A SZABAD SZOFTVEREK SZEREPE AZ IT BIZTONSÁGBAN

zero IT Lab

Bemutatkozás

- CTO @ Zero
- Member @ Crysys Lab & C0r3dump
- RE is love, RE is life
- Opening expensive calculators at times
- <3 capturing the flag

Szabadság? Biztonság? Magánélet?

- Egymástól függenek!
  - “Security by obscurity” nem működik!
    Zárt rendszerek != nagyobb biztonság
- A széleskörű közösségi kontrol a forráskód fölött megenged
  - Célzott *szabad* auditálást a kódbázis részei fölött
  - Fuzzolást (Instrumentáláshoz fontos a kód megléte)
  - Jobb hibakeresést (pl Address/Thread/Memory Sanitizer)

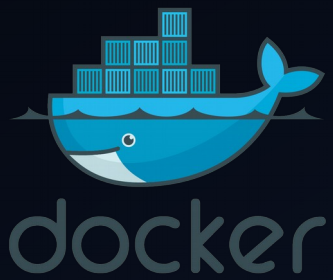
https://puri.sm/posts/why-freedom-is-essential-to-security-and-privacy/

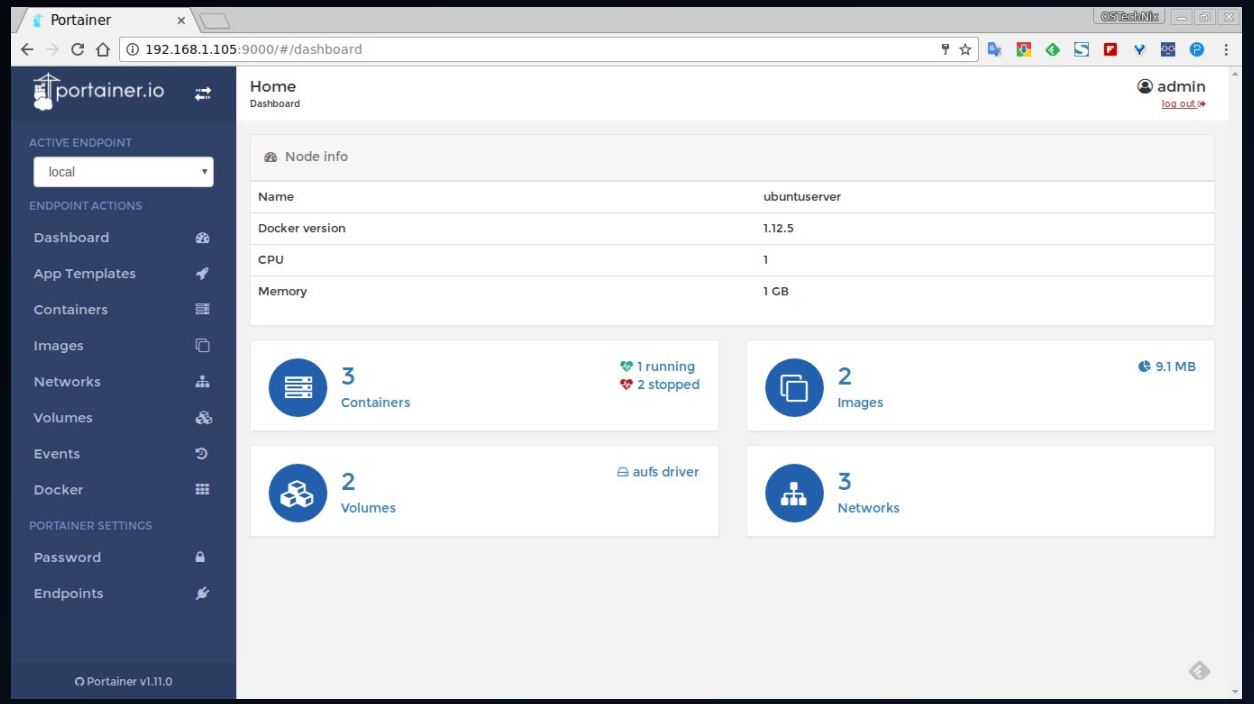
Szabad szoftverek

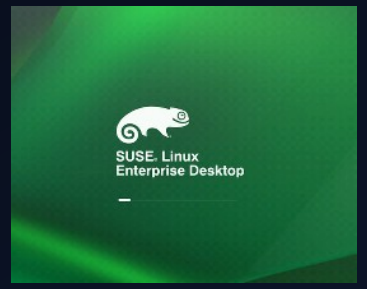
- 0: Futtatható ahogy a felhasználó szeretné
- 1: A program forráskódja elérhető, szabadon megváltoztatható
- 2: A program forráskódjának másolatai szabadon megoszthatóak
- 3: A programmal kapcsolatos változtatások szabadon megoszthatóak

http://www.gnu.org/philosophy/free-sw.en.html

Szabad != profitálható ?



