Introduction

- **Welcome to the Course!**
    - **Domain 1: Planning and Scoping**
        - Planning an engagement
        - Key legal concepts
        - Scoping an engagement
        - Compliance-based assessments
    - **Domain 2: Information Gathering and Vulnerability Identification**
        - Information gathering techniques
        - Vulnerability scanning
        - Analyzing scan results
        - Preparing for exploitation
        - Weaknesses in specialized systems
    - **Domain 3: Attacks and Exploits**
        - Social engineering attacks
        - Exploiting vulnerabilities
            - Network-based
            - Wireless and RF-based
            - Application-based
            - Local host-based
            - Physical security
        - Post-exploitation techniques
    - **Domain 4: Penetration Testing Tools**
        - Use Nmap for information gathering
        - Know the use case for various tools
        - Analyze tool output or data
        - Analyze basic scripts
            - Bash, Python, Ruby, and Powershell
    - **Domain 5: Reporting and Communication**
        - Report writing and best practices
        - Post-report delivery activities
        - Recommending mitigations
        - Importance of communication in testing

Exam Foundations (PT0-001)

- **CompTIA Pentest+**

…verifies that candidates have the knowledge and skills required to plan and scope an assessment, understand legal and compliance requirements, perform vulnerability scanning and penetration testing, analyze data, and effectively report and communicate results.

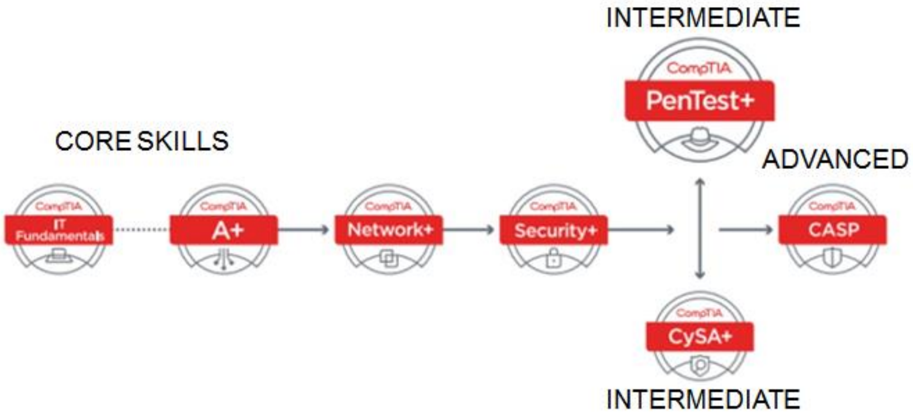
-CompTIA.org

- **Exam Description**
  - CompTIA Pentest+ covers:
    - White hat hacking tools and techniques
    - Legal and compliance requirements
    - Information gathering, vulnerability scanning, exploitation, and reporting results
- **The Five Domains**

1	**Planning and Scoping**	**15%**
2	**Information Gathering and Vulnerability Identification**	**22%**
3	**Attacks and Exploits**	**30%**
4	**Penetration Testing Tools**	**17%**
5	**Reporting and Communication**	**16%**

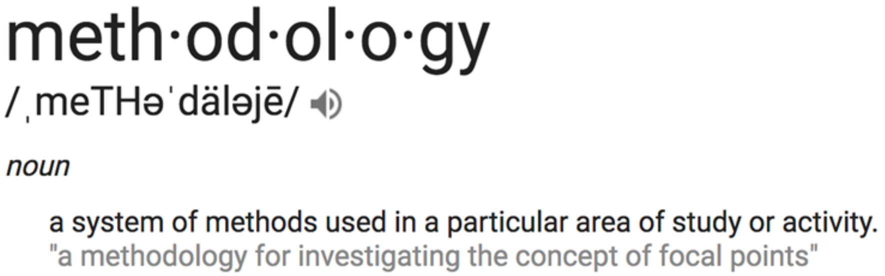
- **Exam Details**
  - Up to 85 questions in 165 minutes
  - Requires a 750 out of 900 (83.33%)
  - Recommended Experience:
    - CompTIA Network+ and/or Security+
    - 3-4 years of hands-on InfoSec
  - Cost: $346 (US Dollars)
  - Released: July 31, 2018

- **Cybersecurity Career Path**



Domain 1 - Planning and Scoping

- **Planning and Scoping**
  - **Domain 1: Planning and Scoping**
    - Planning an engagement
    - Key legal concepts
    - Scoping an engagement
    - Compliance-based assessments
  - **What kinds of questions can I expect on test day?**
    - All objectives for Domain 1 are listed as “explain” only by CompTIA
    - Therefore, no simulations will come from this domain…
- **Penetration Testing Methodology**
  - **Methodology**



meth·od·ol·o·gy

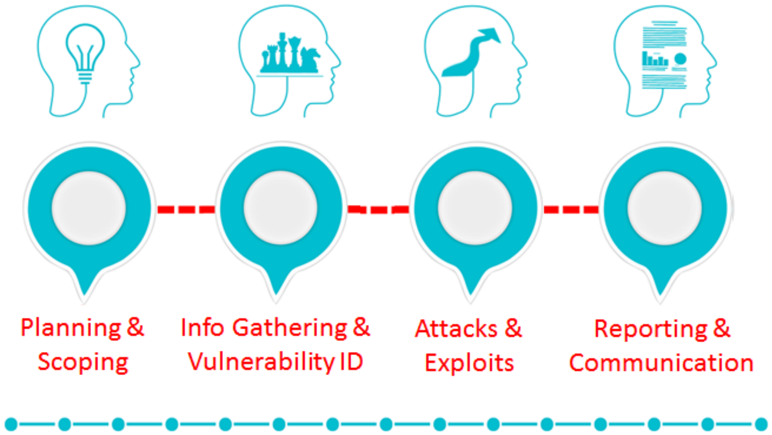
/ˌmeTHəˈdäləjē/

noun

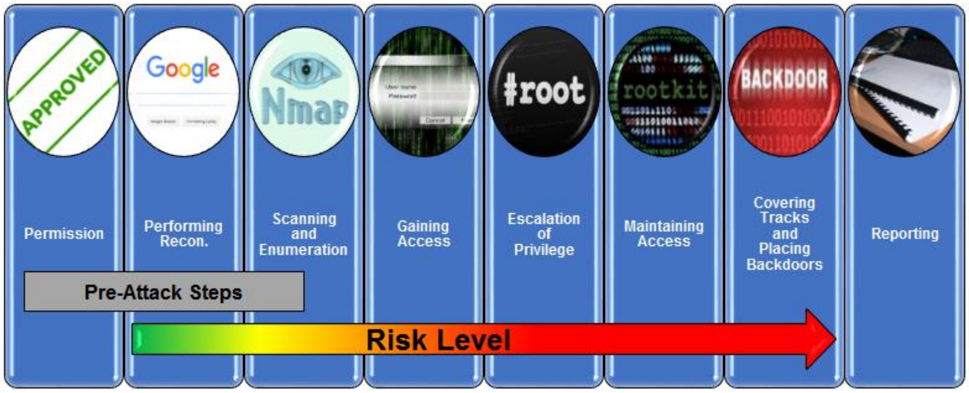
a system of methods used in a particular area of study or activity.

"a methodology for investigating the concept of focal points"

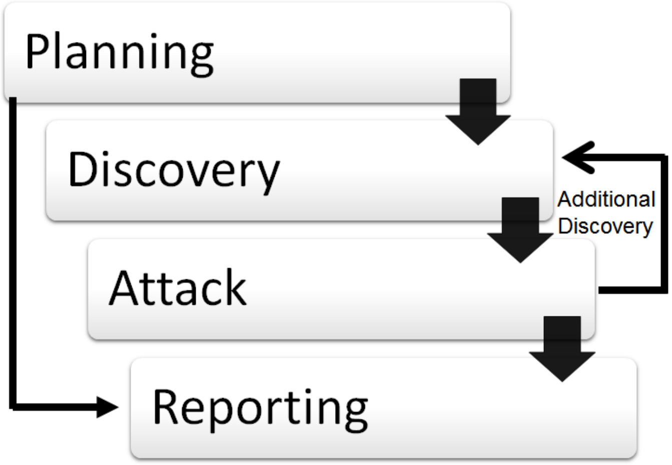
  - **Pentest Methodology**



- **Ethical Hacker's Methodology**



- **NIST SP 800-115 Methodology**



- **Planning a Penetration Test**
  - **Why Is Planning Important?**
  - **Who is the Target Audience?**
    - Need to know to properly plan the pentest
    - What does the business do?
    - What are their objectives?
  - **Budgeting**
    - Controls many factors in a test
    - If you have a large budget, you can perform a more in-depth test
      - Increased timeline for testing
      - Increased scope
      - Increased resources (people, tech, etc.)
  - **Resources and Requirements**
    - What resources will the assessment require?
    - What requirements will be met in the testing?
      - Confidentiality of findings

      - Known vs. unknown vulnerabilities
      - Compliance-based assessment
  - **Communication Paths**
    - Who do we communicate with about the test?
    - What info will be communicated and when?
    - Who is a trusted agent if testing goes wrong?
  - **What is the End State?**
    - What kind of report will be provided after test?
    - Will you provide an estimate of how long remediations would take?
  - **Technical Constraints**
    - What constraints limited your ability to test?
    - Provide the status in your report
      - Tested
      - Not Tested
      - Can't Be Tested
  - **Disclaimers**
    - Point-in-Time Assessment
      - Results were accurate when the pentest occurred
    - Comprehensiveness
      - How complete was the test?
      - Did you test the entire organization or only specific objectives?
- **Rules of Engagement**
  - **Rules of Engagement (RoE)**
    - Timeline
    - Locations
    - Time restrictions
    - Transparency
    - Test boundaries
  - **RoE: Timeline**
    - How long will the test be conducted?
      - A week, a month, a year
    - What tasks will be performed and how long will each be planned for?
  - **RoE: Locations**
    - Where will the testers be located?
      - On-site or remote location
    - Does organization have numerous locations?
    - Does it cross international borders?
  - **RoE: Time Restrictions**
    - Are there certain times that aren't authorized?
    - What about days of the week?
    - What about holidays?

- **RoE: Transparency**
    - Who will know about the pentest?
    - Will the organization provide resources to the testers (white box test)?
- **RoE: Boundaries**
    - What will be tested?
    - Is social engineering allowed to be used?
    - What about physical security testing?
    - How invasive can the pentest be?

- **Legal Concepts**
    - **Local and National Restrictions**
        - Laws and regulations regarding cybercrime vary from country to country, check the local laws before conducting an assessment

Consult your attorney before performing any penetration testing work to ensure you are within the legal bounds for the countries laws where you are operating

- **CRIME AND CRIMINAL PROCEDURE**
    - Hacking is covered under United States Code, Title 18, Chapter 47, Sections 1029 and 1030 (Crimes and Criminal Procedure)
    - §1029 Fraud & related activity w/ access devices
        - Prosecute those who knowingly and with intent to defraud produce, use, or traffic in one or more counterfeit access devices.
        - Access devices can be an application or hardware that is created specifically to generate any type of access credentials
    - §1030 Fraud and related activity with computers
        - Covers just about any computer or device connected to a network
        - Mandates penalties for anyone who accesses a computer in an unauthorized manner or exceeds one's access rights
        - Can be used to prosecute employees using capability and accesses provided by their company to conduct fraudulent activity
- **Obtain Written Authorization**
    - White hat hackers always get permission
    - *This is your get out of jail free card…*
    - Penetration tests can expose confidential information so permission must be granted
- **Third-Party Authorization**
    - If servers and services are hosted in the cloud, you must request permission from the provider prior to conducting a penetration test
- **Contracts**
    - Statement of Work (SOW)

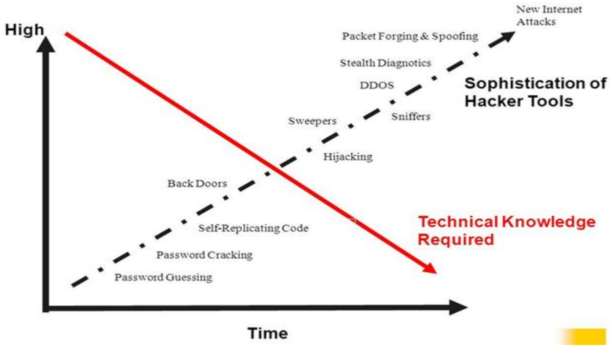
      - Formal document stating scope of what will be performed during a penetration test
    - Master Service Agreement (MSA)
      - Contract where parties agree to most of the terms that will govern future actions
    - Non-Disclosure Agreement (NDA)
      - Legal contract outlining confidential material or information that will be shared during the assessment and what restrictions are placed on it
  - **Corporate Policies**
    - What do corporate policies allow you to do?
    - Have employees waived their privacy?
    - What policies should be tested?
      - Password strength/reuse
      - Bring Your Own Device (BYOD)
      - Encryption
      - Update frequency
  - **Export Restrictions**
    - Wassenaar Agreement precludes the transfer of technologies considered "dual-use"
    - Strong encryption falls under this restriction
    - Penetration testing tools could be considered surveillance tools and fall under these rules
- **Testing Strategies**
  - **Penetration Testing Strategies**
    - Black Box
    - Gray Box
    - White Box
  - **Black Box (No Knowledge Test)**
    - No prior knowledge of target or network
    - Simulates an outsider attack
    - Only focuses on what external attacks see and ignores the insider threat
    - Takes more time and is much more expensive
  - **White Box (Full Knowledge Test)**
    - Full knowledge of network, systems, and the infrastructure
    - Spend more time probing vulnerabilities and less time gathering information
    - Tester is given support resources from the organization
  - **Gray Box (Partial Knowledge Test)**
    - Partial knowledge of target

        - Can be used as an internal test to simulate an insider attack with minimal knowledge
        - Can also be used to decrease the information gathering stage so more time can be spent on identifying vulnerabilities
        - Examples
            - IP ranges provided
            - Company emails to create phishing campaigns
- **White Box Support Resources**
    - **Support Resources**
        - Generally provided only for a white box penetration test
            - Architectural diagrams
            - Sample application requests
            - SDK documentation
            - SOAP project files
            - Swagger document
            - WSDL/WADL
            - XSD
    - **Architectural Diagrams**
        - Network diagrams, software flow charts, physical maps of organizational facilities
        - Assists the tester in mapping out network topologies, location of switch closets, and where key information systems are located
    - **Sample Application Requests**
        - Generally used for testing web applications or other applications developed by organization
    - **SDK Documentation**
        - Software Developer's Kit (SDK) provides a set of tools, libraries, documentation, code samples, processes, or guides to allow faster development of a new app on a platform
        - SDK provides code libraries for use
    - **SOAP Project File**
        - Simple Objective Access Protocol (SOAP) is a messaging protocol specification for exchanging structured information in the implementation of web services
        - SOAP project files are created from WSDL files or a single service call
    - **Swagger Document**
        - Open-source framework with a large system of tools to help design, build, document, test, and standardize REST Web Services
        - Representational State Transfer (REST) has been replacing SOAP in most web applications in recent years
        - REST is a web application architectural style based on HTTP

