From MS08-067 to EternalBlue

BsidesMCR, 2017

#whoami

- Denis Isakov a.k.a vinegrep
- Exploit development and RE enthusiast
- Doing pentest as a daily job
- OSCP/OSWP/OSCE/GXPN/GREM/BBQ/WTF and other acronyms

What this talk is about?

- Follow growing exploit development difficulty based on 3 “zero-to-hero” exploits.
- Analyze how these exploits influenced exploit mitigation techniques.
- Should we expect something similar in recent time?

What this talk is not about?

- Not going to blame Microsoft.
- Not going to discuss conspiracy theories why Microsoft patch MS17-010 so quickly.
- Step-by-step exploit development.
- SMB protocol internals


MS08-067
imgflip.com

MS08-067 general info

- “Wormable” vulnerability in SMB that was triggered by malformed RPC request.
- Buffer overflow was in the way how Server service processes the PathName argument to the NetprPathCanonicalize function.
- Problem was in directory traversal sequences result in traversing past the root path.
- Kudos:

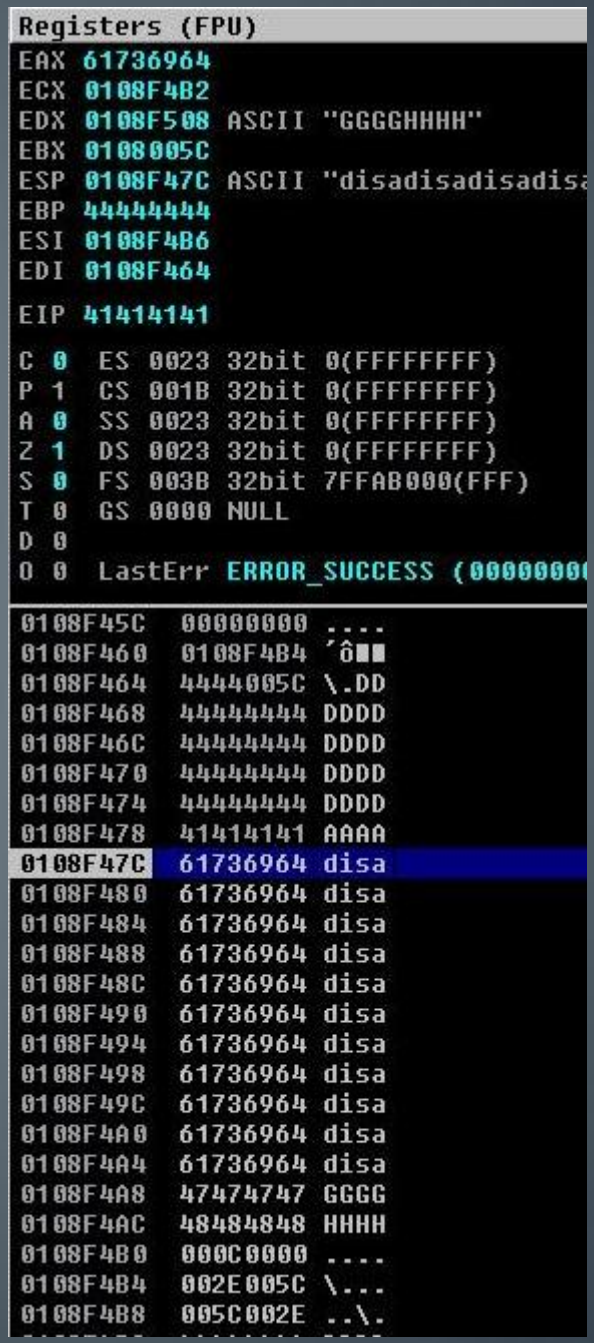
https://blogs.technet.microsoft.com/johnla/2015/09/26/the-inside-story-behind-ms08-067/

https://labs.mwrinfosecurity.com/assets/BlogFiles/hello-ms08-067-my-old-friend.pdf