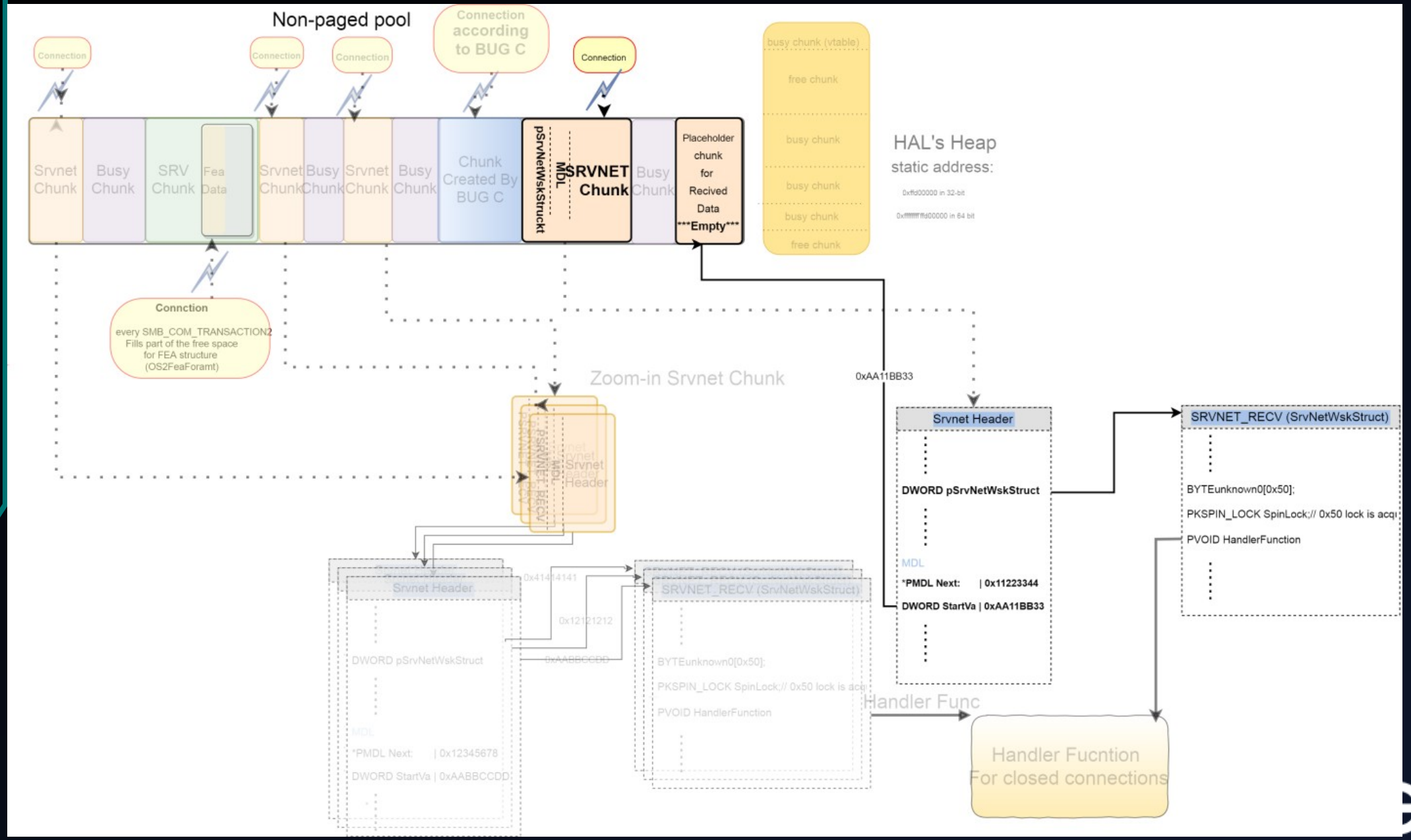