- **WSDL and WADL**
  - Web Services Description Language
    - XML-based interface definition language used for describing the functionality offered by a web service such as a SOAP server
    - Flexible and allows binding options
    - Not useful for REST services with WSDL 1.1
  - Web Application Description Language
    - XML-based machine-readable description of HTTP-based web services
    - Easier to write than WSDL but not as flexible
    - Typically used for REST services
- **XML Schema Definition (XSD)**
  - World Wide Web Consortium (W3C) recommendation that specifies how to formally describe elements in an Extensible Markup Language (XML) document

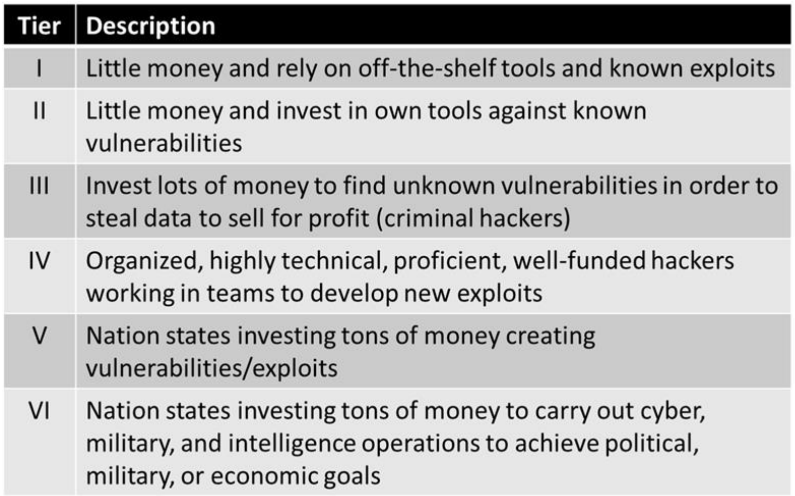
- **Types of Assessments**
  - **Goal-based Pentests**
    - Specific goals are defined before testing starts
    - Pentester may attempt to find many unique methods to achieve the specific goals
  - **Objective-based**
    - Objective-based pentests seek to ensure the information remains secure
    - Testing occurs using all methods and more accurately simulates a real attack
    - **Compliance-based**
    - Risk-based compliance assessment that is required to ensure policies or regulations are being followed properly
    - Regulations and policies provide checklists, for example the PCI-DSS compliance assessment
    - Objectives are clearly defined
    - Focus is on password policies, data isolation, limited network/storage access, and key management
  - **Premerger**
    - Before two companies perform a merger, it is common to conduct penetration tests on them to identify weaknesses being inherited
    - Can be a part of the due diligence efforts
  - **Supply Chain**
    - Pentest may be required of your suppliers to ensure they are meeting their cybersecurity requirements
    - Can be required prior to allowing an interconnection between the supplier's systems and your organization's systems

    - Minimize risk by purchasing only from trusted vendors
  - **Red Team**
    - Penetration test conducted by internal pentesters of an organization during security exercise to ensure defenders (blue team) can perform their jobs adequately
- **Threat Actors**
  - **Tiers of Adversaries**
    - Not all threat actors are created equal
    - Some are structured, some are unstructured
    - Some are more skilled than others
  - **Advanced Persistent Threat (APT)**
    - Group with great capability and intent to hack a particular network or system
    - Target organizations for business or political motives and usually funded by nation states
    - Conduct highly covert hacks over long periods of time
  - **Hacktivist**
    - Conduct activities against governments, corporations, or individuals
    - Can be an individual or member of a group
  - **Insider Threat**
    - Already have authorized user access to the networks, making them extremely dangerous
    - May be a skilled or unskilled attacker
    - Might be a former or current employee
  - **Script Kiddies**
    - Low-skilled attackers who use other's tools
    - Use freely available vulnerability assessment and hacking tools to conduct attacks
  - **Capabilities**



Less technical knowledge is required to perform attacks because of the increased sophistication of hacking tools

- **What is the Intent?**
  - Greed or monetary gain
  - Power, revenge, or blackmail
  - Thrills, reputation, or recognition
  - Espionage or political motivation
- **Threat Modeling**
  - What threat are you trying to emulate?
  - Will you use open-source and openly available tools like a script kiddie, or create custom hacks like an Advanced Persistent Threat?
  - Will you be given insider knowledge or perform a white box penetration test?
- **Tiers of Adversaries**

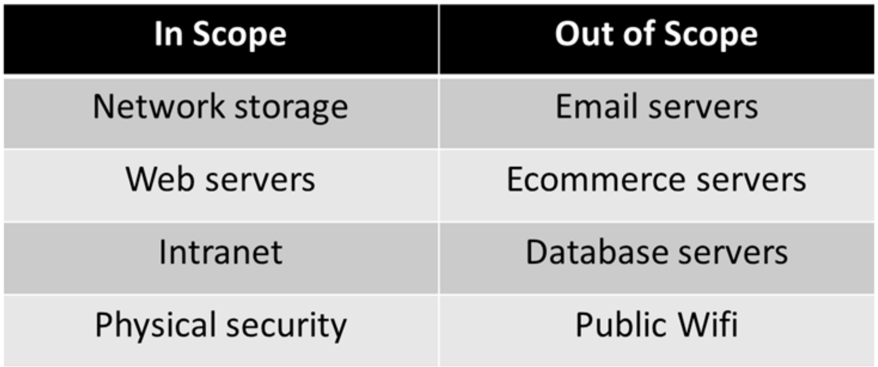


Tier	Description
I	Little money and rely on off-the-shelf tools and known exploits
II	Little money and invest in own tools against known vulnerabilities
III	Invest lots of money to find unknown vulnerabilities in order to steal data to sell for profit (criminal hackers)
IV	Organized, highly technical, proficient, well-funded hackers working in teams to develop new exploits
V	Nation states investing tons of money creating vulnerabilities/exploits
VI	Nation states investing tons of money to carry out cyber, military, and intelligence operations to achieve political, military, or economic goals

- **Target Selection**
  - **Target Selection**
    - Internal or External
    - First-party or Third-party hosted
    - Physical
    - Users
    - SSIDs
    - Applications
  - **Internal or External**
    - Internal focuses on targets inside the firewall
      - Can be on-site or off-site
      - Logically internal
    - External focuses on publicly facing targets
      - Webservers in the DMZ
      - Outside the protected LAN

  - **First-party or Third-party**
    - Are the targets hosted by the organization or by a third-party service provider?
    - DionTraining.com is hosted by Thinkific and might be outside the penetration test scope
  - **Physical**
    - Are we contracted to test physical security?
    - Should we attempt to break into the facility?
  - **Users**
    - Is social engineering authorized?
    - Are particular users being targeted or not considered part of the assessment?
  - **Wireless and SSIDs**
    - Is wireless pentesting being conducted?
    - Are any SSID's out of scope?
      - Guest or public networks
  - **Applications**
    - Are we focused on a particular application?
    - Is a particular application mission critical and cannot be targeted?
      - Credit card processing system
      - Health care systems
- **Other Scoping Considerations**
  - **Whitelist vs Blacklist**
    - Will your pentest systems be put on a list?
    - Whitelist will allow you access, but blacklist will prevent your system from connecting
  - **Security Exceptions**
    - Intrusion Prevention System (IPS)
    - Web Application Firewall (WAF)
    - Network Access Control
    - Certificate Pinning
      - Required if the organization relies on digital certificates as part of their security
    - Company policies
  - **Risk**
    - What is the risk tolerance of the organization?
    - Avoidance
      - Actions taken to eliminate risk completely
    - Transference
      - Risk is moved to another entity
    - Mitigation

            - Controls and countermeasures are put into place
        - Acceptance
            - Risk is identified, analyzed, and within limits
    - **Tolerance to Impact**
        - What is the impact to operations going to be?
        - Balance the assessment needs with the operational needs of the organization by placing things in or out of scope



In Scope	Out of Scope
Network storage	Email servers
Web servers	Ecommerce servers
Intranet	Database servers
Physical security	Public Wifi

    - **Schedule**
        - Will the timing of the penetration test be known by the organization's defenders?
        - Will it be performed during peak or off-peak hours?
        - What about holidays?

List of events

Date and time restrictions

Client stakeholder notifications

        - **Scope Creep**
            - Condition when a client requests additional services after the SOW and project scope have been agreed to and signed
            - How will scope be contained?
            - Document any changes to the scope of test
            - Recommend signing a change order to SOW

More devices = More time = More resources

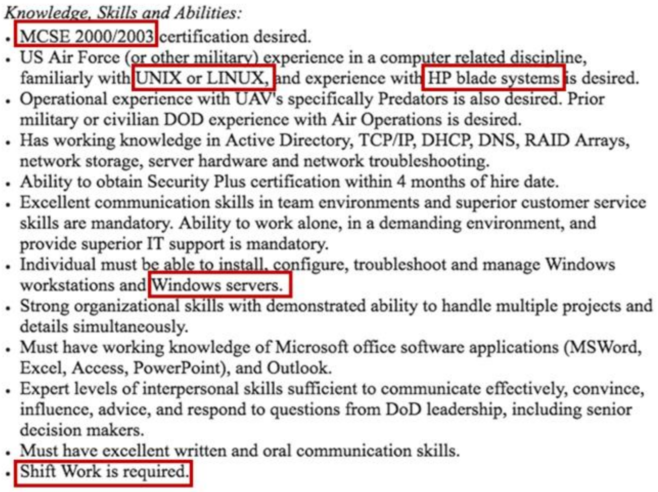
Domain 2 - Information Gathering and Vulnerability Identification

- **Info Gathering & Vulnerability Identification**
    - **Domain 2: Information Gathering and Vulnerability Identification**
        - Conducting information gathering
        - Performing vulnerability scanning
        - Analyzing results of vulnerability scans
        - Leveraging information for exploitation
        - Weaknesses in specialized systems
    - **What kinds of questions can I expect on test day?**
        - First 3 objectives are "given a scenario"
        - Other 2 objectives are "explain" only
        - You could see a simulation on Nmap…
        - Given a scenario, conduct information gathering using appropriate techniques.
        - Given a scenario, perform a vulnerability scan.
        - Given a scenario, analyze a vulnerability scan result.
- **Information Gathering**
    - **Reconnaissance**
        - Systematic attempt to locate, gather, identify, and record information about a target
        - Also known as *footprinting* the organization
        - Techniques include:
            - Internet or open-source research
            - Social engineering
            - Dumpster diving
            - Email harvesting
    - **What kind of information are we looking to find?**
        - Phone numbers
        - Contact names
        - Email addresses
        - Security-related information
        - Information systems used
        - Job postings
        - Resumes

- **Job Postings**

System Administrator II
Government Contractors, LLC

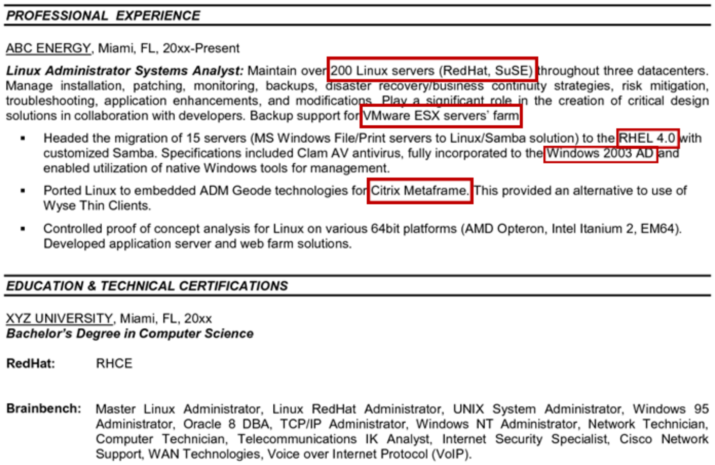
Job Title:	System Administrator II
Clearance:	TS/SCI
Location	Cannon AFB, NM
Reports To:	Program Manager
FLSA Status:	Exempt, Full Time, Regular



Knowledge, Skills and Abilities:

- MCSE 2000/2003 certification desired.
- US Air Force (or other military) experience in a computer related discipline, familiarly with UNIX or LINUX, and experience with HP blade systems is desired.
- Operational experience with UAV's specifically Predators is also desired. Prior military or civilian DOD experience with Air Operations is desired.
- Has working knowledge in Active Directory, TCP/IP, DHCP, DNS, RAID Arrays, network storage, server hardware and network troubleshooting.
- Ability to obtain Security Plus certification within 4 months of hire date.
- Excellent communication skills in team environments and superior customer service skills are mandatory. Ability to work alone, in a demanding environment, and provide superior IT support is mandatory.
- Individual must be able to install, configure, troubleshoot and manage Windows workstations and Windows servers.
- Strong organizational skills with demonstrated ability to handle multiple projects and details simultaneously.
- Must have working knowledge of Microsoft office software applications (MSWord, Excel, Access, PowerPoint), and Outlook.
- Expert levels of interpersonal skills sufficient to communicate effectively, convince, influence, advice, and respond to questions from DoD leadership, including senior decision makers.
- Must have excellent written and oral communication skills.
- Shift Work is required.

- **Resumes**



PROFESSIONAL EXPERIENCE

ABC ENERGY, Miami, FL, 20xx-Present

Linux Administrator Systems Analyst: Maintain over 200 Linux servers (RedHat, SuSE) throughout three datacenters. Manage installation, patching, monitoring, backups, disaster recovery/business continuity strategies, risk mitigation, troubleshooting, application enhancements, and modifications. Play a significant role in the creation of critical design solutions in collaboration with developers. Backup support for VMware ESX servers' farm

- Headed the migration of 15 servers (MS Windows File/Print servers to Linux/Samba solution) to the RHEL 4.0 with customized Samba. Specifications included Clam AV antivirus, fully incorporated to the Windows 2003 AD and enabled utilization of native Windows tools for management.
- Ported Linux to embedded ADM Geode technologies for Citrix Metaframe. This provided an alternative to use of Wyse Thin Clients.
- Controlled proof of concept analysis for Linux on various 64bit platforms (AMD Opteron, Intel Itanium 2, EM64). Developed application server and web farm solutions.

EDUCATION & TECHNICAL CERTIFICATIONS

XYZ UNIVERSITY, Miami, FL, 20xx
Bachelor's Degree in Computer Science

RedHat:	RHCE
Brainbench:	Master Linux Administrator, Linux RedHat Administrator, UNIX System Administrator, Windows 95 Administrator, Oracle 8 DBA, TCP/IP Administrator, Windows NT Administrator, Network Technician, Computer Technician, Telecommunications IK Analyst, Internet Security Specialist, Cisco Network Support, WAN Technologies, Voice over Internet Protocol (VoIP).

