Tools and Techniques to automate the discovery of Zero Day Vulnerabilities

A.K.A – Fuzzing 101

Agenda

- GEEKZONE
- Overview of fuzzing techniques
- Tutorials on specific open-source fuzzers
- Demonstrations
- DIY fuzzing!

Who are we?

- Mark Rowe, Joe Moore
- IT Security Consultants and Researchers
- Pentest Limited
- Independent IT Security Consultancy

Software Security Assessment

- Information Gathering
- Decomposition of application
- Information Analysis and Planning
- Testing of application components
- Analysis and Reporting

Information Gathering

- Design documentation
- RFC's
- Security requirement specifications
- Data flows
- Source code
- Reverse Engineering
- Informal interviews with key personnel (e.g. developers / product managers)
- Runtime analysis
- Goal is to obtain a detailed picture of the product's composition, which technologies it uses, how it is typically deployed and how it integrates into its environment

Decomposition of an application

- Produce a list of interfaces and features
- Understand how end users and other systems interact with the application
- Identify the application’s attack surface

Information Analysis and Planning

- Develop security test scenarios (thinking like an attacker)
- Understanding of how vulnerabilities get into an application
- Threat/Risk modelling
  - Is the component security critical?
  - Ease of attack.
  - Impact.
  - Is the component or feature enabled by default?
  - Known vulnerabilities in similar products, technologies or components.
  - How potential attackers are likely to view the product.
- Prioritise based on risk

Testing of application components

- Use knowledge obtained from previous phases
- Uncover design and implementation flaws
- Regular progress meetings
- Discuss findings with developers

Analysis and Reporting

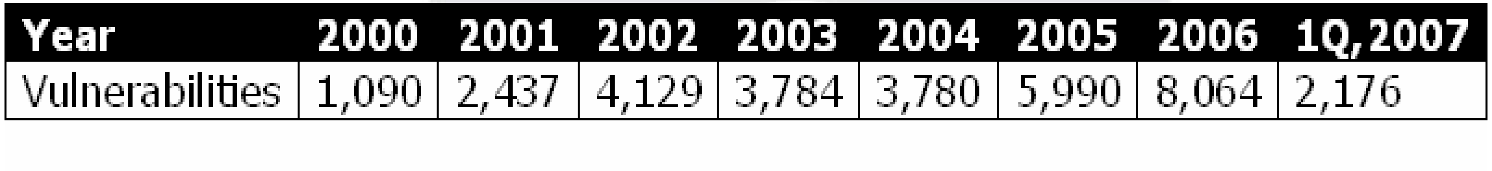
- Bug reports throughout the assessment
- Final written report
  - Details of discovered problems
  - Highlighting possible solutions
  - Prioritised issues
- Presentation
  - Senior management
  - Architects/Developers
- Or sell your 0days ☺

Which box is it in?

- White Box
- Black Box
- Grey Box
- Fuzzing complements more traditional testing

CERT statistics

- **Vulnerabilities identified and cataloged**
- **2000-2007**



Year	2000	2001	2002	2003	2004	2005	2006	1Q,2007
Vulnerabilities	1,090	2,437	4,129	3,784	3,780	5,990	8,064	2,176

Fault Injection

- Understand how the application works
- Enumerate all inputs – “Attack Surface”
- Design tests with input that the application may struggle to handle
- Prioritise tests

What is Fuzzing?

- Sending invalid data to inputs of a program with the purpose of highlighting software defects
- Based on fault injection
- Often automated
- Barton Miller, University of Wisconsin-Madison first person credited with carrying out a rudimentary form of fuzzing (1990)

What can fuzzers discover?

- Buffer overflows
- Integer overflows
- Format string vulnerabilities
- Race condition vulnerabilities
- SQL injection
- Cross Site Scripting (XSS)
- Remote command execution
- Filesystem attacks (reverse traversal, etc)
- Information leaking vulnerabilities
- Memory/Resource exhaustion (DoS)
- Null pointer dereferences

Who uses Fuzzers?

- Security researchers (0days, exploit dev.)
- Software QA
- Developers
- Has gained in popularity over the last few years
- Vendors such as Microsoft have adopted fuzz testing as part of their SDL http://msdn2.microsoft.com/en-us/library/ms995349.aspx

What can you Fuzz?

- Network protocols
- Files
- IPC methods
- Command line arguments
- Environment variables
- APIs
- Network stacks
- Anything that uses a structured data format

Fuzzing process

- Choose your target
- Identify inputs (attack surface)
- Prioritise
- Develop fuzzer or fuzz test cases
- Supply to inputs
- Monitor for exceptions
- Determine exploitability (optional)

Deciding what to Fuzz

- You can’t test everything at once
- Need to be systematic
- Decide which areas to mutate with fuzz data
- Still relies on human expertise!

Runtime analysis

- Processes (ps, ProcessExplorer)
- Network ports (netstat, TCPView, portscanners)
- Network Sniffing (Wireshark, tcpdump)
- Proxies (Paros, WebScarab, ITR)
- Files (Filemon, lsof)
- IPC (OLEView, strace)
- Registry keys (Regmon)
- Debugging (gdb, ollydebug)

Approaches to fuzzing

- Manual
- Semi-automated
- Fully automated

Approaches to automated Fuzzing

- Generate valid inputs from scratch or work from captured inputs (e.g. RFC versus Sniffed traffic)
- Insert fuzz data to produce faulty inputs
- Random fuzzing
- Pre-generated test cases e.g. Protos
- Brute force – bit flipping, raw byte manipulation
- “Intelligent” Fuzzing

Fuzz data

- Bit flipping, random byte changes
- Varying length strings (larger than buffer)
- Large integers, zero, negative integers
- Format strings %n, %25n
- Metacharacters
- ../../../
- <script>alert(‘eek’)</script>
- ‘ OR 1=1 etc.
- , ‘ “)] } NULL
- 0x00

Block based fuzzing

- Originated from Dave Aitel, SPIKE
- Simple and flexible (not Dave! ☺)
- Decompose protocol into length fields and data fields
- Avoids fuzz data being ignored

HTTP POST

POST /path/script.cgi HTTP/1.0

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4

Content-Type: application/x-www-form-urlencoded

Content-Length: 32

postcode=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA&county=cheshire

Spike

- <block_size><data_block><block_size><data_block>… ……..
- Data blocks and sizes (Word, Halfword, Little Endian etc.)
  - s_block_start(), s_block_end(),
  - s_blocksize_halfword_bigendian();
- String data
  - s_string("Hello\r\n")
- Binary data
  - s_binary("41 41")
- Fuzzing
  - s_string_variable
  - s_string_repeat("A",30)

