DEEPSEC 2012

Multilayer Fuzzing With Evader

OLLI-PEKKA NIEMI

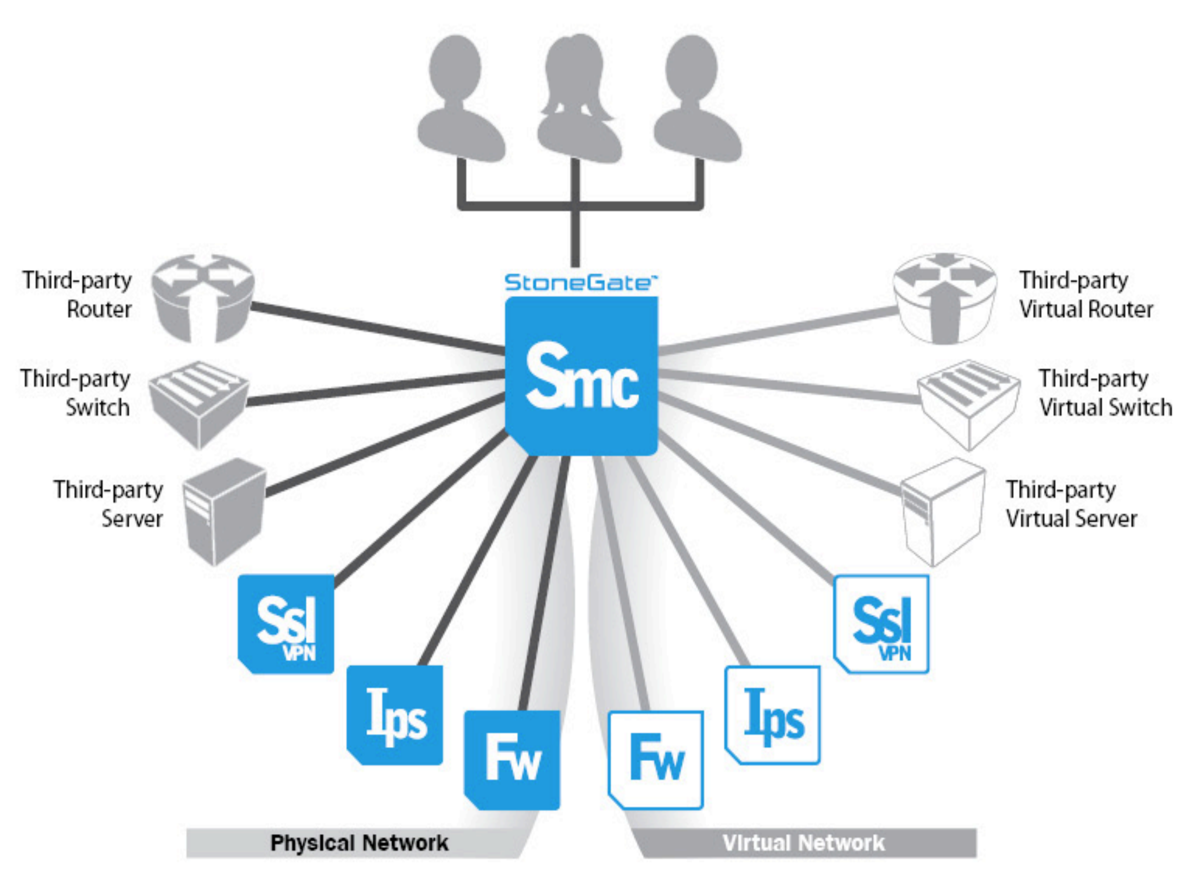
STONESOFT

Who’s Talking

- Olli-Pekka Niemi
    - Pentest @KPMG 1997-2000
    - R&D, Product mgmt, Pre-sales 2000-future @Stonesoft
  - Job:
    - Head of Stonesoft Vulnerability Analysis Group
      - Lead Stonesoft’s signature writing R&D Team
    - Involved NSS/ICSA/CC/other tests/certification processes
    - Analyze threats and write signatures for Stonesoft’s deep inspection products IPS,IDS,FW
    - Research evasions to harden Stonesoft products

Who's Talking

- Stonesoft
- Globally operating security appliance and software vendor from Finland



What is Evader?

- A tool to test (NG)?((I[DP]S|FW)) protocol analysis and reassembly capabilities by applying evasions to attacks
- Does not use simulations, use real attack against real target
- Simple Test Scenario: #shell does not lie
  - Send attack, if we got the shell back, evasion was successful and DUT failed

```
root@evader:~/evader_0_9_8_559#  ./evader

Evader - The IPS testing tool
Copyright (C) 2010-2012 Stonesoft Corporation. All rights reserved.

Usage: ./evader [options]
Network configuration:
        --if=<name>                Name of the interface (e.g. eth0).
        --src_ip=<ip>              Source IP address.
        --src_port=<port>          Source port, if applicable. Defaults to random.
        --src_mask=<mask>          IPv4 source netmask, if applicable. Defaults to 24.
        --gw=<ip>                  Gateway IP if needed. Defaults to unset.
        --dst_ip=<ip>              Destination IP address.
        --dst_port=<ip>            Destination port.
Attack configuration:
        --attacks | -a             List supported attacks and exit.
        --info=<name>              Print more detailed information.
        --attack=<name>            Select the attack to use.
        --clean                    Send only a non-malicious payload to check victim availability.
        --shell                    Send a payload that opens a command shell.
        --fireworks                Send a payload that displays something on the victim hosts screen.
        --obfuscate                Set all available obfuscation flags in the exploit.
        --extra=<str>              Attack specific optional extra configuration.
Evasion configuration:
        --evasions | -e            List supported evasions and exit. Attack must be selected.
        --evasion=<name>           Use named evasion. Attack must be selected.
Other:
        --version | -v             Print version and exit.
        --cfg_file=<filename>      Read configuration also from a configuration file.
        --autoclose                No interaction, automatically close shells.
        --shell_tcp                If defined, shell control channel is opened to a TCP socket instead of standard IO.
        --summary | -s             Print a summary before exiting.
        --verifydelay=<num>        Milliseconds to wait before verifying attack result. Defaults to 500.
        --randseed=<str>           Set the random seed to use.
        --record=<fname>           Record all generated traffic in PCAP format to file <fname>.
        --enable_mmap              Enable MMAPed raw sockets.

root@evader:~/evader_0_9_8_559#
```

What is Mongbat?