- **Reconnaissance Tools**
  - Nslookup
  - Traceroute
  - Ping
  - Whois

        - Domain Dossier
        - Email Dossier
        - Google
        - Social Networking
        - Discover.sh
        - Maltego
    - **Putting It All Together…**
        - You've collected examples of emails, names, phone numbers, servers' addresses, documents, presentations, and more
        - Use the emails to draft potential spearphishing emails to be more realistic
            - Use target's PDF, Word, Excel, and PowerPoint files to embed malware
            - Use real employee names, positions, and writing styles to mimic real email traffic
    - **Taking It Further…**
        - Use domain name squatting
            - Targeting titancipher.com by using titancypher.com
            - Make the site look as close to the original as possible but host malware there
        - Identify any subdomains (developer sites, mail servers, etc.) for exploitation
- **Scanning and Enumeration**
    - **Scanning**
        - Actively connecting to the system and get a response to identify open ports and services
    - **Types of Scanning**
        - Hosts
        - Systems
        - Networks
        - Computers
        - Mobile Devices
        - Applications
        - Printers
    - **Enumeration**
        - Actively connecting to the systems to determine open shares, user accounts, software versions, and other detailed info
    - **Types of Enumeration**
        - Hosts
        - Networks
        - Domains

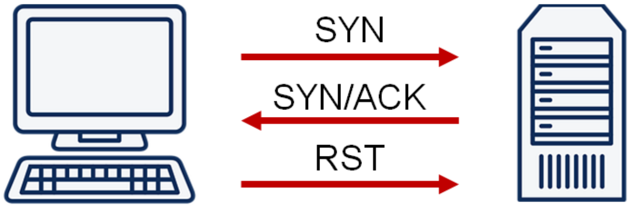
        - Users/Groups
        - Network shares
        - Web pages
        - Applications
        - Services
        - Tokens
        - Social networks
    - **How Do We Scan and Enumerate?**
        - Use specialized scanning/enumeration tools and public information sources
- **Fingerprinting**
    - **Fingerprinting**
        - Identification of the operating system, service, software versions being used by a host
    - **Banner Grabbing**
        - Manual enumeration and fingerprinting
        - Use telnet or Netcat to connect to target host
        - Commonly used for FTP, SSH, Telnet, & HTTP
    - **Packet Crafting**
        - Also known as *packet manipulation*
        - Sending modified packet headers to gather information from a system or host
        - Tools:
            - Nmap
            - Netcat (nc)
            - Ncat (ncat)
            - Hping
    - **Packet Inspection**
        - Manual enumeration performed by analyzing the captured packets to determine information
- **Cryptographic Inspection**
    - **Cryptographic Inspection**
        - Determine the encryption is being used during your information gathering
        - Do they have web servers with SSL or TLS?
        - What about Wireless Networks using WEP, WPA, WPA2, or a WPS handshake?
        - Are files encrypted on the network shares?
    - **Certificate Inspection**
        - Web-servers will identify the type of encryption they support (SSL 2.0, SSL 3.0, or TLS)

        - Tools exists to automate this process
SSLyze script comes with Kali Linux

- **Eavesdropping**
    - **Eavesdropping**
        - Part of your information gathering can include eavesdropping on wired and wireless networks
            - Check if this is in the scope of your assessment
        - Radio Frequency monitoring can be performed to determine the type of devices used in the facility (Cellular, WiFi, Bluetooth, etc)
        - Radio frequencies can be captured and analyzed using specialized tools
    - **Sniffing Network Traffic**
        - Intercepts and logs network traffic that can be seen via the wired or wireless network interface
        - If you gain access to one host computer, you could use it to capture traffic on other parts of the network, too!
    - **Packet Capture**
        - Use Wireshark or TCPDump to conduct packet capturing of wired or wireless networks
        - Connect to a mirrored port to capture wired network traffic
        - Wireless networks can be captured and their encryption cracked to access the data using Aircrack-ng
- **Decompiling and Debugging**
    - **Decompiling**
        - Reverse engineering of software using a decompiler
        - Reverses the processes of a compiler but not as cleanly…
        - Decompilers cannot always turn executables back into their source code but can it back to byte code or assembly
    - **Debugging**
        - Used to identify and remove errors from hardware, software, or systems
    - **Decompiling vs Debugging**
        - Decompiling uses a static analysis of code
        - Debugging often uses a dynamic approach that allows code to be run
            - Code is run step by step through the program
            - Code can be run until a break point
        - Both techniques can be useful when conducting a penetration test or assessment of custom-built applications
- **Open Source Intelligence**
    - **Research Sources**
        - Penetration testers must keep up to date on the latest techniques and vulnerabilities
        - Lots of different sources available and this lesson is not an exhaustive list

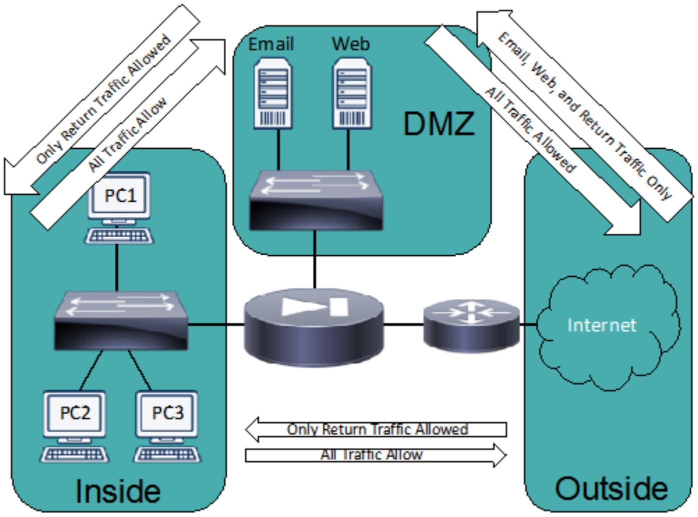
  - **CERT - Computer Emergency Response Team**
    - https://www.us-cert.gov
  - **JPCERT - Japan's CERT**
    - https://www.jpcert.or.jp/english/vh/project.html
  - **NIST - National Vulnerability Database**
    - https://nvd.nist.gov/
  - **CVE - Common Vulnerabilities & Exposures**
    - https://cve.mitre.org/
  - **CWE - Common Weakness Enumeration**
    - https://cwe.mitre.org/
  - **CAPEC - Common Attack Pattern Enumeration & Classification**
    - https://capec.mitre.org/
  - **Full Disclosure - Mailing List from Nmap**
    - http://seclists.org/fulldisclosure/
- **Vulnerability Scans**
  - **Vulnerability Scans**
    - Scans of a host, system, or network to determine what vulnerabilities exist
    - Numerous tools used by both defenders and attackers to identify vulnerabilities
    - Tools are only as good as their configuration
  - **Credentialed vs Non-credentialed Scans**
    - Credentialed scans
      - Scanner uses an authorized user or admin account
      - Closer to the system administrator's perspective
      - Finds more vulnerabilities
    - Non-credentialed scans
      - Scanner doesn't have a user or admin account
      - Closer to the hacker's perspective
  - **Types of Scans**
    - Discovery scan
    - Full scan
    - Stealth scan
    - Compliance scan
  - **Discovery Scan**
    - Least intrusive scan (like a ping sweep)
    - Used to create a network map to show connected devices in the architecture
  - **Full Scan**
    - In-depth scan including port, services, and vulnerabilities
    - Easy to see in network traffic when performed

- **Stealth Scan**
  - Conducts scans by sending a SYN packet and then analyzing the response
  - If SYN/ACK is received, the destination is trying to establish the connection (port is open) and the scanner sends a packet with RST



- **Compliance Scan**
  - Used to identify vulnerabilities that may affect compliance with regulations or policies
  - Commonly setup as a scanning template in your vulnerability scanner (PCI-DSS)
- **QualysGuard Vulnerability Scanner**
- **Tenable's Nessus Vulnerability Scanner**
- **Rapid7's Nexpose**
- **OpenVAS (Open-source Scanner)**
- **Nikto (Web Application Scanner)**

- **Scanning Considerations**
  - **When Do You Run the Scans?**
    - Scanning the systems can take up valuable resources and slow down the network
    - Are you trying to be sneaky?
    - When is the best time to run the scans?
  - **What Protocols Will Be Used?**
    - Each protocol scanned takes time/resources
    - Will you scan every port and services?
    - Consult scope of assessment and objectives
  - **Where Do You Scan From?**
    - Network topology is important, are you inside or outside the network?
    - PCI-DSS requires both internal and external scanning to be performed



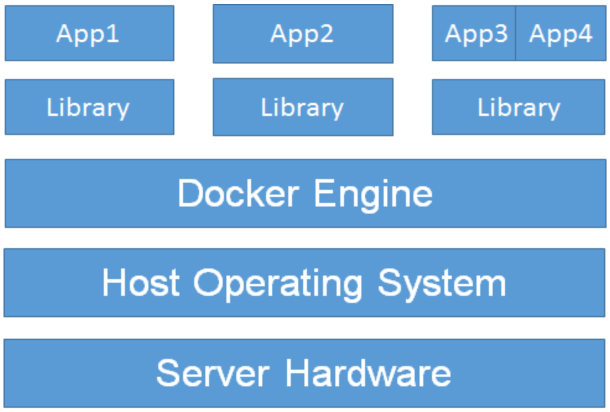
- **Bandwidth Limitations**
  - How much bandwidth is dedicated to the scan?
  - Throttle the queries if needed
    - Nmap –T option sets the timing
- **Fragile or Non-Traditional Systems**
  - Should we scan these?
  - Should we exempt these?

- **Application and Container Scans**
  - **Application Scanning**
    - Dynamic Analysis
      - Occurs while a program is running
      - Program is run in a sandbox and changed noted
    - Static Analysis
      - Performed in a non-runtime environment
      - Inspects programming code for flaws/vulnerabilities
      - Line by line inspection can be performed
  - **Containers**
    - Containers are like micro virtual machines
    - Each container is built from the base Operating System image with unique applications run on top of them
    - Requires less resources than a typical VM
    - Docker, Puppet, and Vagrant are examples
  - **Containers Require Security**
    - Containers still contain applications which can contain vulnerabilities
    - Still need to be scanned for vulnerabilities
    - If an OS vulnerability is found, it will apply to multiple containers (all based on same OS) and can lead to a large level of exploitation

- **Analyzing Vulnerability Scans**
    - **Asset Categorization**
        - Categorize by Operating System or function
        - Ideally, we identify high-value assets
            - Domain Controllers, Web Servers, Databases, etc.
        - Categorize by most vulnerabilities
        - Categorize by the most critical vulnerability
    - **Adjudication**
        - Must consider which vulnerabilities to attack
        - False positives
            - A vulnerability is identified by the scan but does not really exist on the system
            - False positive should be filtered out of your scans
    - **Prioritize the Vulnerabilities**
        - Consider the most critical vulnerabilities first
        - What target should we focus on first?
    - **Common Themes**
        - Analyze vulnerability scans for common themes that are recurring items
            - Vulnerabilities
                - Do the same vulnerabilities show up on many hosts?
            - Observations
                - Do you see the same types of operating systems and applications being used across the network?
            - Lack of best practices
                - Common misconfigurations
                - Weak passwords
                - Poor security practices
                - Logging disabled
- **Leverage Information for Exploit**
    - **Map Vulnerabilities to Exploits**
        - Research vulnerabilities for items found during enumeration
    - **Prioritize Efforts for Pentest**
        - What will be attacked first?
        - What exploits will we use?
            - Do we need custom made exploits?
        - Does Metasploit or Nmap already have known exploits for the vulnerabilities?
            - Use the ‘search’ function in Metasploit
- **Common Attack Techniques**
    - **Common Attack Techniques**

  - Before moving into exploitation, we need to understand the basics of attack techniques
    - Cross-compiling code
    - Exploit modification
    - Exploit chaining
    - Proof-of-concept development
    - Social engineering
    - Credential brute forcing
    - Dictionary attacks
    - Rainbow tables
    - Deception
  - **Cross-compiling Code**
    - Many pentesters use Kali Linux but many victim systems are Windows-based
    - Exploits for Windows can be compiled on Linux using tools like Mingw-w64
  - **Exploit Modification**
    - If the organization has added security, you may need to modify exploits to get past it
    - Encrypting or encoding an exploit to avoid detection by anti-virus
  - **Exploit Chaining**
    - Involves layering exploits in a series
    - Exploit chain example
      1. Bypass the firewall
      2. Gain access to user system
      3. Escalate privileges
- **Proof-of-Concept Development**
  - New or custom exploits require testing before using in a pentest
  - Build a virtual machine based on the specifications you earned during enumeration
- **Social Engineering**
  - Involves manipulating people to get information or to gain access
  - Often utilizes deception and lies
- **Credential Brute Forcing**
  - Attempt to crack a password or authentication system to gain access
  - Attempt to crack passwords from a hash file
  - Conduct password guessing to login
- **Dictionary Attack**
  - Brute force attack that uses a dictionary of commonly used usernames and passwords

        - Weak passwords and passwords from previous data breaches make a great list
    - **Rainbow Tables**
        - Pre-computed hash values of known usernames and passwords used for offline password file cracking
- **Weakness in Specialized Systems**
    - **ICS, SCADA, and PLC**
        - Industrial Control Systems
        - Supervisory Control and Data Acquisition
        - Programmable Logic Controller
    - **Mobile Devices**
        - Lack of updates (especially Android)
        - Root/Jailbreak (especially iPhone)
        - 3rd party applications
        - Bluetooth, NFC, and WiFi
        - Lack of Mobile Device Management in smaller organizations
    - **Internet of Things (IoT)**
    - **Embedded Devices**
        - Contains a special purpose computing system
        - ICS/SCADA contain embedded devices
        - Cars also contain embedded devices
    - **Point-of-Sale (POS) Systems**
        - Used as registers in stores and restaurants
    - **Biometrics**
        - Fingerprint readers and other biometrics aren't foolproof security measures
    - **Application Containers**
        - Containers rely on the same underlying operating system
        - Breaking out of a container can allow attackers to break into other systems



- Real-Time Operating System (RTOS)
    - Usually found in embedded systems
    - Security is not a primary concern during their development
    - Usually a stripped-down version of Linux
    - Uses limited resources on the machine and can be subjected easily to attacks

Domain 3 - Attacks and Exploits

- **Attacks and Exploits**
    - **Domain 3: Attacks and Exploits**
        - Social engineering attacks
        - Network, wireless/RF, application, and local host exploitation
        - Physical security attacks
        - Post-exploitation techniques
    - **What kinds of questions can I expect on test day?**
        - Majority are "given a scenario"
        - "Compare and Contrast" social engineering attacks
        - "Summarize" physical security attacks
- **Social Engineering**
    - **Phishing**
        - Victims are contacted by email, telephone, or text message by someone posing as a legitimate organization
        - Lures people into providing sensitive data
            - Personal identifiable information
            - Banking information
            - Passwords
    - **Spear Phishing**
        - Occurs when an attacker creates a message to appeal to a specific individual
    - **Whaling**
        - Form of spear phishing that directly targets the CEO, CFO, CIO, CSO, or other high-value targets
    - **SMS Phishing**
        - Short Message Service
        - Phishing that occurs over text message
    - **Vishing (Voice Phishing)**
        - Phishing that occurs over a telephone
        - Involves calling someone and pretending you are someone else
    - **Elicitation**

verb (used with object)

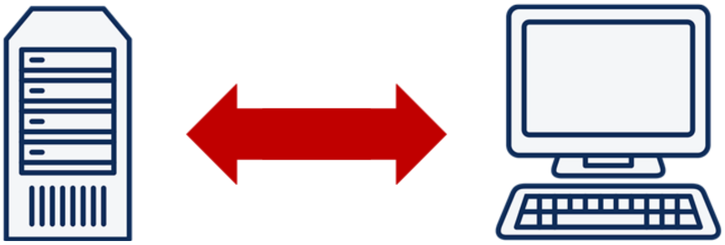
1. to draw or bring out or forth; educe; evoke:
   to elicit the truth; to elicit a response with a question.

