Colby Heilner
Professor Torres
9/22

IT 120

Lab 5

Part A: Netlab1Security+ V3Links to an external site.

- Log onto the Netlab environment and attempt to locate "ALL" accounts on "ALL" systems (except Kali)
- Change the password on "all" accounts
- All accounts should have only 1 admin account, and 1 user account.

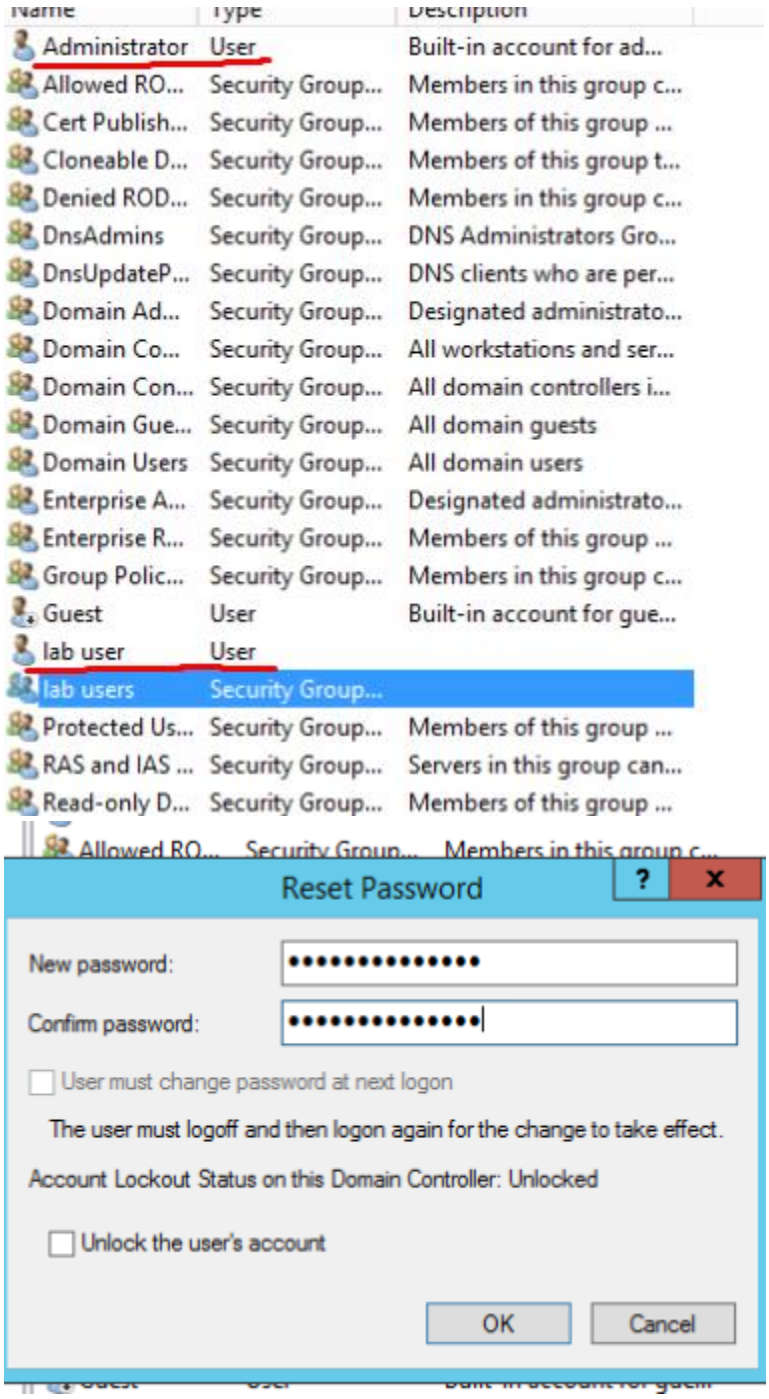
Instead of showing every single account with a password changed. I will show one for each computer and my methods on how I found all the accounts for each.