- “A cross between a monkey and a bat. **Mongbats** are strong and vicious, attacking their victims with strength and without fear.” – Ultimawiki
- An accompanier tool for evader
  - The Mongbat is the brains for the evader allowing users to build test cases that run the evader over and over again until the weaknesses of the middle-box are found.

```
mongbat.rb - uses solo/dual/random evasions to attack target host
Options:
        --mode=(solo|dual|triple|random)                Mode of attack, solo for each supported evasion with some options, dual for combinations of two and random (default) for random options
        --attack=(conficker|http_phpbb_highlight|rdp_dos)       Attack to use (default is conficker)
        --iface=<interface>                                     Interface from which the attacks should originate
        --attacker=<src ip>                                     Starting source IP for attackers; First worker will use this and if more are configured, they will use the following
        --victim=<dst ip>                                       Destination IP for the attackers; Expecting correct host and vulnerable service on default port
        --mask=<netmask or prefix>                              Netmask for IPv4 in CIDR notation, prefix length for IPv6
        --gw=<gw ip>                                                    Gateway address if the victim is not in the local network (defaults to empty)
        --time=<time in seconds, default 60>                    Time in seconds - stop attacking once time is up (--mode=random)
        --workers=<worker count, default 1>                     Use this many workers (and source IP addresses) to do the attacking
        --use_evasions=<evasion>(,evasion)*                     Use only these evasions
        --disable_evasions=<evasion>(,evasion)*                 Do not use these evasions
        --check_victim=(true|false)                             Check that victim allows legal traffic without evasions before attacking (default true)
        --record=<recdir>                                       Record the attacks to dirname in pcap format
        --min_evasions=<min evasions>                           Minimum evasions for random mode (default: 1)
        --max_evasions=<max evasions>                           Maximum evasions for random mode (0 for unlimited) (default: 0)
        --index=<begin(-end)?>                                  Start and optional end index for solo and dual mode
        --stop_on_success                                       Stop if an attack is successful
        --payload=<shell|..>                                    Payloads types. Defaults to 'shell'. Some payload cannot be checked for success
        --stages=<true|false>                                   Use stages when available. Defaults to true
        --all_options=<true|false>                              Enable use of all options (dangerous). Defaults to false
        --validator=<validator>(,validator)*            Use this ruby code validator to evaluate whether the combination is valid (dangerous)
        --randseed=<randseed>                                   This sets the base64 randseed to allow for some repeatability
        --passthrough                                                   Pass remaining unknown arguments directly to evader
Example:
mongbat.rb --attack=conficker --iface=eth1 --attacker=10.0.0.100 --workers=16 --victim=10.0.0.3 --time=3600
root@evader:~/evader_0_9_8_559#
```

Evasion

TO MAKE ATTACKS UNNOTICED BY IPS/NGFW DEVICES, THE ATTACKER CAN USE SO-CALLED EVASION TECHNIQUES, WAYS TO OBFUSCATE THE HARMFUL CONTENT.

EVASION

- The reason that evasions work is the old robustness principle stated by Jon Postel in RFC793

“be conservative in what you do,

be liberal in what you accept from

others”.

Previous Academic Research on Evasions

- Ptacek, Newsham: “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection”, 1998.
- Raffael Marty, Thor – A tool to test intrusion detection systems by variaton of attacks, 2002
- A. Samuel Gorton and Terrence G. Champion, Combining Evasion Techniques to Avoid Network Intrusion Detection Systems, 2004
- Giovanni Vigna William Robertson Davide Balzarotti : Testing Network-based Intrusion Detection Signatures Using Mutant Exploits, 2004
- Shai Rubin, Somesh Jha, and Barton P. Miller: Automatic Generation and Analysis of NIDS Attacks, 2004
- Varghese, et al., Detecting Evasion Attacks at High Speeds without Reassembly, Sigcomm, 2006.

Community Work on Evasions

- Horizon, Defeating Sniffers and Intrusion Detection Systems, Phrack Magazine Issue 54, 1998, article 10 of 12.
- Rain Forest Puppy: A look at whisker's anti-IDS tactics,1999
- NIDS Evasion Method named "SeolMa", Phrack 57, Phile 0x03, 2001
- Daniel J. Roelker , HTTP IDS Evasions Revisited, 2003
- Brian Caswell, H D Moore, Thermoptic Camouflage:Total IDS Evasion, BlackHat, 2006
- Renaud Bidou: IPS Shortcomings, BlackHat 2006

Evasion libraries and tools

- Fragroute(r) by Dug Song ~1999
- Robert Graham, SideStep, 2000
- Rain Forest Puppy: Whisker, libwhisker
- Raffael Marty, Thor – A tool to test intrusion detection systems, 2002
- Metasploit Framework
- Immunity Canvas
- Core Impact
- Breaking Point
- Libnet
- Scapy
- Tcpreplay
- Karalon

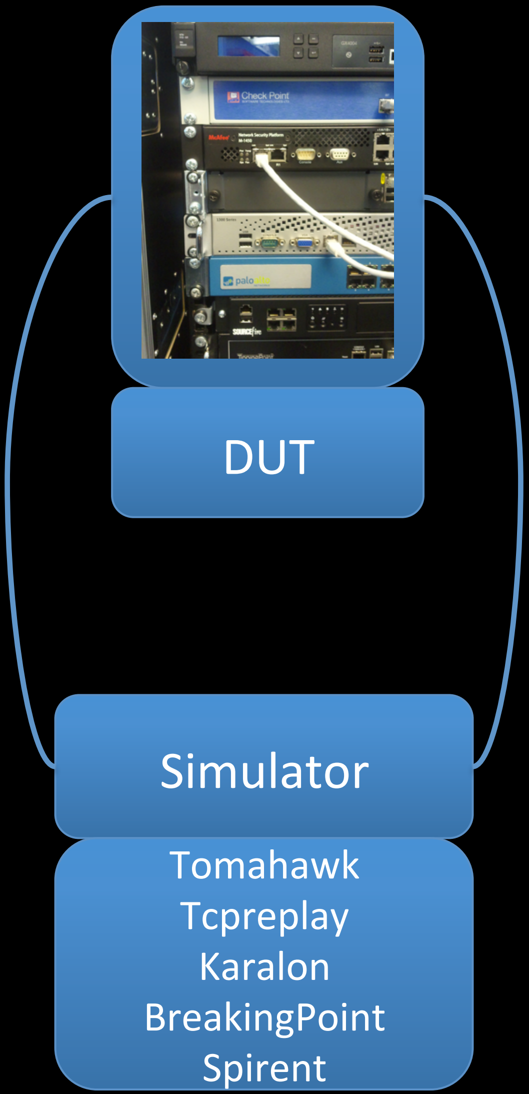
Motivation?

