Red Team
ivan notes 2022

1. Passive Recon

- Definition
  - attempt to gain info about targeted computers and networks without actively engaging with the systems
- Tools
  - Google Hacking/Dorking
    - Filter:
      site: - results on this site
      filetype: - results with this file-type
      inurl: - results with this term in url
      intitle: - results with this term in the title
    - Operators:
      -FILTER_OR_TERM - exclude this
      (FILTERS_OR_TERMS) - group these filters and/or terms
      FILTER | FILTER - results for this or this, can use OR instead of |
      FILTER AND FILTER - results that meet both this and this
  - theharvester
    - finding emails from an org
    - open source tool for acquiring emails from different major resources
  - whois
    - domain registration, track record of ownership
    - info includes registrar, name server and, sometimes, full contact info
  - netcraft
    - hording website of info regarding any particular website
    - info includes general background, networking, DNS, tracking info, CDNs, CMS, and more
  - recon-ng
    - can be used for both passive and active

- module load <module_name>
  options list
  options set <option_name value>
  run

- Summary
  - passive recon relies on being able to gather info about the target without directly contacting the target
  - can gather info through whois and other large DBs that acquire info about particular sites
  - tools such as recon-ng and theharvester allow us to gain some pertinent info including potential email targets for our future exploit
  - don't forget to check all different sources including GitHub when searching for info and misconfig
  - Basic methodology:
    info gathering
    gaining initial access
    priv esc
    post exploitation

2. Active Recon

- Definition
  - type of computer attack in which an intruder engages with the targeted system to gather info about vulns
- Understanding the target
  - -machine or network?
    -how many machines on the network?
    -which machines should i target to find out info about the network?
    -which machines will give me the most access to the rest of the network?
    -which machines are the lowest hanging fruit?
  - -lets write a bash script to find out which machines are on this network
    -lets write a bash script to find out which machines are running web servers on default TCP port 80
- nmap
  - -Be aware of how much data your scans can use
    -Be aware of how long scans can take
    -Always start minimum then expand

- Use cases:
  -check machines available on network
  -check ports open on a machine
  -enumerate services and versions that are running on any particular machine
  -determine OS and other metadata
  -run scripts that check if the machine is vulnerable against particular attacks
  -so much more
- key options
  - -p
    search a particular set of ports on targets, -p- for all ports
  - -O
    enable OS detection
  - -sV
    probe open ports to determine service/version info
  - --script=
    run a particular script against target(s)
  - -Pn
    treat all hosts as if they're online
  - -sU
    perform UDP scan
  - -A
    enable OS detection, version detection, script scanning, and traceroute

- DNS enumeration
  - -consider DNS servers to be first thought when accessing a local network
    -ping prods can be disabled on any particular machine and firewalls may prevent access
    -getting a true network map is essentially like putting all of your targets on a silver platter
    -many tools here: host, dig, nslookup, dnsenum, dnsrecon, and more
  - -what we want: a DNS server willing to give us all the info regarding available hosts on a network
    -what do we need: the domain name and DNS server
- Service Enumeration
  - SMB
    - -Operates on TCP 139, 445
      -allows comms between multiple machines and serves up info pertaining to password policies, usernames, group names, machine names, user and host SIDs and more

- good service if authenticated, if not - we are able to this maliciously
- enum4linux = tool for enumerating info from Windows and Samda systems

- SMTP
  - -server often used to send mail
    -supports several commands such as VRFY and EXPN
    -good tools = python scripts and smtp-user-enum
- SNMP
  - -UDP based
    -overlooked in machines since we tend to focus on TCP switches
    -management info base (MIB)
    -sometimes uses default community string
  - -MID values correspond to certain SNMP params
    -can use onesixtyone and snmpwalk

- Web Enumeration
  - -enumerate available paths/URIs on a website by using dirbuster and nikto
  - tools:
    -gobuster/dirbuster
    -nikto
    -burp suite
    -curl
    -your eyes and thoughts
  - gobuster
    - gobuster dir -u URL -w WORDLIST (x32 - 2018-)
      dirb URL wordlist

      to start, use the command, dirbuster
  - nikto
    niklto - h URL
- Summary
  - with active recon, we are throwing our first pokes at our target
  - be aware how we are poking and what effect this may have on the system
  - look for particular services that we can further enumerate for more information

3. Shells & Payloads

- Shells
  - user interface for access to an OS's services

- Types:
  Bash, Zsh, Sh, Web shells, shells through other services, interactive shells
  - command prompt
    #Meterpreter