DVL
Method: First, I got all users account on the machine.

```
root:x:0:0::/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/log:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/:
news:x:9:13:news:/usr/lib/news:
uucp:x:10:14:uucp:/var/spool/uucppublic:
operator:x:11:0:operator:/root:/bin/bash
games:x:12:100:games:/usr/games:
ftp:x:14:50::/home/ftp:
smmsp:x:25:25:smmsp:/var/spool/clientmqueue:
mysql:x:27:27:MySQL:/var/lib/mysql:/bin/bash
rpc:x:32:32:RPC portmap user:/:/bin/false
sshd:x:33:33:sshd:/:
gdm:x:42:42:GDM:/var/state/gdm:/bin/bash
pop:x:90:90:POP:/:
nobody:x:99:99:nobody:/:
postgres:x:1000:100::/home/postgres:
ftpadmin:x:1001:100::/home/ftp:/bin/false
```

I used passwd username to change all of the system passwords.

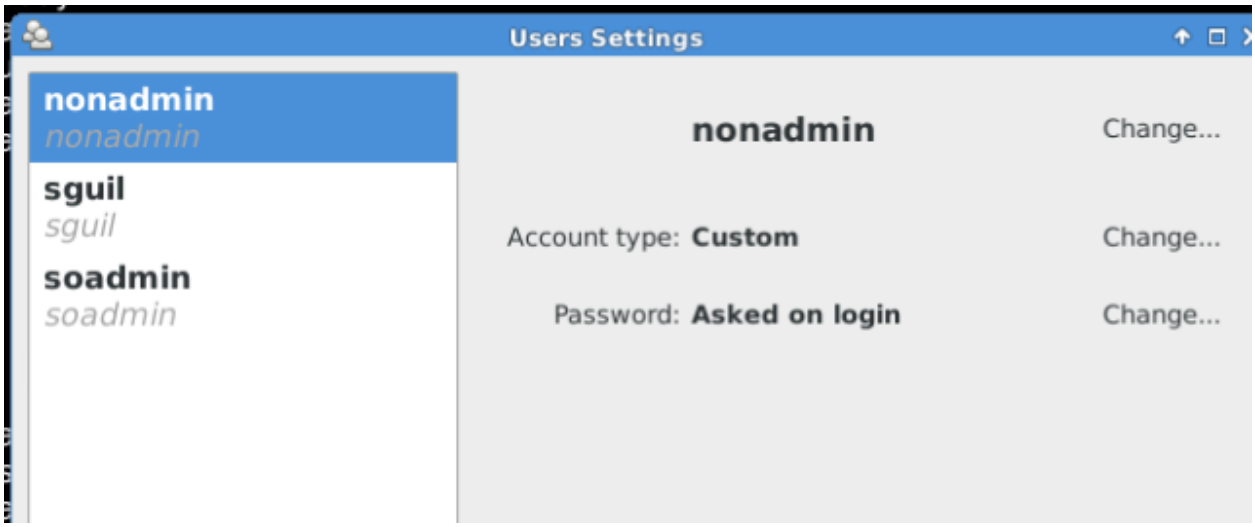
WIN12R2 For this I had to go to active directory and view the users i also deleted all users but one admin and one non



SecOnion

For this one I ran the same command as the DVL box.

But I also found the local account here



This also includes one non admin account I added because the sguil account is disabled.

Ubuntu

Again, Linux so we run our command to view all the services, and their users associated. Then I look for non-machine accounts.