        - Usually uses a series of questions to get employees to tell you valuable or sensitive information
        - If you can compromise one email account then you can elicit more information from other employees by acting like that person

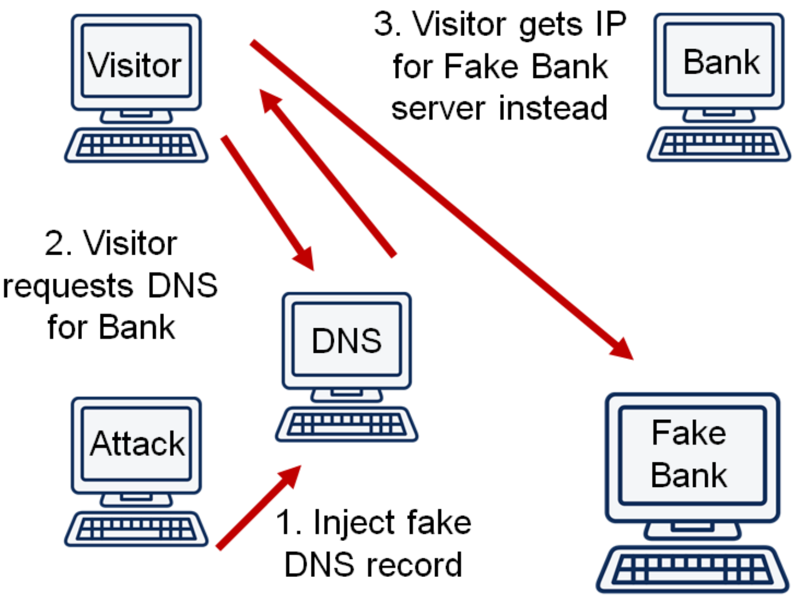
            - *This is your boss's boss and I need you to send me the quarterly financials…*
    - **Interrogation**
        - Interviews used by law enforcement, military, or intelligence agencies
        - Pentesters won't generally use this technique…
    - **Impersonation**
        - Act of pretending to be someone else in order to gain access or gather information
    - **Shoulder Surfing**
        - Reading the screen of another user
        - Looking at a user entering a PIN or password
    - **USB Key Drop**
        - Pentester loads up a USB with malware, backdoors, or a keylogger
        - Drop the USB drive in the parking lot near the organization
- **Motivation Factors**
    - **Authority**
        - People are more willing to comply with a request when they think it is coming from someone in authority
            - CEO or manager
            - Important client
            - Government agencies
            - Financial institutions
    - **Urgency**
        - Humans want to please others by nature…
        - We want to be helpful…
        - I only have a few minutes before the big presentation, can you print this for me?
    - **Social Proof**
        - Social engineering through Facebook or Twitter can be useful
            - Lots of Likes or Shares add to social proof
            - People are more likely to click the link
        - We crave social group interaction and have a need to be included
        - Sometimes we don't fully understand what the inclusion means for us or why we are performing an action
    - **Scarcity**
        - Technique that works well to get people to act fast
        - Signup now for a special offer… supplies are limited!
    - **Likeability**
        - Social engineers are friendly and likeable
            - People will want to help them
        - Find common ground and shared interests

  - **Fear**
    - If you don't do _____ then ______ will happen
      Use threats or demands
    - Anti-virus scams & Ransomware are examples
- **Physical Security Attacks**
  - **Piggybacking/Tailgating**
    - Occurs when a pentester follows an authorized individual into a secure location
    - Authorized person may or may not be complicit
  - **Fence Jumping**
    - Fences provide a physical security boundary for the organization
    - Pentester can go over (or under) a fence to avoid a checkpoint
  - **Dumpster Diving**
    - Pentester looks through the trash of an organization
    - Looking for paperwork, disks, USB drives, badges, files, manuals
  - **Lock Picking**
    - Many areas that the pentester needs access to are locked
    - Learning lock picking is a valuable skill
      for a pentester who focuses on physical security
  - **Lock Bypass**
    - Pentester could jam a lock or bypass it by manipulating the locking function
    - Stop a door from being shut fully by inserting a spacer or wedge
  - **Egress Sensor**
    - Door will automatically unlock and open when a person approaches
    - Sensors could be tricked to allow the door to be opened
    - Some of these "fail open" when power is lost
  - **Badge Cloning**
    - Identification badges are required by many organizations
    - Snap a photo using a digital camera and reproduce the security badge
      - Works visually but won't make it past a reader
    - Badge cloners can reproduce magnetic swipe
      or RFID badges
- **Network-based Vulnerabilities**
  - **NETBIOS Name Service**
    - Often called WINS on Windows systems
    - NetBIOS Name Service (NBNS) is part of the NetBIOS-over-TCP protocol suite
    - Serves much the same purpose as DNS to translate human-readable names to IP addresses using a 16-character (ASCII) name
    - NETBIOS name is the host name of a system

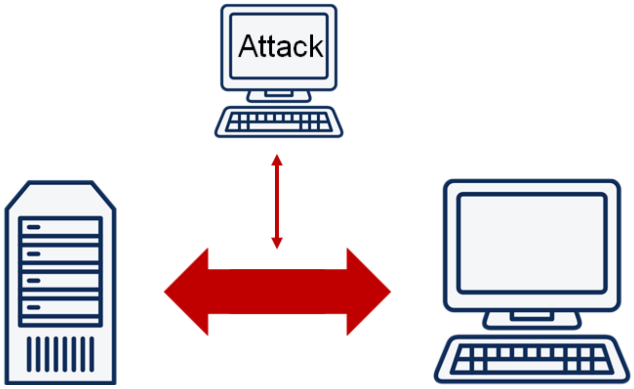
- **Link-Local Multicast Name Resolution (LLMNR)**
  - Protocol based on the DNS packet format allowing both IPv4 and IPv6 hosts to perform name resolution for hosts on same local link
  - Often used when there is not DNS server on the network
  - Included in Windows Vista and newer versions
  - Linux implements LLMNR using system
  - Useful when a temporary network is created, such as Ad-Hoc WiFi networks
- **Server Message Block (SMB)**
  - Transport protocol used by Windows machines for many purposes
    - File sharing
    - Printer sharing
    - Access to remote Windows services
  - Operates over TCP ports 139 and 445
  - EternalBlue exploits and WannaCry ransomware utilized flaws in the SMB protocol
- **Simple Network Management Protocol (SNMP)**
  - Three versions of SNMP exist
  - SNMPv1 has port security and includes authentication using a shared “community string” sent in cleartext when set to “public”
  - Community string operates like a password and is valid for EVERY node on the network
- **Simple Mail Transfer Protocol (SMTP)**
  - Internet standard for electronic mail transmissions
  - Focus can be on:
    - Direct exploits of the protocol
    - Using open relays
    - Using local relays
    - Phishing attacks
    - SPAM
- **File Transfer Protocol (FTP)**
  - Was the internet standard for file sharing?
  - Insecure protocol that sends data and authentication in cleartext over the network



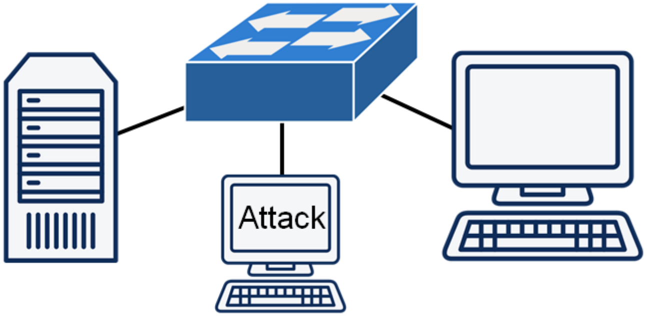
- **DNS Poisoning**


3. Visitor gets IP for Fake Bank server instead
Visitor
Bank
2. Visitor requests DNS for Bank
DNS
Attack
Fake Bank
1. Inject fake DNS record

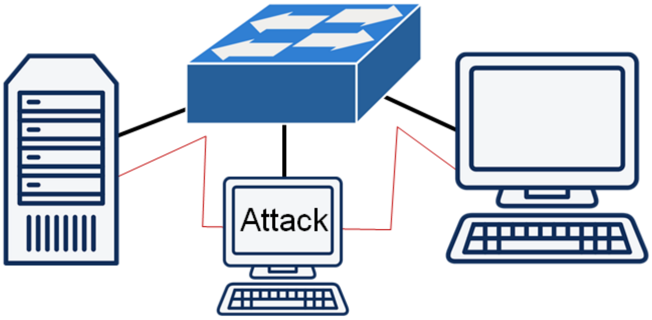
- **DNS Cache Poisoning**
  - Works like DNS poisoning but the poisoning occurs in the DNS cache of the local computer or server
- **Pass the Hash**
  - Attack against the NT LAN Manager (NTLM) authentication system
  - Attacker steals a hashed user credential and reuses it in the Windows authentication system to create a new authenticated session
- **Man-in-the-Middle**
  - ARP spoofing
  - Replay
  - Relay
  - SSL stripping
  - Downgrade



- **ARP Spoofing**
    - Attacker sends falsified ARP messages over the local area network
    - Results in the attacker's MAC being associated with the IP of a valid computer



- **Replay**
    - Attack occurs when valid data is captured by an attacker and is repeated or delayed
    - For example, they could capture a wireless authentication handshake and replay it to gain access to the wireless network as an authenticated client
- **Relay**
    - Attack occurs when the attacker is able to become the man-in-the-middle and acts as a middle man in a communication session

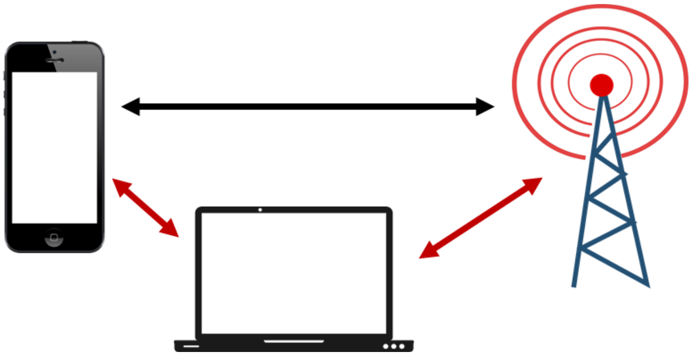


- **SSL Stripping**
    - Attack where a website's encryption is tricked into presenting the user with a HTTP connection instead of a HTTPS connection
- **Downgrade**
    - Attack that attempts to have a client or server abandon a higher security mode to use a lower security mode
    - TLS 1.2 is more secure than SSL 2.0
        - Downgrade attack will cause session to attempt to establish an SSL 2.0 connection
- **Denial of Service**
    - Called a *stress test* in penetration testing

        - Attack that denies resources or a service to an authorized user by exhausting resources
        - Distributed Denial of Service is a variant of this
    - **Network Access Control (NAC) Bypass**
        - NAC can prevent you from gaining access to the network
        - NAC can often be bypassed by spoofing the MAC address of a VOIP device
            - Many VOIP devices don't support 802.1x
            - Their MAC addresses are often whitelisted for NAC
    - **Virtual Local Area Network (VLAN) Hopping**
        - VLANs are often used as logical separation
        - Attack host on a different VLAN to gain access
        - Double tagging the VLAN tag in 802.1Q
        - Switch Spoofing
            - Attempt to auto negotiate with a targeted switch by setting your device to act as a switch
            - Switches get copies of all VLAN traffic and separate them based on tags
- **Wireless-based Vulnerabilities**
    - **Wireless-based Vulnerabilities**
        - Evil Twin
        - Deauthentication attacks
        - Fragmentation attacks
        - Credential harvesting
        - WPS implementation weakness
        - Bluejacking
        - Bluesnarfing
        - RFID cloning
        - Jamming
        - Repeating
    - **Evil Twin**
        - Rogue access point that appears to be legitimate but is setup to eavesdrop on wireless communication
        - Karma Attack
            - Karma Attacks Radio Machines Automatically
            - Devices listen for SSID requests and respond as if they were the legitimate access point
        - Deauthentication (DeAuth) Attack
            - Type of denial of service that targets communication between a user and a wireless access point

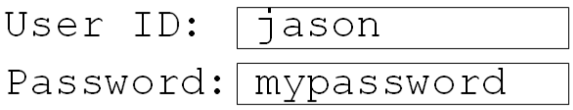
- **Fragmentation Attack**
  - Attacker exploits a network by using datagram fragmentation mechanisms against it
  - A small amount of keying material is obtained from the packet then attempts to send ARP and/or LLC packets with known content to the access point (AP)
  - If the packet is successfully echoed back by the AP then a larger amount of keying information can be obtained from the returned packet
- **Credential Harvesting**
  - Attack that focuses on collecting usernames and passwords from its victims
  - In wireless, this is usually performed by creating a fake Captive Portal
  - ESPortalV2 can be used to setup a fake portal and redirect all WiFi devices connected to the portal for authentication
- **WPS Implementation Weakness**
  - Wi-Fi Protected Setup (WPS) uses a push button configuration method to setup devices
  - Uses an 8-digit WPS Pin to configure them
  - Can be easily brute force attacked because the PIN is authenticated by breaking it in two
  - Reaver and Bully are common attack tools
- **Bluetooth Attacks**
  - Bluejacking
    - Sending unsolicited messages over Bluetooth to Bluetooth-enabled devices such as mobile phones, PDAs, or laptops
  - Bluesnarfing
    - Theft of information from a wireless device through a Bluetooth connection
- **RFID Cloning**
  - Attacker captures the Radio Frequency (RF) signal from a badge or device and can copy it for reuse
- **Jamming**
  - Wireless denial of service attack that prevents devices from communicating with each other by occupying taking over frequency
- **Repeating**
  - Used to capture the existing wireless signal and rebroadcast it to extend the range
  - If not properly configured by the network administrators, this can be an attack vector

- Fake Cellphone Towers
  - Used to capture the International Mobile Subscriber Identity (IMSI) number
  - Can be used to create a man-in-the-middle



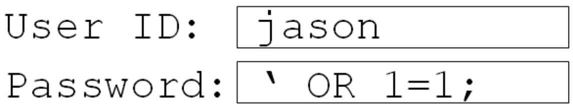
- **Wireless (Wi-Fi) Attack Tools**
  - AirCrack-ng is the most popular suite of tools
    - Monitoring
    - Attacking
    - Testing
    - Cracking

- **Application-based Vulnerabilities**
  - **Application-based Vulnerabilities**
    - Injections
    - Authentication
    - Authorization
    - Cross-site scripting (XSS)
    - Cross-site request forgery (CSRF/XSRF)
    - Clickjacking
    - Security misconfiguration
    - File inclusion
    - Unsecure coding practices
  - **Injection Attacks**
    - Insertion of additional information or code via a data input from a client to the application
    - Most commonly done as SQL inject, but can also be HTML, Command, or Code
    - Prevent this through input validation and using least privilege for the databases
  - **SQL Injection (Structured Query Language)**



```
User ID:  jason
Password: mypassword
```

select * from Users where user_id= 'jason' and password = 'mypassword'



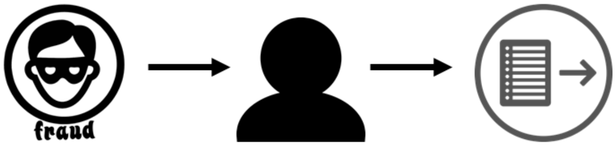
```
User ID:  jason
Password: ` OR 1=1;
```

select * from Users where user_id= 'jason' and password = ' ` OR 1 = 1;

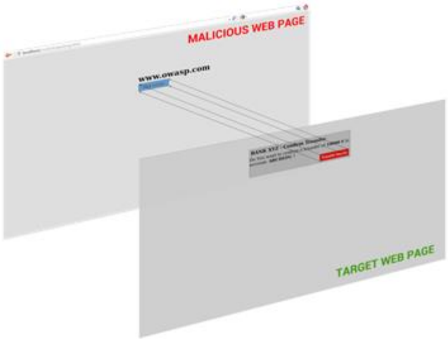
- **Authentication**
  - Credential brute forcing
  - Session hijacking
    - Attacks the web session control mechanism by taking over a session by guessing session token
  - Redirect
    - Sends user to login page to capture credentials
  - Default credentials
  - Weak credentials
    - Easy to crack using dictionary or brute force
- **Kerberos Authentication**
  - Kerberos is a system of tickets that grant devices permission to communicate over a non-secure network and identify themselves
  - Golden Tickets
    - Kerberos Ticket-Granting Tickets (TGT)
    - Can be used to access any Kerberos service
  - Silver Tickets
    - Kerberos Ticket Granting Service (TGS) tickets
    - Can only be used for a specific Kerberos service
- **Parameter Pollution (Authorization)**
  - HTTP parameters are modified in order to conduct a malicious attack
- **Insecure Direct Object Reference (Authorization)**
  - Application provides direct access to an object based on the user-supplied input
- **Cross-Site Scripting (XSS)**
  - Attacker embeds malicious scripting commands on a trusted website
  - Victim in this case is user not the server
    - Stored/persistent
      - Data provided by attacker is saved on server

- Reflected
  - Non-persistent, activated through link on site
- DOM
  - Document Object Model (DOM) is vulnerable
  - Victim's browser is exploited (client-side XSS)

- **Cross-Site Request Forgery**
  - Attacker forces a user to execute actions on web server which they authenticated
  - Attacker cannot see web server's response but this attack can be used to have victim transfer funds, change their password, and more.