Middle-Box Testing is difficult

- False positive testing
  - Where do you get realistic traffic?
- False negative testing
  - Where do you get attack traffic?
- Simulation
  - Pathological traffic -> problems in false positive testing (the traffic was abnormal -> terminate)
  - Pathological traffic -> problems in false negative testing (the traffic does not constitute working attack)
- Automation is difficult without simulation

Pcap replay test procedure

- Replay pcap using one interface for client and one for server
- Verify that all packets are received in both interfaces
- If the pcap was an attack, receiving all packets means successful attack



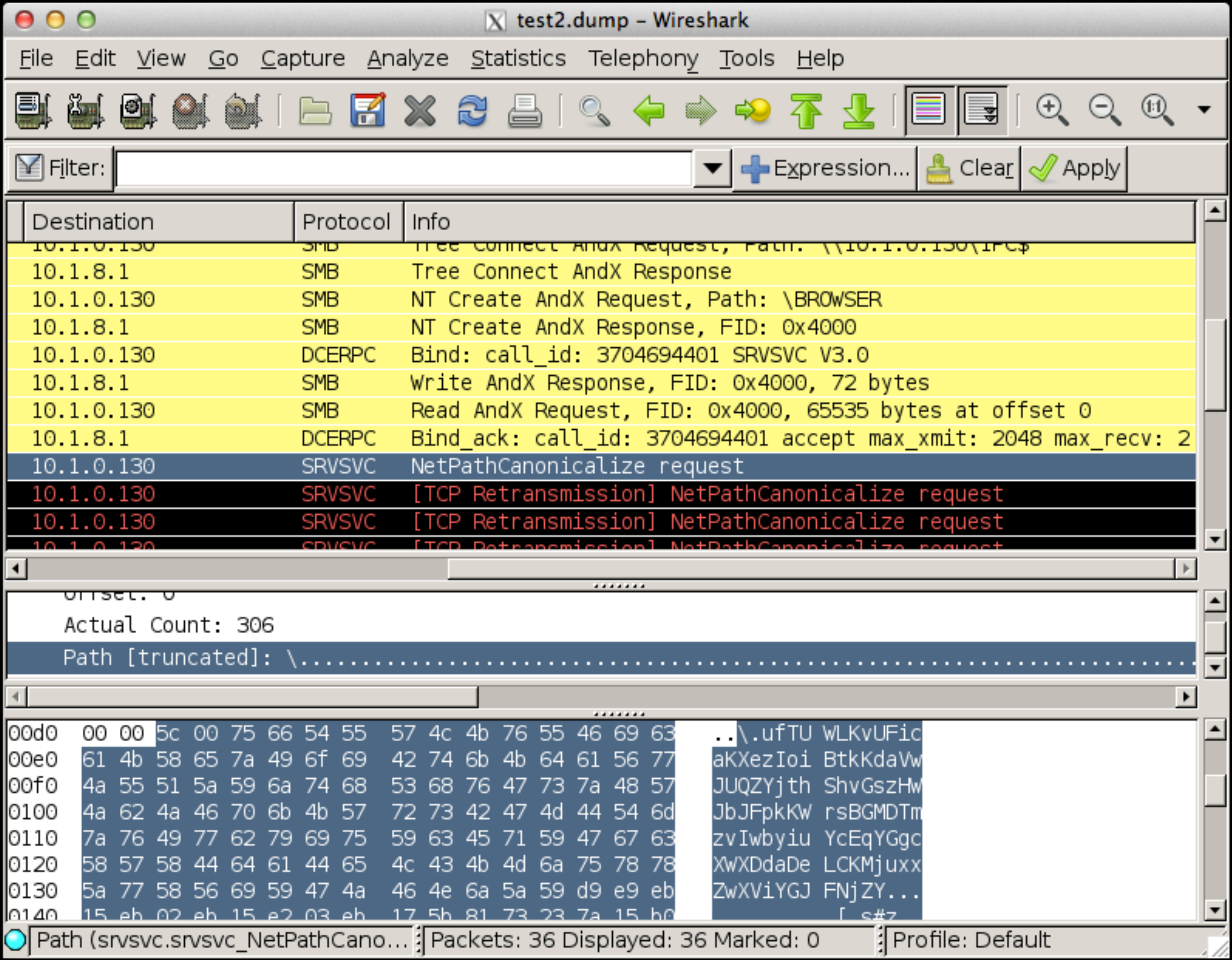
DUT

Simulator

Tomahawk
Tcpreplay
Karalon
BreakingPoint
Spirent

Example

- Exploiting MS08-067 CVE-2008-4250
- PCAP taken between Attacker and DUT (IPS)



test2.dump – Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression...
Clear
Apply

```
Destination     Protocol  Info
10.1.0.130      SMB       Tree Connect AndX Request, Path: \\10.1.0.130\IPC$
10.1.8.1        SMB       Tree Connect AndX Response
10.1.0.130      SMB       NT Create AndX Request, Path: \BROWSER
10.1.8.1        SMB       NT Create AndX Response, FID: 0x4000
10.1.0.130      DCERPC    Bind: call_id: 3704694401 SRVSVC V3.0
10.1.8.1        SMB       Write AndX Response, FID: 0x4000, 72 bytes
10.1.0.130      SMB       Read AndX Request, FID: 0x4000, 65535 bytes at offset 0
10.1.8.1        DCERPC    Bind_ack: call_id: 3704694401 accept max_xmit: 2048 max_recv: 2
10.1.0.130      SRVSVC    NetPathCanonicalize request
10.1.0.130      SRVSVC    [TCP Retransmission] NetPathCanonicalize request
10.1.0.130      SRVSVC    [TCP Retransmission] NetPathCanonicalize request
10.1.0.130      SRVSVC    [TCP Retransmission] NetPathCanonicalize request
```

```
    Offset: 0
    Actual Count: 306
    Path [truncated]: \..................................................................
```