I saw FileZilla and made sure to check that the accounts were disallowed unless specify allowed

```
# /etc/ftpusers: list of users disallowed FTP access. See ftpusers(5).

root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
```

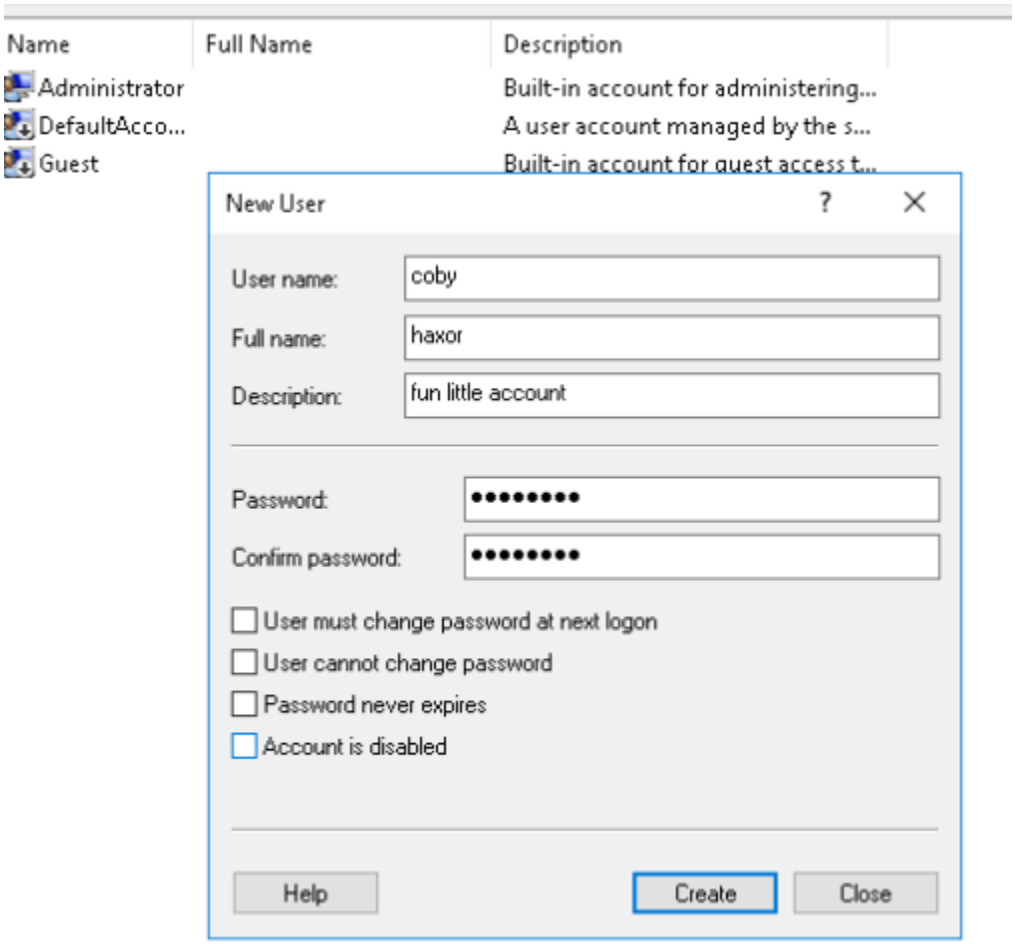
WIN16

For this i navigated to PowerShell to have some fun.

```
PS C:\Users\Administrator> get-localuser

Name           Enabled Description
----           ------- -----------
Administrator  True    Built-in account for administering the computer/domain
DefaultAccount False   A user account managed by the system.
Guest          False   Built-in account for guest access to the computer/domain


PS C:\Users\Administrator>
```

I made sure it was disabled and went into the gui to make another non admin user non guest

```
PS C:\Users\Administrator> disable-localuser "guest"
PS C:\Users\Administrator> get-localuser

Name           Enabled Description
----           ------- -----------
Administrator  True    Built-in account for administering the computer/domain
DefaultAccount False   A user account managed by the system.
Guest          False   Built-in account for guest access to the computer/domain
```

