

CSC427 tutorial: Eternal Blue

file location: <https://bit.ly/2q0dpLC>

April 3, 2018

What you will do

You are given a pcap file of an attempt at an EternalBlue exploit. Look through this file and fill out the submit.txt which should contain the following tasks:

1. Label what part of the exploit process is each set of packets a part of
2. What did the attacker do after they got in?

You will need the following to analyze the file

- WireShark, which is provided in the Kali Linux virtual machine found in [\[utorid\]@dh2020pc01.utm.utoronto.ca:/virtual/csc427_EternalBlue/kali.zip](mailto:[utorid]@dh2020pc01.utm.utoronto.ca:/virtual/csc427_EternalBlue/kali.zip) or you can download it yourself
- packet total, which is a free online pcap analyzer: <https://packettotal.com/>

EternalBlue exploit process

1. **Initial SMB connection and Transaction2:** you can recognize this because it will be sending a lot of bytes which is why it will be processed by SMB.COM.TRANSACTION2, It sends most of the FEA data which is going to be a buffer so that the last bit of it which is sent later can be overflowed by a bug in SMB.
2. **Request and reply for Windows version:** there will be one request that asks version of Windows
3. **Creating the Bug C struct:** This bug C struct upon disconnection will free up space so that when we send the last bit of the FEA data which will then be miscalculated and overflow into one of the structs in the next step
4. **Heap Spraying the non-paged pool with svrnet structs:** continuous connections to Windows creating these structs with a function to handle disconnection, which is what EternalBlue will try to overflow.
5. **Send last bit of FEA data:** This is the last bit of FEA data being sent which will cause the overflow because it will try to convert the list size to read and the actual list size to differ causing the overflow
6. **Disconnection:** the handler functions are called and hopefully one of them is overwritten with our shellcode, this won't show up on packet total

hints:

- It might be possible for steps 3 and 4 to be done multiple times, everything else should only occur once in Packet Total
- In WireShark, the actions are displayed as plain text, if you view the packet bytes in ascii