Spike (2)

- Create .spk file
- Sending data
  - generic_send_tcp
  - generic_send_udp
- Lots of examples in audit directory
- Other useful tools like “dcedump”
- Spike Proxy for web apps

Bug Detection

- Segmentation faults
- Debuggers (can sometimes mask the presence of a bug)
- Search for core dumps
- Network port closure
- Processes restarting
- High CPU usage
- Memory usage
- Errors in error logs
- Other activity that you wouldn’t expect during normal operation
- Need to match with test case

Problems you may encounter

- Application becomes slow or unresponsive
- Encryption, checksums, compression, obfuscation
- One bug hides another bug
- Combinations of tests cause problems, single test doesn't trigger the bug!
  - Memory depletion/leaking
  - Process exhaustion
  - Timing issues

Advantages

- Allows fast detection of exploitable security bugs, often serious
- Identify implementation errors not discovered during code reviews or other testing
- Useful when time is limited
- Reusable
- You don’t need source code
- Can make testing of complex environments easier
- Fire and Forget

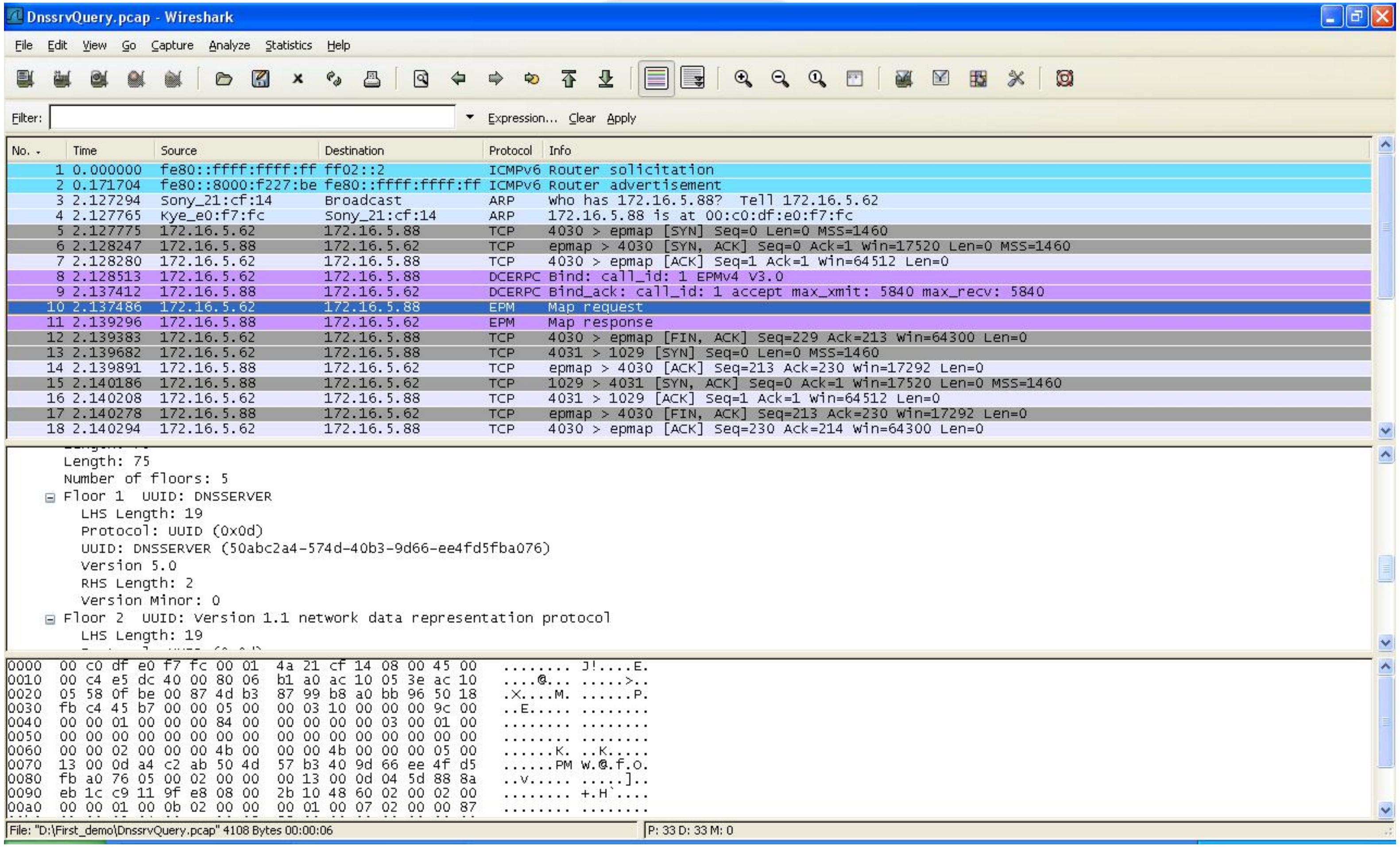
Disadvantages

- Modelling complex protocols can be difficult and time consuming especially if they aren’t documented
- Maintaining state is often difficult
- Not guaranteed to expose all bugs
- Poor code coverage
- Low yield, simple faults
- Tedious to watch!

Fun Stuff!

- Putting it all together
- MS07-029 RPC DNS vulnerability
- Start with *dnscmd.exe*

Fun Stuff!



