Ethical Student Hackers

Operating System Security

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

Code of Conduct

- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.
- If you have any doubts or need anything clarified, please ask a member of the committee.
- Breaching the Code of Conduct = immediate ejection and further consequences.
- Code of Conduct can be found at https://shefesh.com/downloads/SESH%20Code%20of%20Conduct.pdf

Difference between Windows and Linux

Comparison Table

	Windows	Linux
How are passwords stored?	SAM Hives	/etc/shadow
How are firewalls set?	Windows Defender Firewall	iptables, nftables
How are processes scheduled?	Task Scheduler	systemd
How are file permissions managed?	Explorer	Kernel, chmod, chown, chroot
How are user privileges managed?	UAC, Privileges	Sudo, kernel, su
How are services run?	Service accounts	Systemctl, user accounts
What antivirus is available?	Defender, proprietary	Proprietary

CVE Bingo!

How many recent vulnerabilities in Windows/Linux OS or services can you name?

Windows CVEs

Defender Priv Esc
(CVE-2021-24092)

PrintNightmare
(CVE-2021-34527)

Exchange Server
Exploit Chain
(CVE-2021-27065,
CVE-2021-26855,
CVE-2021-26857,
CVE-2021-26858)

OMIGOD
(CVE-2021-38647)

HiveNightmare
(CVE-2021-36934)

EternalBlue
(CVE-2017-0143)

PetitPotam
(CVE-2021-36942)

SambaCry
(CVE-2017–7494)

PrintSpoofer
(CVE-2020-1048)

Linux CVEs

Dirtycow
(CVE-2016-5195)

Rowhammer
(CVE-2015-0565)

Sudoedit
(CVE-2021-3156)

Shellshock
(CVE-2014-6271)

HeartBleed
(CVE-2014-0160)

PolKit
(CVE-2021-3560)

sequoia /proc
Filesystem Overflow
(CVE-2021-33909)

Windows

Windows Security Features

(a very brief overview)

Users & Groups

- Like in Unix, each User has an account, and Users can be grouped
- Identified by a Security Identifier (SID) - this is included in Access Tokens granted when a user logs on
- Read more: https://www.digitalcitizen.life/simple-questions-what-user-group-windows-what-does-it-do/

Service Accounts

- Services, such as webservers on IIS or MSSQL servers, often run under a separate account
- They can 'impersonate' users' Kerberos tokens (more on this later) and act as that user
- This enables a whole branch of security issues... more on this later, too
- Read more: https://docs.microsoft.com/en-us/windows/security/identity-protection/access-control/service-accounts

NTFS/Share Permissions

- Apply to files and folders (locally) or shared folders
- NTFS: Read, Read and Execute, Write, Modify, List Folder Contents, and Full Control
- Shares: Read, Change, and Full Control
- Read more: https://www.dell.com/support/kbdoc/en-uk/000137238/understanding-file-and-folder-permissions-in-windows

Windows Security Features (cont.)

Privileges

- Give an account the ability to perform a certain action (such as shutdown, impersonation etc)
- Common privilege escalation vector if an account has SeImpersonatePrivilege
- Read more: https://docs.microsoft.com/en-us/windows/win32/secauthz/privileges, https://blog.palantir.com/windows-privilege-abuse-auditing-detection-and-defense-3078a403d74e, and https://foxglovesecurity.com/2016/09/26/rotten-potato-privilege-escalation-from-service-accounts-to-system/

User Account Control (UAC)

- Operations that require administrative privileges require prompts for consent
- Processes are assigned ‘integrity’ levels - this indicates the level of trust
- Read more: https://docs.microsoft.com/en-us/windows/security/identity-protection/user-account-control/how-user-account-control-works

