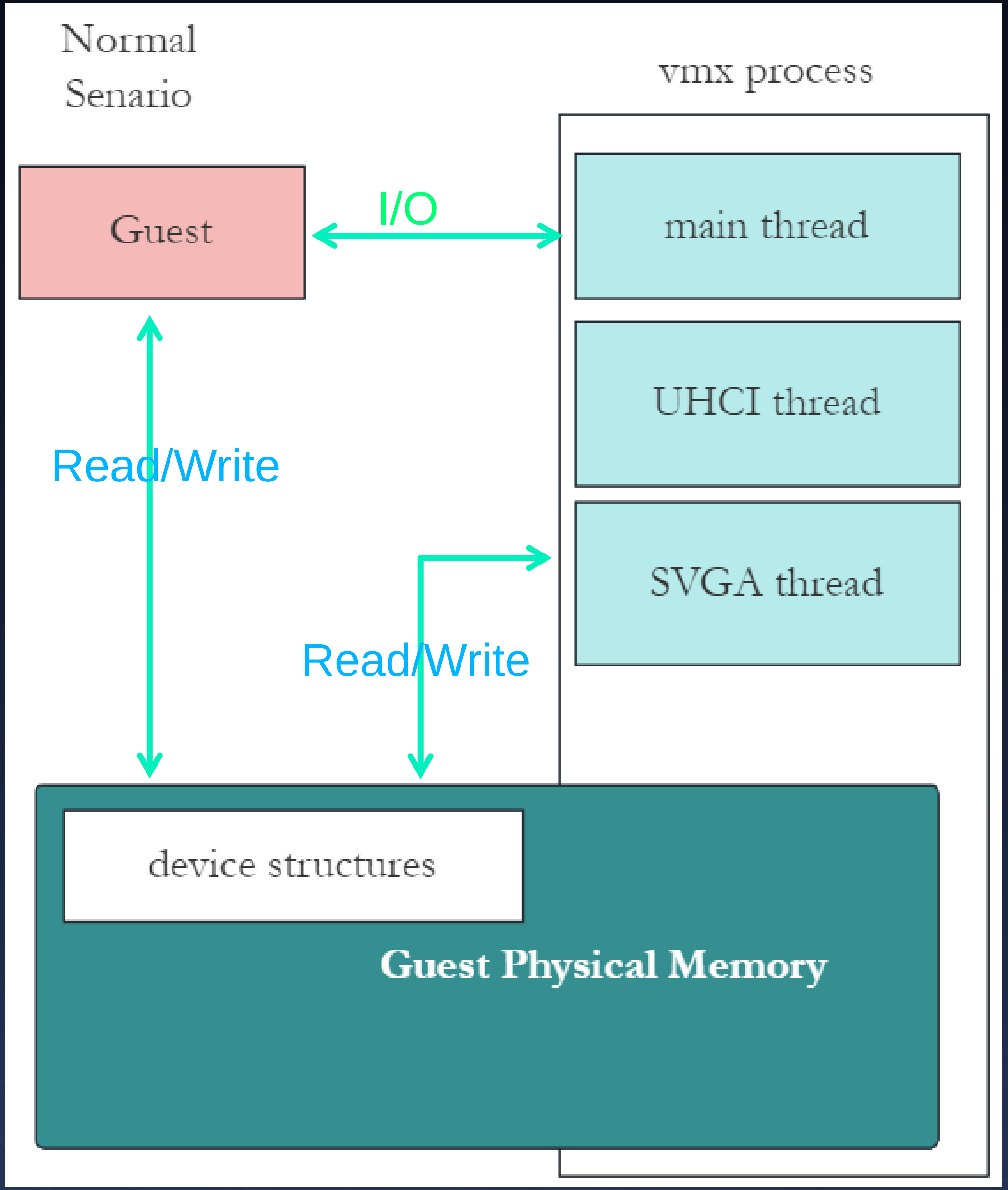


Virtualization Basic Info
Virtual Device and Guest Driver Interaction


Guest System
IO Port
Insb/Inb/outb/outsb
IO Memory
Map to GVA, Directly read and write
IO port handler functions
IO Memory handler functions
VMX process

Virtualization Basic Info
VM Escape and RCE exploit





Virtualization Basic Info
USB Controller

USB 1.x
UHCI

USB 2.0
EHCI

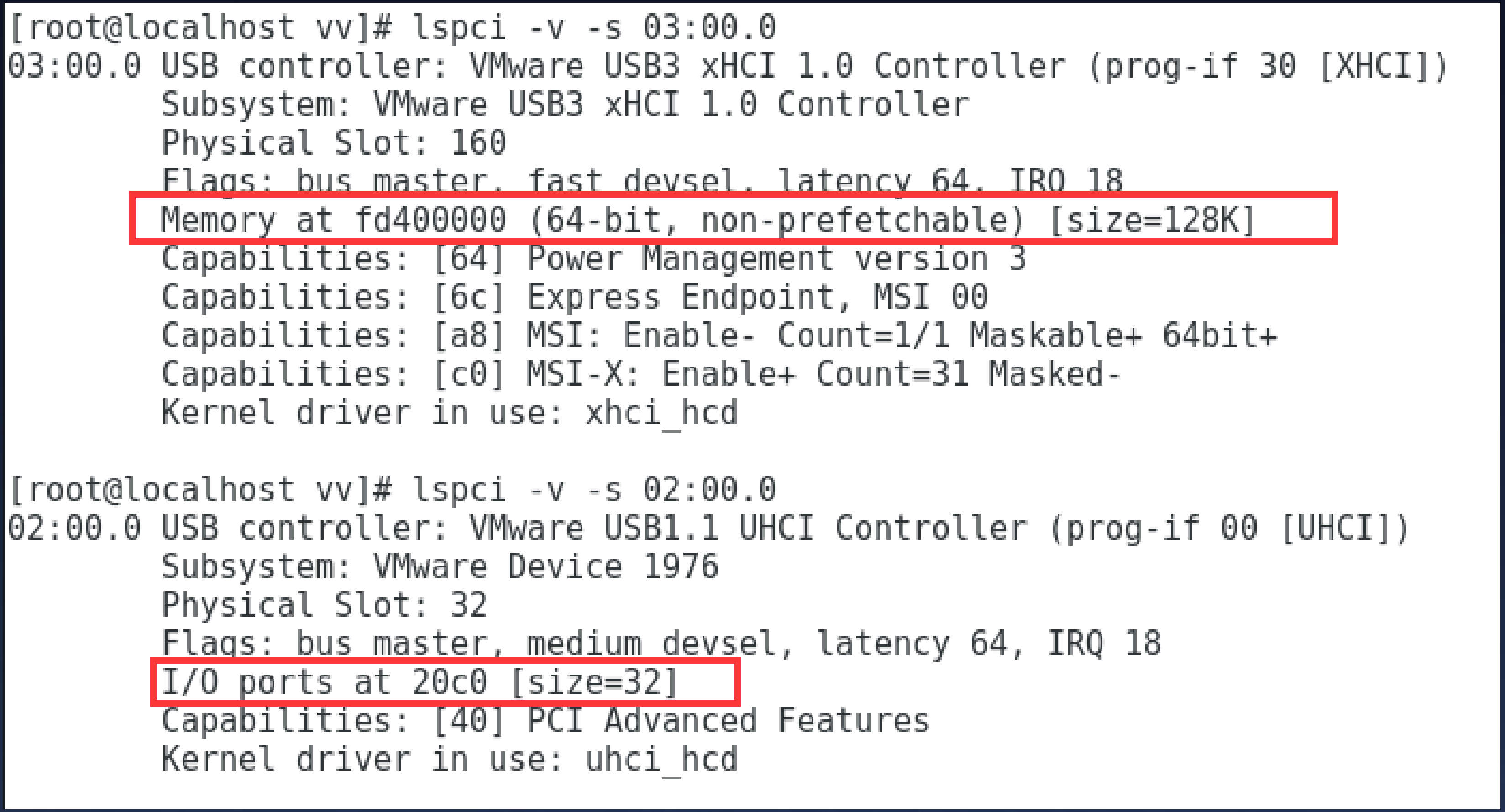
USB 3.x
XHCI

USB 4.0
Future

CVE-2021-22041	CVE-2022-31705 …	CVE-2024-22252
CVE-2019-5519		CVE-2021-22040
CVE-2019-5518		CVE-2020-4004
CVE-2023-20870 …		CVE-2020-3968
		CVE-2017-4904 …

