Attacking and Defending

Cybersecurity Boot Camp

Lesson 1.2

Class Objectives

By the end of today's class, you will be able to:

List different types of user, web, server, and database cybersecurity attacks.

Identify risk mitigation plan frameworks for user, web, server, and database cybersecurity attacks.

Set up a virtual machine lab environment that you will use throughout the course.


Recap

Last class, we described cybersecurity as centering on two concepts. What were they?

Recap

The two concepts cybersecurity is centered on are:

How would you define threat assessment?

Threat assessment is the structured process of identifying the threats posed to a group or system.

How would you define risk mitigation?

Risk mitigation is the systematic reduction of the impact and/or likely occurrence of a negative event.

Recap

In other words...

Threat assessment:
What could happen?

Risk mitigation:
How do we handle it?

Last class, we introduced a framework that captures the fundamental goal of information security.

What was the framework?

The framework that captures the fundamental goal of information security is the CIA triad.

What are the three elements of the CIA triad?

The three elements of the **CIA triad** are:

Define each of the three elements in the context of information security.

Confidentiality

Ensuring sensitive information is protected from access by unauthorized persons.

Integrity

Protecting information from being modified or tampered by unauthorized persons.

Availability

Ensuring that all operating systems, equipment, and data are functioning correctly and accessible by those who need it.

Provide an example of how each of the three elements of the CIA triad can be adversely affected.

Confidentiality

Ensuring sensitive information is protected from access by unauthorized persons.

Example:
Banking breach releases credit card info to the public.

Integrity

Protecting information from being modified or tampered by unauthorized persons.

Example:
Students modify official grades for themselves and their friends.

Availability

Ensuring that all operating systems, equipment, and data are functioning correctly and accessible by those who need it.

Example:
Attackers disable a website through a denial of service attack.

Quick Review

Activity: Oh look, a phone!

Suppose in the last class, two students left their phones unattended.

As a class, let's **identify as many exploits as possible** that could result from a stolen cell phone.

Hint: Be creative! Think like an attacker.

- What is the worst possible damage that could happen?
- Think about **real** *damage*.
- Think beyond the value of the phone itself.

Activity Review: Oh look, a phone!

Potential Adverse Events

1. The phone is wiped and resold.
2. Phone memory is harvested and photos and sensitive material are used for blackmail.
3. Credentials for email and social media accounts are used to extract financial gain.
4. Installed applications are used to make purchases.
5. Malware software is directly installed to track future activity.
6. Phone contacts are socially engineered to provide money.
7. The phone is used to conduct illegal activity.
8. The owner's identity is stolen.