https://www.mysonicwall.com/sonicalert/searchresults.aspx?ev=article&id=74

https://dontstuffbeansupyournose.com/2008/10/23/looking-at-ms08-067/

MS08-067 exploitation



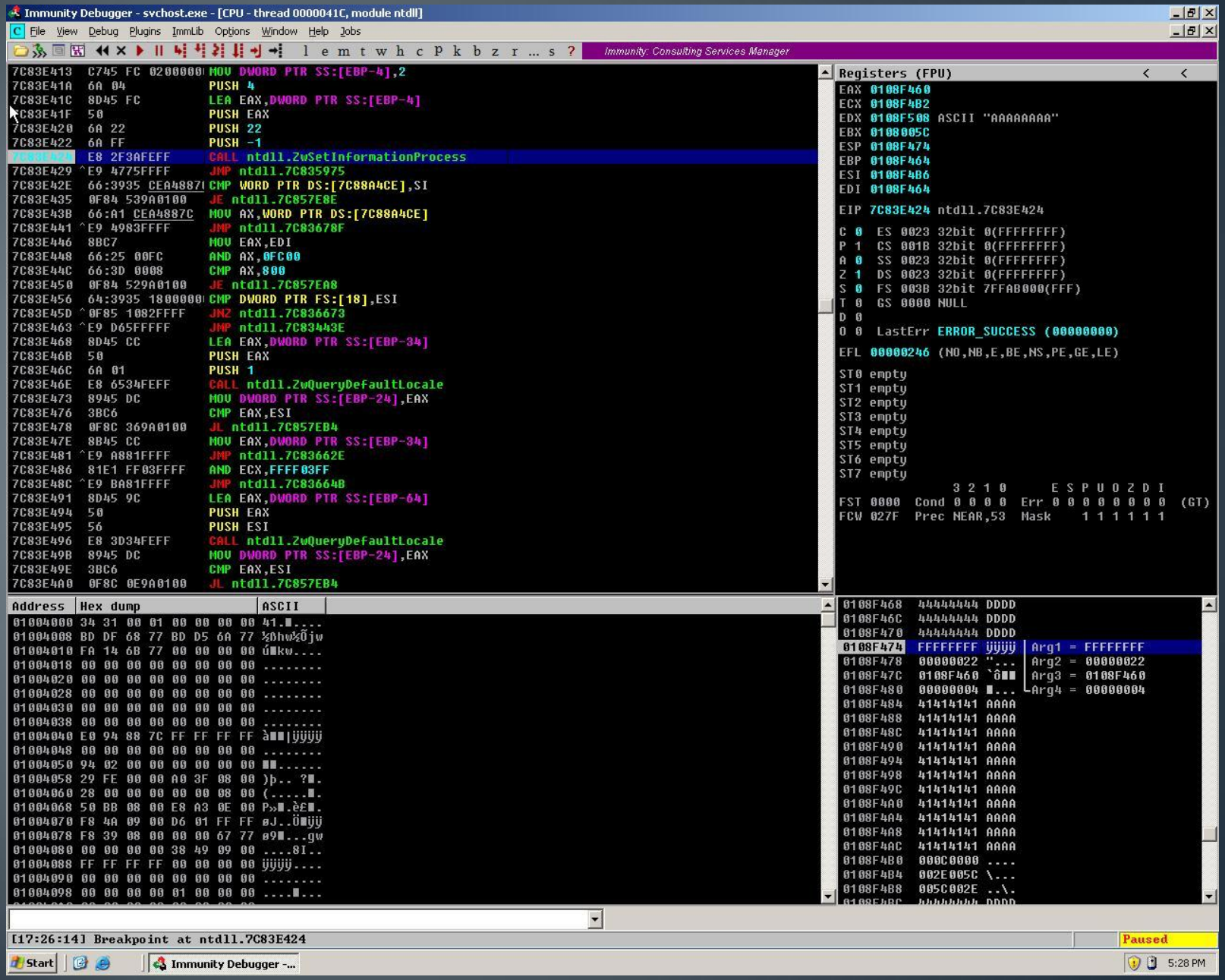
MS08-067 exploitation (2)

- Crash service by overwriting EIP at 18 byte offset.
- ESP and EDX points to our buffer, EAX and EBP was overwritten with our buffer content.
- Overwrite EIP with address of instruction “JMP EDX”, as there is enough space between EIP and address where EDX points.
- Store NOP sled + egghunter after our return address.
- Jump to the end of our controller buffer.
- There is a short jump back instruction that will pass execution to egghunter.
- Egghunter will find final shellcode and PROFIT.