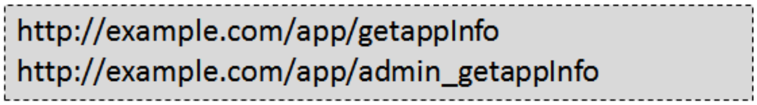


- **Clickjacking**
  - Attack that uses multiple transparent layers to trick a user into clicking on a button or link on a page when they were intending to click on the actual page
  - Conceals hyperlinks under legitimate clickable content



- **Security Misconfiguration**
  - Attacks that rely on the application or server using insecure settings
  - Directory traversal
    - Attack that allows access to restricted directories and for command execution outside of the webserver's root directory
      - http://testsite.com/get.php?f=/var/www/html/get.php
  - Cookie Manipulation
    - DOM-based cookie manipulation that allows a script to write data into the value of a client-stored cookie
- **File Inclusion**
  - Attack that includes a file into a targeted application by exploiting a dynamic file inclusion mechanism

    - Usually occurs due to improper input validation by application
    - File can be included
      - Local
        - ../../uploads/malware.exe
      - Remote
        - https://www.xyz.com/malware.exe
  - **Unsecure Code Practices**
    - Comments in source code
      - Programmers are taught to fully document code
      - Great for developers for maintainability
      - Horrible for security
    - Lack of error handling
      - Applications should fail cleanly on errors
      - Prevents information leakage about the server
    - Verbose error handling
      - Errors can display too much information
      - Great for debugging…horrible for security
    - Hard-coded credentials
      - Source code of a web application has the username and password written into the code instead of using an inclusion file
      - Common issue for applications using PHP, databases, or WordPress
    - Race conditions
      - Flaw that produces unexpected results when the timing of actions can impact other actions
      - Can occur when multithreaded operations are occurring on the same piece of data
    - Unauthorized use of function/unprotected API
      - Allows anyone with network access to send your application a request
      - Designers should implement function-level access control

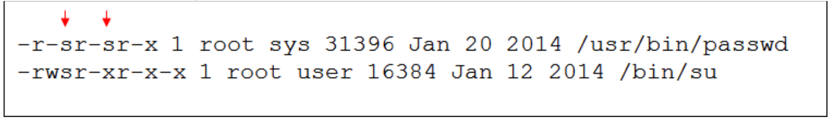


```
http://example.com/app/getappInfo
http://example.com/app/admin_getappInfo
```

    - Hidden elements
      - HTML forms often use hidden elements
        - Fields using <INPUT TYPE=HIDDEN>
      - Could allow sensitive data to be stored in the DOM
    - Lack of code signing
      - Without code signing it is easy for an attacker to modify the code and it go unnoticed

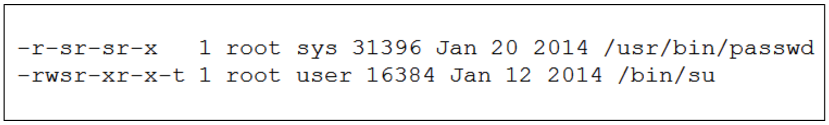
      - Code signing ensures it is digitally signed, which uses a hash digest that is encrypted with a private key certificate to ensure changes have not occurred

- **Local Host Vulnerabilities**
  - **Operating System Vulnerabilities**
    - Windows
    - Mac OS
    - Unix
    - Linux
    - Android
    - iOS
  - **Unsecure Service and Protocol Configuration**
    - Services and daemons run programs constantly in the background of the OS
    - Unsecure services are vulnerable
      - FTP, Telnet, TFTP, and many others
    - Misconfigurations introduce vulnerabilities in secure protocols
      - SSH downgraded to support SSHv1
      - SNMPv3 downgraded to support SMPv1
      - Using WPA instead of WPA2
      - Allow webservers to autonegotiate older SSL certs
      - Default usernames and passwords
- **Privilege Escalation (Linux)**
  - **Privilege Escalation in Linux**
    - Allows a user to run a program or process as a different user with additional permissions
  - **SUID/SGID**
    - Set-User Identification (SUID)
    - Set-Group Identification (SGID)



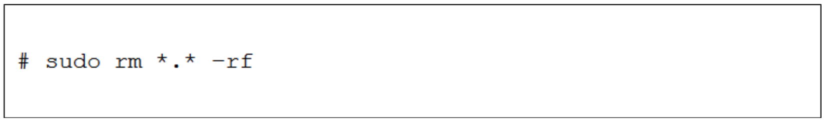
```
-r-sr-sr-x 1 root sys 31396 Jan 20 2014 /usr/bin/passwd
-rwsr-xr-x-x 1 root user 16384 Jan 12 2014 /bin/su
```

    - The ‘s’ in the permissions indicates the program is run as the user or the group
    - Could allow a program to be used for privilege escalation
  - **Sticky Bit**
    - Used for shared folders like /tmp
    - Allows users to create files, read, and execute files owned by other users
    - Attack cannot remove files owned by others



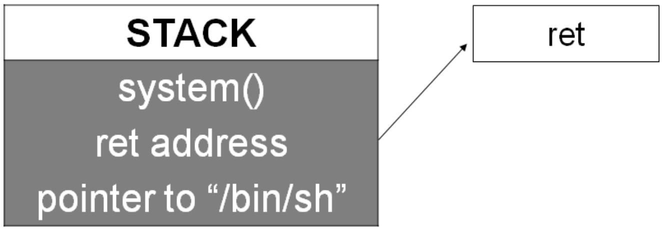
```
-r-sr-sr-x   1 root sys 31396 Jan 20 2014 /usr/bin/passwd
-rwsr-xr-x-t 1 root user 16384 Jan 12 2014 /bin/su
```

- **Unsecure SUDO**
    - SUDO is a program for Unix/Linux systems
    - Allows users to run programs with the privileges of another user
    - By default, the other user is 'root'
    - Works like "Run as Administrator" on Windows



```
# sudo rm *.* -rf
```

- **Ret2libc**
    - An attack technique that relies on overwriting the program stack to create a new stack frame that calls the system function
    - Stands for "return to library call"



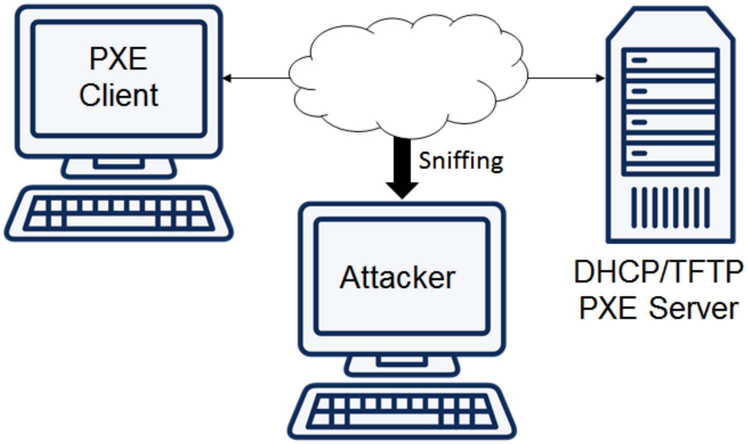
- **Privilege Escalation (Windows)**
    - **Privilege Escalation in Windows**
        - Allows a user to run a program or process as a different user with additional permissions
    - **Cpassword**
        - Name of the attribute that stores the passwords in a Group Policy preference item
        - Stored in the SYSVOL folder on the Domain Controllers in encrypted XML file
        - Easily decrypted by any authenticated user in the domain, though
    - **Clear Text Credentials in LDAP**
        - If SSL is not enabled for LDAP, credentials are sent over the network in clear text
        - Use the Insecure LDAP Bind script to check for this in PowerShell

```
.\Query-InsecureLDAPBinds.ps1 -ComputerName dc1.corp.com -Hours 24
```

- You receive a CSV file as output showing which accounts are vulnerable

```
"IPAddress","Port","User","BindType"
"10.0.0.3","60901",”CORP\Administrator","Simple"
"[::1]","65445",”CORP\Administrator","Simple"
```

- **Kerberoasting**
  - Any domain user account that has a service principal name (SPN) set can have a service ticket (TGS)
  - Ticket can be requested by any user in the domain and allows for offline cracking of the service account plaintext password
- **Credentials in LSASS**
  - Local Security Authority Subsystem Service
  - Process in Windows that enforces the security policy of the system
  - Verifies users when logging on to a computer or server
  - Performs password changes
  - Creates access token (ie, Kerberos)
- **Unattended Installation**
  - Clear text credentials of Preboot Execution Environment (PXE) could be captured using network sniffers

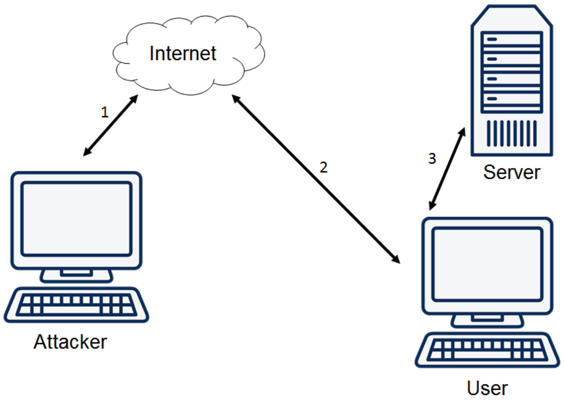


- **SAM Database**
  - Security Account Manager is a database file that stores the user passwords in Windows as a LM hash or NTLM hash
  - File is used to authenticate local users and remote users
  - Passwords can be cracked offline if the SAM file is stolen
- **DLL Hijacking**
  - Dynamic Link Library (DLL) provides a method for sharing code and allows a program to upgrade its functionality without requiring re-linking or re-compiling of the application
  - Hijacking is a technique used to load a malicious DLL in the place of an accepted DLL

        - Commonly used by malware to achieve persistence on the victim machine
    - **Exploitable Services**
        - Attacker uses the way services normally operate to cause an unintended program to run
        - Examples
            - Unquoted service path call in file system
                - C:\Dion\My Files\server.exe Normal
                - C:\Dion\My\server.exe Malicious
            - Writable services
                - Using PSExec, a service can be replaced with a custom service that runs a command shell (cmd.exe)
    - **Unsecure File and Folder Permissions**
        - Older versions of Windows allow administrators to access any non-admin user’s files and folders
        - Can lead to DLL hijacking and malicious file installations on a non-admin targeted user
    - **Keylogger**
        - Surveillance technology used to monitor and record the keystrokes of a victim user
        - Can be software or hardware-based
    - **Scheduled Tasks**
        - Attacker uses the Windows Task Scheduler to create callbacks and retain persistence
        - Arbitrary code could be executed at a certain time or in response to an event.
- **Privilege Escalation**
    - **Kernel Exploits**
        - Unpatched Windows and Linux systems are vulnerable to many different exploits
        - Search CVE’s for various versions of Windows or Linux to determine what exploits exist
        - Metasploit has a library of existing exploits
    - **Default Account Settings**
        - Default administrator accounts can be exploited
        - Guest accounts should be disabled, but are enabled by default on most systems
    - **Sandbox Escape**
        - Shell upgrade
            - Restricted shells (like rbash) are exploited to gain an upgraded shell

- Virtual Machines
  - Escaping the VM sandbox can lead to exploit of the underlying hardware and puts other hosted VMs are risk
- Container
  - Share a common operating system
    If you can compromise that system, you can compromise every container that relies upon it
- **Physical Service Security**
  - Cold boot attack
  - JTAG debug
  - Serial console
- **Cold Boot Attack**
  - A side channel attack where an attacker has physical access to the system
  - User is able to retrieve the encryption keys from a running operating system after using a cold reboot to restart the machine
- **JTAG Debug**
  - JTAG is a standard for verifying designs and testing printed circuit boards
    - Diagnostic connection
  - Port use for debugging, probing, and programming
  - With breakpoints setup, the JTAG can be used to read registers from motherboard and read arbitrary memory locations
- **Serial Console**
  - Many network devices still have serial console connections (routers and switches)
  - If attacker can get physical access to the device then they can connect to the device over the serial port
  - Lower security enabled (if any) on these ports

- **Lateral Movement**
  - **Lateral Movement**

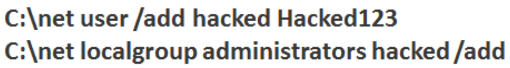


  - **Remote Procedure Call & Distributed Component Object Model**

        - Remote Procedure Call (RPC)
            - Protocol used in Windows to allow the remote execution of code on a remote computer or server
        - Distributed Component Object Model (DCOM)
            - Proprietary Microsoft technology for communication between software components on networked computers
        - PsExec
            - Light-weight telnet-replacement that lets you execute processes on other systems with full interactivity for console applications without having to manually install client software
        - Windows Management Instrumentation (WMI)
            - Set of specifications from Microsoft for consolidating the management of devices and applications in a network from Windows computing systems
    - **PS Remoting and WinRM**
        - PowerShell Remoting
            - Allows a computer to receive Windows PowerShell remote commands
        - Windows Remote Management (WinRM)
            - Allows administrators to remotely run management scripts using the WS-Management Protocol (based on SOAP)
            - Windows Remote Management is run on server
            - Windows Remote Shell (WinRS) is run on client
        - Server Message Block (SMB)
            - Mostly Windows PCs but Linux using Samba too.
    - **Remote Desktop Protocol (RDP)**
        - Allows remote access to a machine over the network as if you were sitting right in front of it
        - Provides GUI access through an RDP client
    - **Apple Remote Desktop**
        - Allows remote access to a machine over the network through a GUI
        - Recent versions allow for an encrypted AES 128-bit tunnel to be created from the machine being controlled
    - **Virtual Network Computing (VNC)**
        - Originally used in thin client architectures
        - Operates much like RDP, but a cross-platform solution for Windows, Linux, and OS X
    - **X11 Forwarding**
        - X-windows/X-server is the GUI for Linux
            - Known collectively as X11
        - X11 forwarding provides a GUI by forwarding the