DnssrvQuery.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

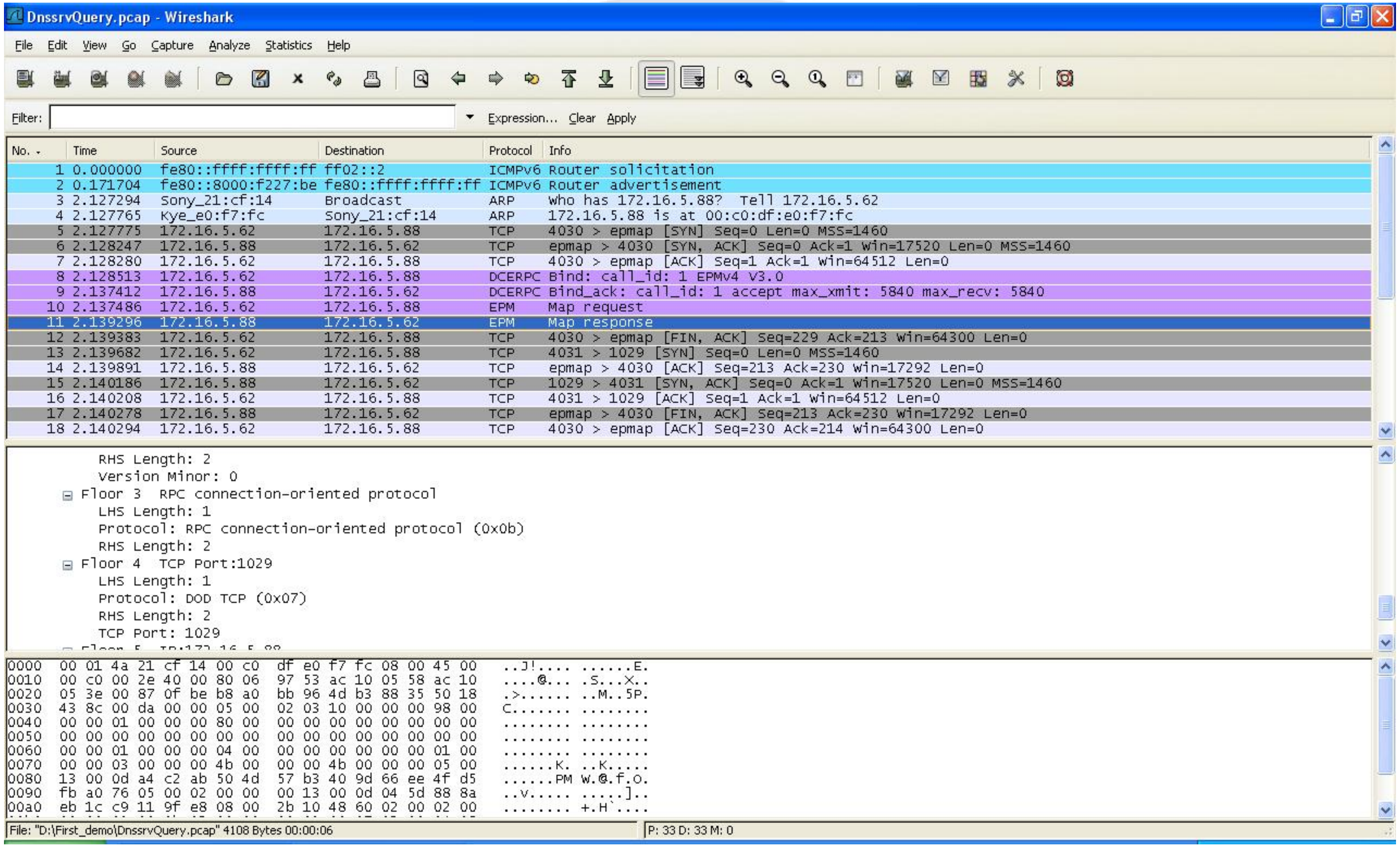
```
No. -  Time      Source                Destination           Protocol Info
     1 0.000000  fe80::ffff:ffff:ff    ff02::2               ICMPv6 Router solicitation
     2 0.171704  fe80::8000:f227:be    fe80::ffff:ffff:ff    ICMPv6 Router advertisement
     3 2.127294  Sony_21:cf:14         Broadcast             ARP    Who has 172.16.5.88?  Tell 172.16.5.62
     4 2.127765  Kye_e0:f7:fc          Sony_21:cf:14         ARP    172.16.5.88 is at 00:c0:df:e0:f7:fc
     5 2.127775  172.16.5.62           172.16.5.88           TCP    4030 > epmap [SYN] Seq=0 Len=0 MSS=1460
     6 2.128247  172.16.5.88           172.16.5.62           TCP    epmap > 4030 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460
     7 2.128280  172.16.5.62           172.16.5.88           TCP    4030 > epmap [ACK] Seq=1 Ack=1 Win=64512 Len=0
     8 2.128513  172.16.5.62           172.16.5.88           DCERPC Bind: call_id: 1 EPMv4 V3.0
     9 2.137412  172.16.5.88           172.16.5.62           DCERPC Bind_ack: call_id: 1 accept max_xmit: 5840 max_recv: 5840
    10 2.137486  172.16.5.62           172.16.5.88           EPM    Map request
    11 2.139296  172.16.5.88           172.16.5.62           EPM    Map response
    12 2.139383  172.16.5.62           172.16.5.88           TCP    4030 > epmap [FIN, ACK] Seq=229 Ack=213 Win=64300 Len=0
    13 2.139682  172.16.5.62           172.16.5.88           TCP    4031 > 1029 [SYN] Seq=0 Len=0 MSS=1460
    14 2.139891  172.16.5.88           172.16.5.62           TCP    epmap > 4030 [ACK] Seq=213 Ack=230 Win=17292 Len=0
    15 2.140186  172.16.5.88           172.16.5.62           TCP    1029 > 4031 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460
    16 2.140208  172.16.5.62           172.16.5.88           TCP    4031 > 1029 [ACK] Seq=1 Ack=1 Win=64512 Len=0
    17 2.140278  172.16.5.88           172.16.5.62           TCP    epmap > 4030 [FIN, ACK] Seq=213 Ack=230 Win=17292 Len=0
    18 2.140294  172.16.5.62           172.16.5.88           TCP    4030 > epmap [ACK] Seq=230 Ack=214 Win=64300 Len=0
```

```
      Length: 75
      Number of floors: 5
    ⊟ Floor 1  UUID: DNSSERVER
        LHS Length: 19
        Protocol: UUID (0x0d)
        UUID: DNSSERVER (50abc2a4-574d-40b3-9d66-ee4fd5fba076)
        Version 5.0
        RHS Length: 2
        Version Minor: 0
    ⊟ Floor 2  UUID: Version 1.1 network data representation protocol
        LHS Length: 19
```

```
0000  00 c0 df e0 f7 fc 00 01  4a 21 cf 14 08 00 45 00   ........ J!....E.
0010  00 c4 e5 dc 40 00 80 06  b1 a0 ac 10 05 3e ac 10   ....@... .....>..
0020  05 58 0f be 00 87 4d b3  87 99 b8 a0 bb 96 50 18   .X....M. ......P.
0030  fb c4 45 b7 00 00 05 00  00 03 10 00 00 00 9c 00   ..E..... ........
0040  00 00 01 00 00 00 84 00  00 00 00 00 03 00 01 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 02 00 00 00 4b 00  00 00 4b 00 00 00 05 00   ......K. ..K.....
0070  13 00 0d a4 c2 ab 50 4d  57 b3 40 9d 66 ee 4f d5   ......PM W.@.f.O.
0080  fb a0 76 05 00 02 00 00  00 13 00 0d 04 5d 88 8a   ..v..... .....]..
0090  eb 1c c9 11 9f e8 08 00  2b 10 48 60 02 00 02 00   ........ +.H`....
00a0  00 00 01 00 0b 02 00 00  00 01 00 07 02 00 00 87   ........ ........
```

File: "D:\First_demo\DnssrvQuery.pcap" 4108 Bytes 00:00:06 | P: 33 D: 33 M: 0

Fun Stuff!