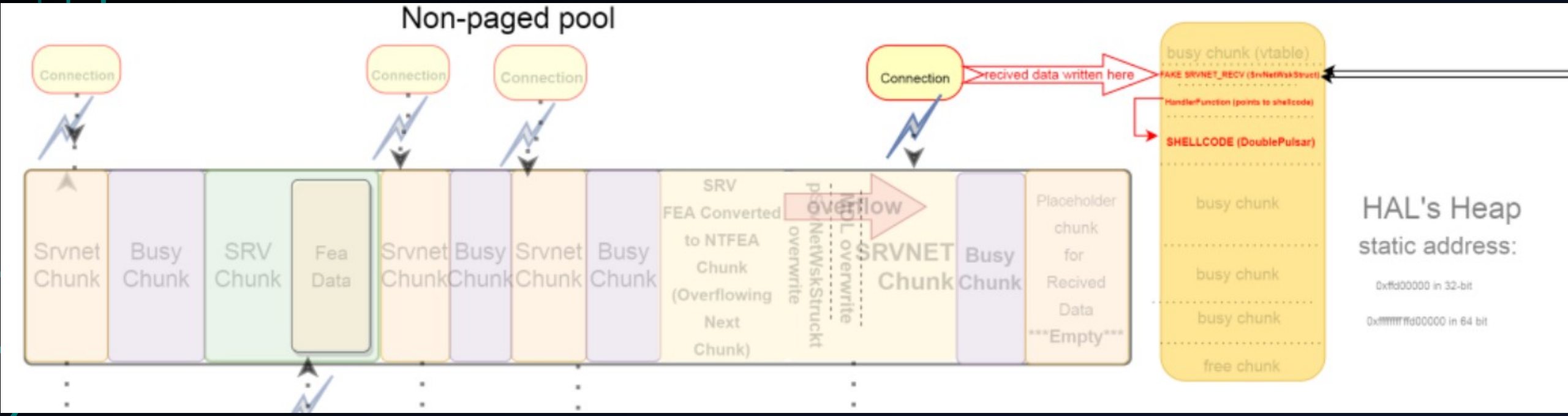
Case study: CVE-2017-0144 aka ETERNALBLUE