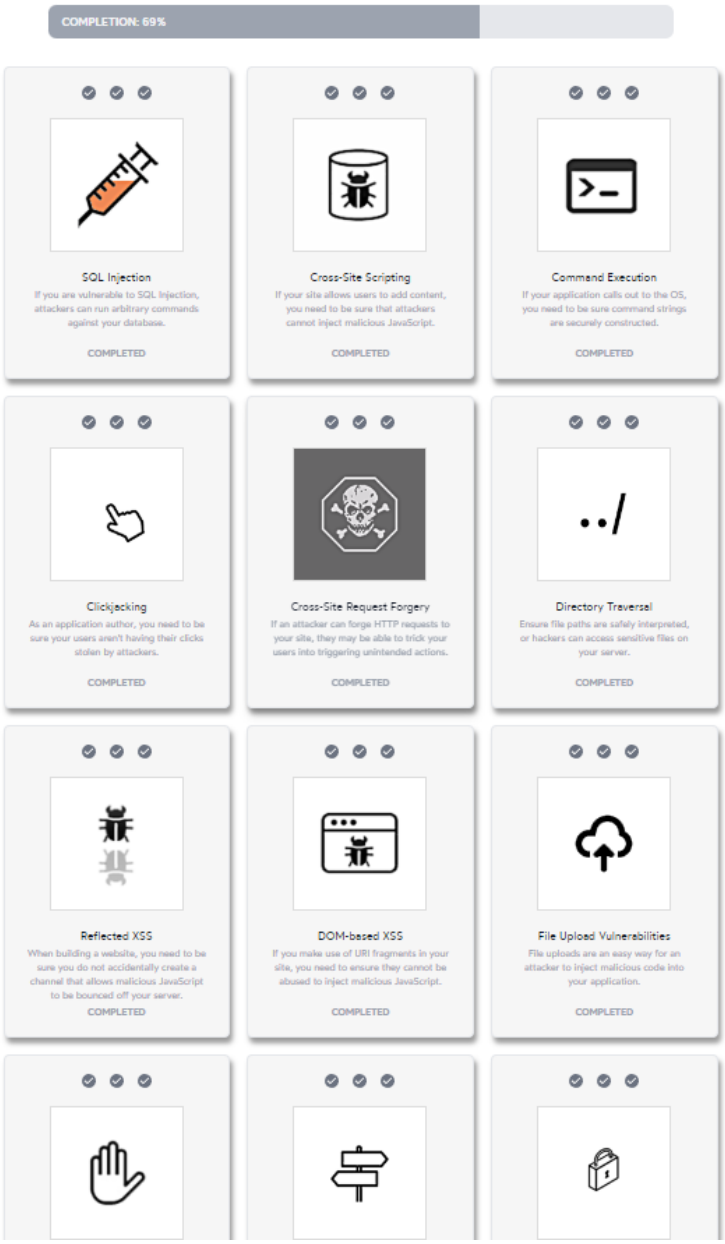


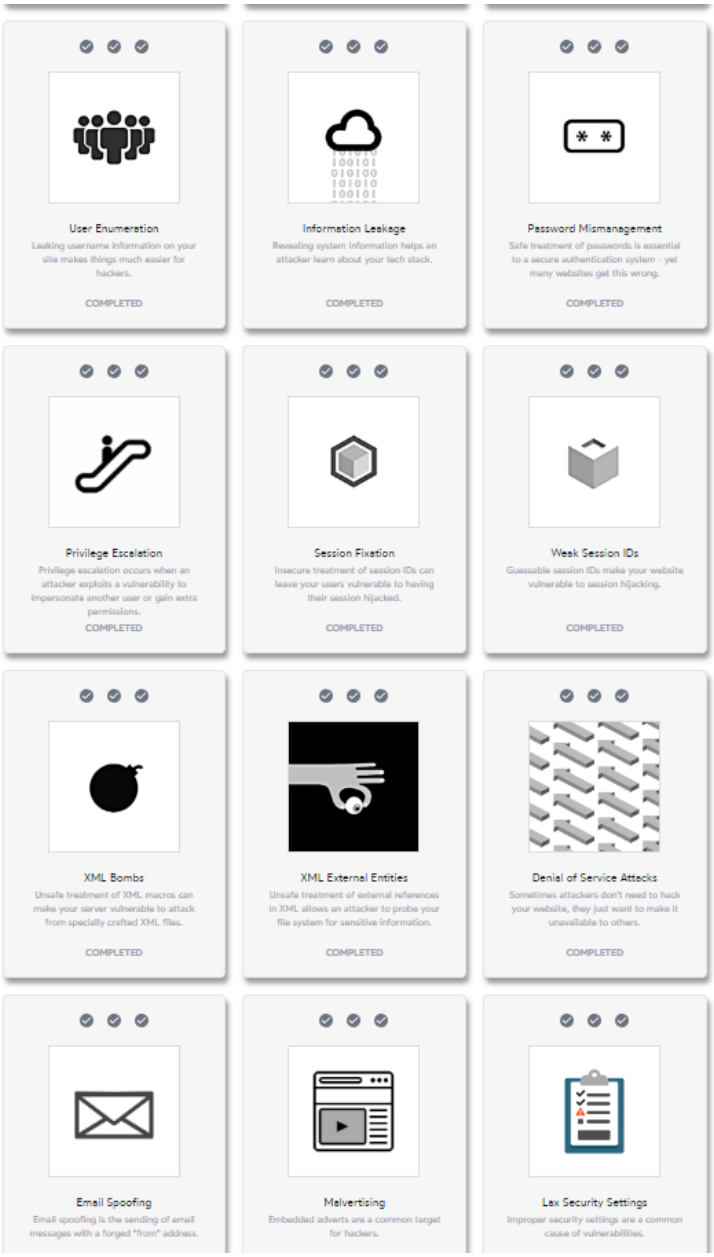

```
Name           Enabled Description
----           ------- -----------
Administrator  True    Built-in account for administering the computer/domain
coby           True    fun little account
DefaultAccount False   A user account managed by the system.
Guest          False   Built-in account for guest access to the computer/domain
```

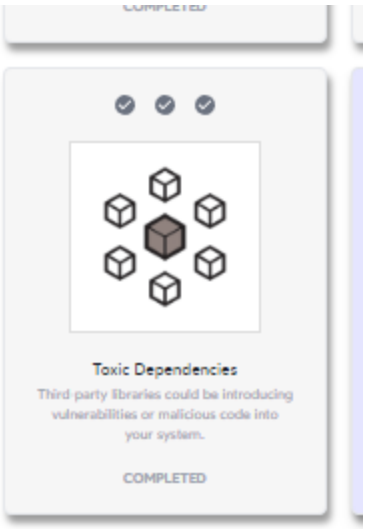
Part B:

- Sign up for a free account at: https://www.hacksplaining.com/lessons (Links to an external site.)
  - this website explains the top 10 in detail
  - There are 25 links explaining the top 10 in lab format.
  - Do ANY 10 of the links, and screenshot your final page showing completion of at least 10 labs. If you do all 25, you will receive an extra 5 points of extra credit.

    I did 25 and broke it up into three screenshots.




User Enumeration
Leaking username information on your site makes things much easier for hackers.
COMPLETED
Information Leakage
Revealing system information helps an attacker learn about your tech stack.
COMPLETED
Password Mismanagement
Safe treatment of passwords is essential to a secure authentication system - yet many websites get this wrong.
COMPLETED
Privilege Escalation
Privilege escalation occurs when an attacker exploits a vulnerability to impersonate another user or gain extra permissions.
COMPLETED
Session Fixation
Insecure treatment of session IDs can leave your users vulnerable to having their session hijacked.
COMPLETED
Weak Session IDs
Guessable session IDs make your website vulnerable to session hijacking.
COMPLETED
XML Bombs
Unsafe treatment of XML macros can make your server vulnerable to attack from specially crafted XML files.
COMPLETED
XML External Entities
Unsafe treatment of external references in XML allows an attacker to probe your file system for sensitive information.
COMPLETED
Denial of Service Attacks
Sometimes attackers don't need to hack your website, they just want to make it unavailable to others.
COMPLETED
Email Spoofing
Email spoofing is the sending of email messages with a forged "from" address.
Malvertising
Embedded adverts are a common target for hackers.
Lax Security Settings
Improper security settings are a common cause of vulnerabilities.