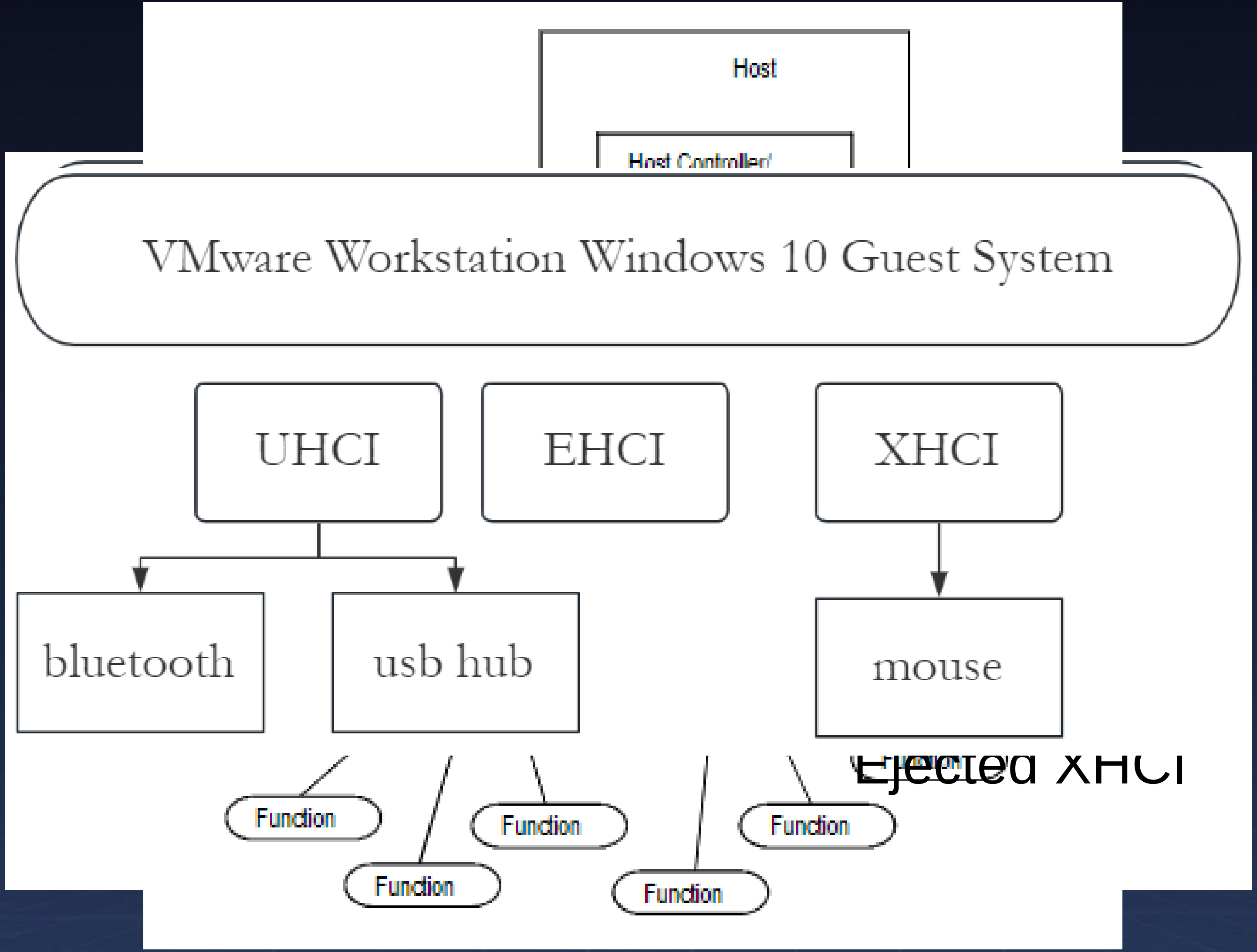
Virtualization Basic Info

Virtual USB Controller Device Info

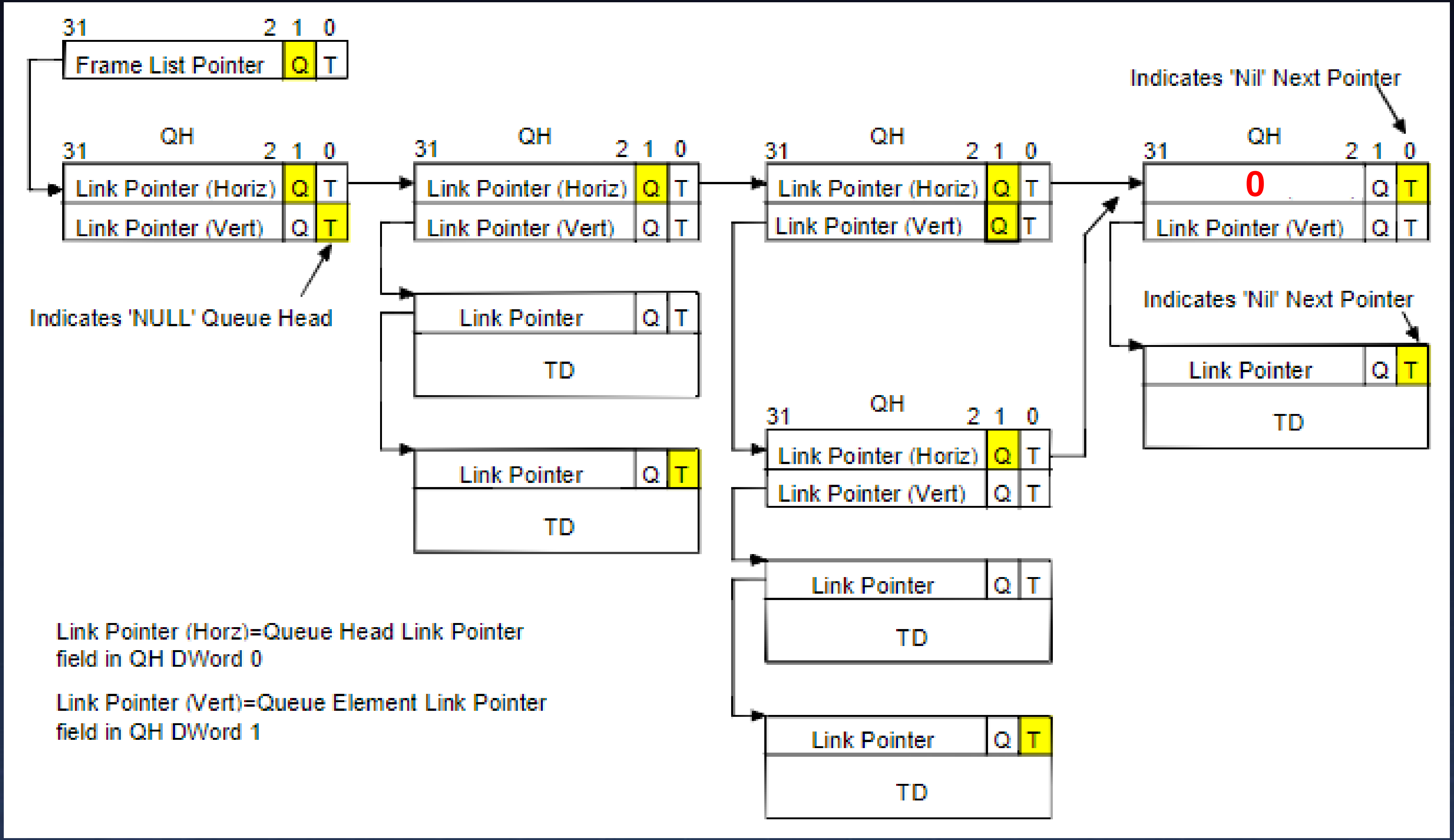


```
[root@localhost vv]# lspci -v -s 03:00.0
03:00.0 USB controller: VMware USB3 xHCI 1.0 Controller (prog-if 30 [XHCI])
        Subsystem: VMware USB3 xHCI 1.0 Controller
        Physical Slot: 160
        Flags: bus master, fast devsel, latency 64, IRQ 18
        Memory at fd400000 (64-bit, non-prefetchable) [size=128K]
        Capabilities: [64] Power Management version 3
        Capabilities: [6c] Express Endpoint, MSI 00
        Capabilities: [a8] MSI: Enable- Count=1/1 Maskable+ 64bit+
        Capabilities: [c0] MSI-X: Enable+ Count=31 Masked-
        Kernel driver in use: xhci_hcd

[root@localhost vv]# lspci -v -s 02:00.0
02:00.0 USB controller: VMware USB1.1 UHCI Controller (prog-if 00 [UHCI])
        Subsystem: VMware Device 1976
        Physical Slot: 32
        Flags: bus master, medium devsel, latency 64, IRQ 18
        I/O ports at 20c0 [size=32]
        Capabilities: [40] PCI Advanced Features
        Kernel driver in use: uhci_hcd
```

Virtualization Basic Info
UHCI Controller



Virtualization Basic Info
UHCI Controller



Link Pointer (Horz)=Queue Head Link Pointer field in QH DWord 0

Link Pointer (Vert)=Queue Element Link Pointer field in QH DWord 1

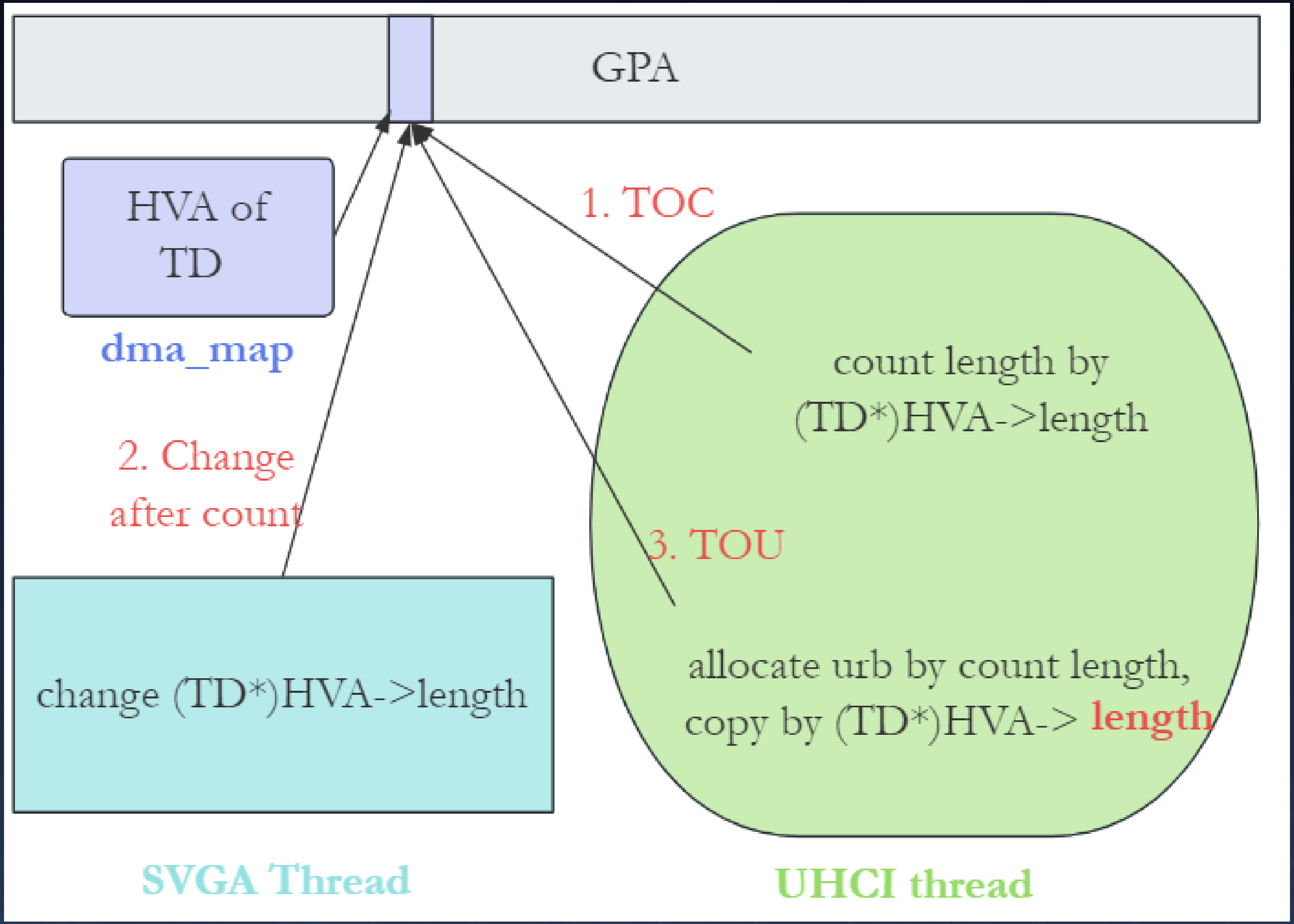
Virtualization Basic Info
UHCI Controller

```
          u32 * TD = dmaAlloc(0x10, &TD_GPA);
          buffer = dmaAlloc(0x10, &buffer_GPA);

          frame_list[0] = TD_GPA | 1;

          TD[0] = 1;// end
          TD[1] = 1 << 23;// active
          TD[2] = (2 << 8) | (0 << 15) | (7 << 21) | 0x2d;
//dev_id: 2, ep_id: 0, length: 8(7+1), type: setup(0x2d)
          TD[3] = buffer_GPA;

          buffer[0] =XXX;
```