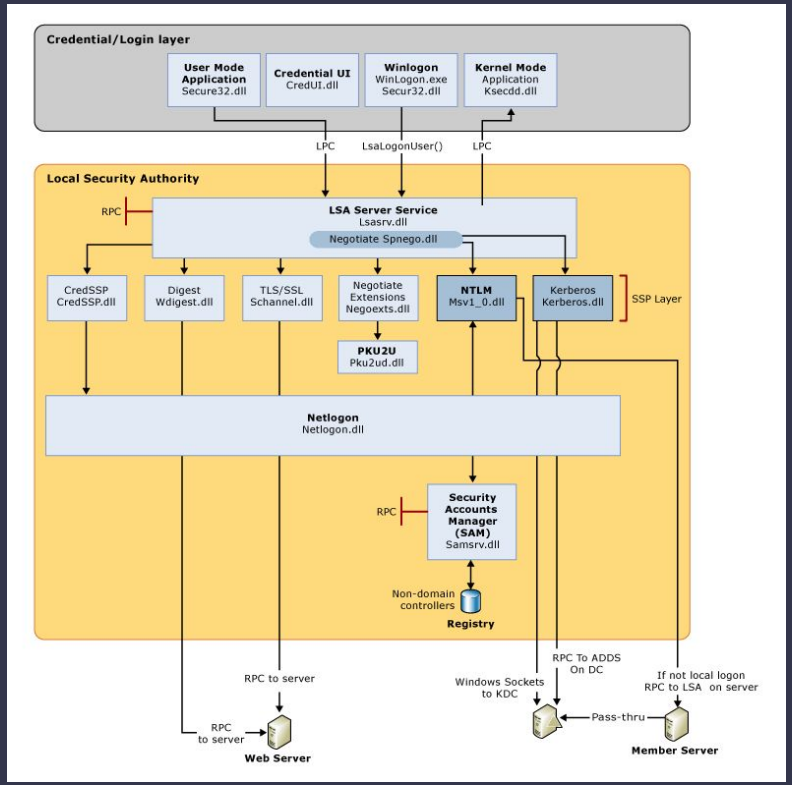
Windows Authentication

Security Accounts Manager (SAM)

- Windows Credential Store Database
- Stores several hashed passwords and Usernames
- SAM ‘hives’ can be extracted with tools like mimikatz and hashes dumped with impacket-secretsdump

NTLM Authentication Protocol

- NT hashes used to identify users
- Kerberos, the authentication protocol used in Active Directory, makes use of this
- Hashes can be used to authenticate on their own in pass-the-hash and NTLM relay attacks



Source: https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/credentials-processes-in-windows-authentication

What's Wrong with Windows?

A core Windows feature ties together a lot of Windows CVEs - NTLM Authentication

Passing the Hash

- Plaintext passwords are rarely used to authenticate in Windows - usually it is the NT Hash
- This means stealing the NT Hash is roughly equivalent to stealing a plaintext password
- Hashes can be extracted from the SAM or NTDS.DIT databases
- Hashes can be stolen by tricking accounts into authenticating - for example, Responder's rogue SMB server
- One compromised machine can lead to multiple if they have the same passwords
- impacket-psexec: psexec.py -hashes [HASH] Administrator@ip
- Read more: https://en.hackndo.com/pass-the-hash/#protocol-ntlm and https://www.securify.nl/en/blog/living-off-the-land-stealing-netntlm-hashes/

NTLM Relays

- NTLM authentication can be Man-in-the-Middled to authenticate against Active Directory
- This is the basis for PetitPotam and several other attacks - read more: https://www.csoonline.com/article/3632090/ntlm-relay-attacks-explained-and-why-petitpotam-is-the-most-dangerous.html

Final note: https://github.com/cfalta/MicrosoftWontFixList/blob/main/README.md

Windows Services

Here are some of the services you might see when scanning a Windows Machine:

- RPC + RPCBind (port 111 and 135): Lists services, allows remote interaction, enum with rpcclient
- Netbios + SMB (port 139 and 445): File sharing services, checking version numbers is crucial
- LDAP (port 389, 3268): Active Directory querying service
- Kerberos (port 88): Authentication Service
- RDP (port 3389): Remote Desktop Protocol
- WinRM (port 5985): Another remote access protocol

Many of these services make use of NTLM for authentication - SMB is used in Responder, and is the service used by psexec. NT hashes can also be used with tools like evil-winrm

Activity - SAM Hash Extraction

Download the SAM file zip from here: https://shefesh.com/assets/demos/SAM%20Files.zip

- On Linux: wget https://shefesh.com/assets/demos/SAM%20Files.zip
- Files were extracted with reg.exe save hklm\sam c:\windows\temp\sam.save (etc for hklm\security, hklm\system)

Extract the hashes using impacket-secretsdump

- secretsdump.py -sam SAM -security SECURITY -system SYSTEM LOCAL

Optional: crack the hashes using hashcat

- hashcat -a 0 -m 1000 hashes --wordlist /usr/share/wordlists/rockyou.txt

Quick Break

Linux

File System

umask

- Umask sets the default permission for newly created files by the user. By default it is set to 0022, which means anyone can read the newly created file, however the owner has read and write permissions. To change it so that only the owner of the file can read/write to the file, put umask 0077 or 0027 in the /etc/login.defs file.