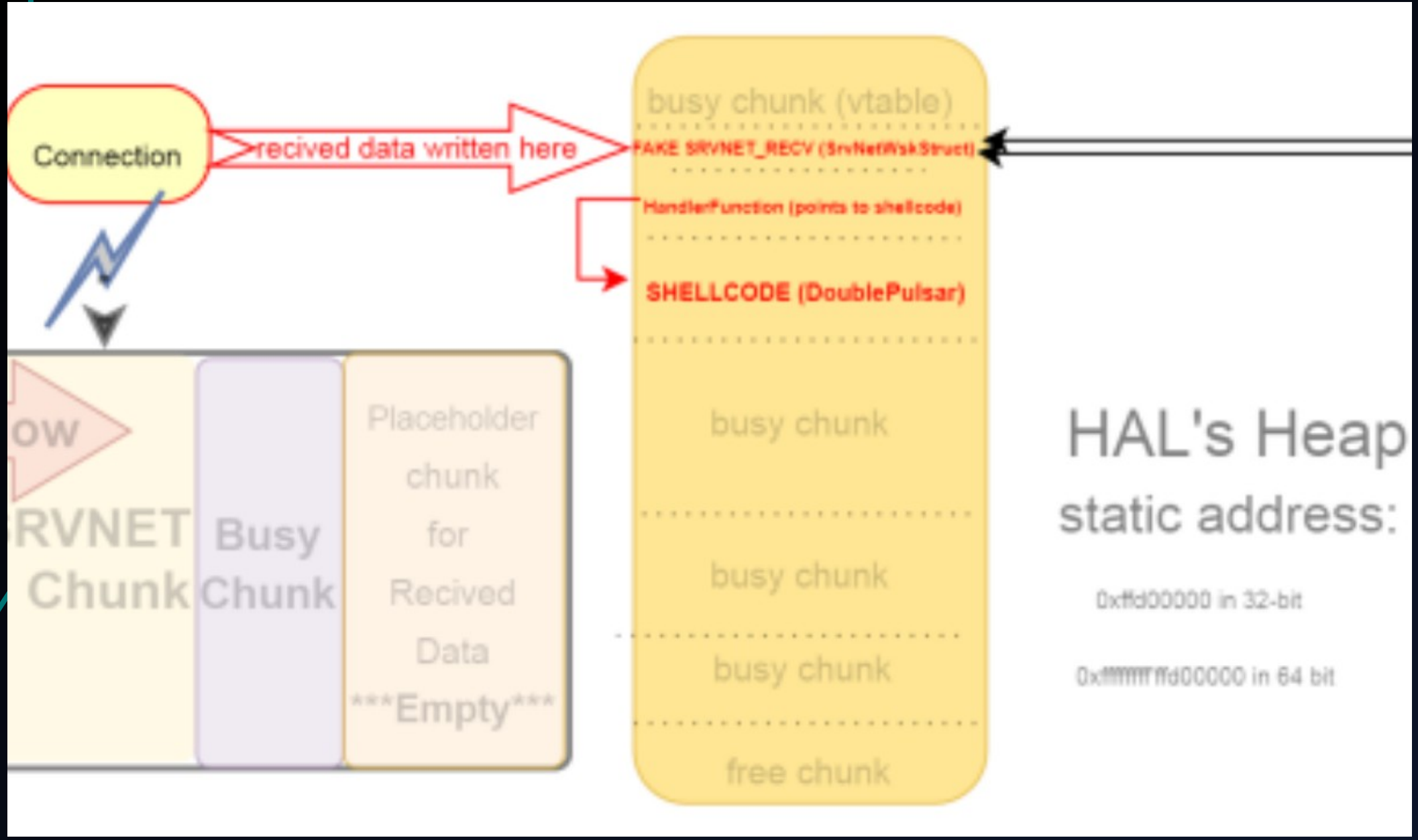
- SMB sérülékenység (srv.sys); lényegében egyszerű matematikai hiba “Srv!SrvOs2FeaListSizeToNt” → Heap based buffer overflow
- WannaCry, NotPetya, Bad Rabbit, Retefe (>200k eszköz, 150 országban, X*100M→4B USD kár; csak 92M GBP az NHS-nek)
- Megtalálható lett volna; de nem is ezért érdekes igazán az eset; hanem:
- ASLR defeat by HAL’s heap

https://research.checkpoint.com/eternalblue-everything-know/


Non-paged pool
Connection
Connection
Connection
Connection according to BUG C
Connection
Srvnet Chunk
Busy Chunk
SRV Chunk
Fea Data
Srvnet Chunk
Busy Chunk
Srvnet Chunk
Busy Chunk
Chunk Created By BUG C
pSrvNetWskStruckt
MDL
SRVNET Chunk
Busy Chunk
Placeholder chunk for Recived Data
Empty
busy chunk (vtable)
free chunk
busy chunk
busy chunk
busy chunk
free chunk
HAL's Heap
static address:
0xffd00000 in 32-bit
0xfffffffffd00000 in 64 bit
Connction
every SMB_COM_TRANSACTION2 Fills part of the free space for FEA structure (OS2FeaForamt)
Zoom-in Srvnet Chunk
0xAA11BB33
Srvnet Header
DWORD pSrvNetWskStruct
MDL
*PMDL Next: | 0x11223344
DWORD StartVa | 0xAA11BB33
SRVNET_RECV (SrvNetWskStruct)
BYTEunknown0[0x50];
PKSPIN_LOCK SpinLock;// 0x50 lock is acq
PVOID HandlerFunction
0x41414141
0x12121212
0xAABBCCDD
Srvnet Header
DWORD pSrvNetWskStruct
MDL
*PMDL Next: | 0x12345678
DWORD StartVa | 0xAABBCCDD
SRVNET_RECV (SrvNetWskStruct)
BYTEunknown0[0x50];
PKSPIN_LOCK SpinLock;// 0x50 lock is acq
PVOID HandlerFunction
Handler Func
Handler Fucntion For closed connections