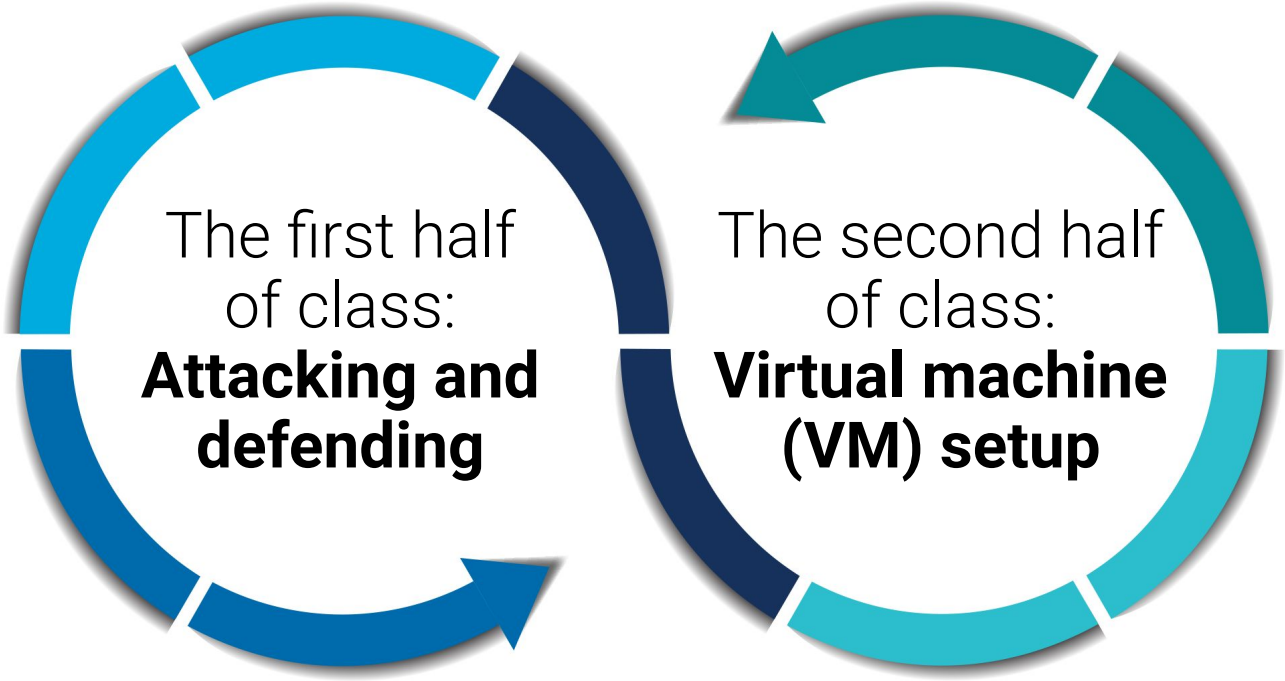
Today’s Class

Today's Class

We will continue with assessing threats and mitigating risks by evaluating specific attacks and vulnerabilities of:

- Users
- Web applications
- Servers
- Databases

Today's Class



Today's Class: Attacking and Defending

You will have the opportunity to think like offensive and defensive cybersecurity professionals evaluating the attacks and defenses of levels of information within a company.

To assess threats and mitigate risks, we need to look at each component of an organization, and understand how malicious actors can exploit weaknesses and damage the stakeholders' finances, reputations, and well-being.

Today's Class: VM Setup

You will set up VirtualBox and Vagrant, two programs needed to run VMs on local machines. You should have a basic familiarity with the need for VMs from our technical overview in the first day of class.

Today, we will dive into more detailed installation instructions.

A Note on Troubleshooting

As we set up our VMs, we may have issues that require troubleshooting.

Troubleshooting is the process of problem solving.

In this course, troubleshooting will often involve ensuring that our VMs and lab services are running smoothly.

A Note on Troubleshooting

Whether you are a penetration tester, system administrator, security operations center (SOC) analyst, network admin, or IT help desk, you will most likely have to troubleshoot technology on a regular basis.

Troubleshooting will be a common theme throughout this course, and we'll do it alongside various activities, such as:

- Tinkering with scripts;
- Configuring Azure Lab setups; and
- Navigating access controls.

Just as troubleshooting is **necessary** in the professional environment, it will be **necessary** in this learning environment.

Professional environment

Learning environment

Security Task #1:
Attacking the Wall

We will now work on two security challenges related to assessing threats and mitigating risks.

SECURITY TASK #1

Attacking the Wall

In this first activity, we will consider various strategies that attackers can use to penetrate an insecure login.

While you may be new to this type of thinking, this exercise should help you think creatively about all the ways a system can be penetrated, from user attacks to physical break-ins.

To successfully complete this exercise, you must think through creative options.

Let's examine the scenario...

SECURITY TASK #1

Congratulations! You and your team have just been hired by a very successful startup that runs a Bitcoin dating exchange.

While their founding team is brilliant, like many startups, **they don't know the first thing about security.**

SECURITY TASK #1

They just handed you a lot of money to solve their **most critical problem.**

Their login process is **totally insecure.** Attackers are routinely logging in as users (and administrators) and gaining access to company data and financial assets.