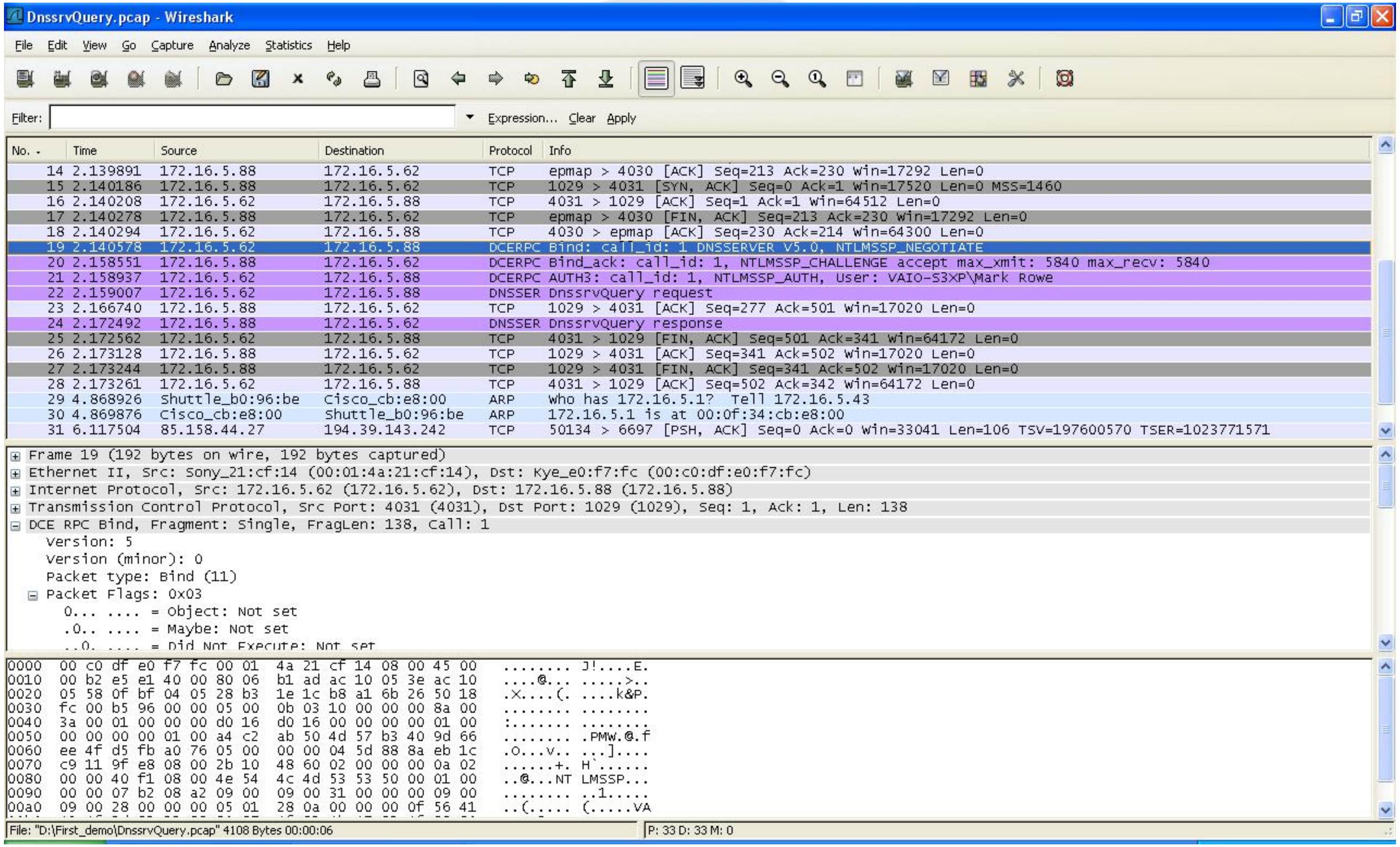
```
No. -  Time      Source              Destination         Protocol Info
    1 0.000000  fe80::ffff:ffff:ff  ff02::2             ICMPv6 Router solicitation
    2 0.171704  fe80::8000:f227:be  fe80::ffff:ffff:ff  ICMPv6 Router advertisement
    3 2.127294  Sony_21:cf:14       Broadcast           ARP    Who has 172.16.5.88?  Tell 172.16.5.62
    4 2.127765  Kye_e0:f7:fc        Sony_21:cf:14       ARP    172.16.5.88 is at 00:c0:df:e0:f7:fc
    5 2.127775  172.16.5.62         172.16.5.88         TCP    4030 > epmap [SYN] Seq=0 Len=0 MSS=1460
    6 2.128247  172.16.5.88         172.16.5.62         TCP    epmap > 4030 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460
    7 2.128280  172.16.5.62         172.16.5.88         TCP    4030 > epmap [ACK] Seq=1 Ack=1 Win=64512 Len=0
    8 2.128513  172.16.5.62         172.16.5.88         DCERPC Bind: call_id: 1 EPMv4 V3.0
    9 2.137412  172.16.5.88         172.16.5.62         DCERPC Bind_ack: call_id: 1 accept max_xmit: 5840 max_recv: 5840
   10 2.137486  172.16.5.62         172.16.5.88         EPM    Map request
   11 2.139296  172.16.5.88         172.16.5.62         EPM    Map response
   12 2.139383  172.16.5.62         172.16.5.88         TCP    4030 > epmap [FIN, ACK] Seq=229 Ack=213 Win=64300 Len=0
   13 2.139682  172.16.5.62         172.16.5.88         TCP    4031 > 1029 [SYN] Seq=0 Len=0 MSS=1460
   14 2.139891  172.16.5.88         172.16.5.62         TCP    epmap > 4030 [ACK] Seq=213 Ack=230 Win=17292 Len=0
   15 2.140186  172.16.5.88         172.16.5.62         TCP    1029 > 4031 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460
   16 2.140208  172.16.5.62         172.16.5.88         TCP    4031 > 1029 [ACK] Seq=1 Ack=1 Win=64512 Len=0
   17 2.140278  172.16.5.88         172.16.5.62         TCP    epmap > 4030 [FIN, ACK] Seq=213 Ack=230 Win=17292 Len=0
   18 2.140294  172.16.5.62         172.16.5.88         TCP    4030 > epmap [ACK] Seq=230 Ack=214 Win=64300 Len=0
```

```
      RHS Length: 2
      Version Minor: 0
  ⊟ Floor 3  RPC connection-oriented protocol
      LHS Length: 1
      Protocol: RPC connection-oriented protocol (0x0b)
      RHS Length: 2
  ⊟ Floor 4  TCP Port:1029
      LHS Length: 1
      Protocol: DOD TCP (0x07)
      RHS Length: 2
      TCP Port: 1029
  ⊟ Floor 5  IP:172.16.5.88
```

```
0000  00 01 4a 21 cf 14 00 c0  df e0 f7 fc 08 00 45 00   ..J!.... ......E.
0010  00 c0 00 2e 40 00 80 06  97 53 ac 10 05 58 ac 10   ....@... .S...X..
0020  05 3e 00 87 0f be b8 a0  bb 96 4d b3 88 35 50 18   .>...... ..M..5P.
0030  43 8c 00 da 00 00 05 00  02 03 10 00 00 00 98 00   C....... ........
0040  00 00 01 00 00 00 80 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 01 00 00 00 04 00  00 00 00 00 00 00 01 00   ........ ........
0070  00 00 03 00 00 00 4b 00  00 00 4b 00 00 00 05 00   ......K. ..K.....
0080  13 00 0d a4 c2 ab 50 4d  57 b3 40 9d 66 ee 4f d5   ......PM W.@.f.O.
0090  fb a0 76 05 00 02 00 00  00 13 00 0d 04 5d 88 8a   ..v..... .....]..
00a0  eb 1c c9 11 9f e8 08 00  2b 10 48 60 02 00 02 00   ........ +.H`....
```