zero IT Lab


Non-paged pool
Connection
Connection
Connection
Connection
recived data written here
busy chunk (vtable)
FAKE SRVNET_RECV (SrvNetWskStruct)
HandlerFunction (points to shellcode)
SHELLCODE (DoublePulsar)
Srvnet Chunk
Busy Chunk
SRV Chunk
Fea Data
Srvnet Chunk
Busy Chunk
Srvnet Chunk
Busy Chunk
SRV FEA Converted to NTFEA Chunk (Overflowing Next Chunk)
overwrite pNetWskStruckt
overflow
SRVNET Chunk
Busy Chunk
Placeholder chunk for Recived Data ***Empty***
busy chunk
busy chunk
busy chunk
free chunk
HAL's Heap
static address:
0xffd00000 in 32-bit
0xffffffffffd00000 in 64 bit


Connection
recived data written here
busy chunk (vtable)
FAKE SRVNET_RECV (SrvNetWskStruct)
HandlerFunction (points to shellcode)
SHELLCODE (DoublePulsar)
busy chunk
busy chunk
busy chunk
free chunk
HAL's Heap
static address:
0xffd00000 in 32-bit
0xffffffffffd00000 in 64 bit
ow
RVNET
Chunk
Busy
Chunk
Placeholder
chunk
for
Recived
Data
Empty

HAL's Heap

static address:

0xffd00000 in 32-bit

0xffffffffd00000 in 64 bit

Case study: CVE-2019-0708 aka Bluekeep

- Újabb pre-auth zero-click RCE windows ellen
- Forrás: Use-After-Free in Remote-Desktop-Services
- A javitócsomag alapján már bárki reverselhetette és exploitálhatta Május eleje óta.
- 2019 Júniusában >1M eszköz érintett

```
VOID GenerateSMI() // Flip some bits in memory to generate an SMI
{
    DWORD dwSmiValue;
    DWORD dwTriggerAddress;

    struct SmiTriggerInfo *pSmiTriggerInfo = &GLOBAL_SMI_TRIGGER_INFO;
    if (1 << pSmiTriggerInfo->ValueSize != 4)
        goto err;

    /// ... redacted for brevity...
    DWORD size = 1 << pSmiTriggerInfo->ValueSize;
    dwTriggerAddress = dma_map(pSmiTriggerInfo->Address, size, &dwTriggerAddress);

    memcpy(&dwSmiValue, dwTriggerAddress, size);
    dwSmiValue = dwSmiValue & pSmiTriggerInfo->AndMask;
    dwSmiValue = dwSmiValue | pSmiTriggerInfo->OrMask;
    memcpy(dwTriggerAddress, &dwSmiValue, size);
    // ….
}
```

Credits: CTS Labs https://youtu.be/YoMQxb0ZdK0

Ryzenfall-4

- Double fetch leads to stack overflow
- No stack cookies, no ASLR or other exploit mitigations

Credits: CTS Labs https://youtu.be/YoMQxb0ZdK0

„A végső tragédia nem az elnyomás és a kegyetlenség a rossz emberek által, hanem a jó emberek csendje efölött.” ~ Martin Luther King Jr.

- Hiszem, hogy jobbá tudjuk tenni az IT jövőjét. Fogjunk össze és fektessünk be, válasszuk, használjuk és ajánljuk azokat a termékeket és szolgáltatásokat amik tisztelik a társadalmat.

ZERO IT LAB