MS08-067 NX bypass

- From XP SP2 and 2003 SP1 we have NX bit enabled. NX bit separates between code and data.
- Excellent paper from skape and Skywing how to bypass it - http://uninformed.org/?v=2&a=4&t=txt
- Idea: disable DEP for process during execution.
- Search memory for opcodes and check whether they are in executable area.
- LdprCheckNXCompatibility in ntdll.dll is used to check whether NX should be enabled and then based on this check call to ZwSetInformationProcess is used to enable/disable DEP.
- Return to controlled buffer, jump back, egg hunt and final shellcode.

MS08-067 NX bypass before call



MS08-067 NX bypass after call

MS08-067 conclusions

- Wrong path validation leads to overflow.
- Easy to exploit vulnerability without any protections.
- Even on NX-enabled versions also exploitable.


MS09-050
imgflip.com

MS09-050 general info

- Found by Laurent Gaffié in 3 seconds of fuzzing SMBv2. Reliable exploit by Piotr Bania, first MSF version by Stephen Fewer.
- EAX register is initialized with a word from the SMB2 packet. In the next instruction our controlled value is used as an array index with one safety check whether the value is NULL. Accidentally there is no check if array index exceeds the number of elements in the array. Later, the location pointed by our controlled value is executed using the call instruction.
- Kudos:

·http://g-laurent.blogspot.be/2009/09/windows-vista7-smb20-negotiate-protocol.html

·http://g-laurent.blogspot.be/2009/10/more-explication-on-cve-2009-3103.html

·http://piotrbania.com/all/articles/smb2_351_packets.pdf

MS09-050 exploitation

```
kd> g
Breakpoint 3 hit
srv2!SrvSnapShotScavengerTimer:
8e32eea0 6a01            push    1
kd> g
Breakpoint 2 hit
srv2!SrvProcCompleteRequest+0xd2:
8e33bb91 ffd0            call    eax
kd> r eax
eax=ffd00d09
kd> g
Breakpoint 0 hit
ffd00d09 5e              pop     esi
kd> u
ffd00d09 5e              pop     esi
ffd00d0a c3              ret
ffd00d0b 7c58            jl      ffd00d65
ffd00d0d 0fb75102        movzx   edx,word ptr [ecx+2]
ffd00d11 0fb77002        movzx   esi,word ptr [eax+2]
ffd00d15 663bd6          cmp     dx,si
ffd00d18 7fed            jg      ffd00d07
ffd00d1a 7c49            jl      ffd00d65
kd> p
ffd00d0a c3              ret
kd> p
84910bf8 00544d42        add     byte ptr [ebp+ecx*2+42h],dl
kd> g
```

MS09-050 conclusions

- Quick to find, fuzzer code from Laurent is straight-forward.
- Requires efforts to make exploitation reliable: Piotr's trampoline vs hard-coded addresses.
- ASLR were bypassed as address from kernel HAL memory region was used.
- NX HAL heap and NonPagedPoolNx from Windows 8


MS17-010
imgflip.com

MS17-010 general info

- Released by Shadow Brokers, developed by Equation Group.
- Complicated exploit requires several steps to trigger vulnerability.

Kudos:

http://risksense.com/_api/filesystem/466/EternalBlue_RiskSense-Exploit-Analysis-and-Port-to-Microsoft-Windows-10_v1_2.pdf