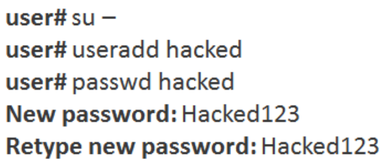
        - X-windows/X-server over an SSH connection
    - **Telnet**
        - Permits sending commands to remote devices
        - Information is sent in plain text
        - Telnet should never be used over an insecure connection and is a huge security risk to use
    - **SSH**
        - Works like telnet, but uses encryption to create a secure channel between the client and the server
        - SSH should always be used instead of telnet
    - **RSH and Rlogin**
        - Remote Shell (RSH)
            - Command line program used to execute shell commands as another user on another computer over the network
            - Rsh is unsecure because it doesn't use encryption, therefore SSH should be used instead
        - Rlogin
            - Rsh created as part of rlogin package in BSD Unix
            - Allowed a user to login and issue commands on another Unix computer over a TCP/IP network
- **Persistence**
    - **Persistence**
        - Method to maintain access to a victim machine
    - **Scheduled Jobs or Tasks**
        - Scheduled Jobs (cron)
            - Cron jobs are used in Unix, Linux, and OS X
            - Allows a script or command to be run at periodic times, dates, or intervals
            - Export_dump.sh is run Every Saturday (6) @ 23:45
        - Scheduled Tasks (at)
            - Windows command-line program to schedule tasks
            - Task Scheduler is the GUI version of the program
    - **Daemons**
        - Background process that exists for the purpose of handling periodic service requests that a computer system expects to receive
        - For example, sshd is the SSH daemon
        - In Windows, these are called "services"
    - **Backdoors**
        - Method to bypass normal authentication or encryption in a computer system

        - May take the form of a hidden part of a program (such as a trojan or rootkit)
        - Default passwords are considered a
          backdoor when they are not changed by the user
    - **Trojans**
        - Any malicious computer program that misleads users to their true intent
        - A piece of software that pretends to be a game but allows the attacker access to the system
        - Used as a technical form of social engineering
    - **Creating New Users**
        - Attacker creates new user accounts
            - Can be created as regular or admin level users
        - Command Line (Windows)
            - net user /add [username] [password]
                - Net localgroup administrators [username] /add



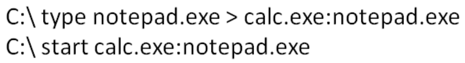
```
C:\net user /add hacked Hacked123
C:\net localgroup administrators hacked /add
```

        - Shell (Linux)



```
user# su –
user# useradd hacked
user# passwd hacked
New password: Hacked123
Retype new password: Hacked123
```

- **Covering Your Tracks**
    - **Covering Your Tracks**
        - Erase, Modify, or Disable the Evidence
        - Clear Log Files
        - Hiding files and folders
    - **Erase, Modify, or Disable the Evidence**
        - Removing any unneeded files or tools that were added to the victim's machine
        - Hiding other files and resources in hidden or uncommon locations
            - Linux, Unix, OS X
                - Create a folder beginning with .
            - Windows
                - Hide stuff in the System32 or User folders
                - Apply hidden attribute
                - Use alternate Data Streams



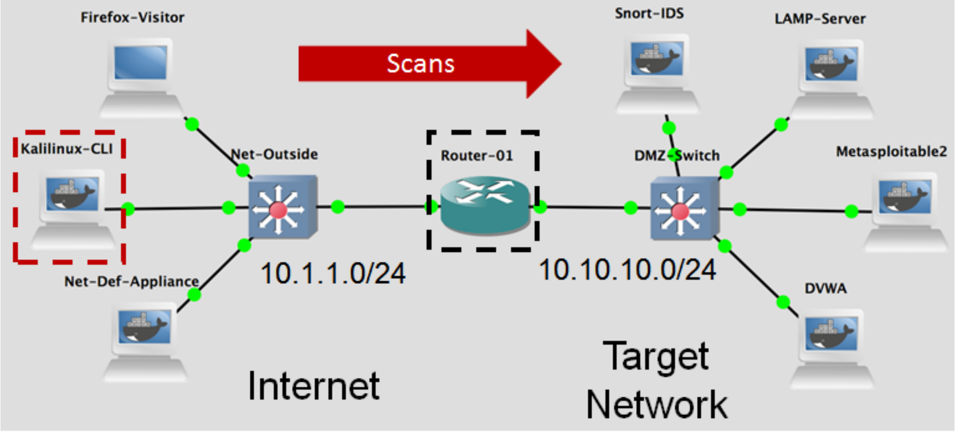
```
C:\ type notepad.exe > calc.exe:notepad.exe
C:\ start calc.exe:notepad.exe
```

            - Hide files in the slack space
    - **Clearing the Log Files**
        - Cleaning up traces of our activities in various log files
            - Windows
                - System logs, Application logs, Security logs, Event logs
            - Linux
                - Logs are usually stored in /var/logs
        - IMPORTANT
            - Penetration testers DO NOT usually modify or delete any of the logs…check your scope of work!
    - **Modifying the Log Files**
        - Log files are just text (they can be edited)
        - Timestomp can be used to modify the access time of a file
        - Change the files ownership to original user
        - IMPORTANT
            - Penetration testers DO NOT usually modify or delete any of the logs…check your scope of work!
    - **Timestomp**
        - touch (Linux, Unix, OSX)
            - Updates time to the current time
        - ctime (Linux, Unix, OSX)
            - Change the time to a given date/time
        - Meterpreter has built-in tool

Domain 4 - Penetration Testing Tools

- **Penetration Testing Tools**
    - **Domain 4: Penetration Testing Tools**
        - Use Nmap to conduct information gathering
        - Compare and contrast various tools
        - Analyze tool output or data
            - Password cracking, pass the hash, injections, setting up bind or reverse shells, upload web shell
        - Analyze a basic script
            - Bash, Python, Ruby, and Powershell
    - **What kinds of questions can I expect on test day?**
        - Majority are "given a scenario"
        - "Compare and Contrast" various tools
        - Expect to get simulations from here
- **Nmap Usage**
    - **Nmap**
        - Command line tool that sends specially crafted packets to the target host(s) of a network
        - Discovers the hosts and services being run based on the responses received
    - **Port Scan Results**
        - Open
            - Application is accepting connections
        - Closed
            - No application is listening
        - Filtered
            - Probes aren't reaching the port
            - Usually indicates a firewall
    - **Nmap -sS**
        - SYN Scan (default and most popular)
        - Can scan 1000 ports per second
        - Never completes the TCP connection
    - **Nmap -sT**
        - TCP Connect Scan
        - Uses the Operating System to send packets
        - Completes the TCP connection (less stealthy)
    - **Nmap -p**
        - Specifies the port to scan (override defaults)

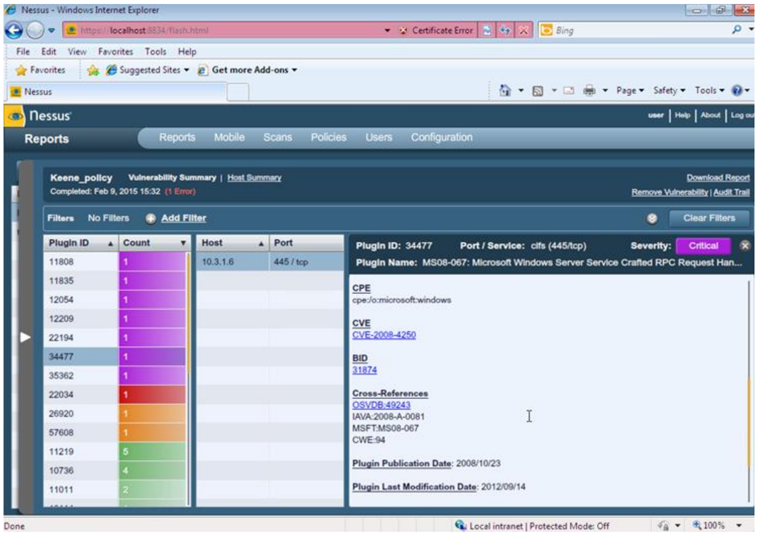
  - **Nmap -O**
    - Enables OS detection by using fingerprinting of the TCP/UDP packet received
  - **Nmap -Pn**
    - Skips the host discovery
    - Treats all hosts in the range as online
  - **Nmap –iL**
    - Scan targets from a text file
  - **Nmap –T**
    - Sets the timing for the scan
      - T0 – Paranoid (one port every five minutes)
      - T1 – Sneaky (one port every 15 seconds)
      - T2 – Polite
      - T3 - Normal
      - T4 - Aggressive
      - T5 - Insane
  - **Nmap Output**
    - -oN Normal output format
      - *nmap -oN outputfile.txt target*
    - -oG Grepable output format
      - *nmap -oG outputfile.txt target*
    - -oX XML output format
      - *nmap -oX outputfile.xml target*
    - -oA Combined format with all of the above
      - *nmap -oA outputfile target*
- **Nmap Usage (Demo)**
  - **Nmap Demo Environment**
    - Start all our scans from the Kali Linux CLI



- **Use Cases for Tools**
  - **Reconnaissance**
    - Collecting information before attacking an IT system
    - Usually conducted using open source research or passive collection
    - Tools
      - Whois, Nslookup, Theharvester, Shodan, Recon-NG, Censys, Aircrack-NG, Kismet, WiFite(2), Wireshark, Hping, SET, Nmap, Metasploit framework
  - **Enumeration**
    - Establishes an active connection to the targets to discover potential attack vectors
    - Usually conducted active techniques and fingerprinting
    - Tools
      - Nslookup, Wireshark, Hping, Nmap
  - **Vulnerability Scanning**
    - In-depth scanning of a target to determine its vulnerabilities
    - Uses automated tools to determine missing patches and incorrect configurations
    - Tools
      - Nikto, OpenVAS, Nessus, SQLmap, W3AF, OWASP ZAP, Nmap, Metasploit Framework
  - **Credential Attacks**
    - Offline password cracking
      - John the Ripper, Mimikatz, Cain and Abel, Hashcat, AirCrack-NG
    - Brute-forcing services
      - SQLmap (for databases), Medusa, Hydra, W3AF, Mimikatz, Cain and Abel, Patator, Aircrack-NG
  - **Persistence**
    - Maintaining a foothold into the network or victim system
    - Tools
      - SET, BeEF, SSH, NCAT, NETCAT, Drozer, Powersploit, Empire, Metasploit framework
  - **Configuration Compliance**
    - Ensuring a system meets a given security baseline or policy
    - Tools
      - Nikto, OpenVAS, Nessus, SQLmap, Nmap
  - **Evasion**
    - Hide from system administrators or defenders
    - Tools
      - Proxychains, SET, Metasploit Framework, Route
      -

- ○ **Decompilation**
  - ▪ Reversing an executable into human readable code
  - ▪ Tools
    - IDA, Hopper, Immunity debugger, APK Studio, APKX
- ○ **Forensics**
  - ▪ Tools used to collect and analyze digital evidence for crimes and analysis
  - ▪ Tools
    - foremost, FTK, EnCase, Tableau
- ○ **Debugging**
  - ▪ Process of finding and resolving defects in a computer program
  - ▪ Tools
    - Ollydbg, Immunity debugger, GDB, WinDBG, IDA Pro, APK Studio, APKX
- ○ **Software Assurance**
  - ▪ Fuzzing
    - Peach and AFL
  - ▪ Security Testing
    - Static Application Security Testing (SAST)
    - Dynamic Application Security Testing (DAST)
    - Findsecbugs, SonarQube, and YASCA (Yet Another Source Code Analyzer)

- **Scanners**
  - ○ **Scanners**
    - ▪ Nessus, OpenVAS, Nikto, SQLmap
  - ○ **Nessus**


Nessus - Windows Internet Explorer
https://localhost:8834/flash.html
Certificate Error
Bing
File Edit View Favorites Tools Help
Favorites
Suggested Sites
Get more Add-ons
Nessus
Page Safety Tools
Nessus
user | Help | About | Log out
Reports
Reports Mobile Scans Policies Users Configuration
Download Report
Remove Vulnerability | Audit Trail
Add Filter
Clear Filters
Done
Local intranet | Protected Mode: Off
100%

Keene_policy Vulnerability Summary | Host Summary
Completed: Feb 9, 2015 15:32 (1 Error)

Filters No Filters

Plugin ID	Count
11808	1
11835	1
12054	1
12209	1
22194	1
34477	1
35362	1
22034	1
26920	1
57608	1
11219	5
10736	4
11011	2

Host	Port
10.3.1.6	445 / tcp

Plugin ID: 34477 Port / Service: cifs (445/tcp) Severity: Critical
Plugin Name: MS08-067: Microsoft Windows Server Service Crafted RPC Request Han...

CPE
cpe:/o:microsoft:windows

CVE
CVE-2008-4250

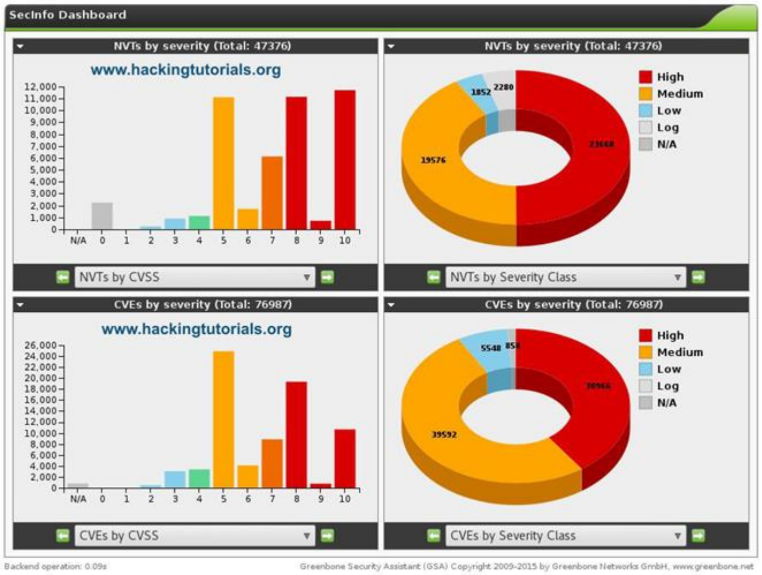
BID
31874

Cross-References
OSVDB:49243
IAVA:2008-A-0081
MSFT:MS08-067
CWE:94

Plugin Publication Date: 2008/10/23

Plugin Last Modification Date: 2012/09/14

- **OpenVAS**



- **Nikto**
  - Open-source web server scanner that identifies outdated versions and server misconfigurations such as multiple index files, and HTTP server options
  - Run as a Perl script with the target IP
- **SQLmap**
  - Open-source pentest tool to automate detecting\exploiting SQL injection flaws

- **Credential Testing Tools**
  - **Credential Testing Tools**
    - Hashcat, Medusa, Hydra, CeWL, John the Ripper, Cain and Abel, Mimikatz, Patator, Dirbuster, W3AF
  - **Hashcat**
    - One of the fastest password recovery tools
    - Runs on:
      - Linux
      - OS X
      - Windows
    - Replies on CPU or GPU to crack passwords
  - **Hydra**
    - Brute-force network log-on cracking tool
    - Repeatedly attempts to login to a system
  - **Medusa**
    - Brute-force password attack tool