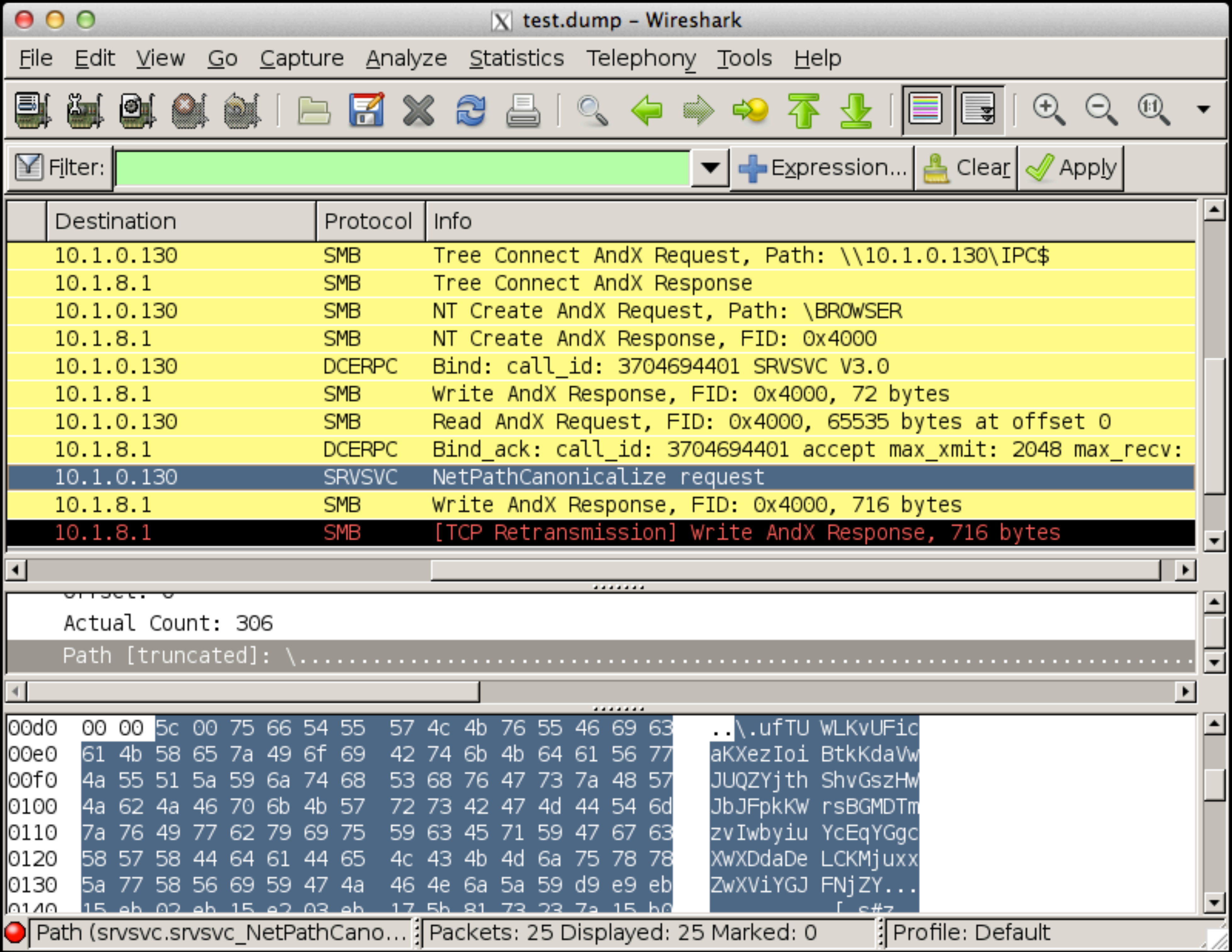
```
00d0  00 00 5c 00 75 66 54 55  57 4c 4b 76 55 46 69 63   ..\.ufTU WLKvUFic
00e0  61 4b 58 65 7a 49 6f 69  42 74 6b 4b 64 61 56 77   aKXezIoi BtkKdaVw
00f0  4a 55 51 5a 59 6a 74 68  53 68 76 47 73 7a 48 57   JUQZYjth ShvGszHW
0100  4a 62 4a 46 70 6b 4b 57  72 73 42 47 4d 44 54 6d   JbJFpkKW rsBGMDTm
0110  7a 76 49 77 62 79 69 75  59 63 45 71 59 47 67 63   zvIwbyiu YcEqYGgc
0120  58 57 58 44 64 61 44 65  4c 43 4b 4d 6a 75 78 78   XWXDdaDe LCKMjuxx
0130  5a 77 58 56 69 59 47 4a  46 4e 6a 5a 59 d9 e9 eb   ZwXViYGJ FNjZY...
0140  15 eb 02 eb 15 e2 03 eb  17 5b 81 73 23 7a 15 b0            [ s#z
```

Path (srvsvc.srvsvc_NetPathCano... Packets: 36 Displayed: 36 Marked: 0 Profile: Default

Problems in pcap replay tests

- The test scenario does not measure, when the connection was terminated
- It only measures whether everything in the pcap was received or not. It could have been that enough data was sent through before termination occurs -> in real life this would have been a successful attack
  - Even if the DUT logs say that attack was detected and terminated

- Exploiting MS08-067 CVE-2008-4250
- PCAP taken between DUT (IPS) and Target



```
Destination    Protocol  Info
10.1.0.130     SMB       Tree Connect AndX Request, Path: \\10.1.0.130\IPC$
10.1.8.1       SMB       Tree Connect AndX Response
10.1.0.130     SMB       NT Create AndX Request, Path: \BROWSER
10.1.8.1       SMB       NT Create AndX Response, FID: 0x4000
10.1.0.130     DCERPC    Bind: call_id: 3704694401 SRVSVC V3.0
10.1.8.1       SMB       Write AndX Response, FID: 0x4000, 72 bytes
10.1.0.130     SMB       Read AndX Request, FID: 0x4000, 65535 bytes at offset 0
10.1.8.1       DCERPC    Bind_ack: call_id: 3704694401 accept max_xmit: 2048 max_recv:
10.1.0.130     SRVSVC    NetPathCanonicalize request
10.1.8.1       SMB       Write AndX Response, FID: 0x4000, 716 bytes
10.1.8.1       SMB       [TCP Retransmission] Write AndX Response, 716 bytes
```