http://blog.trendmicro.com/trendlabs-security-intelligence/ms17-010-eternalblue/

http://blogs.360.cn/360safe/2017/04/17/nsa-eternalblue-smb/

https://github.com/worawit/MS17-010

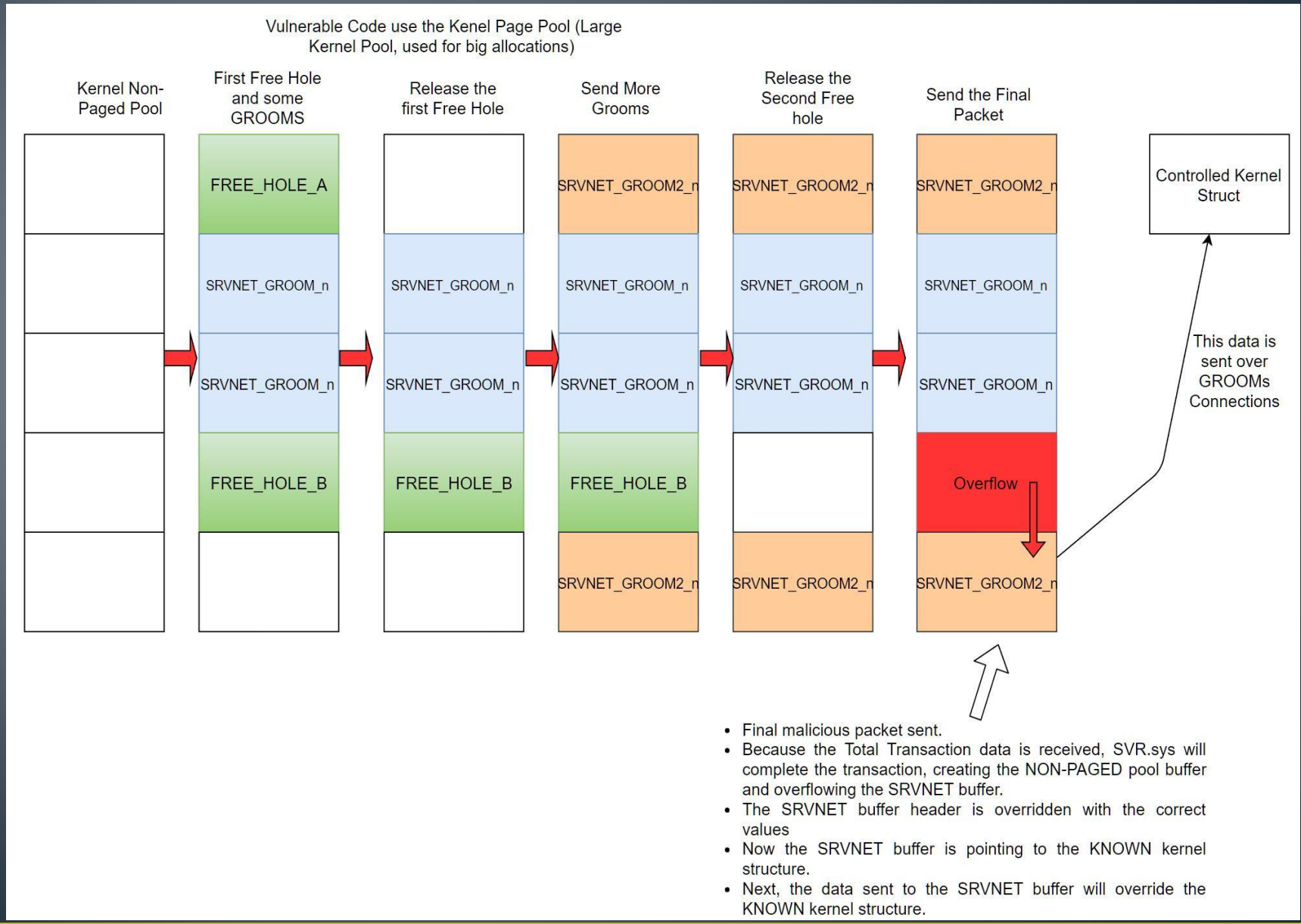
https://packetstormsecurity.com/files/142548/MS17-010-EternalBlue-SMB-Remote-Windows-Kernel-Pool-Corruption.html