File: "D:\First_demo\DnssrvQuery.pcap" 4108 Bytes 00:00:06 | P: 33 D: 33 M: 0

Fun Stuff!


DnssrvQuery.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

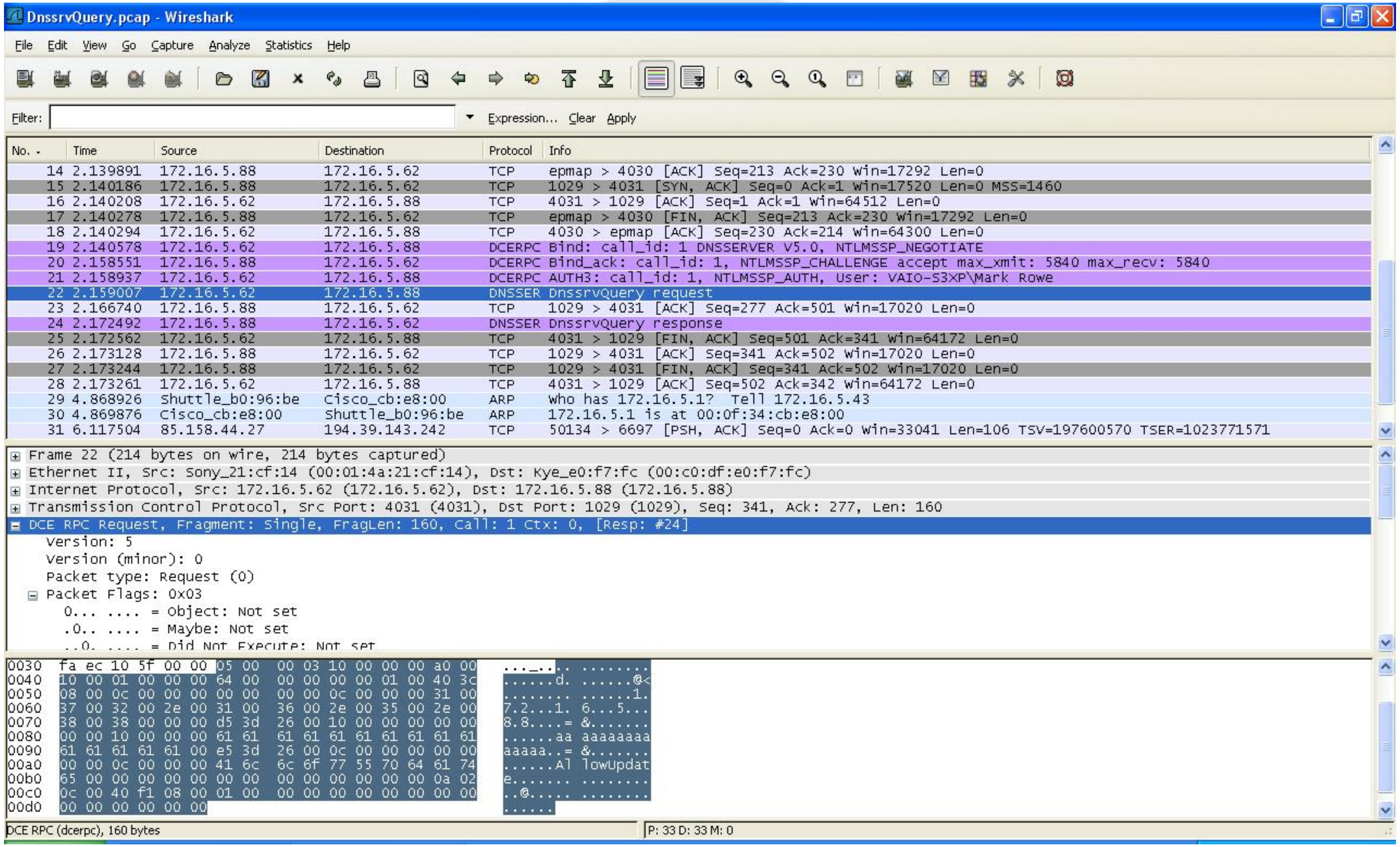
```
No. -   Time        Source              Destination         Protocol Info
     14 2.139891    172.16.5.88         172.16.5.62         TCP      epmap > 4030 [ACK] Seq=213 Ack=230 Win=17292 Len=0
     15 2.140186    172.16.5.88         172.16.5.62         TCP      1029 > 4031 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460
     16 2.140208    172.16.5.62         172.16.5.88         TCP      4031 > 1029 [ACK] Seq=1 Ack=1 Win=64512 Len=0
     17 2.140278    172.16.5.88         172.16.5.62         TCP      epmap > 4030 [FIN, ACK] Seq=213 Ack=230 Win=17292 Len=0
     18 2.140294    172.16.5.62         172.16.5.88         TCP      4030 > epmap [ACK] Seq=230 Ack=214 Win=64300 Len=0
     19 2.140578    172.16.5.62         172.16.5.88         DCERPC   Bind: call_id: 1 DNSSERVER V5.0, NTLMSSP_NEGOTIATE
     20 2.158551    172.16.5.88         172.16.5.62         DCERPC   Bind_ack: call_id: 1, NTLMSSP_CHALLENGE accept max_xmit: 5840 max_recv: 5840
     21 2.158937    172.16.5.62         172.16.5.88         DCERPC   AUTH3: call_id: 1, NTLMSSP_AUTH, User: VAIO-S3XP\Mark Rowe
     22 2.159007    172.16.5.62         172.16.5.88         DNSSER   DnssrvQuery request
     23 2.166740    172.16.5.88         172.16.5.62         TCP      1029 > 4031 [ACK] Seq=277 Ack=501 Win=17020 Len=0
     24 2.172492    172.16.5.88         172.16.5.62         DNSSER   DnssrvQuery response
     25 2.172562    172.16.5.62         172.16.5.88         TCP      4031 > 1029 [FIN, ACK] Seq=501 Ack=341 Win=64172 Len=0
     26 2.173128    172.16.5.88         172.16.5.62         TCP      1029 > 4031 [ACK] Seq=341 Ack=502 Win=17020 Len=0
     27 2.173244    172.16.5.88         172.16.5.62         TCP      1029 > 4031 [FIN, ACK] Seq=341 Ack=502 Win=17020 Len=0
     28 2.173261    172.16.5.62         172.16.5.88         TCP      4031 > 1029 [ACK] Seq=502 Ack=342 Win=64172 Len=0
     29 4.868926    Shuttle_b0:96:be    Cisco_cb:e8:00      ARP      Who has 172.16.5.1?  Tell 172.16.5.43
     30 4.869876    Cisco_cb:e8:00      Shuttle_b0:96:be    ARP      172.16.5.1 is at 00:0f:34:cb:e8:00
     31 6.117504    85.158.44.27        194.39.143.242      TCP      50134 > 6697 [PSH, ACK] Seq=0 Ack=0 Win=33041 Len=106 TSV=197600570 TSER=1023771571
```

```
⊞ Frame 19 (192 bytes on wire, 192 bytes captured)
⊞ Ethernet II, Src: Sony_21:cf:14 (00:01:4a:21:cf:14), Dst: Kye_e0:f7:fc (00:c0:df:e0:f7:fc)
⊞ Internet Protocol, Src: 172.16.5.62 (172.16.5.62), Dst: 172.16.5.88 (172.16.5.88)
⊞ Transmission Control Protocol, Src Port: 4031 (4031), Dst Port: 1029 (1029), Seq: 1, Ack: 1, Len: 138
⊟ DCE RPC Bind, Fragment: Single, FragLen: 138, Call: 1
    Version: 5
    Version (minor): 0
    Packet type: Bind (11)
  ⊟ Packet Flags: 0x03
      0... .... = Object: Not set
      .0.. .... = Maybe: Not set
      ..0. .... = Did Not Execute: Not set
```

```
0000  00 c0 df e0 f7 fc 00 01  4a 21 cf 14 08 00 45 00   ........ J!....E.