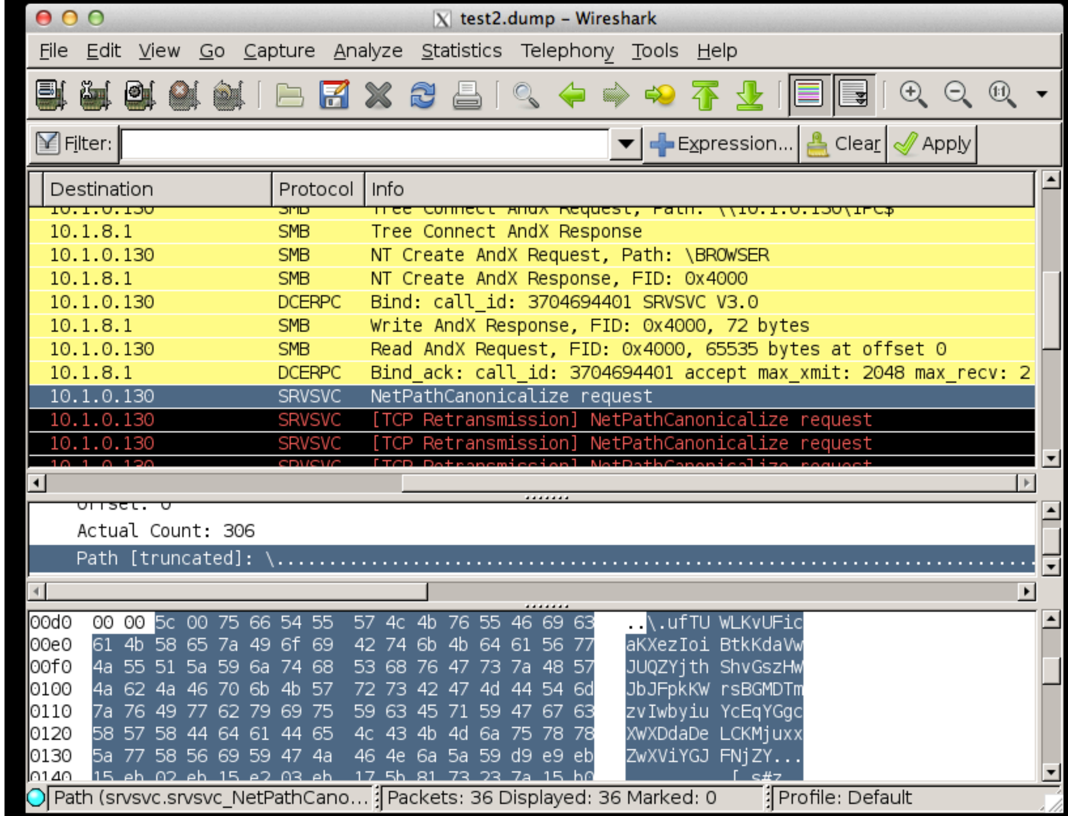
```
    Offset: 0
    Actual Count: 306
    Path [truncated]: \.....................................................................
```

```
00d0  00 00 5c 00 75 66 54 55  57 4c 4b 76 55 46 69 63   ..\.ufTU WLKvUFic
00e0  61 4b 58 65 7a 49 6f 69  42 74 6b 4b 64 61 56 77   aKXezIoi BtkKdaVw
00f0  4a 55 51 5a 59 6a 74 68  53 68 76 47 73 7a 48 57   JUQZYjth ShvGszHW
0100  4a 62 4a 46 70 6b 4b 57  72 73 42 47 4d 44 54 6d   JbJFpkKW rsBGMDTm
0110  7a 76 49 77 62 79 69 75  59 63 45 71 59 47 67 63   zvIwbyiu YcEqYGgc
0120  58 57 58 44 64 61 44 65  4c 43 4b 4d 6a 75 78 78   XWXDdaDe LCKMjuxx
0130  5a 77 58 56 69 59 47 4a  46 4e 6a 5a 59 d9 e9 eb   ZwXViYGJ FNjZY...
0140  15 eb 02 eb 15 e2 03 eb  17 5b 81 73 23 7a 15 b0   ........ .[.s#z..
```

Path (srvsvc.srvsvc_NetPathCano... Packets: 25 Displayed: 25 Marked: 0 Profile: Default