PART TWO

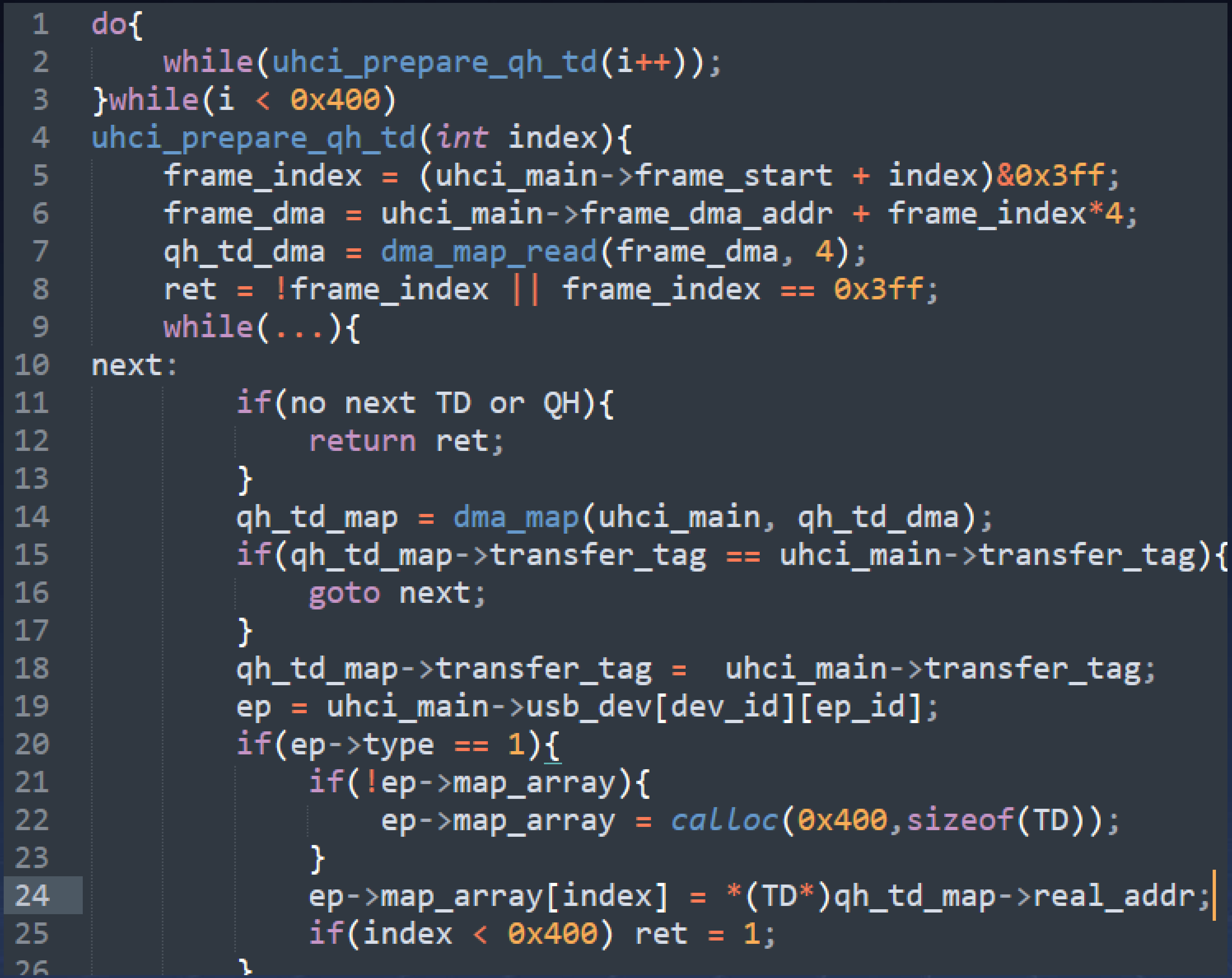
Historic Bugs In UHCI

CVE-2019-5519 TOCTOU



Found by Amat Cama and Richard Zhu

CVE-2021-22041 TOCTOU



```
do{
    while(uhci_prepare_qh_td(i++));
}while(i < 0x400)
uhci_prepare_qh_td(int index){
    frame_index = (uhci_main->frame_start + index)&0x3ff;
    frame_dma = uhci_main->frame_dma_addr + frame_index*4;
    qh_td_dma = dma_map_read(frame_dma, 4);
    ret = !frame_index || frame_index == 0x3ff;
    while(...){
next:
        if(no next TD or QH){
            return ret;
        }
        qh_td_map = dma_map(uhci_main, qh_td_dma);
        if(qh_td_map->transfer_tag == uhci_main->transfer_tag){
            goto next;
        }
        qh_td_map->transfer_tag =  uhci_main->transfer_tag;
        ep = uhci_main->usb_dev[dev_id][ep_id];
        if(ep->type == 1){
            if(!ep->map_array){
                ep->map_array = calloc(0x400,sizeof(TD));
            }
            ep->map_array[index] = *(TD*)qh_td_map->real_addr;
            if(index < 0x400) ret = 1;
        }
```

If frame_start = 0x3ff,
i=0x400,

frame_index =
(0x400+0x3ff)&0x3ff = 0x3ff;

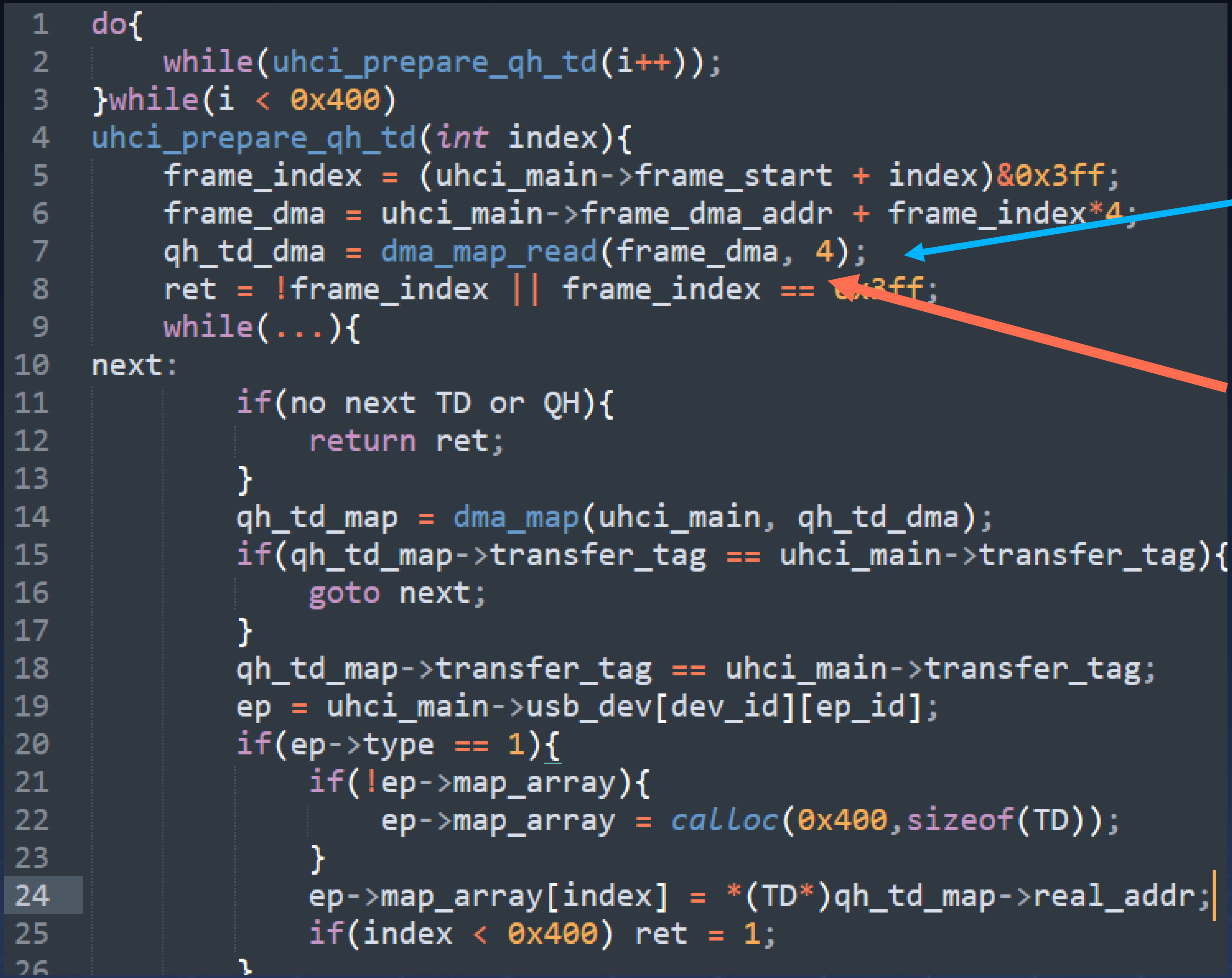
ret = 1

frame[(0+0x3ff)&0x3ff] ==
frame[(0x400+0x3ff)&0x3ff]

transfer_tag will match

Found by me, used in TianfuCup 2021

CVE-2021-22041 TOCTOU



```c
do{
    while(uhci_prepare_qh_td(i++));
}while(i < 0x400)
uhci_prepare_qh_td(int index){
    frame_index = (uhci_main->frame_start + index)&0x3ff;
    frame_dma = uhci_main->frame_dma_addr + frame_index*4;
    qh_td_dma = dma_map_read(frame_dma, 4);
    ret = !frame_index || frame_index == 0x3ff;
    while(...){
next:
        if(no next TD or QH){
            return ret;
        }
        qh_td_map = dma_map(uhci_main, qh_td_dma);
        if(qh_td_map->transfer_tag == uhci_main->transfer_tag){
            goto next;
        }
        qh_td_map->transfer_tag == uhci_main->transfer_tag;
        ep = uhci_main->usb_dev[dev_id][ep_id];
        if(ep->type == 1){
            if(!ep->map_array){
                ep->map_array = calloc(0x400,sizeof(TD));
            }
            ep->map_array[index] = *(TD*)qh_td_map->real_addr;
            if(index < 0x400) ret = 1;
        }
```