- Payloads
  - the executable code, often contains a shell
  - Shell payloads:
    BIND - shell to be shared is bound by the listener, we are responsible for connecting to it
    REVERSE - shell to be shared is bound by the connector, we are responsible for opening a port and listening for an incoming connection
  - with nc, use -e option
  - Generation:
    Target OS, target architecture, type of payload, type of shell, destination of file execution, connection type
    - OS:
      Win, Lin, OSX, Unix, Android, IOS
    - Arch:
      x86, x64
  - Root
    - Windows Executable
      Linux Elf
      Bash
      Web Service / Other Network Service
  - Types
    - Unstaged
      - -entire payload is sent at one time
        -can be caught by nc or metasploit
        -generate a ton of traffic
        -fairly eas to detect by firewall or AC
    - Staged
      - -payloads are initially incomplete
        -require mechanism to distribute remainder of payload over time
        -good for avoiding AV and firewall detection
        -must be caught my metasploit

- msfvenom

- -p
  set a payload
  -p linux/x64/sell_reverse_tcp
- -f
  output format/file type
  elf, exe, py, c, etc
- LPORT=
  ip of port to connect back to on connecting machine or IP of port to open on hose machine (bind shells)
- LHOST=
  ip of machine to connect back to (usually your IP address)
- --list
  list options in a particular category
  --list payloads

- Summary
  - active recon = poking target
  - be aware how poking is impacting the system/network
  - look for services to further enum for more info

4. Web

- Fundamental Concerns
  - Authentication
    - trusting that someone is who they say they are
  - Communication
    - transferring data through potentially unreliable middlemen
  - Authorization
    - giving resource access to the right people
  - Control
    - limiting or understanding the capabilities of agents
- OWASP Top 10
  - injection
    - server-side code exec
      - -assume attacker is non-admin client
        -defense vulnerable if client can execute code on a server

- broken authentication
  - allows for impersonation
    - identity theft
      -assume attacker has access to middlemen and db
      -defense vulnerable if comm or storage exposes passwords
- XSS
  - client-side code execution
    - -assume attacker is non-admin client
      -defense vulnerable if client can execute code on another client
      -stored and reflected
      - Stored
        persistent XSS, is the more damaging of the two, It occurs when a malicious script is injected directly into a vulnerable web application, the application instead stores the input and embeds it into a later response in an unsafe way
      - Reflected
        reflecting of a malicious script off of a web application, onto a user's browser, when an application takes some input from an HTTP request and embeds that input into the immediate response in an unsafe way
- direct reference
  - access control can be circumvented
- security misconfig
  - vulnerable default/inherited settings
    - -assume attacker has access to codebase
      -defense is vulnerable if secrets are easy to discover, files are improperly shared
      -bad stuff: app impersonation, decryption
- data exposure
  - data is insecurely transmitted, stored, or overshared
- missing access control
  - users can do things they shouldn't be allowed to do
    - -assume attacker is client
      -defense vulnerable if client can act outside of authorization
      -frontend access control = good UX
      -true access control come from backend, considerations: requested resource, requested action, agent making the request

- XSRF
  - abuse target website's trust in the browser
    - also known as one-click attack or session riding and abbreviated as CSRF or XSRF, is a type of malicious exploit of a website where unauthorized commands are submitted from a user that the web application trusts
- vulnerable components
  - 3p tools are vulnerable
- unvalidated redirects
  - abusable open-ended forwarding

- HTTPS
  - SSL(encryption + authentic server) = HTTPS
    - Authentic Server
      -has SSL cert
      -digitally signed
      -forms a web of trust
      -self-signed in development

- Good Auth
  - -comms are secure
    -storage is secure
    -cannot set priv via signup
    -logging in requires username and pass
    -data is not inadvertently shared

- Things to look for
  - Default creds
    - did i try admin admin?
  - request freedom
    - can i just try to upload a file with a post request?
  - improper authorization
    - can i just try to upload a file with a standard user account?
  - data over exposure
    - is there a way i can see something im not supposed to
  - injection opportunities
    - can i use any termination characters to attempt to write my own code?