Before IPS

After IPS

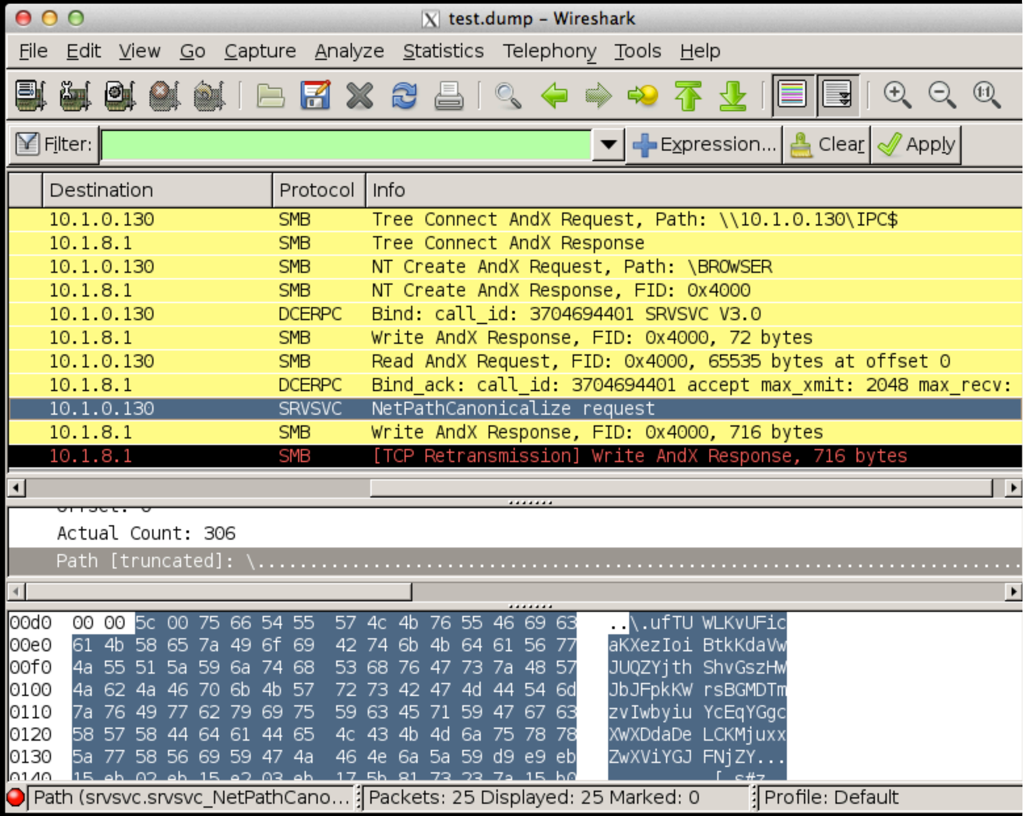

test2.dump - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression...
Clear
Apply

```
Destination        Protocol  Info
10.1.0.130         SMB       Tree Connect AndX Request, Path: \\10.1.0.130\IPC$
10.1.8.1           SMB       Tree Connect AndX Response
10.1.0.130         SMB       NT Create AndX Request, Path: \BROWSER
10.1.8.1           SMB       NT Create AndX Response, FID: 0x4000
10.1.0.130         DCERPC    Bind: call_id: 3704694401 SRVSVC V3.0
10.1.8.1           SMB       Write AndX Response, FID: 0x4000, 72 bytes
10.1.0.130         SMB       Read AndX Request, FID: 0x4000, 65535 bytes at offset 0
10.1.8.1           DCERPC    Bind_ack: call_id: 3704694401 accept max_xmit: 2048 max_recv: 2
10.1.0.130         SRVSVC    NetPathCanonicalize request
10.1.0.130         SRVSVC    [TCP Retransmission] NetPathCanonicalize request
10.1.0.130         SRVSVC    [TCP Retransmission] NetPathCanonicalize request
10.1.0.130         SRVSVC    [TCP Retransmission] NetPathCanonicalize request
```

```
    Offset: 0
    Actual Count: 306
    Path [truncated]: \.........................................................................
```

```
00d0  00 00 5c 00 75 66 54 55  57 4c 4b 76 55 46 69 63   ..\.ufTU WLKvUFic
00e0  61 4b 58 65 7a 49 6f 69  42 74 6b 4b 64 61 56 77   aKXezIoi BtkKdaVw
00f0  4a 55 51 5a 59 6a 74 68  53 68 76 47 73 7a 48 57   JUQZYjth ShvGszHW
0100  4a 62 4a 46 70 6b 4b 57  72 73 42 47 4d 44 54 6d   JbJFpkKW rsBGMDTm
0110  7a 76 49 77 62 79 69 75  59 63 45 71 59 47 67 63   zvIwbyiu YcEqYGgc
0120  58 57 58 44 64 61 44 65  4c 43 4b 4d 6a 75 78 78   XWXDdaDe LCKMjuxx
0130  5a 77 58 56 69 59 47 4a  46 4e 6a 5a 59 d9 e9 eb   ZwXViYGJ FNjZY...
0140  15 eb 02 eb 15 e2 03 eb  17 5b 81 73 23 7a 15 b0   .........[.s#z..
```

Path (srvsvc.srvsvc_NetPathCano… Packets: 36 Displayed: 36 Marked: 0 Profile: Default