        - Supports numerous remote authentication protocols (rlogin, ssh, telnet, http, etc)
        - Supports multi-threading
        - Faster than Hydra
    - **CeWL**
        - Tool to create a custom wordlist or dictionary
        - Searches a target website for words meeting criteria set as inputs to CeWL
    - **John the Ripper**
        - Password cracking tool
        - Runs on Linux/Unix, OS X, Windows, & more
        - Supports dictionary and brute-force attacks
    - **Cain and Abel**
        - Password cracker on Windows
        - Conducts network sniffing and hash cracking
    - **Mimikatz**
        - Targets Windows machines to extract plaintext passwords, hashes, PIN codes, and Kerberos tickets from the machine's memory
        - Can be used for pass-the-hash, pass-the-ticket, and creating Golden Tickets
    - **Patator**
        - Multi-purpose brute-force attack tool
        - Supports modules for different target services
    - **Dirbuster**
        - Brute-force tool for directories and file names on web/application servers
    - **W3AF**
        - Web Application Attack and Audit Framework
        - Tool to find web app vulnerabilities
- **Debuggers**
    - **Debuggers**
        - Ollydbg, Immunity Debugger, GDB, WinDBG, IDA
    - **Ollydbg**
        - Assembler level debugger for Windows
        - Useful for binary code analysis without source code being available
    - **Immunity Debugger**
        - Used to write exploits, analyze malware, and reverse engineer binary files
        - Supports Python APIs and execution
    - **GDB (GNU Debugger)**
        - Runs on Unix and Linux systems
            - Supports Ada, C, C++, Obj-C, Pascal, Fortran, Go, Java, and other languages

  - **WinDBG**
    - Debugger for Windows (created by Microsoft)
  - **IDA (Interactive Disassembler)**
    - Generates assembly language code from executable code
    - Graphical user interface and supports executables from multiple operating systems
- **Software Assurance**
  - **Software Assurance**
    - Findbugs and Findsecbugs, Peach, AFL, SonarQube, YASCA
  - **Findbugs and Findsecbugs**
    - Used to conduct security audits of Java apps before deployment
  - **Peach**
    - Automated security testing platform to identify vulnerabilities by conducting fuzzing
  - **AFL (American Fuzzy Lop)**
    - Open-source, text-based security fuzzer that requires nearly no configuration to operate
  - **SonarQube**
    - Open-source platform that performs automatic static code reviews to find vulnerabilities and bugs in over 20 programming languages
  - **YASCA (Yet Another Source Code Analyzer)**
    - Open-source software code scanner that uses plug-ins to add languages and features
- **OSINT**
  - **OSINT**
    - Whois, Nslookup, Foca, The Harvester, Shodan, Maltego, Recon-NG, Censys
  - **Whois**
    - Query and response protocol for internet resources
  - **Nslookup**
    - Command-line tool for querying DNS
  - **Foca**
    - Fingerprinting Organizations with Collected Archives (FOCA)
    - Used to find metadata and hidden info in docs
  - **The Harvester**
    - Gathers emails, subdomains, hosts, employee names, open ports, and banners
  - **Shodan**
    - Search engine that lets you find webcams, routers, servers, and more on the internet

- **The Maltego**
  - Commercial software for conducting open-source intelligence and visually connecting the relationships
- **Recon-NG**
  - Open-source web reconnaissance framework written in Python
- **Censys**
  - Search engine for hosts and networks across the internet with data about their configuration
  - Contains search interface, report builder, and SQL engine

- **Wireless**
  - **Wireless**
    - Aircrack-NG, Kismet, WiFite
  - **Aircrack-NG**
    - Wireless hacking suite that consists of scanner, packet sniffer, and password cracker
  - **Kismet**
    - Wireless hacking suite that consists of scanner, packet sniffer, and IDS
  - **WiFite**
    - Automated wireless attack tool
    - Menu-driven Python script
- **Web Proxies**
  - **Web Proxies**
    - OWASP ZAP, Burp Suite
  - **OWASP ZAP**
    - Open-source web application security scanner
    - Can be used as a proxy to manipulate traffic running through it (even https)
  - **Burp Suite**
    - Graphical tool for web application security
    - Allows for the interception, inspection, and modification of raw traffic passing through it
- **Social Engineering Tools**
  - **Social Engineering Tools**
    - SET, BeEF
  - **Social Engineer Toolkit (SET)**
    - Open-source penetration testing framework for social engineering
  - **BeEF (Browser Exploitation Framework)**
    - Pentest tool focused on the web browser
    - Used to hook a web browser for launching command modules and attacks

- **Remote Access Tools**
    - **Remote Access Tools**
        - SSH, Netcat, Ncat, Proxychains
    - **SSH (Secure Shell)**
        - Works like telnet, but uses encryption to create a secure channel between the client and the server
        - SSH should always be used instead of telnet
    - **Netcat**
        - Command-line tool for reading, writing, redirecting, and encrypting data on a network
        - Referred to as the swiss army knife of pentest
    - **Ncat**
        - Command-line tool for reading, writing, redirecting, and encrypting data on a network
        - From makers of Nmap as update to Netcat
    - **Proxychains**
        - Tool that forces TCP connections from all applications to run through a proxy
        - Can be TOR or other HTTP/SOCKS proxy
- **Networking Tools**
    - **Networking Tools**
        - Wireshark, Hping
    - **Wireshark**
        - Open-source packet analyzer
    - **Hping**
        - Command-line TCP/IP packet assembler and analyzer
        - Can use TCP, UDP, ICMP, RAW-IP protocols
- **Mobile Tools**
    - **Mobile Tools**
        - Drozer, APKX, APK Studio
    - **Drozer**
        - Provides tools to use and share public exploits for the Android operating system
        - Complete security audit and attack framework
    - **APKX (Android APK Decompilation for the Lazy)**
        - Python wrapper to extract Java source code directly from Android APK files
    - **APK Studio**
        - Cross-platform IDE for reverse engineering and recompiling Android application binaries

- **Miscellaneous Tools**
    - **Miscellaneous Tools**
        - Searchsploit, Powersploit, Responder, Impacket, Empire, Metasploit Framework (MSF)
    - **Searchsploit**
        - Command-line search tool for the Exploit-DB
        - Allows for offline searches through local repo
    - **Powersploit**
        - Collection of Microsoft PowerShell modules for use in penetration testing
        - Considered a post-exploitation framework
    - **Responder**
        - LLMNR, NBT-NS, and MDNA poisoner
        - Used to answer specific queries based on name suffix on the network
    - **Impacket**
        - Collection of Python classes for working with network protocols
        - Focused on low-level program access for SMB and MSRPC protocol implementation
    - **Empire**
        - PowerShell and Python post-exploitation agent
    - **Metasploit Framework (MSF)**
        - Open-source framework that provides scanners, payloads, and other tools
- **Intro to Programming**
    - **What is programming?**
        - Creating a sequence of instructions to tell a computer how to perform a specific task
        - Instructions are stored as a program or script
    - **What is a script?**
        - Short program that is used to automate tasks
    - **What kinds of questions can I expect on test day?**
        - Given a scenario, analyze a basic script (Bash, Python, Ruby, and PowerShell)
            - Logic and Common operations
            - Encoding/decoding
            - I/O
            - Error handling
            - Variables, Arrays, Substitutions
- **Programming Concepts**
    - **Comments**
        - Bash, Python, Ruby, and PowerShell all use a # to signify the code is commented

BASH

```
#!/bin/bash
#Hello World Bash Script
echo "Hello World!"
```

PowerShell

```
#Hello World PowerShell Script
echo "Hello World!"
```

Python

```
#Hello World Python Script
print (“Hello World!”)
```

Ruby

```
#!/usr/bin/ruby
#Hello World Ruby Script
puts "Hello World!"
```

- **Variable**
  - Variables are used to represent any value and can be changed during the execution of the program

BASH

```
#BASH variable declaration
CustomerName = Jason
```

Note: Would use $CustomerName to reference

PowerShell

```
#PowerShell variable declaration
$CustomerName = Jason
```

Python

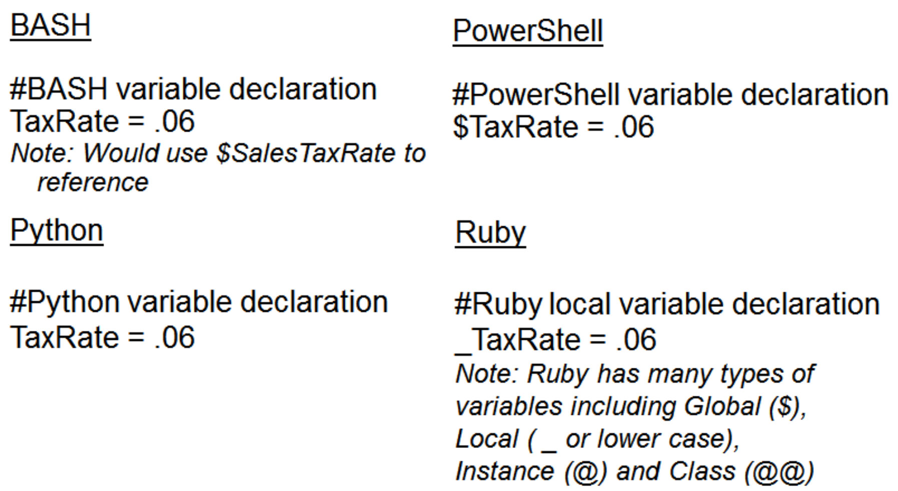
```
#Python variable declaration
CustomerName = Jason
```

Ruby

```
#Ruby local variable declaration
_CustomerName = Jason
```

Note: Ruby has many types of variables including Global ($), Local (_ or lower case), Instance (@) and Class (@@)

- **Constants**
  - Constants are used to define a set value across the entire program and cannot be changed



BASH

```
#BASH variable declaration
TaxRate = .06
```

Note: Would use $SalesTaxRate to reference

PowerShell

```
#PowerShell variable declaration
$TaxRate = .06
```

Python

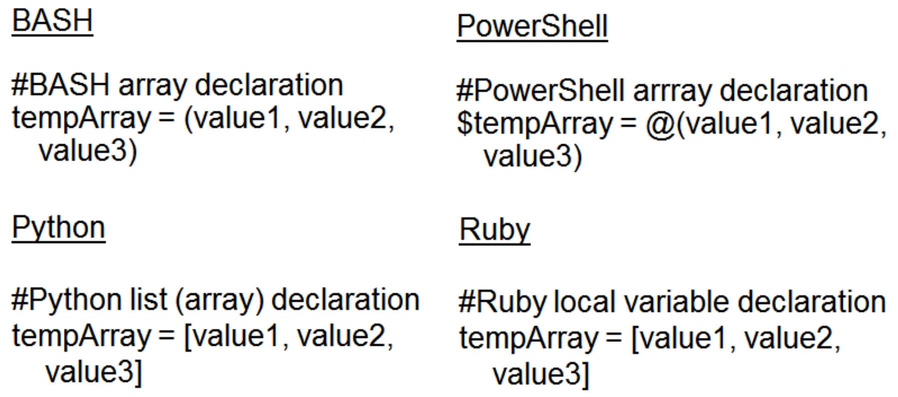
```
#Python variable declaration
TaxRate = .06
```

Ruby

```
#Ruby local variable declaration
_TaxRate = .06
```

Note: Ruby has many types of variables including Global ($), Local (_ or lower case), Instance (@) and Class (@@)

- **Arrays (Basic or Indexed)**
  - Can store multiple values and be referenced from a single name (like a list of variables)



BASH

```
#BASH array declaration
tempArray = (value1, value2,
  value3)
```

PowerShell

```
#PowerShell arrray declaration
$tempArray = @(value1, value2,
  value3)
```

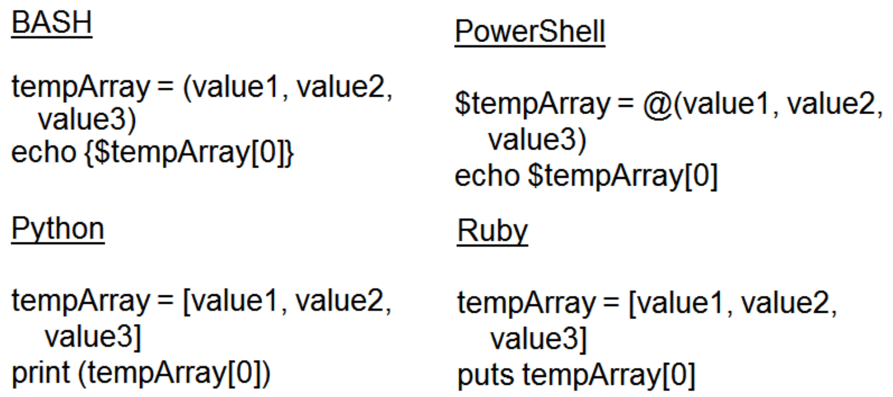
Python

```
#Python list (array) declaration
tempArray = [value1, value2,
   value3]
```

Ruby

```
#Ruby local variable declaration
tempArray = [value1, value2,
   value3]
```

- **Getting Information Out Of An Array**
  - How do you recall the information stored in an array?