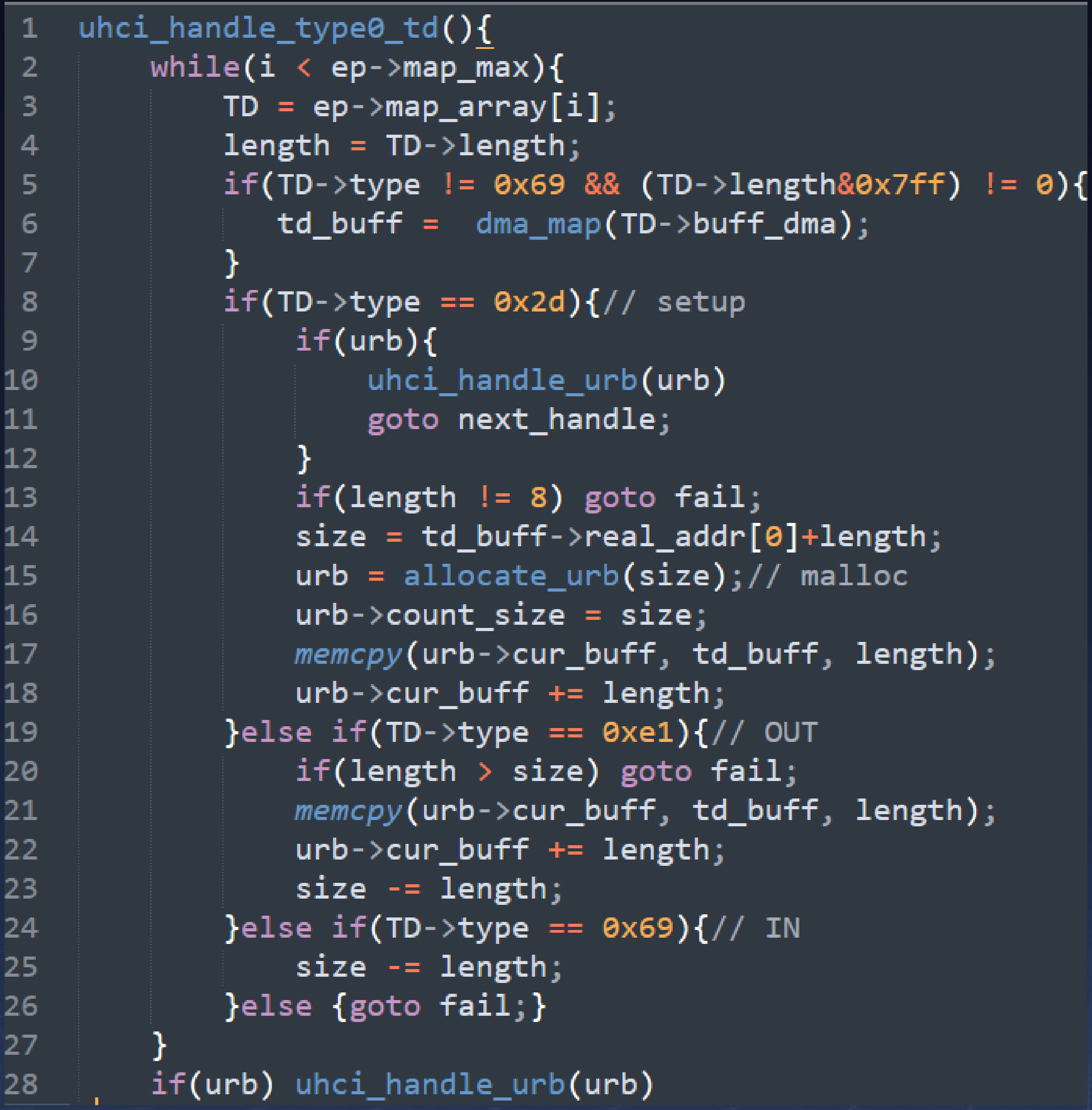
1. Access frame[0x3ff]

2. Change frame[0x3ff] in SVGA thread

3. Access frame[(0x400+0x3ff)&0x3ff] again
Get a new GPA

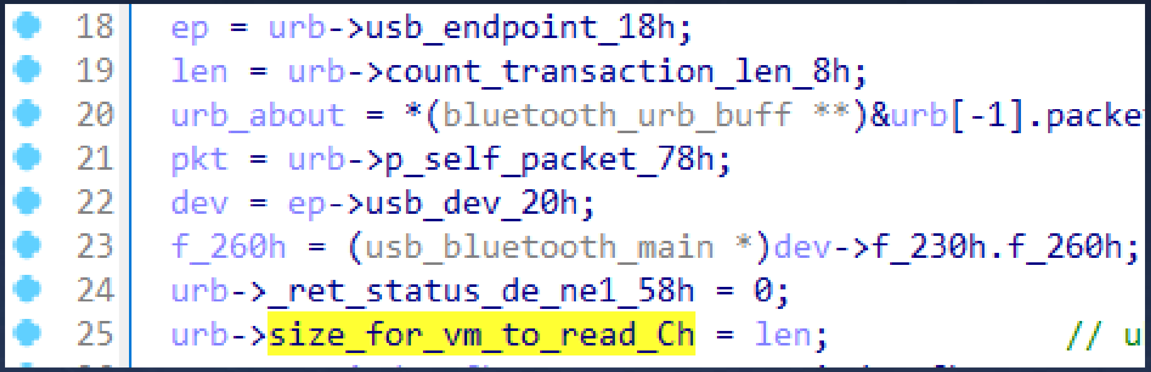
Found by me, used in TianfuCup 2021

CVE-2023-20870 Uninitialize Leak



```
uhci_handle_type0_td(){
    while(i < ep->map_max){
        TD = ep->map_array[i];
        length = TD->length;
        if(TD->type != 0x69 && (TD->length&0x7ff) != 0){
           td_buff =  dma_map(TD->buff_dma);
        }
        if(TD->type == 0x2d){// setup
            if(urb){
                uhci_handle_urb(urb)
                goto next_handle;
            }
            if(length != 8) goto fail;
            size = td_buff->real_addr[0]+length;
            urb = allocate_urb(size);// malloc
            urb->count_size = size;
            memcpy(urb->cur_buff, td_buff, length);
            urb->cur_buff += length;
        }else if(TD->type == 0xe1){// OUT
            if(length > size) goto fail;
            memcpy(urb->cur_buff, td_buff, length);
            urb->cur_buff += length;
            size -= length;
        }else if(TD->type == 0x69){// IN
            size -= length;
        }else {goto fail;}
    }
    if(urb) uhci_handle_urb(urb)
```

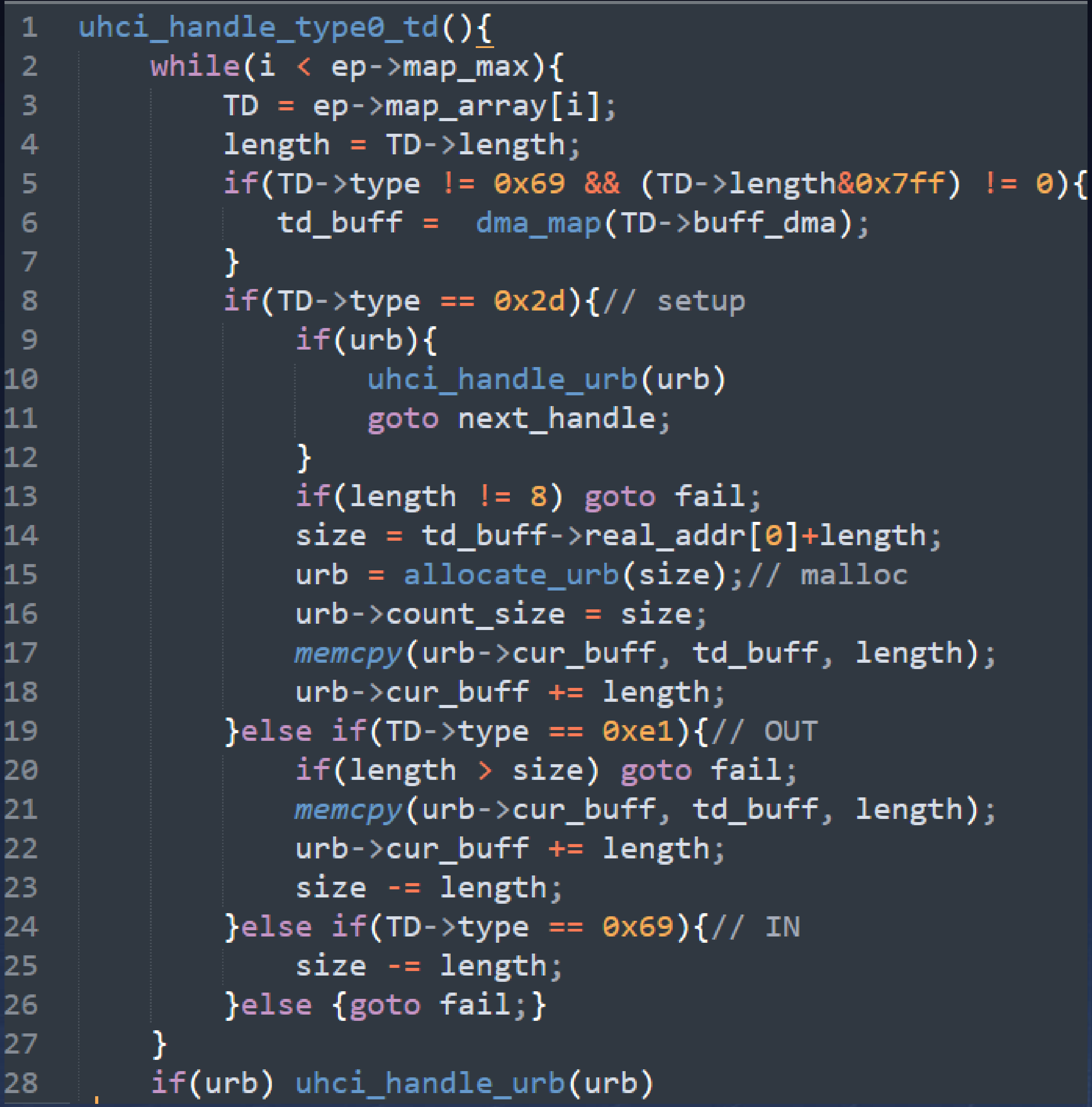
```
struct urb{
    +0h reference;
    +4h buffer size;
    +8h count size;
    +Ch size can read to vm; default 0
    ...
    +18h endpoint;
    ...
    +78h buffer start;
    +80h cur_buff;
    char buffer[xxx]; size is determined by input size
}
```



```
ep = urb->usb_endpoint_18h;
len = urb->count_transaction_len_8h;
urb_about = *(bluetooth_urb_buff **)&urb[-1].packe
pkt = urb->p_self_packet_78h;
dev = ep->usb_dev_20h;
f_260h = (usb_bluetooth_main *)dev->f_230h.f_260h;
urb->_ret_status_de_ne1_58h = 0;
urb->size_for_vm_to_read_Ch = len;             // u
```

Found by Thach Nguyen Hoang of STAR Labs, Wei and me also found it.