0010  00 b2 e5 e1 40 00 80 06  b1 ad ac 10 05 3e ac 10   ....@... .....>..
0020  05 58 0f bf 04 05 28 b3  1e 1c b8 a1 6b 26 50 18   .X....(. ....k&P.
0030  fc 00 b5 96 00 00 05 00  0b 03 10 00 00 00 8a 00   ........ ........
0040  3a 00 01 00 00 00 d0 16  d0 16 00 00 00 00 01 00   :....... ........
0050  00 00 00 00 01 00 a4 c2  ab 50 4d 57 b3 40 9d 66   ........ .PMW.@.f
0060  ee 4f d5 fb a0 76 05 00  00 00 04 5d 88 8a eb 1c   .O...v.. ...]....
0070  c9 11 9f e8 08 00 2b 10  48 60 02 00 00 00 0a 02   ......+. H`......
0080  00 00 40 f1 08 00 4e 54  4c 4d 53 53 50 00 01 00   ..@...NT LMSSP...
0090  00 00 07 b2 08 a2 09 00  09 00 31 00 00 00 09 00   ........ ..1.....
00a0  09 00 28 00 00 00 05 01  28 0a 00 00 00 0f 56 41   ..(..... (.....VA
```

File: "D:\First_demo\DnssrvQuery.pcap" 4108 Bytes 00:00:06 P: 33 D: 33 M: 0

Fun Stuff!


DnssrvQuery.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. -  Time      Source             Destination        Protocol Info
    14 2.139891  172.16.5.88        172.16.5.62        TCP      epmap > 4030 [ACK] Seq=213 Ack=230 Win=17292 Len=0
    15 2.140186  172.16.5.88        172.16.5.62        TCP      1029 > 4031 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460
    16 2.140208  172.16.5.62        172.16.5.88        TCP      4031 > 1029 [ACK] Seq=1 Ack=1 Win=64512 Len=0
    17 2.140278  172.16.5.88        172.16.5.62        TCP      epmap > 4030 [FIN, ACK] Seq=213 Ack=230 Win=17292 Len=0
    18 2.140294  172.16.5.62        172.16.5.88        TCP      4030 > epmap [ACK] Seq=230 Ack=214 Win=64300 Len=0
    19 2.140578  172.16.5.62        172.16.5.88        DCERPC   Bind: call_id: 1 DNSSERVER V5.0, NTLMSSP_NEGOTIATE
    20 2.158551  172.16.5.88        172.16.5.62        DCERPC   Bind_ack: call_id: 1, NTLMSSP_CHALLENGE accept max_xmit: 5840 max_recv: 5840
    21 2.158937  172.16.5.62        172.16.5.88        DCERPC   AUTH3: call_id: 1, NTLMSSP_AUTH, User: VAIO-S3XP\Mark Rowe
    22 2.159007  172.16.5.62        172.16.5.88        DNSSER   DnssrvQuery request
    23 2.166740  172.16.5.88        172.16.5.62        TCP      1029 > 4031 [ACK] Seq=277 Ack=501 Win=17020 Len=0
    24 2.172492  172.16.5.88        172.16.5.62        DNSSER   DnssrvQuery response
    25 2.172562  172.16.5.62        172.16.5.88        TCP      4031 > 1029 [FIN, ACK] Seq=501 Ack=341 Win=64172 Len=0
    26 2.173128  172.16.5.88        172.16.5.62        TCP      1029 > 4031 [ACK] Seq=341 Ack=502 Win=17020 Len=0
    27 2.173244  172.16.5.88        172.16.5.62        TCP      1029 > 4031 [FIN, ACK] Seq=341 Ack=502 Win=17020 Len=0
    28 2.173261  172.16.5.62        172.16.5.88        TCP      4031 > 1029 [ACK] Seq=502 Ack=342 Win=64172 Len=0
    29 4.868926  Shuttle_b0:96:be   Cisco_cb:e8:00     ARP      Who has 172.16.5.1?  Tell 172.16.5.43
    30 4.869876  Cisco_cb:e8:00     Shuttle_b0:96:be   ARP      172.16.5.1 is at 00:0f:34:cb:e8:00
    31 6.117504  85.158.44.27       194.39.143.242     TCP      50134 > 6697 [PSH, ACK] Seq=0 Ack=0 Win=33041 Len=106 TSV=197600570 TSER=1023771571
```