BASH

```
tempArray = (value1, value2,
  value3)
echo {$tempArray[0]}
```

PowerShell

```
$tempArray = @(value1, value2,
   value3)
echo $tempArray[0]
```

Python

```
tempArray = [value1, value2,
   value3]
print (tempArray[0])
```

Ruby

```
tempArray = [value1, value2,
   value3]
puts tempArray[0]
```

- **Named Arrays or Associative Arrays**
  - Works more like a table in a database
  - Data is stored in an array based on a name

BASH

```
declare –A tempArray
tempArray[name]=Jason
echo ${tempArray[name]}
```

PowerShell

```
$tempArray = @{}
$tempArray.name = ‘Jason’
echo $tempArray.name
```

Python

```
tempArray = [{“name”:”Jason”}]
print (tempArray[“name”])
```

Ruby

```
_tempArray = {“name” => “Jason”}
Puts _tempArray[“name’]
```

- **Comparisons**

	BASH	Python	PowerShell	Ruby
Equal	-eq	==	eq	==
Not Equal	-ne	!= or <>	ne	!=
Greater Than	-gt	>	gt	>
Greater or Equal	-ge	>=	ge	>=
Less Than	-lt	<	lt	<
Less or Equal	-le	<=	le	<=

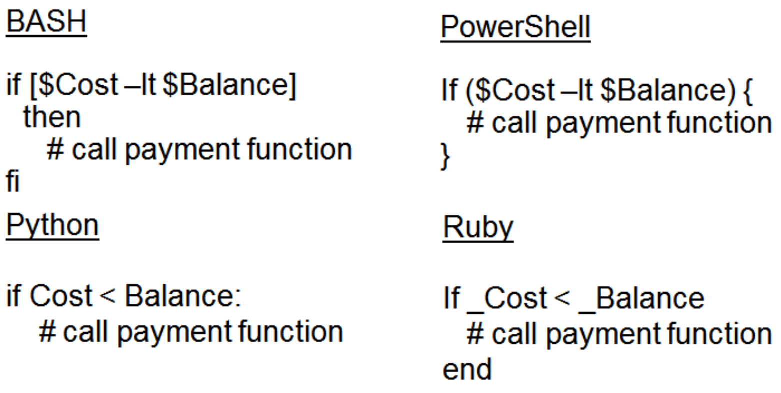
BASH string operations

= or == (equal)
!= (not equal)
> (greater)
< (less)

Ruby

=== (equal in case statements)

- **IF**
  - Conditional statement used to execute a portion of the code only IF the condition is true



BASH

```
if [$Cost –lt $Balance]
  then
     # call payment function
fi
```

PowerShell

```
If ($Cost –lt $Balance) {
   # call payment function
}
```

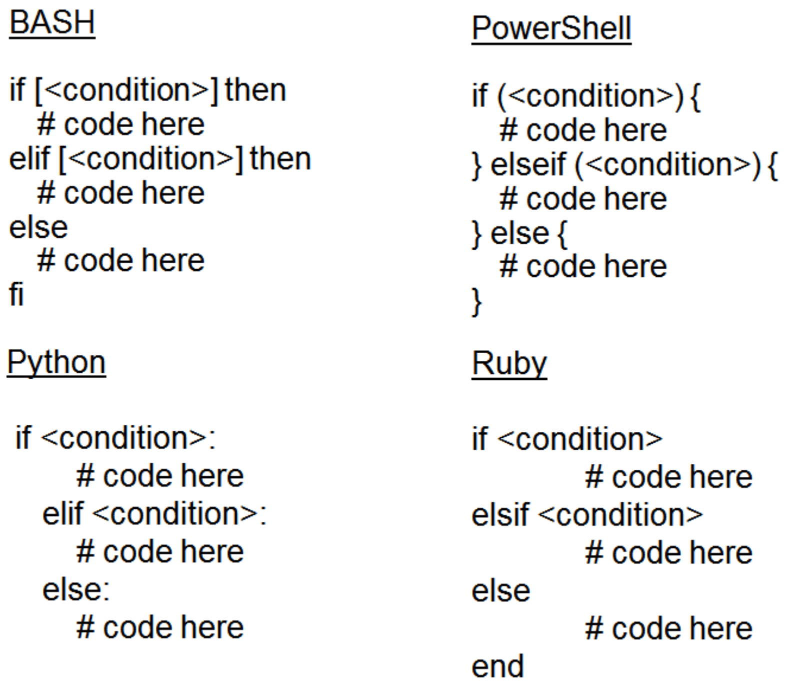
Python

```
if Cost < Balance:
    # call payment function
```

Ruby

```
If _Cost < _Balance
   # call payment function
end
```

- **IF, ELSEIF, ELSE**



BASH

```
if [<condition>] then
   # code here
elif [<condition>] then
   # code here
else
   # code here
fi
```

PowerShell

```
if (<condition>) {
   # code here
} elseif (<condition>) {
   # code here
} else {
   # code here
}
```

Python

```
if <condition>:
       # code here
    elif <condition>:
       # code here
    else:
       # code here
```

Ruby

```
if <condition>
              # code here
elsif <condition>
              # code here
else
              # code here
end
```

- **Loops**
  - For, Do While, and While commands are used to repeat code for a given amount of time

BASH

```
while [$this -eq $that]
do
   # commands
done
```

PowerShell

```
Do {
   # commands
} While ($this -eq $that)
```

Python

```
tempArray = [value1, value2,
value3]
for x in tempArray:
print (x)
```

Ruby

```
tempArray = [value1, value2,
value3]
for x in tempArray
          puts x
end
```

- **String Operations**
    - These commands are used to manipulate data that is of the string format (like words)

BASH

```
testString = “Test String”
echo ${#testString}
echo ${testString:0:4}
echo ${testString/Test/Testing}
```

PowerShell

```
$testString = “Test String”
echo $testString + “2”
$testString.Substring(0,4)
```

Python

```
testString = “Test String”
print testString.uppercase
print testString.replace(“Test”,
  “Testing”)
```

Ruby

```
testString = “Test String”
print testString.length
print testString.upcase
```

- **Input/Output (Keyboard and Monitor)**
    - Functions used to enter data from keyboard and output to the monitor

BASH

```
echo “Please enter your name:”
read userName
echo “Hello $username!”
```

PowerShell

```
$userName = Read-Host -Prompt
  'Please enter your name:’
Echo “Hello “ + $username
```

Python

```
userName = input(‘Please
   enter your name: ')
print (“Hello “, userName)
```

Ruby

```
puts “Please enter your name:”
userName = gets
puts “Hello “ + userName
```

- **Input/Output (Files)**
    - Functions used to input and output data from files

BASH

```
tempFile=$(<test.txt)
echo "$tempFile“
```

PowerShell

```
$tempFile = Get-Content - Path
  C:\test.txt
```

Python

```
tempFile = open(‘test.txt', 'w')
tempFile =
open(‘//Host/share/test.txt', 'w')
```

Ruby

```
File.open(‘test.txt’)
File.open(test.txt, 'w') { |file|
file.write(“test text") }
```

- **Error Handling**
    - Functions used to input and output data from files

BASH

```
cd $some_directory
if [ "$?" = "0" ]; then
  rm *
Else
  echo "Cannot change
  directory!" 1>&2 exit
fi
```

PowerShell

```
Function Do-Something {
 [CmdletBinding()]
 param()
 try {
 if  (-not (Test-Path –Path
  'C:\test.txt')) {
 Write-Host  'The file does not
  exist'
 }
 Write-Host  'The file does exist'
 } catch  {
         }
}
```

Python

```
try:
        print 1/0
except ZeroDivisionError:
        print "You can't divide by zero!"
```

Ruby

```
def raise_and_rescue
   begin
     puts 'I am before the raise.‘
     raise 'An error has occurred.‘
     puts 'I am after the raise.‘
   rescue
     puts 'I am rescued.‘
   end
   puts 'I am after the begin block.‘
 end
 raise_and_rescue
```

- **Encoding/Decoding**
  - Various encoding standards are used in programming
  - ASCII (American Standard Code for Information Interchange)
  - ISO/IEC 10646 (Universal Coded Character Set)
  - Unicode
    - UTF-8
    - UTF-16
    - UTF-32

- **BASH Sample Script**

```
#!/bin/bash

if [ "$1" != "" ]
then
         TARGETFILE=$1
        REPORTDIR="/var/log/nmap"
        for TARGET in $(cat $TARGETFILE)
        do
                echo "Scanning $TARGET ..."
                /usr/bin/nmap -oN $REPORTDIR/$TARGET.nmap -sS -sU -T4 -A -v -PE -PP -PS80,443 -PA3389
                 -PU40125 -PY -g 53 --script "default or (discovery and safe)" $TARGET >>
                 $REPORTDIR/$TARGET.nmap 2>$1
        done
else
         echo
        echo "You did not provide any command line options. Usage:  $PROGRAM [file]"
        echo
echo
fi
```

- **Python Sample Script**

```
#!/usr/bin/env python

import urllib2

def get_public_ip(request_target):
    grabber = urllib2.build_opener()
    grabber.addheaders = [('User-agent','Mozilla/5.0')]
    try:
        public_ip_address = grabber.open(target_url).read()
    except urllib2.HTTPError, error:
        print("There was an error trying to get your Public IP: %s") % (error)
    except urllib2.URLError, error:
        print("There was an error trying to get your Public IP: %s") % (error)
    return public_ip_address

public_ip = "None"
target_url = "http://ip.42.pl/raw"
public_ip = get_public_ip(target_url)

if not "None" in public_ip:
    print("Your Public IP address is: %s") % (str(public_ip))
else:
print("Your Public IP address was not found")
```

- **PowerShell Sample Script**

```
$Access = Get-Date
Write-Output "[***] You ran this script on $Access [***]"

$ComputerName = $env:computername
$OS = (Get-WmiObject -Class Win32_OperatingSystem -ComputerName $ComputerName | select caption |
select-string windows)-split("=", "}", "{")[0] -replace "}"| select-string windows
If ($OS -match "10") {Write-Output "[*] You are running $OS"}
If ($OS -match " 8") {Write-Output "[*] You are running $OS"}
If ($OS -match " 7") {Write-Output "[*] You are running $OS"}
if ($OS -match "2016") {Write-Output "[*] You are running $OS"}
If ($OS -match "2012") {Write-Output "[*] You are running $OS"}
If ($OS -match "2008") {Write-Output "[*] You are running $OS"}

# Query for currently logged in users and whether or not they are active
Write-Output "[*] The following users are currently logged in"
If ($OS -match "7") {$Current = query user | fl | out-host}
# Windows 10 use this
Else {Get-WmiObject -Class Win32_ComputerSystem | select username}

# List shares available
Write-Output "[*] The following shares are available"
PSdrive | select-object * -exclude used, free, provider, credential, currentlocation | fl

# Check whether or not SMBv1 is enabled or disabled.
$SMBCheck = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\Lanmanserver
\Parameters" -Name SMB1 | Select-Object "SMB1")
if ( $SMBCheck -match "0" ) {Write-Host "SMBv1 is currently disabled"}
Else {Write-Host "SMBv1 is enabled!"}
```

- **Ruby Sample Script**

```
# coding: utf-8

require 'socket'

target_ports = 1..65535
target_ips   = ['10.0.0.2', '10.0.0.3', '10.0.0.4']

def grab_banner(ip, port)
  TCPSocket.new(ip, port).recv(1024)
rescue
  "Unable to establish a connection with #{ip}:#{port}"
end

target_ips.each do |ip|
  target_ports.each do |port|
    puts grab_banner(ip, port)
  end
end
```

Domain 5 - Reporting and Communication

- **Reporting and Communication**
    - **Domain 5: Reporting and Communication**
        - Report writing and handling best practices
        - Explain post-report delivery activities
        - Recommend mitigation strategies for discovered vulnerabilities
        - Communication during the penetration testing process
    - **What kinds of questions can I expect on test day?**
        - "Given a scenario" for report writing and mitigation strategies
        - "Explain" for post-report delivery activity and communication during the process
- **Pentest Communications**
    - **Communication**
        - Lots of communication is needed before, during, and after a penetration test
        - Therefore, it is important to understand:
            - Communication paths
            - What triggers communication to occur
            - And the reason for communicating in the first place
    - **Reasons**
        - Situational Awareness
            - A shared common understanding of the network and its current security state
        - De-confliction
            - Determining if detected activity is a hacker or an authorized penetration tester
        - De-escalation
            - Decrease the severity, intensity, or magnitude of a security alert that is being reported
    - **Triggers**
        - Stages
            - Communication often occurs as the assessment moves from one phase to another
        - Critical Findings
            - A vulnerability is found that causes significant risk to occur to the security of the network
        - Indicators of Prior Compromise
            - Attack signatures have been detected and the network has been previously hacked

- **Goal Reprioritization**
  - Have the goals of the assessment changed?
  - Has any new information been found that might affect the goal or desired end state?
- **Communication Paths**
  - How will the pentest team communicate with the organization?
    - Phone, text, chat, email, white paper, etc.
  - Who at the organization can they contact?
      - CEO/CSO/CTO or System Admins

- **Report Writing**
  - **Normalization of Data**
    - Teams collect a lot of data during a test
    - Each tool collects and store data differently
    - All the data must be aggregated, normalized, and correlated in order for it to "make sense"
    - Normalization
      - Process of combining data from multiple sources and in different formats into a common and consistent event format
  - **Written Report of Findings**
    - Executive Summary
    - Methodology
    - Findings and Remediation
      - Consider the risk appetite
    - Metrics and Measures
      - Including risk ratings
    - Conclusion
  - **How Long Do I Keep the Report?**
    - How long should the report be stored?
    - Depends on your organization
    - POA&M is often created from a penetration tester's final report
    - Might have limits on how long to retain the data and the report based on the contract because of privacy and sensitivity concerns
  - **Handling and Disposal**
    - Data from the assessment should always be handled with due diligence and care
    - Findings and recommendations are sensitive in nature and should be treated as confidential
- **Mitigation Strategies**
  - **Mitigation Strategies**
    - Report should contain a list of not just findings, but recommendations on how to mitigate a vulnerability

        - Solutions come from:
            - Technology
                - Add a multifactor authentication system
            - Processes
                - Proper employee off-boarding to minimize an insider threat
            - People
                - Employee cybersecurity training
                - Hire qualified and certified IT professionals
    - **Finding: Shared Local Admin Credentials**
        - Randomize credentials
            - Every system uses a different password
        - Local Administrator Password Solution (LAPS)
            - Microsoft tool that provides centralized storage of passwords in Active Directory
            - Manages the passwords for each workstation when logon without domain credentials is necessary
    - **Finding: Weak Password Complexity**
        - Minimum password requirements/filters
        - Passwords Must…
            - Be at least 14 characters
            - Contain letters, numbers, and special characters
            - Not have repeating characters or digits
    - **Finding: Plain Text Passwords**
        - All passwords must be stored as hashes or another encrypted format
    - **Finding: No Multifactor Authentication**
        - Implement multifactor authentication
            - Something you know
            - Something you have
            - Something you are
            - Something you do
    - **Finding: SQL Injection**
        - Sanitize user input
            - User data checked for expected input type
            - Escape data to avoid SQL injections
        - Parameterize queries
            - Better than user input sanitization
            - Allow prepared statements to be used with bounded variables to access database
            - Each piece of SQL code is static but receives parameters from a separate section of code

    - **Finding: Unnecessary Open Services**
        - System hardening
            - Securing a computer or server by reducing its attack surface
            - Disable unneeded services
            - Close unused ports
            - Uninstall unused programs
- **Post-Report Activities**
    - **Post-Engagement Cleanup**
        - Remove shells, tools, and credentials created
    - **Attestation of Findings**
        - Provide evidence of your findings to the client
        - Provide them detailed reports, explanations, and ensure they understand the risks involved
    - **Client Acceptance**
        - Does the client agree you have fulfilled the scope of work?
        - Is formal acceptance required by the contract?
    - **Follow-up Actions or Retests**
        - What follow-up actions are you required to perform?
        - Will a retest be conducted after 30 or 90 days?
    - **Lessons Learned**
        - Documented information of both the positive and negative experiences that occurred
        - What did you do great on?
        - What could have gone better?
        - How can it go better next time?

Conclusion

- **CompTIA Pentest+**
  - **Domain 1: Planning and Scoping**
    - Planning an engagement
    - Key legal concepts
    - Scoping an engagement
    - Compliance-based assessments
  - **Domain 2: Information Gathering and Vulnerability Identification**
    - Information gathering techniques
    - Vulnerability scanning
    - Analyzing scan results
    - Preparing for exploitation
    - Weaknesses in specialized systems
  - **Domain 3: Attacks and Exploits**
    - Social engineering attacks
    - Exploiting vulnerabilities
      - Network-based
      - Wireless and RF-based
      - Application-based
      - Local host-based
      - Physical security
    - Post-exploitation techniques
  - **Domain 4: Penetration Testing Tools**
    - Use Nmap for information gathering
    - Know the use case for various tools
    - Analyze tool output or data
    - Analyze basic scripts
      - Bash, Python, Ruby, and Powershell
  - **Domain 5: Reporting and Communication**
    - Report writing and best practices
    - Post-report delivery activities
    - Recommending mitigations
    - Importance of communication in testing
  - **Are You Ready?**
    - Take the practice exam in this course
    - Did you score at least 85% or higher?
    - If you need more practice, take additional practice exams to hone your skills