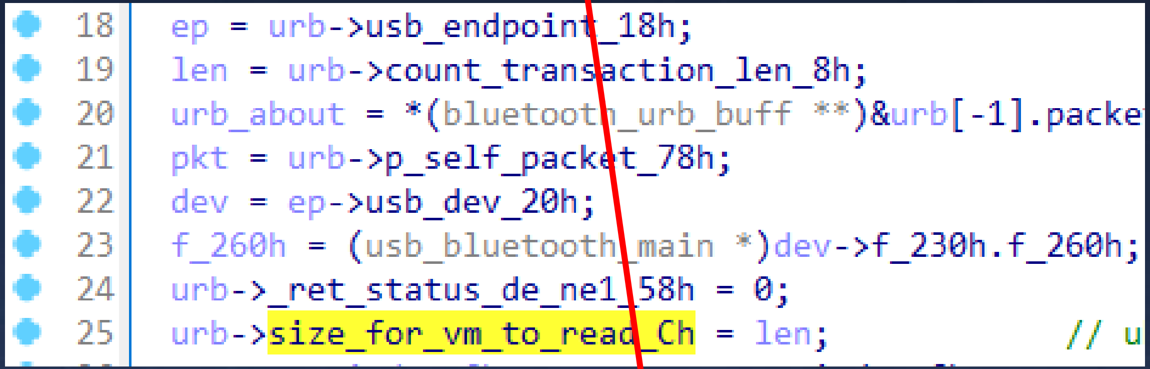
CVE-2023-20870 Uninitialize Leak



```
uhci_handle_type0_td(){
    while(i < ep->map_max){
        TD = ep->map_array[i];
        length = TD->length;
        if(TD->type != 0x69 && (TD->length&0x7ff) != 0){
           td_buff =  dma_map(TD->buff_dma);
        }
        if(TD->type == 0x2d){// setup
            if(urb){
                uhci_handle_urb(urb)
                goto next_handle;
            }
            if(length != 8) goto fail;
            size = td_buff->real_addr[0]+length;
            urb = allocate_urb(size);// malloc
            urb->count_size = size;
            memcpy(urb->cur_buff, td_buff, length);
            urb->cur_buff += length;
        }else if(TD->type == 0xe1){// OUT
            if(length > size) goto fail;
            memcpy(urb->cur_buff, td_buff, length);
            urb->cur_buff += length;
            size -= length;
        }else if(TD->type == 0x69){// IN
            size -= length;
        }else {goto fail;}
    }
    if(urb) uhci_handle_urb(urb)
```

Fix:

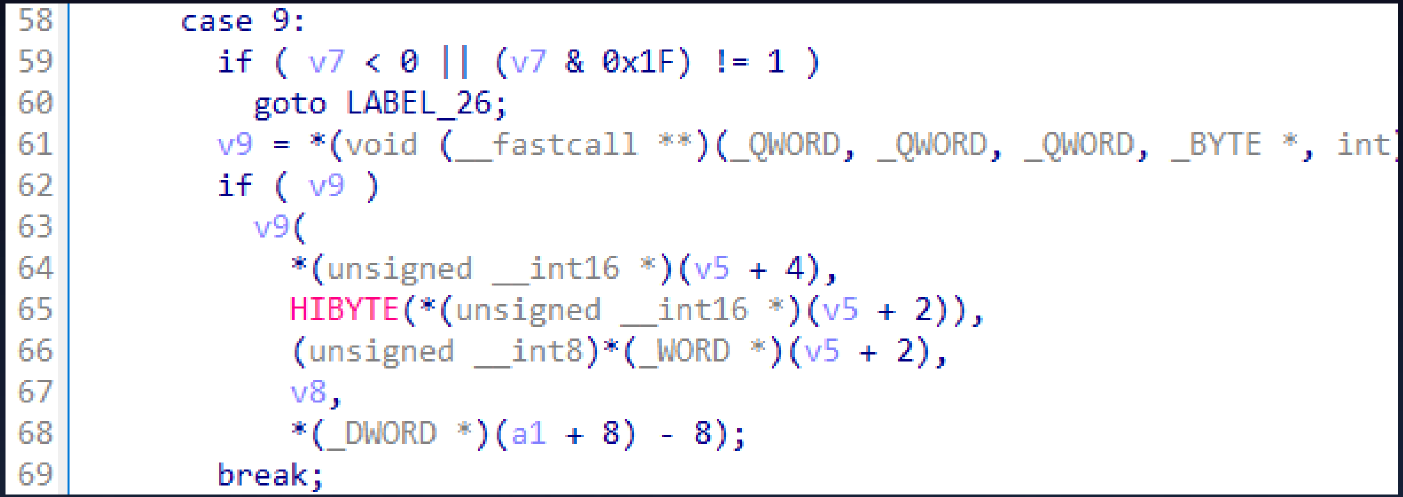
Set urb->Ch = 8 in Bluetooth handler



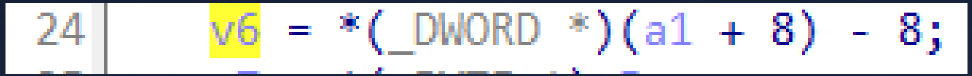
```
ep = urb->usb_endpoint_18h;
len = urb->count_transaction_len_8h;
urb_about = *(bluetooth_urb_buff **)&urb[-1].packe
pkt = urb->p_self_packet_78h;
dev = ep->usb_dev_20h;
f_260h = (usb_bluetooth_main *)dev->f_230h.f_260h;
urb->_ret_status_de_ne1_58h = 0;
urb->size_for_vm_to_read_Ch = len;              // u
```

Found by Thach Nguyen Hoang of STAR Labs, Wei and me also found it.

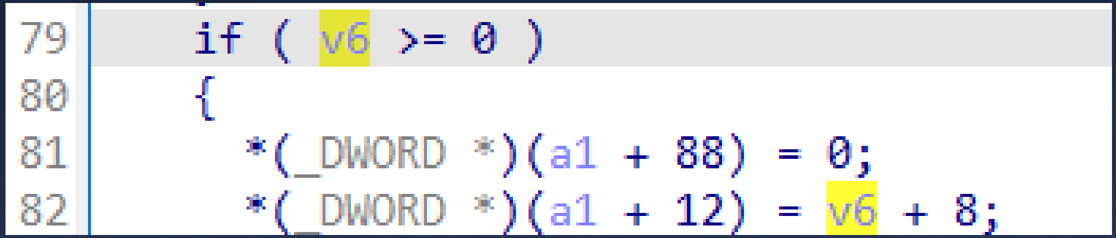
CVE-2024-22255 Uninitialize Leak



```
      case 9:
        if ( v7 < 0 || (v7 & 0x1F) != 1 )
          goto LABEL_26;
        v9 = *(void (__fastcall **)(_QWORD, _QWORD, _QWORD, _BYTE *, int
        if ( v9 )
          v9(
            *(unsigned __int16 *)(v5 + 4),
            HIBYTE(*(unsigned __int16 *)(v5 + 2)),
            (unsigned __int8)*(_WORD *)(v5 + 2),
            v8,
            *(_DWORD *)(a1 + 8) - 8);
        break;
```



```
  v6 = *(_DWORD *)(a1 + 8) - 8;
```



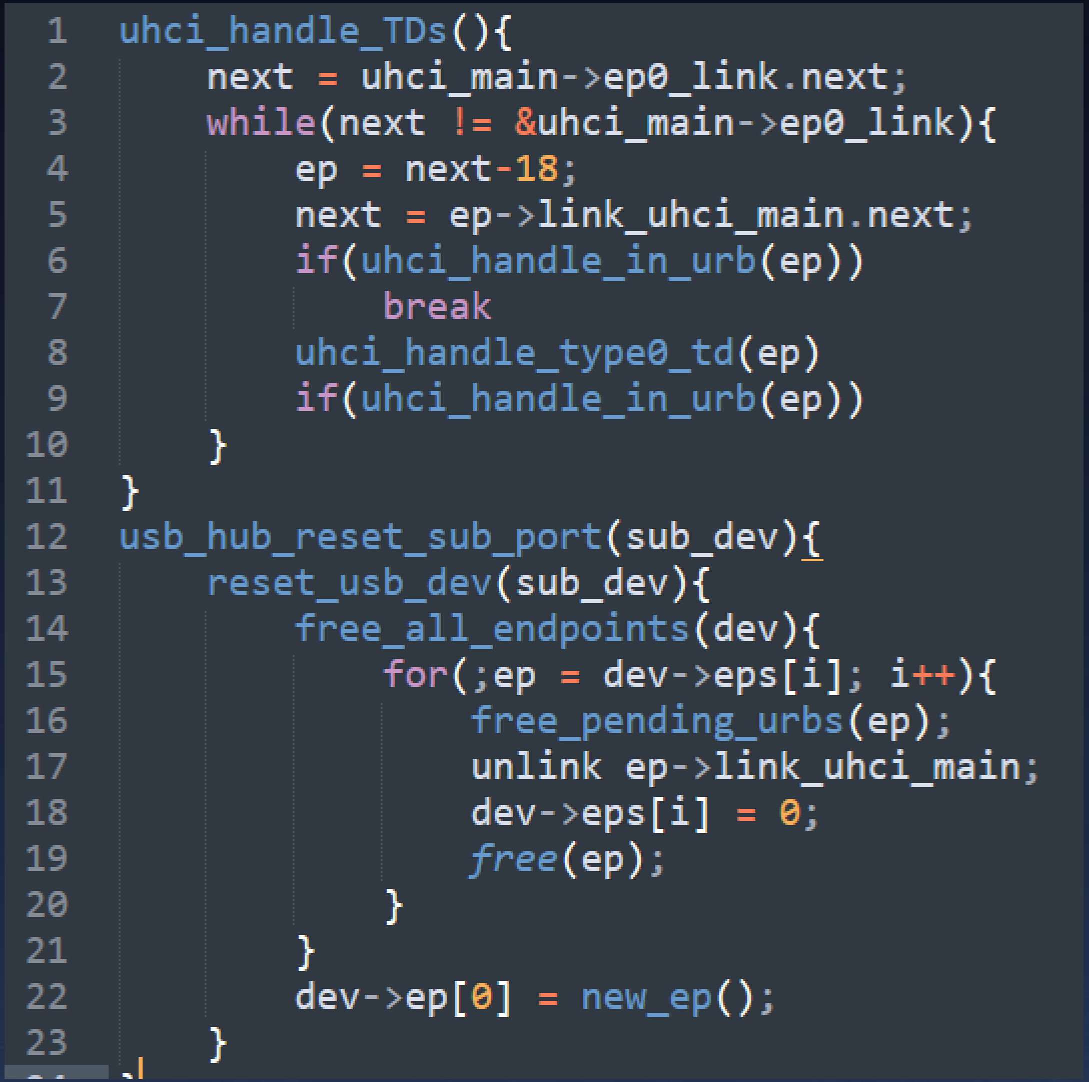
```
  if ( v6 >= 0 )
  {
    *(_DWORD *)(a1 + 88) = 0;
    *(_DWORD *)(a1 + 12) = v6 + 8;
```

```
struct urb{
    +0h reference;
    +4h buffer size;
    +8h count size;
    +Ch size can read to vm; default 0
    ...
    +18h endpoint;
    ...
    +78h buffer start;
    +80h cur_buff;
    char buffer[xxx]
}
```