```
⊞ Frame 22 (214 bytes on wire, 214 bytes captured)
⊞ Ethernet II, Src: Sony_21:cf:14 (00:01:4a:21:cf:14), Dst: Kye_e0:f7:fc (00:c0:df:e0:f7:fc)
⊞ Internet Protocol, Src: 172.16.5.62 (172.16.5.62), Dst: 172.16.5.88 (172.16.5.88)
⊞ Transmission Control Protocol, Src Port: 4031 (4031), Dst Port: 1029 (1029), Seq: 341, Ack: 277, Len: 160
⊟ DCE RPC Request, Fragment: Single, FragLen: 160, Call: 1 Ctx: 0, [Resp: #24]
    Version: 5
    Version (minor): 0
    Packet type: Request (0)
  ⊟ Packet Flags: 0x03
      0... .... = Object: Not set
      .0.. .... = Maybe: Not set
      ..0. .... = Did Not Execute: Not set
```

```
0030  fa ec 10 5f 00 00 05 00  00 03 10 00 00 00 a0 00   ..._.... ........
0040  10 00 01 00 00 00 64 00  00 00 00 00 01 00 40 3c   ......d. ......@<
0050  08 00 0c 00 00 00 00 00  00 00 0c 00 00 00 31 00   ........ ......1.
0060  37 00 32 00 2e 00 31 00  36 00 2e 00 35 00 2e 00   7.2...1. 6...5...
0070  38 00 38 00 00 00 d5 3d  26 00 10 00 00 00 00 00   8.8....= &.......
0080  00 00 10 00 00 00 61 61  61 61 61 61 61 61 61 61   ......aa aaaaaaaa
0090  61 61 61 61 61 00 e5 3d  26 00 0c 00 00 00 00 00   aaaaa..= &.......
00a0  00 00 0c 00 00 00 41 6c  6c 6f 77 55 70 64 61 74   ......Al lowUpdat
00b0  65 00 00 00 00 00 00 00  00 00 00 00 00 00 0a 02   e....... ........
00c0  0c 00 40 f1 08 00 01 00  00 00 00 00 00 00 00 00   ..@..... ........
00d0  00 00 00 00 00 00                                  ......
```

DCE RPC (dcerpc), 160 bytes | P: 33 D: 33 M: 0

User supplied string

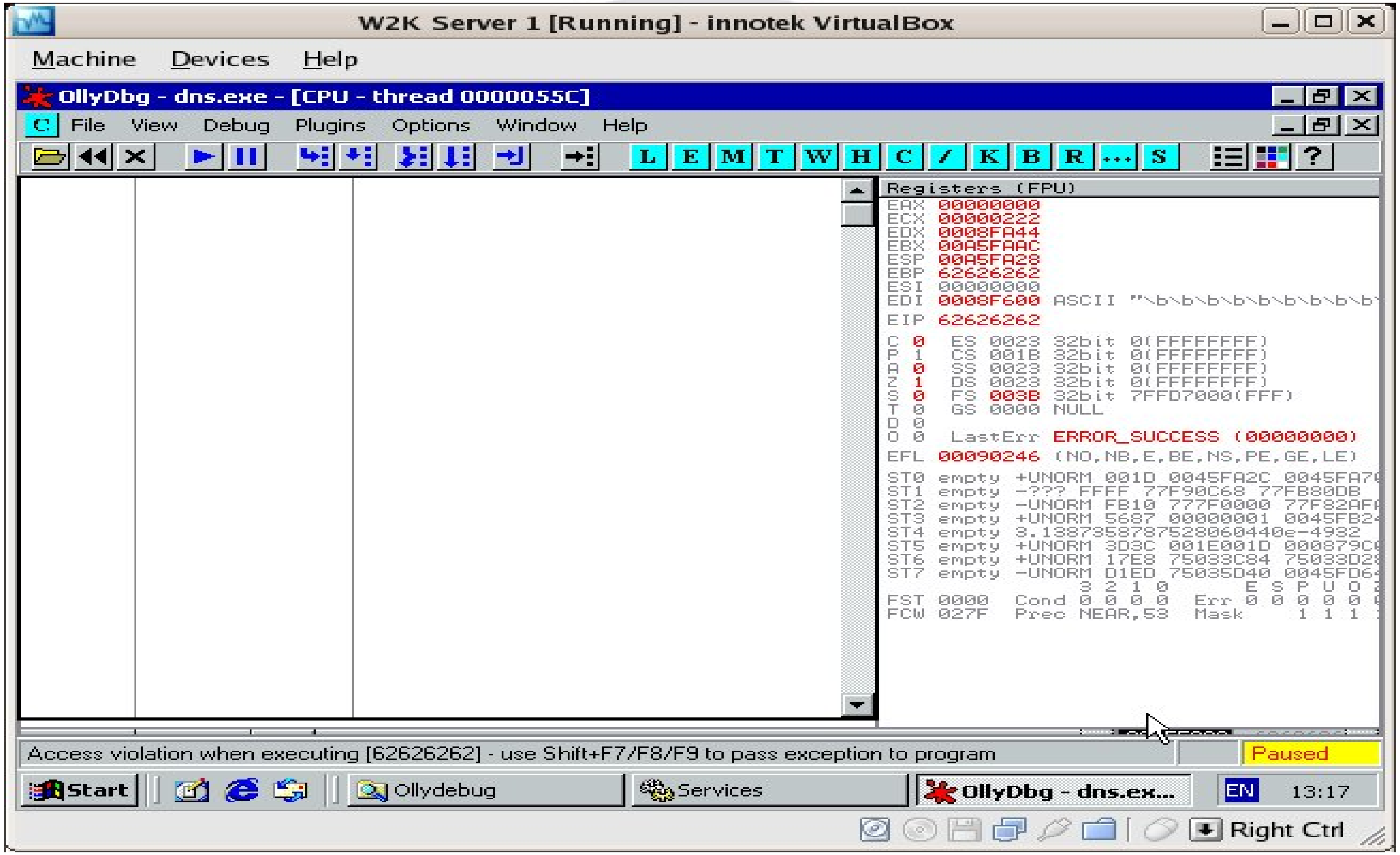
DCE Bind (DNS)