Part C:

- Discuss each section of the OWASP top 10, how are they exploited and how do you protect against it.

  - **Broken Access Control**
  - **Cryptographic Failures** (formerly Sensitive Data Exposure)
  - **Injection** (like SQL, NoSQL, etc.)
  - **Insecure Design** (new in 2021)
  - **Security Misconfiguration**
  - **Vulnerable and Outdated Components**
  - **Identification and Authentication Failures**
  - **Software and Data Integrity Failures** (new in 2021)
  - **Security Logging and Monitoring Failures**
  - **Server-Side Request Forgery (SSRF)** (new in 2021)

  - **Broken Access Control**

  - **Risk**: Attackers gain unauthorized access to resources.

  - **Protection**: Enforce role-based access controls (RBAC), use least privilege, and thoroughly validate access control rules.

My own words: A regular user can acess an admin one my changing thr URL of a website.

- **Cryptographic Failures**

Risk: Sensitive data is exposed due to weak encryption or improper handling.

Protection: Use strong, modern encryption algorithms and secure transport layers (e.g., TLS). Ensure key management practices are secure.

My own words: Password are stored in plaintext

- **Injection**

Risk: Malicious input is sent to interpreters like SQL or NoSQL databases.

Protection: Use parameterized queries, prepared statements, and input validation. Never concatenate user input into queries.

My own words: If you know SQL well enough, sadly I do not yet. You can string together SQL to pull certain things from tables. Select * FROM users where users = ?

- **Insecure Design**

Risk: Inherent weaknesses in system design expose applications to attacks.

Protection: Apply secure design principles early in the development lifecycle, including threat modeling and architectural risk analysis.

My own words: you allow infinite password attempt to log into accounts allowing for brute force attacks.

- **Security Misconfiguration**

Risk: Poorly configured systems are vulnerable to attack.

Protection: Regularly update software, automate security configurations, disable unused services, and follow the principle of least privilege.

My own words: A development API is left exposed on a production server

- **Vulnerable and Outdated Components**

Risk: Applications use old, unpatched software.

Protection: Regularly patch software, use trusted components, and continuously monitor for vulnerabilities in dependencies.

My own words: A web application uses an outdated version of a library with known vulnerabilities

- **Identification and Authentication Failures**

Risk: Weak authentication mechanisms allow unauthorized users to access systems.

Protection: Use strong multi-factor authentication (MFA), secure session management, and implement secure password policies.

My own words: Organizations might have no password complexity, age or length requirements for users.

- **Software and Data Integrity Failures**

Risk: Software or data is altered maliciously without detection.

Protection: Implement code signing, integrity checks, and continuous monitoring. Use secure update processes.

My own words: Malicious code is injected during a software update

- **Security Logging and Monitoring Failures**

Risk: Inadequate logging prevents detection of malicious activity.

Protection: Enable logging for key events, monitor logs regularly, and implement real-time alerts for suspicious activities.

My own words: If someone compromises your system and it is not logged. How do you expect to learn from it?

- **Server-Side Request Forgery (SSRF)**

Risk: Attackers can manipulate server requests to access unauthorized internal systems.

Protection: Validate and sanitize URLs, disable unused protocols, and enforce allowlists for external requests.

My own words: You can manipulate a web application to fetch data from an internal network by sending a false request.