- XSS opportunities
  - can i write code that will impact another user?
- File Inclusion
  - when the attack tricks web server into including and unintended file, file must be in a language the server speaks to be executed
  - purpose
    -allows attacker to access sensitive files and execute malicious code
  - Two types
    -LFI - included file is hosted on the web server
    -RFI - included file is hosted remotely (anywhere)
    - RFI example
      -attacker injects malicious script to webapp
      -malicious code is executed from attacker's website
      -server download malicious file from attacker's website
      -attacker gets control over the webapp
- SQLi
  - common SQL keywords
    - SELECT
      - which COLUMNS to include in output table (shrinks results horizontally)
    - FROM
      - which TABLE to pull data from
    - JOIN
      - another TABLE to glue/concatenate to the output
    - ON
      - what COLUMNS must match when joining two tables
    - WHERE
      - which ROWS to include in the output table (shrinks the result vertically)

5. Attacks

- Passwords
  - Hashing
    - plain text > hash function > hashed text

- -one-way functions
  -DBs and other password storage mediums should hash their passwords
  -good hash funcs avoid collisions, have high entropy, and involve salting
- SHA
  NTLM
  LM
  MD
- can gather password hashed and try to crack them

- Attack methods
  - Brute Force
    - -tries a pass, compares to hash
      -can be used against salted passwords
      -works for local and remote
      -small (computed at run time)
      -can take a very long time
  - Rainbow table
    - -list of pre-computed hashes linking password to hash
      -potentially extremely large
      -very quick with a big enough list
      -can NOT be used against salted passwords
- tools
  - Wordlists
    - rockyou.txt
      cewl
      crunch
  - Crackers
    - john
      hashcat
      hydra
      medusa
  - Rainbows
    - crackstation.net

- Exploit finding & Metasploit
  - what are exploits?

- a piece of software, chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic

- where can i find exploits?
  - exploitdb.com
    searchsploit
    google
    metasploit

- ms
  - -comprehensive hacking framework consisting of thousands of modules
    -requires postgresql to cache exploits and improve performance
    - console run: msfconsole
      postgres: service postgresql start
      cache in the console: db_rebuild_cache
  - composition
    - exploits
      - use search func, can also download modules online
    - auxiliary
      - -set of tasks that are generally not used to compromise systems
        -can be used to verify, check, soft/hard test, scope out, and more
        -over 1000 aux modules
    - post
      - -generally use post exploitation
        -can assist in recon or escalation
    - additional tools
  - Command format
    - use
      - load a module into ms
    - exploit
      - launch a module
    - show
      - show some particular thing within ms, show payloads, show options
    - set

- set a value of an option within ms

- sessions
  - bring up list off current sessions within ms

6. Priv Esc

- act of exploiting a bug, design flaw, or config oversight in an os or app to gain elevated access to resources that are normally protected from an app or user
  - -use everything found in recon
    -pay attention to versions
    -transfer helper tools over to your machine
    /use/share/windows-resources
    -upgrade to a meterpreter shell
    -find writeable directories:
    lin: /tmp
    win: %TEMP% , %PUBLIC%
- compiling exploits
  - many written in C or C++, generally dont just find .exe or .elf laying around
    - be aware of how exploit is written, read instructions for compilation from exploit, was is written originally from lin or win? do you need dos2unix?
  - how do we compile?
    linux - gcc or g++
    win - ARCH-ming32-gcc or ARCH-ming32-g++
    apt update && apt upgrade
    install mingw-w64
- general techniques
  - kernel exploit
    - typically involves making a syscall with arguments designed specially to cause unintended behavior, despite the syscall attempting to only allow valid arguments
    - often intended result is spawning a root shell
      - can also affect certain kernel defenses (modify permissions or /etc/shadow)
    - helps to check for missing patches: wmic qme get Caption, Description, HotFixID, InstalledOn
  - service exploit
    - takes advantage of a service that is running with higher permissions

- can i run certain services with higher privs?
- linux
  -sudo -l (also check /etc/groups)
- win
  -net localgroup administrators

- this service can be running already or started as another user with a SUID bit
  - trick the service into doing something it shouldnt
  - lin
    -ps aux
  - win
    -tasklist

- pass dumping / cracking / reuse
  - pwddump/fgdump/secretsdump
  - pass the hash
  - mimikatz
    - priv escalate
      more modules at the github

- data leaking
  - did you look in all of the .xtx/.doc/.docx/.pdf files?
  - did you look for cookies or stores passwords?
    - Windows: C:
      \Users\username\AppData\Roaming\Microsoft\Windows\Cookies
    - Metasploit: enum_ie, enum_crome
  - did you look at all of the log files you can see?
  - did you look for config files (.conf), data files (.json / .xml), or initialization files (.sql)?
  - did you try locally connecting to the db?
    - use built-in mysql command, maybe there is just a username and no password?

- misconfigured permissions
  - did you check which apps you can run without a password?
    - did you check what programs have an SUID bit set as root?

- on windows: icacls

- process hijacking (usually win only)
  - attempt to gain control of a service that has higher permissions
    - unquoted service path
    - HotPotato / JuicePotato
- Powershell
  - Empire
  - PowerUp
    - powershell.exe -exec bypass
      Import-Module .\PATH-TO-FILE\PowerUp.ps1
      Invoke-AllChecks