```
                05 00 0b 03 10 00 00 00 48 00  ................
0040   00 00 01 00 00 00 d0 16 d0 16 00 00 00 00 01 00  :...............
0050   00 00 00 00 01 00 a4 c2 ab 50 4d 57 b3 40 9d 66  .........PMW.@.f
0060   ee 4f d5 fb a0 76 05 00 00 00 04 5d 88 8a eb 1c  .O...v.....]....
0070   c9 11 9f e8 08 00 2b 10 48 60 02 00 00 00
```

DnssrvQuery ***a0 00*** packet size

```
                05 00 00 03 10 00 00 00 ***a0 00***  ..._...........
0040   00 00 01 00 00 00 64 00 00 00 00 00 01 00 40 3c  ......d.......@<
0050   08 00 0c 00 00 00 00 00 00 00 0c 00 00 00 31 00  ..............1.
0060   37 00 32 00 2e 00 31 00 36 00 2e 00 35 00 2e 00  7.2...1.6...5...
0070   38 00 38 00 00 00 d5 3d 26 00 10 00 00 00 00 00  8.8....=&.......
0080   00 00 10 00 00 00 61 61 61 61 61 61 61 61 61 61  ......aaaaaaaaaa
0090   61 61 61 61 61 00 e5 3d 26 00 0c 00 00 00 00 00  aaaaa..=&.......
00a0   00 00 0c 00 00 00 41 6c 6c 6f 77 55 70 64 61 74  ......AllowUpdat
00b0   65 00                                            ......
```

EIP: 62626262!


W2K Server 1 [Running] - innotek VirtualBox
Machine Devices Help
OllyDbg - dns.exe - [CPU - thread 0000055C]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Start
Ollydebug
Services
OllyDbg - dns.ex...
EN 13:17
Right Ctrl

```
Registers (FPU)
EAX 00000000
ECX 00000222
EDX 0008FA44
EBX 00A5FAAC
ESP 00A5FA28
EBP 62626262
ESI 00000000
EDI 0008F600 ASCII "\b\b\b\b\b\b\b\b\b\b

EIP 62626262

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFD7000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00090246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty +UNORM 001D 0045FA2C 0045FA7
ST1 empty -??? FFFF 77F90C68 77FB80DB
ST2 empty -UNORM FB10 777F0000 77F82AF
ST3 empty +UNORM 5687 00000001 0045FB2
ST4 empty 3.1387358787528060440e-4932
ST5 empty +UNORM 3D3C 001E001D 000879C
ST6 empty +UNORM 17E8 75033C84 75033D2
ST7 empty -UNORM D1ED 75035D40 0045FD6
               3 2 1 0      E S P U O
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0
FCW 027F  Prec NEAR,53  Mask    1 1 1
```

```
Access violation when executing [62626262] - use Shift+F7/F8/F9 to pass exception to program
```

Paused

Open Source Fuzzers

- SPIKE
- Autodafé
- PEACH
- And many, many more…

Commercial Fuzzers

- BeStorm protocol fuzzer
- OULU commercial fuzzer
- Codenomicon
- Mu-4000
- BreakingPoint

Homegrown Fuzzers

- Specific purpose
- Often quick, not very comprehensive
- Modify other (open source) fuzzers
- Not really something you could sell(!)

Autodafé

- An act of software *“torture”*
- Similar to SPIKE
  - Block based scripts
- File and Network fuzzing
- Monitoring tools built in
- Weighting attacks (very cool!)

Autodafé

```
block_begin("rmf_header");
hex(2e 52 4d 46);
block_size_b32("rmf_header"); /* chunk size */
hex(00 01); /* chunk version */
hex(00 00 00 00); /* file version (0) */
hex(00 00 00 06); /* number of headers */
block_end("rmf_header");
```

PEACH

- Written by Michael Eddington (IOActive)
- Python based framework
    - Cross-platform
- Can fuzz just about anything!
- Syntax and concepts needs to be learnt
- Easily re-usable code

PEACH (continued)

- Four components to a PEACH script
  - Generators
  - Transformers
  - Protocols
  - Publishers
- Sounds complicated, really isn’t!

File Fuzzing

- Targets
  - Common application formats
  - One format many targets
- Manual approach
  - Create a series of corrupted files (hex editor for binary protocols)
  - Open each file with the application
  - Very slow
  - Boring!

Automated File Fuzzing

- Binary file formats can be complicated
- In depth knowledge may be required
- Often not documented
- Makes intelligent fuzzing difficult
- Good news is dumb fuzzing often yields results ☺
- Randomly overwrite bytes or perform bit flipping
- File headers are often a good place to start

DIY fuzzing

- Modified BackTrack live-cd
- Real world example – RealPlayer 10 .smil file stack overflow
- Suggest you use SPIKEfile
- Sample .smil file in /usr/local/examples

BackTrack CD

- Should autoboot
  - Possible problems :
    - IRQ - bt irqpoll
    - PCMCIA – bt *nopcmcia*
    - ACPI – bt *acpi=off*
    - DHCP – bt *nodhcp*

Questions?

Useful resources

- http://www.threatmind.net/secwiki/FuzzingTools
- http://www.owasp.org/index.php/Fuzzing
- http://www.owasp.org/index.php/OWASP_Testing_Guide_Appendix_C:_Fuzz_Vectors
- http://www.immunitysec.com/downloads/advantages_of_block_based_analysis.pdf
- http://www.immunitysec.com/resources-freesoftware.shtml
- http://autodafe.sourceforge.net
- Fuzzing mailing list http://www.whitestar.linuxbox.org/mailman/listinfo/fuzzing