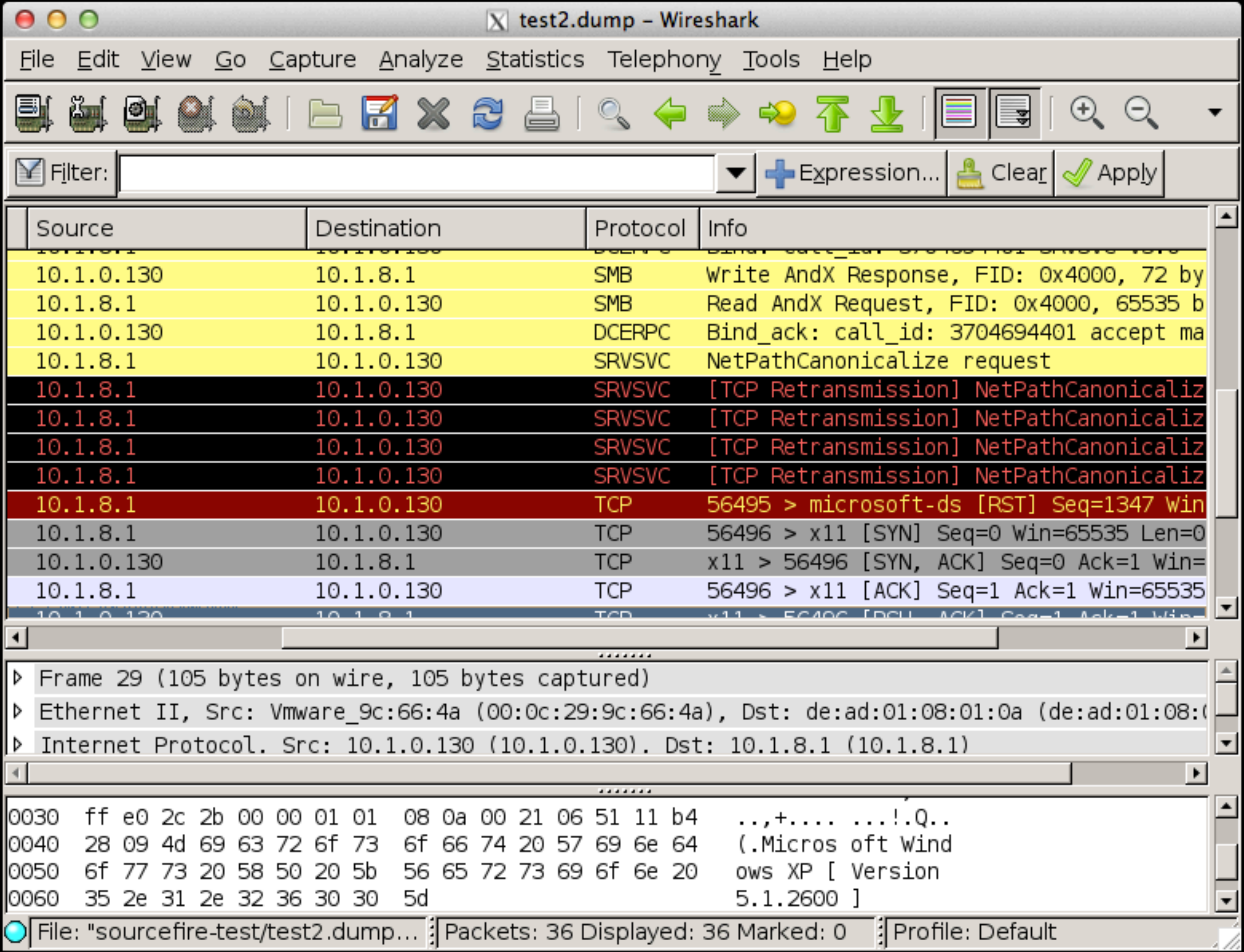

test.dump - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression...
Clear
Apply

```
Destination        Protocol  Info
10.1.0.130         SMB       Tree Connect AndX Request, Path: \\10.1.0.130\IPC$
10.1.8.1           SMB       Tree Connect AndX Response
10.1.0.130         SMB       NT Create AndX Request, Path: \BROWSER
10.1.8.1           SMB       NT Create AndX Response, FID: 0x4000
10.1.0.130         DCERPC    Bind: call_id: 3704694401 SRVSVC V3.0
10.1.8.1           SMB       Write AndX Response, FID: 0x4000, 72 bytes
10.1.0.130         SMB       Read AndX Request, FID: 0x4000, 65535 bytes at offset 0
10.1.8.1           DCERPC    Bind_ack: call_id: 3704694401 accept max_xmit: 2048 max_recv:
10.1.0.130         SRVSVC    NetPathCanonicalize request
10.1.8.1           SMB       Write AndX Response, FID: 0x4000, 716 bytes
10.1.8.1           SMB       [TCP Retransmission] Write AndX Response, 716 bytes
```

```
    Offset: 0
    Actual Count: 306
    Path [truncated]: \.........................................................................
```

```
00d0  00 00 5c 00 75 66 54 55  57 4c 4b 76 55 46 69 63   ..\.ufTU WLKvUFic
00e0  61 4b 58 65 7a 49 6f 69  42 74 6b 4b 64 61 56 77   aKXezIoi BtkKdaVw
00f0  4a 55 51 5a 59 6a 74 68  53 68 76 47 73 7a 48 57   JUQZYjth ShvGszHW
0100  4a 62 4a 46 70 6b 4b 57  72 73 42 47 4d 44 54 6d   JbJFpkKW rsBGMDTm
0110  7a 76 49 77 62 79 69 75  59 63 45 71 59 47 67 63   zvIwbyiu YcEqYGgc
0120  58 57 58 44 64 61 44 65  4c 43 4b 4d 6a 75 78 78   XWXDdaDe LCKMjuxx
0130  5a 77 58 56 69 59 47 4a  46 4e 6a 5a 59 d9 e9 eb   ZwXViYGJ FNjZY...
0140  15 eb 02 eb 15 e2 03 eb  17 5b 81 73 23 7a 15 b0   .........[.s#z..
```

Path (srvsvc.srvsvc_NetPathCano… Packets: 25 Displayed: 25 Marked: 0 Profile: Default



Source	Destination	Protocol	Info
10.1.0.130	10.1.8.1	SMB	Write AndX Response, FID: 0x4000, 72 by
10.1.8.1	10.1.0.130	SMB	Read AndX Request, FID: 0x4000, 65535 b
10.1.0.130	10.1.8.1	DCERPC	Bind_ack: call_id: 3704694401 accept ma
10.1.8.1	10.1.0.130	SRVSVC	NetPathCanonicalize request
10.1.8.1	10.1.0.130	SRVSVC	[TCP Retransmission] NetPathCanonicaliz
10.1.8.1	10.1.0.130	SRVSVC	[TCP Retransmission] NetPathCanonicaliz
10.1.8.1	10.1.0.130	SRVSVC	[TCP Retransmission] NetPathCanonicaliz
10.1.8.1	10.1.0.130	SRVSVC	[TCP Retransmission] NetPathCanonicaliz
10.1.8.1	10.1.0.130	TCP	56495 > microsoft-ds [RST] Seq=1347 Win
10.1.8.1	10.1.0.130	TCP	56496 > x11 [SYN] Seq=0 Win=65535 Len=0
10.1.0.130	10.1.8.1	TCP	x11 > 56496 [SYN, ACK] Seq=0 Ack=1 Win=
10.1.8.1	10.1.0.130	TCP	56496 > x11 [ACK] Seq=1 Ack=1 Win=65535

```
▷ Frame 29 (105 bytes on wire, 105 bytes captured)
▷ Ethernet II, Src: Vmware_9c:66:4a (00:0c:29:9c:66:4a), Dst: de:ad:01:08:01:0a (de:ad:01:08:(
▷ Internet Protocol. Src: 10.1.0.130 (10.1.0.130). Dst: 10.1.8.1 (10.1.8.1)
```

```
0030   ff e0 2c 2b 00 00 01 01  08 0a 00 21 06 51 11 b4   ..,+.... ...!.Q..
0040   28 09 4d 69 63 72 6f 73  6f 66 74 20 57 69 6e 64   (.Micros oft Wind
0050   6f 77 73 20 58 50 20 5b  56 65 72 73 69 6f 6e 20   ows XP [ Version
0060   35 2e 31 2e 32 36 30 30  5d                        5.1.2600 ]
```

File: "sourcefire-test/test2.dump... | Packets: 36 Displayed: 36 Marked: 0 | Profile: Default

More Problems

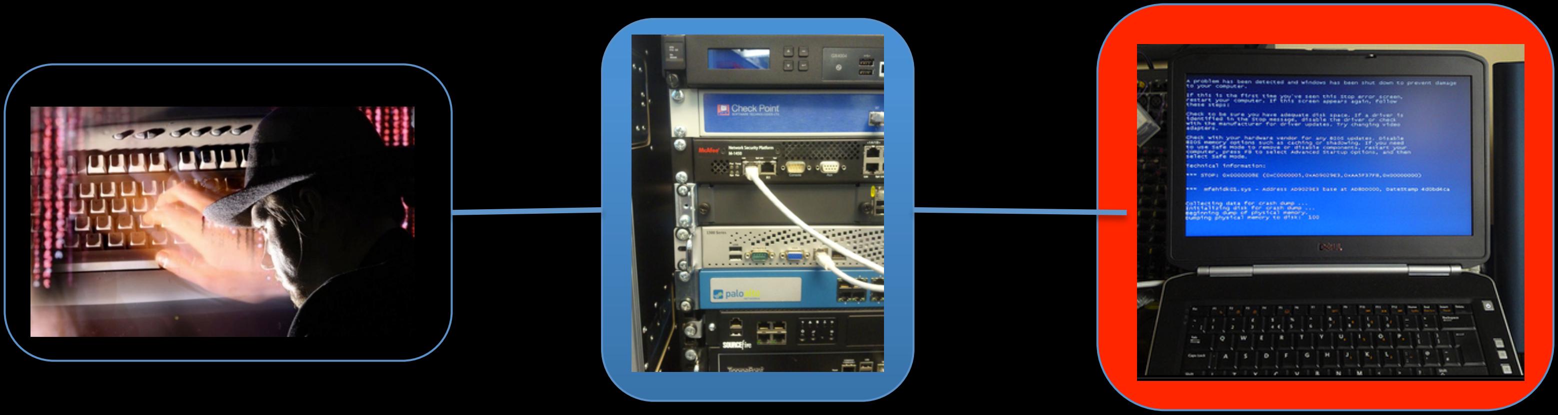
- Even real attack toolkits like metasploit are fooled.
- Remember, the DUT terminates the connection. Even metasploit does not know when the connection is terminated
- But if your shellcode opens backdoor in high port, it could be there even metasploit said that attack was terminated