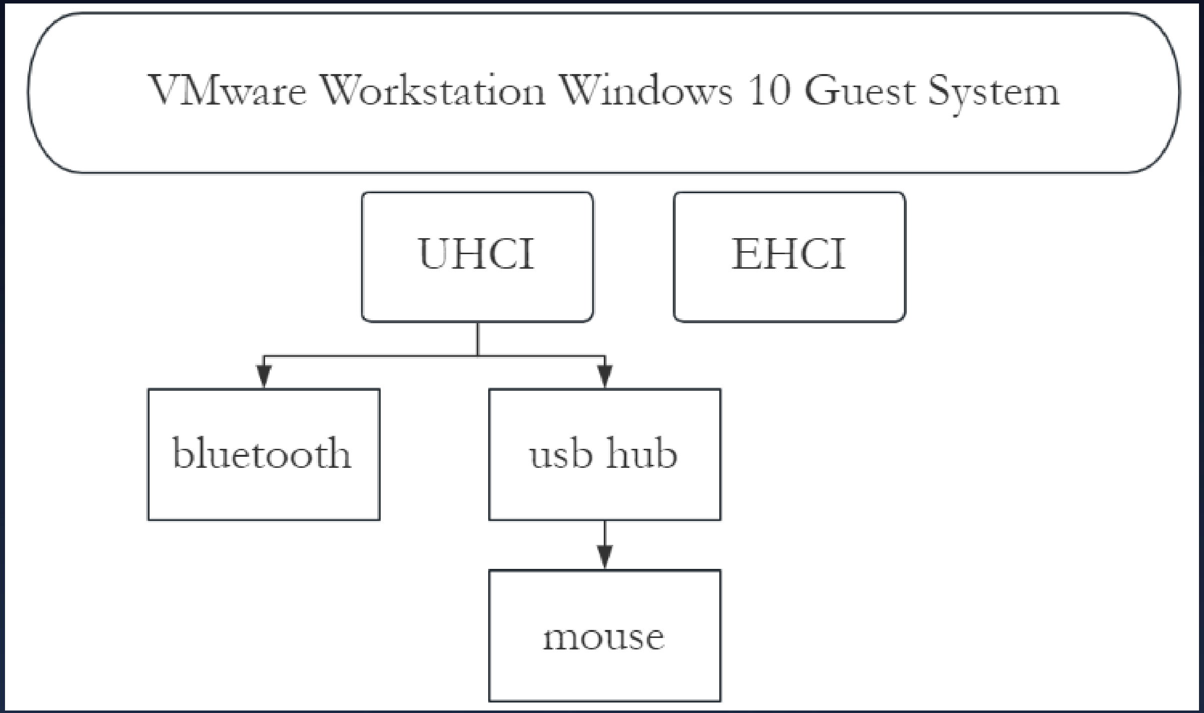
```
U8(buffer, 0) = 0x21;
U8(buffer, 1) = 9;// CASE
U16(buffer, 6) = buffer size  - 8;
```

Found by Wei and me

CVE-2024-22253 UAF



```
uhci_handle_TDs(){
    next = uhci_main->ep0_link.next;
    while(next != &uhci_main->ep0_link){
        ep = next-18;
        next = ep->link_uhci_main.next;
        if(uhci_handle_in_urb(ep))
            break
        uhci_handle_type0_td(ep)
        if(uhci_handle_in_urb(ep))
    }
}
usb_hub_reset_sub_port(sub_dev){
    reset_usb_dev(sub_dev){
        free_all_endpoints(dev){
            for(;ep = dev->eps[i]; i++){
                free_pending_urbs(ep);
                unlink ep->link_uhci_main;
                dev->eps[i] = 0;
                free(ep);
            }
        }
        dev->ep[0] = new_ep();
    }
```