MS17-010 general concept

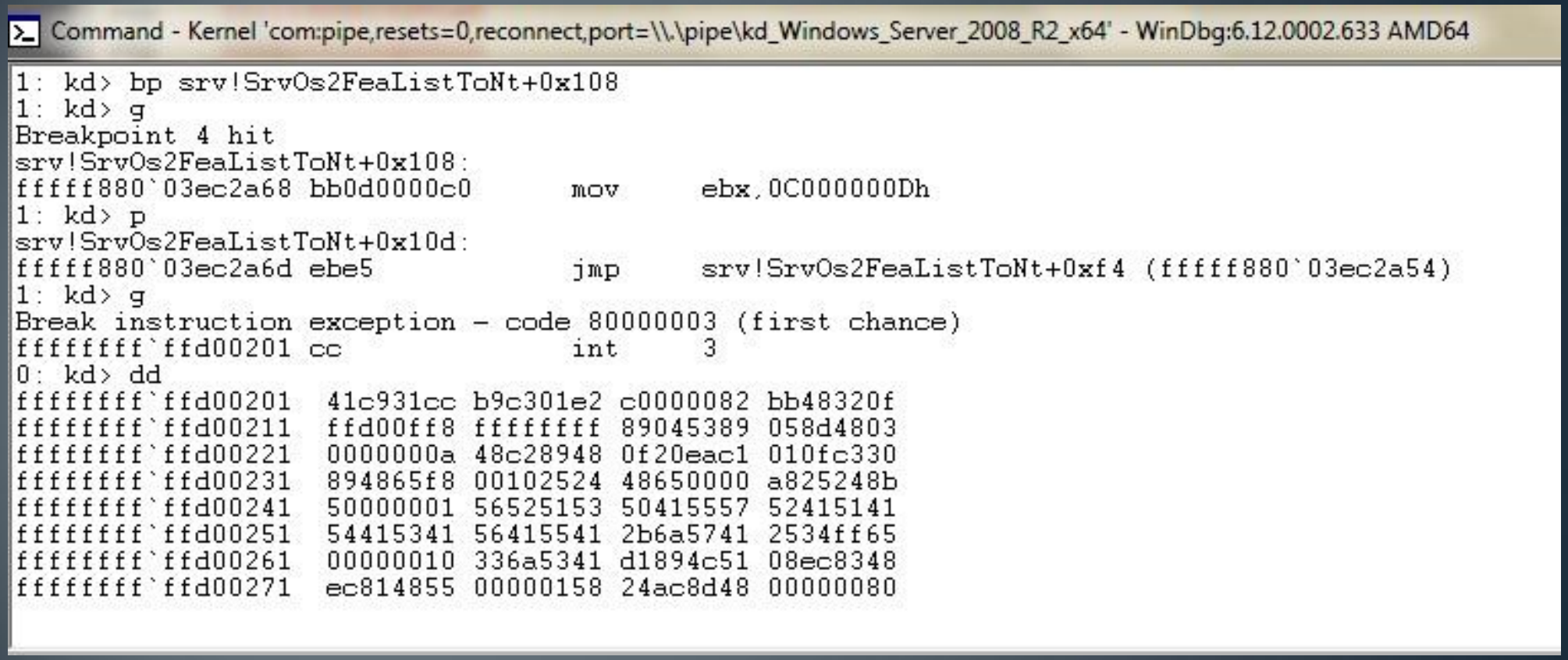
- There is a buffer overflow before second memmove/memcpy operation in Srv!SrvOs2FeaToNt. The size is calculated in Srv!SrvOs2FeaListSizeToNt, with mathematical error where a DWORD is subtracted into a WORD.
- Size of NTFEA list requires Large Kernel Pool Allocation in SRV.sys
- Last iteration overwrite next memory area (SRVNET.sys). To achieve it both buffers should be aligned.
- Kernel Pool should be sprayed.
- Server returns 0xC000000D (status_invalid_parameter).
- Exploit may require several tries.

MS17-010 exploitation

From TrendMicro:



MS17-010 exploitation (2)



MS17-010 conclusions

- Sophisticated exploit which requires several steps to achieve reliable execution.
- Most of kernel structures used by exploit are undocumented.
- In-depth SMB protocol knowledge required.
- Redstone 1 update randomization for page table entries prevents the DEP bypass.
- Redstone 2 update (after the MS17-010 patch), the HAL heap is randomized, defeating the ASLR bypass.

Final thoughts

- Comparing these exploits we can see how Microsoft improved OS security in past few years.
- New exploit mitigation techniques make reliable exploitation much harder to achieve.
- On recently updated Windows 10 I don't expect that something similar will arise in future until new class of vulnerabilities would be introduced.
- Great work by Microsoft to introduce new mitigations based on vulnerability classes from previous OS releases.

Any questions?

Thank you

- Email: vinegrep@protonmail.com
- Twitter: @vinegrep