Activity: Security Task #1: Attacking the Wall

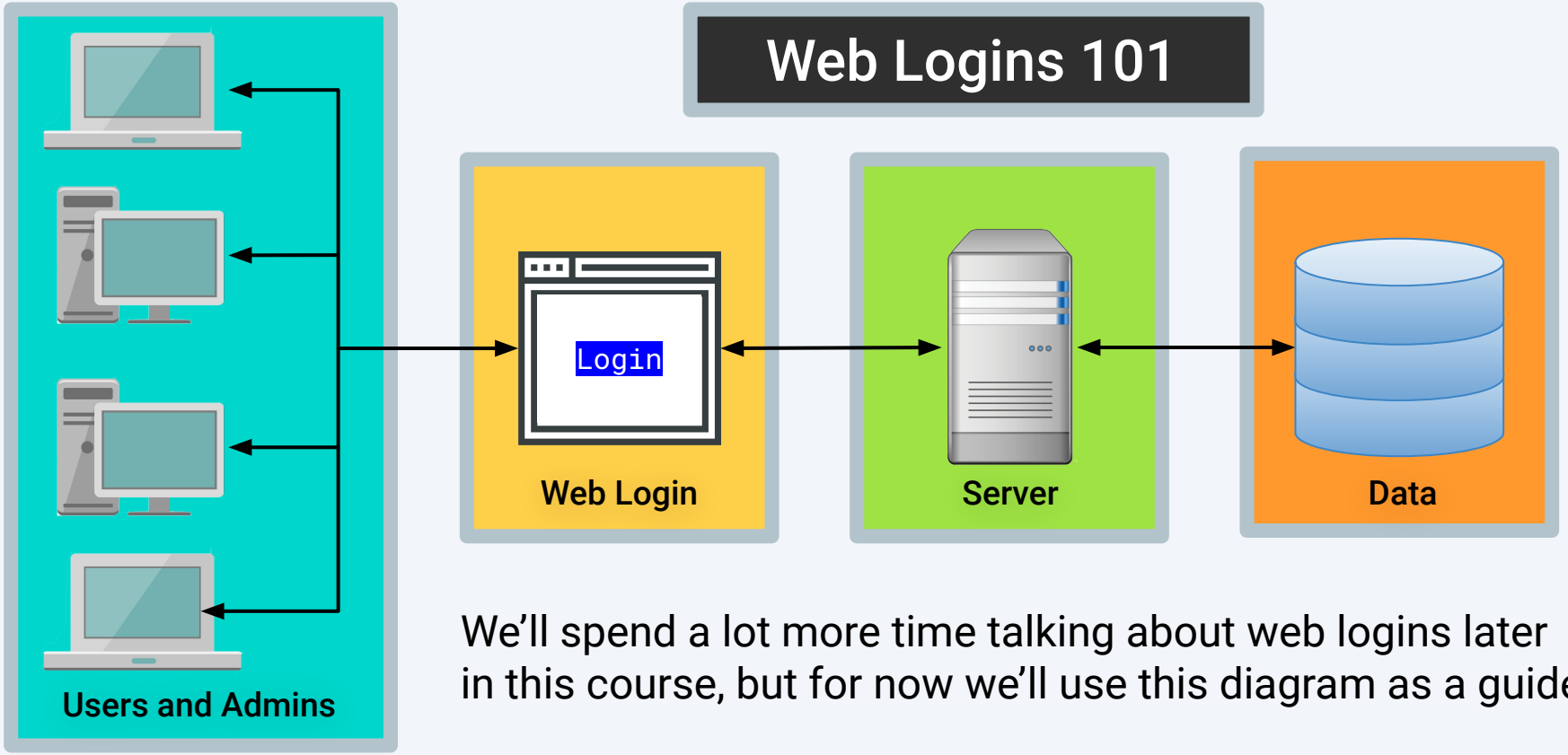
In this security task, you and your group will play the role of security professionals tasked with handling a real-world situation.

See the next two slides for more instructions.