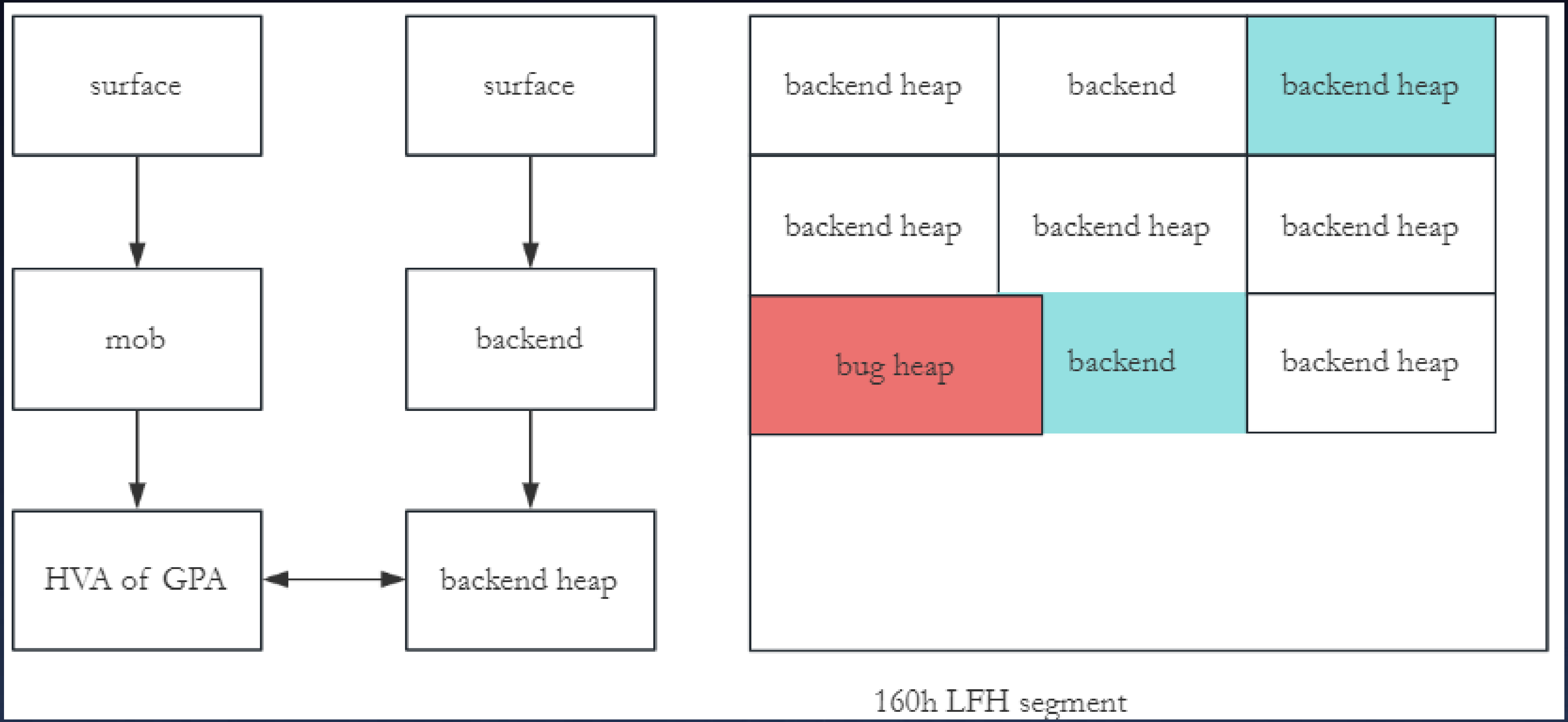


Found by me, used at TianfuCup 2023

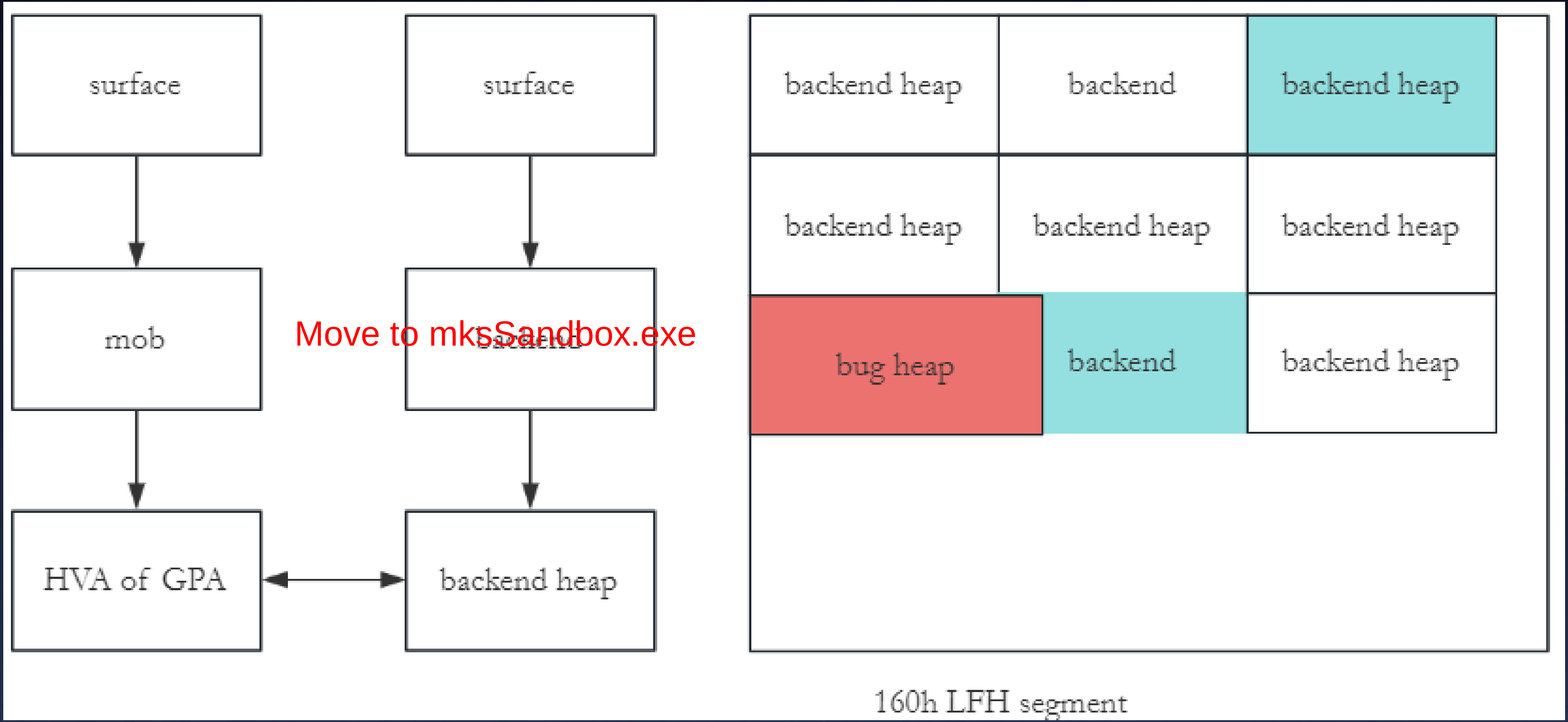
PART THREE

Exploit for TianfuCup 2023

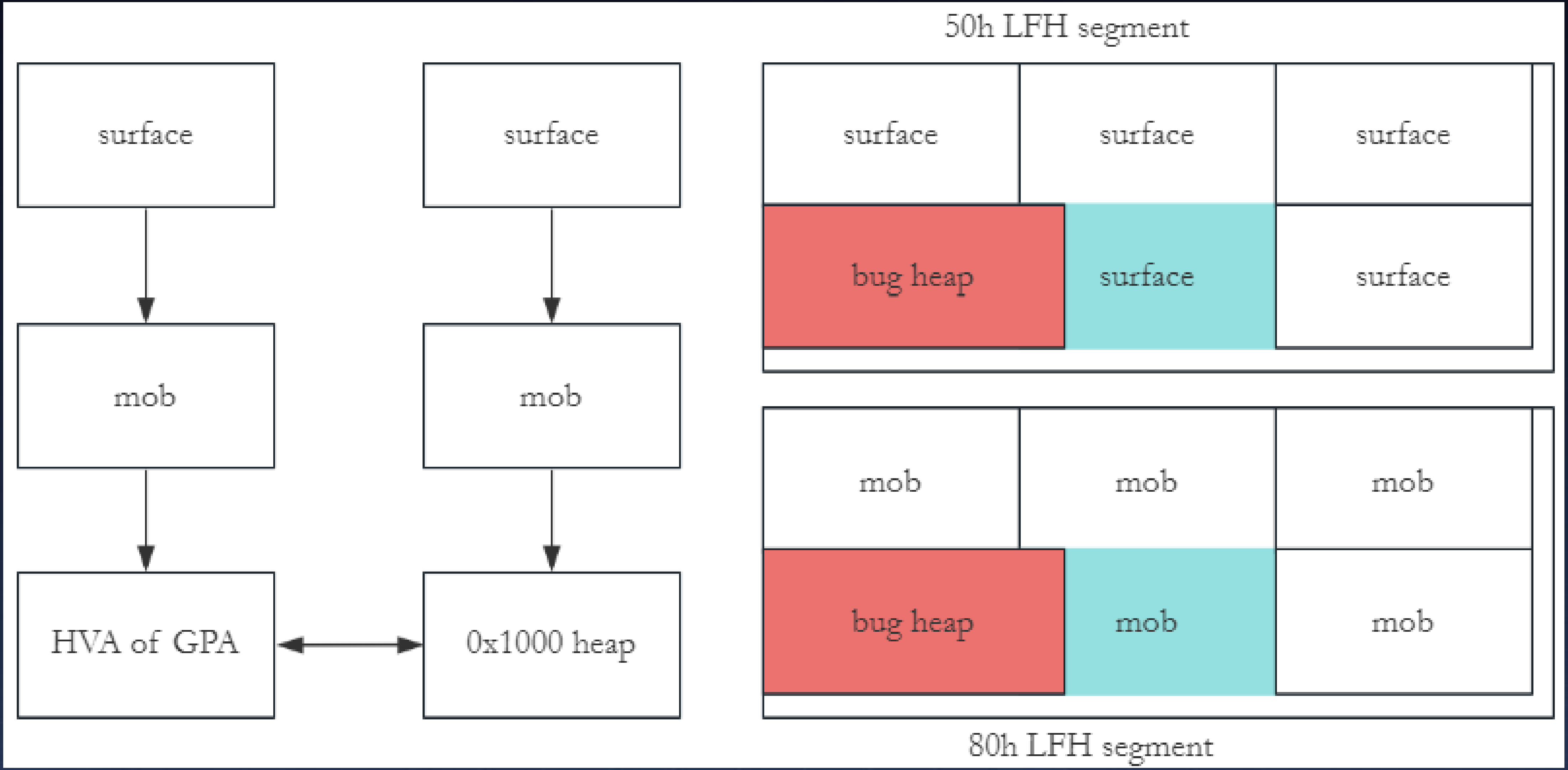
Old Exploit primitives-Straight outta VMware



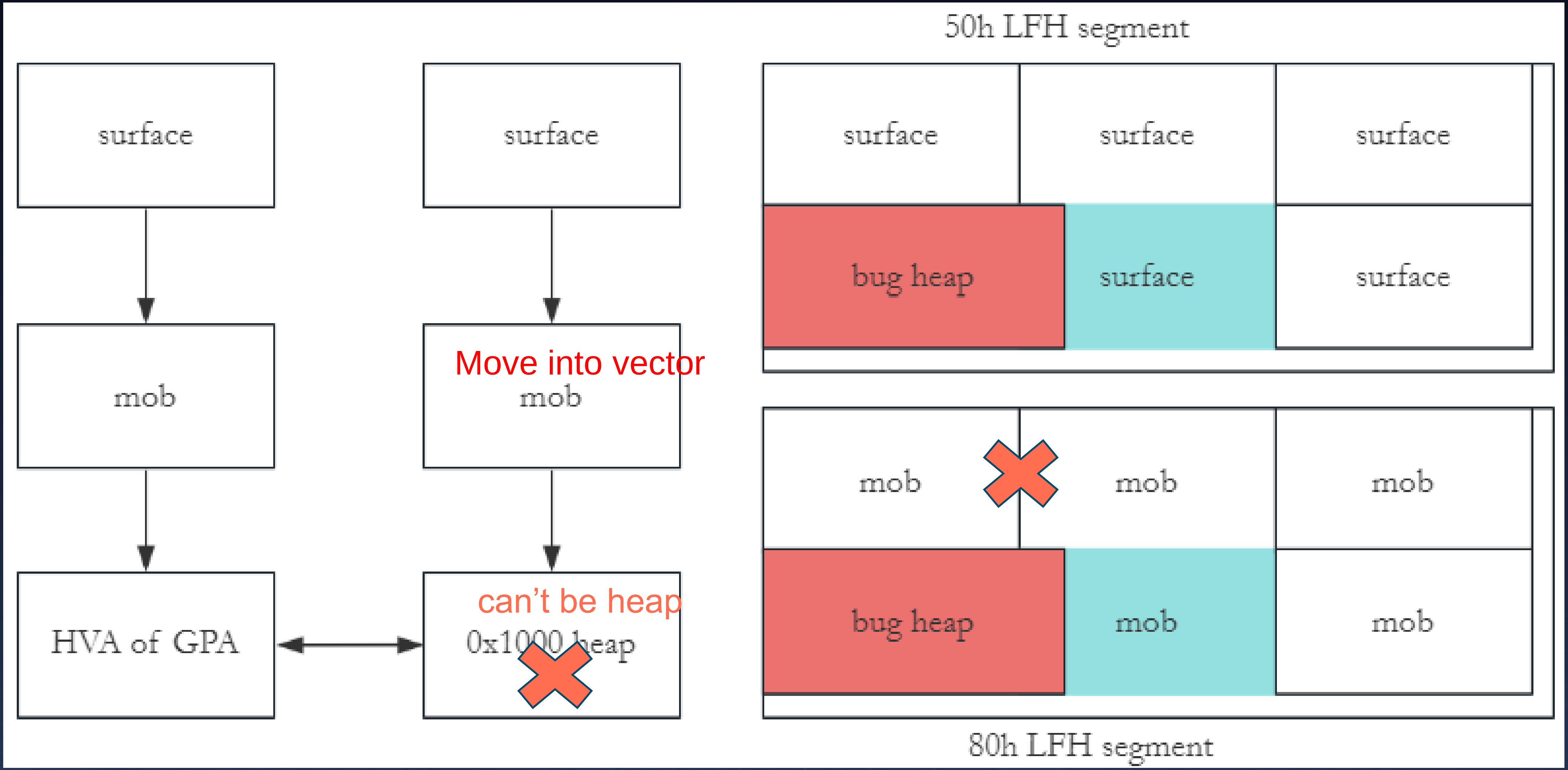
Old Exploit primitives-Straight outta VMware



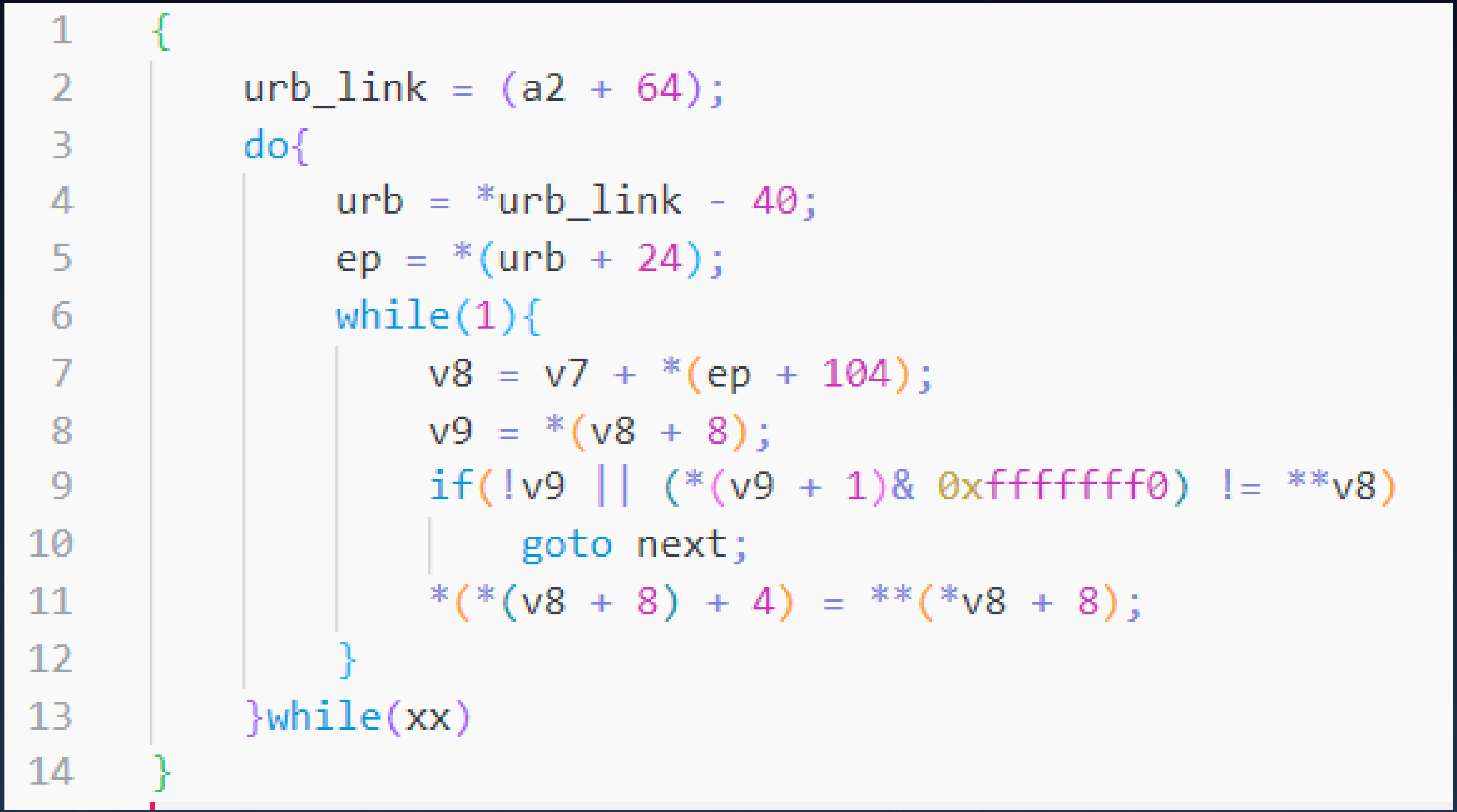
Old Exploit primitives-Breakout Script of the Westworld