Mount options

- For file storage systems or unknown USB drives, mount using mount -o noexec,nosuid,rw /dev/sda1 /mnt/usbDevice
- This makes sure that files cannot be executed from the usb, and that no suid can elevate privs

Password Policy

Linux uses PAM (Pluggable Authentication Modules) to handle authentication. There are multiple modules available to use, however we'll look at some for hardening the password policy. Use the command sudo nano /etc/pam.d/passwd to edit the password policy file.

https://wiki.archlinux.org/title/Security#Enforcing_strong_passwords_with_pam_pwquality

Setting up a Firewall

A firewall is an application that sits and listens to your network traffic, it follows a set of rules that allow it to accept and block traffic.

- Firewalls are useful for stopping services that shouldn't be running from being accessed by potentially unwanted users

There are multiple firewall applications available, however two common ones for Linux are iptables and nftables.

Nftables is basically the newer and more scalable version of iptables, however we will go over iptables for this demonstration. You can install the iptables-nftables-compat package to do a 1-1 translation of iptables commands into nftable commands.

https://wiki.archlinux.org/title/Simple_stateful_firewall

IPTables

Traffic on a network is made up of lots of packets (small segments of data), iptables reads the header of these packets and based on a set of rules decides how to process the packets.

Iptables uses tables, Chains, rules and targets to filter the packets.

- Tables are a collection of chains that join similar actions
- Chains contain a set of rules, iptables iterates through each chain until it finds a rule that suits the case
- Rules define what iptables should do with the packet, for example forward the packet.
- Targets are what ends up happening to the packet, for example dropping the packet.

https://phoenixnap.com/kb/iptables-tutorial-linux-firewall

IPTables demo

Hardening SSH

SSH (Secure Shell)

- Is the go-to method for remotely accessing a Linux device.
- By default requires us to supply a username and password to authenticate
    - Therefore is only as strong as the weakest password (Hence previous password policy slide)
- We can set up private keys
- Specify certain users to be allowed to login

https://www.sshaudit.com/hardening_guides.html#ubuntu_20_04_lts

https://linuxhandbook.com/ssh-hardening-tips/

Logging

Autitd

- Auditd allows us to view critical events that have occurred on the machine
- It logs administrative commands, new logins and other things we can configure it to log
- Auditd is a tool for checking logs AFTER an incident has occured, it's not preventative
- It can be used to find an existing and previously exploited hole in your Linux security

https://github.com/Neo23x0/auditd

aureport - Shows a quick summary of the events on the system

ausearch - Search through the events that have occurred, allows for filtering

sudo less /var/log/audit/audit.log - View the whole log file

AppArmor/SELinux

AppArmor and SELinux are tools for mandating the access controls for applications. They can accept/deny applications from accessing certain files/folders, stop privilege escalation attempts and more.

SELinux has the mentality of rejecting all permissions and allowing where necessary while AppArmor accepts all permissions and has rules to restrict access. This means that SELinux is more secure by design but AppArmor can be easier to manage.

- Both can actively stop/mitigate potential attacks, if configured well
- Both can and do log events similar to Auditd

System Auditing software

There are tools available to us that can tell us if we have weaknesses in our Linux setup. They help to point out improvements that we can make to secure the system.

One of these tools is called Lynis, Lynis is a commonly used audit tools that checks for misconfigurations and missing security software. https://cisofy.com/downloads/lynis/

- `./lynis audit system`
- At the end of the analysis it gives a rundown of things to patch/change on the system

Upcoming Sessions

What's up next?

www.shefesh.com/sessions

Next week

The week after that

And the week after that

And the week after that

And the week after that

Any Questions?

www.shefesh.com

Thanks for coming!