Our Approach

- There's no publicly / commercially available single tool that
  - Does not simulate connections, but runs real exploits against real targets
  - Is suitable for automatic testing and is capable to rerun every testcase
  - Implements most known evasion research into reliable evasion methods
  - Is designed to use multiple evasions techniques in multiple protocol layers from IP to Application layer at the same time to test IPS and NGFW deep inspection's detection and prevention capabilities
  - Support also payload mutation to differentiate exploit and vulnerability based detection. Supports applying evasions on "normal traffic" to identify anomaly based detection
- Until Evader…

Evader

- We have implemented a tool we call Evader.
- It applies network level evasions to send a payload into a remote host through the IPS/NGFW
- Evader first sends non-malicious payloads that should not be prevented. This is called the false positive test.



Attacker

IPS

Victim

- If this is successful, the malicious payload will be sent. Depending on the selected malicious payload, the remote system is either crashed or compromised via remote code execution.
- If this happens we know that the evasion was functional.

Evader

- Evader contains known exploits that every IPS should detect
  - CVE-2004-1315
    - viewtopic.php phpBB remote code execution
    - selected because it can be continuously exploited without reboot or restart -> suitable for automatic testing
  - CVE-2008-4250, MS08-067
    - Msrcp server service buffer overflow, exploited by worms like conficker and stuxnet
    - selected because it can be continuously exploited without reboot or restart -> suitable for automatic testing
  - CVE-2012-0002, MS12-020
    - Remote Desktop Denial of Service
    - Relatively new vulnerability that most IPS/NGFW claim to protect against exploits

Evader

- Evader contains a multilayer network protocol stack.
- When sending the payload, Evader can apply multiple evasions on various protocols
- If the payload exploits some HTTP server vulnerability, we can apply evasions in the IP layer but also in TCP layer and HTTP layer. For msrpc, evader can built evasion combinations using IP/TCP/NetBIOS/SMB/MSRPC layers
- Evader can divide the connection into several stages and every stage can have its own evasions applied.

- In theory, for the selected exploit, the Evader can produce every possible data stream transmitting the payload, but in practice this cannot be tested since there are virtually endless amount of combinations and stage permutations.

- When evasions are not used, IPS/NGFW devices detect and terminate the attack
- With proper evasions applied, IPS/ NGFW start to Fail
  - Does not detect anything
  - Detect something that cannot be terminated due to risk for false positive
  - Detect attack, claims to terminate but fails termination

- Evader can be automated with another tool called mongbat
- Mongbat runs evader with different evasion combinations and collects results
  - Full evader command line is saved with random seed to allow exactly same evasion attack run at a later time
  - Takes pcaps
  - Basically Mongbat+Evader=Evader Fuzzer

Key differentiators

- Designed for automatic testing to systematically find weaknesses in middle-box security devices, specifically IPS and NGFW deep inspection
- Plugin interface to give hints on successful evasions and to disable pathological cases
- Does not simulate exploits, runs real exploits against real targets
- Complete stack visibility due own TCP/IP stack with built in application layers
- Not a proxy, so it knows the context of what it is going to send and can apply evasions for the whole session, or split the session in stages and apply different set of evasions per stage, or apply evasions per packet.
- Fuzzing
- Records everything (pcap, command line, randseed) allowing results to be analysed for what ever purpose...and repeated (or known working evasions applied into different exploit...)

Simple Test…

- We run evader using RDP exploit
- We used following evasions
  - Base = No evasion
  - Seg = Segment Size 8
  - Reverse = Segment Size 8 + Reverse
  - Time-Wait, re-use socket/source port before timer expires, no other evasions
  - Paws = Abuse Protection Against Wrapped Segment Numbers with timestamp option mangling, no other evasions

RDP CVE-2012-0002 Results

	DUT	Base	Seg 8	+REV	TWait	PAWS
A	xxxxx	FAIL	FAIL	FAIL	FAIL	FAIL
B	xxxxx	OK	FAIL	FAIL	FAIL	FAIL
C	xxxxx	OK	FAIL	FAIL	OK	OK
D	xxxxx	OK	OK	FAIL	OK	FAIL
E	xxxxx	OK	OK	OK	OK	FAIL
F	xxxxx	OK	FAIL	FAIL	OK	OK
G	xxxxx	OK	OK	OK	FAIL	OK
H	xxxxx	OK	OK	FAIL	OK	FAIL

- Tests were run in May 2012
- Every device were running latest software with latest updates and patches installed.
- All DUT were deployed inline
- All were running hardened policy with TCP/IP reassembly applied to RDP when not in default configuration
- OK = Attack was detected and blocked
- FAIL = Attack was not detected and Remote host was crashed

Stonesoft IPS was also tested, but its results were left out of the paper as we were unable to evade it at all (even with mongbat). It is also our belief that it is by far most difficult IPS to evade.

The RDP Exploit against Win7 was tested through these devices

- PaloAlto
- Fortigate
- SourceFire
- McAfee
- Juniper
- Cisco
- HP TippingPoint
- IBM Proventia

Conclusion

- Our tests prove that even 14 years after Ptacek – Newsham paper several IPS/NGFW are still vulnerable to TCP/IP reassembly attacks. We believe that proper TCP/IP reassembly is difficult to implement and expensive in terms of performance. Also the lack of proper testing tools helps to hide these incapabilities.
  - We know of cases where hardened policy that improves evasion resistance drops performance dramatically, for example to 10% of original throughput performance
- We have released a version of the evader tool for everyone to use and verify our findings. The tool can also be use to verify and harden IPS/NGFW policies in case there is some configurations available to improve detection rate when evasions are in place

http://Evader.stonesoft.com

- Freely available