Old Exploit primitives-Breakout Script of the Westworld



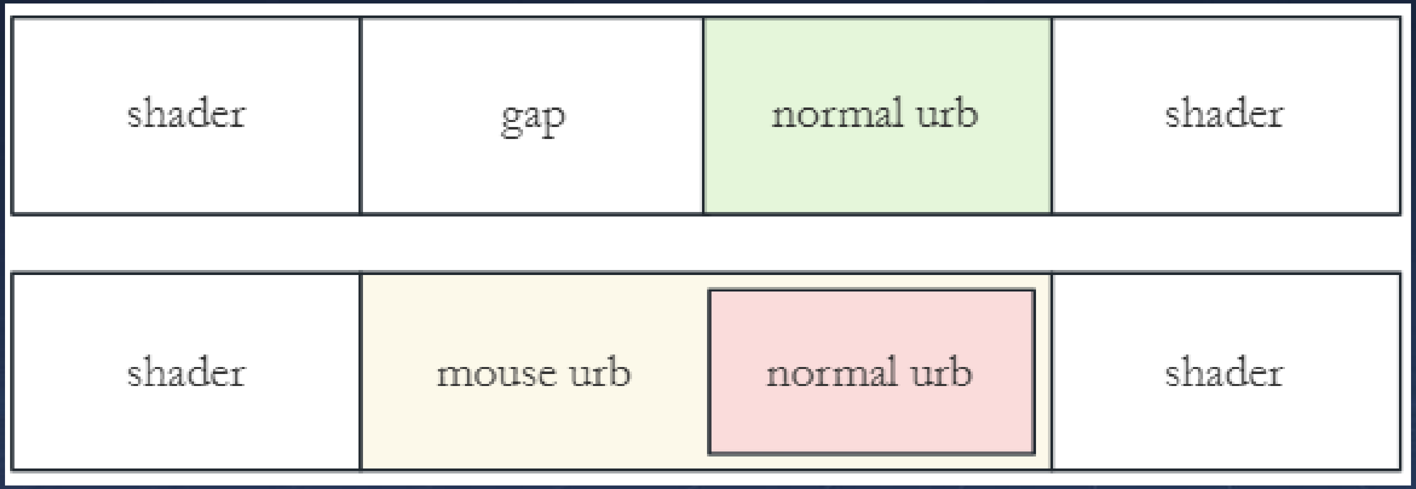
Exploit primitives-UHCI Endpoint



```
{
    urb_link = (a2 + 64);
    do{
        urb = *urb_link - 40;
        ep = *(urb + 24);
        while(1){
            v8 = v7 + *(ep + 104);
            v9 = *(v8 + 8);
            if(!v9 || (*(v9 + 1)& 0xfffffff0) != **v8)
                goto next;
            *(*(v8 + 8) + 4) = **(*v8 + 8);
        }
    }while(xx)
}
```

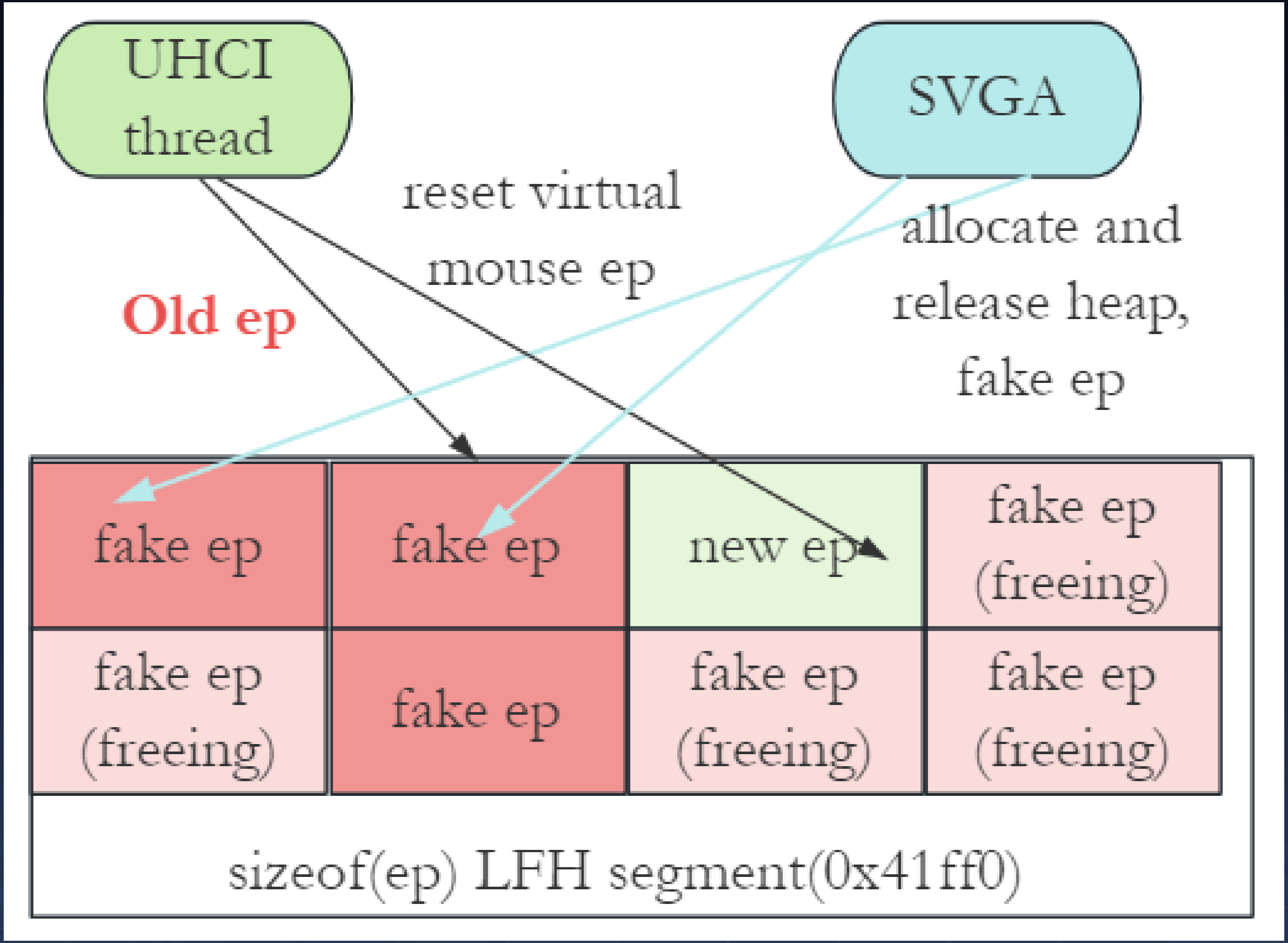
Exploit primitives-Leak address by Urb bug

```
struct urb{
    ...
    +70h vmx related process address
    +78h buffer start;
    +80h cur_buff;
    char buffer[xxx]; size is determined by input
size
}
```

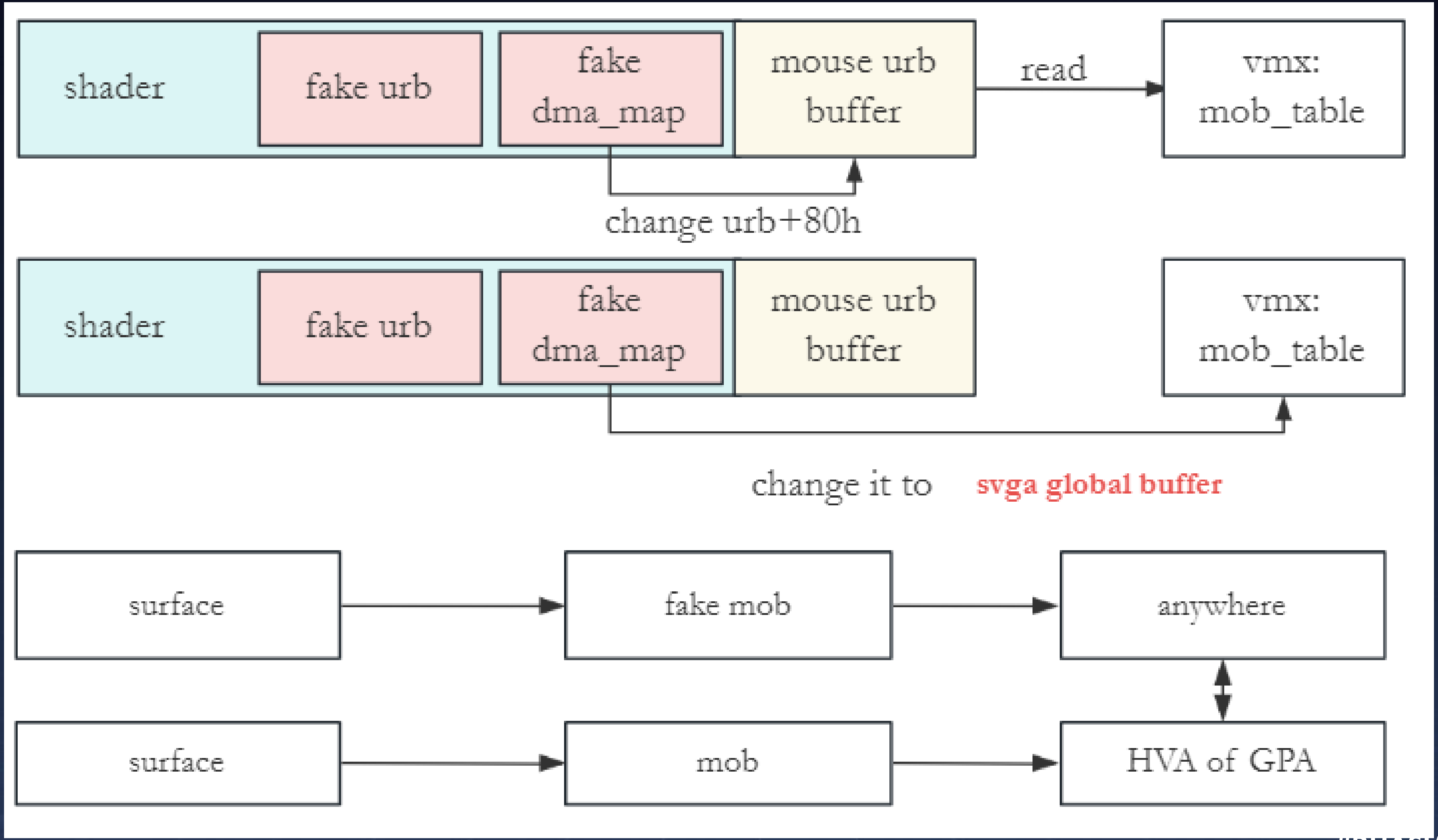


Get a urb heap address
Get VMX related address

Exploit primitives-R/W Everywhere by Endpoint primitive and urb



Exploit primitives-R/W Everywhere by Endpoint primitive and urb



Exploit Demo

Windows 11 22H2 - VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
主页
W1809
Windows 11 WIP 25951
Centos7
Windows 11 x64 test
Windows 11 22H2
回收站
Microsoft Edge
kdu
搜索
英
10:35
2023/10/26
要将输入定向到该虚拟机，请将鼠标指针移入其中或按 Ctrl+G。

PART FOUR

Summary

Black Hat Sound Bytes

Bug

Bug Research Tips

- TOCTOU, data of HVA can complete
- UAF, Notice reset operation, similar bug: cve-2020-4004

Exp

Exploit Tips

- Urb to leak data
- Endpoint to write arbitrary anywhere

Defense

Defense Escape Attack

- Remove unnecessary virtual devices: Usb, Sound, CDrom
- Disable SVGA 3D
- Keep your software newest

参考

https://census-labs.com/media/straightouttavmware-wp.pdf
Zero Day Initiative — Taking Control of VMware Through the Universal Host Control Interface: Part 2
https://github.com/474172261/slides/blob/main/Breakout%20Script%20of%20the%20Westworld-new%5B1088%5D.pdf
Universal Host Controller Interface (UHCI) Design Guide

Q&A

THANKS