Suggested Time:

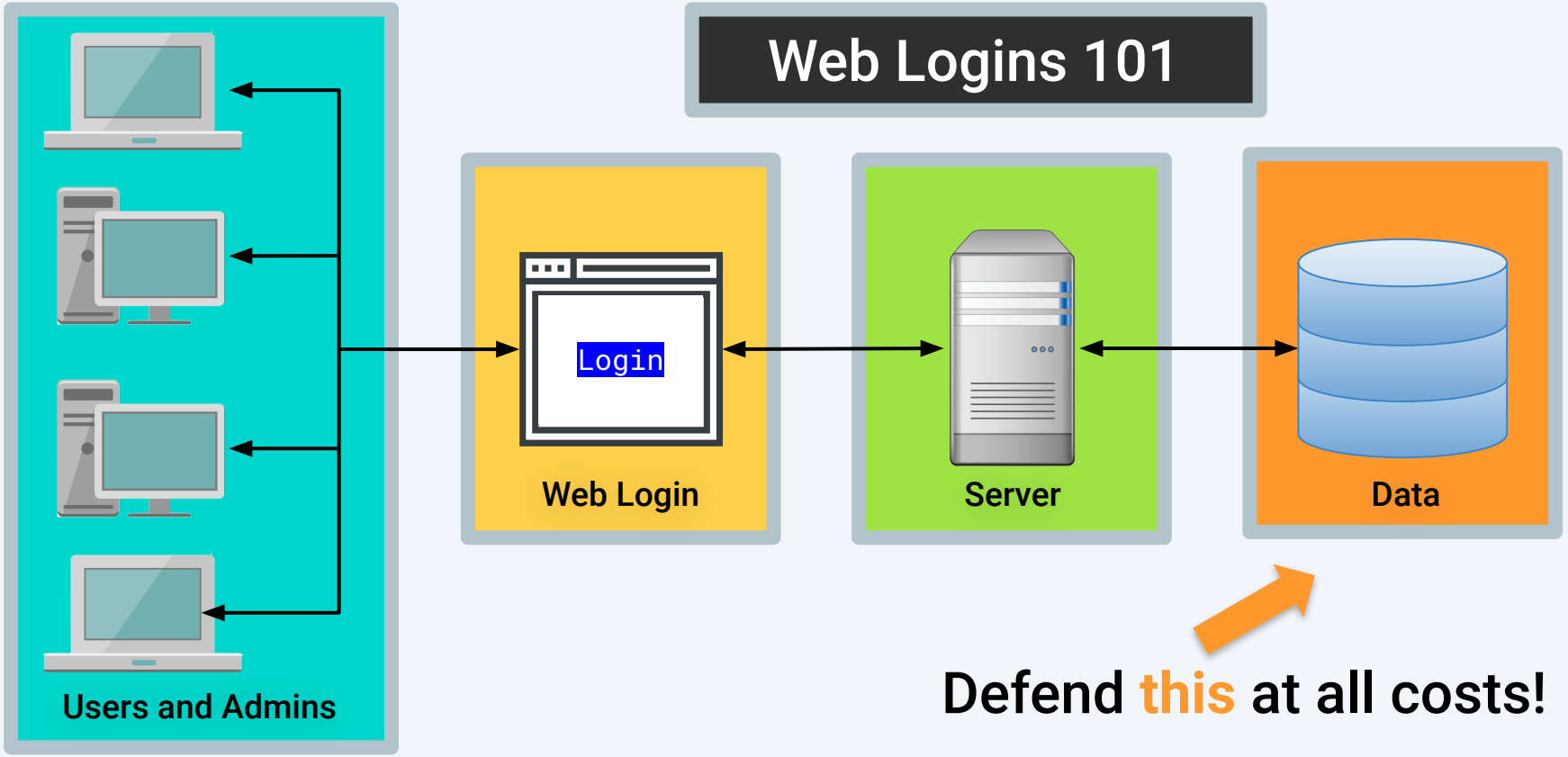
20 Minutes

Activity: Security Task #1: Attacking the Wall



We'll spend a lot more time talking about web logins later in this course, but for now we'll use this diagram as a guide.

Activity: Security Task #1: Attacking the Wall

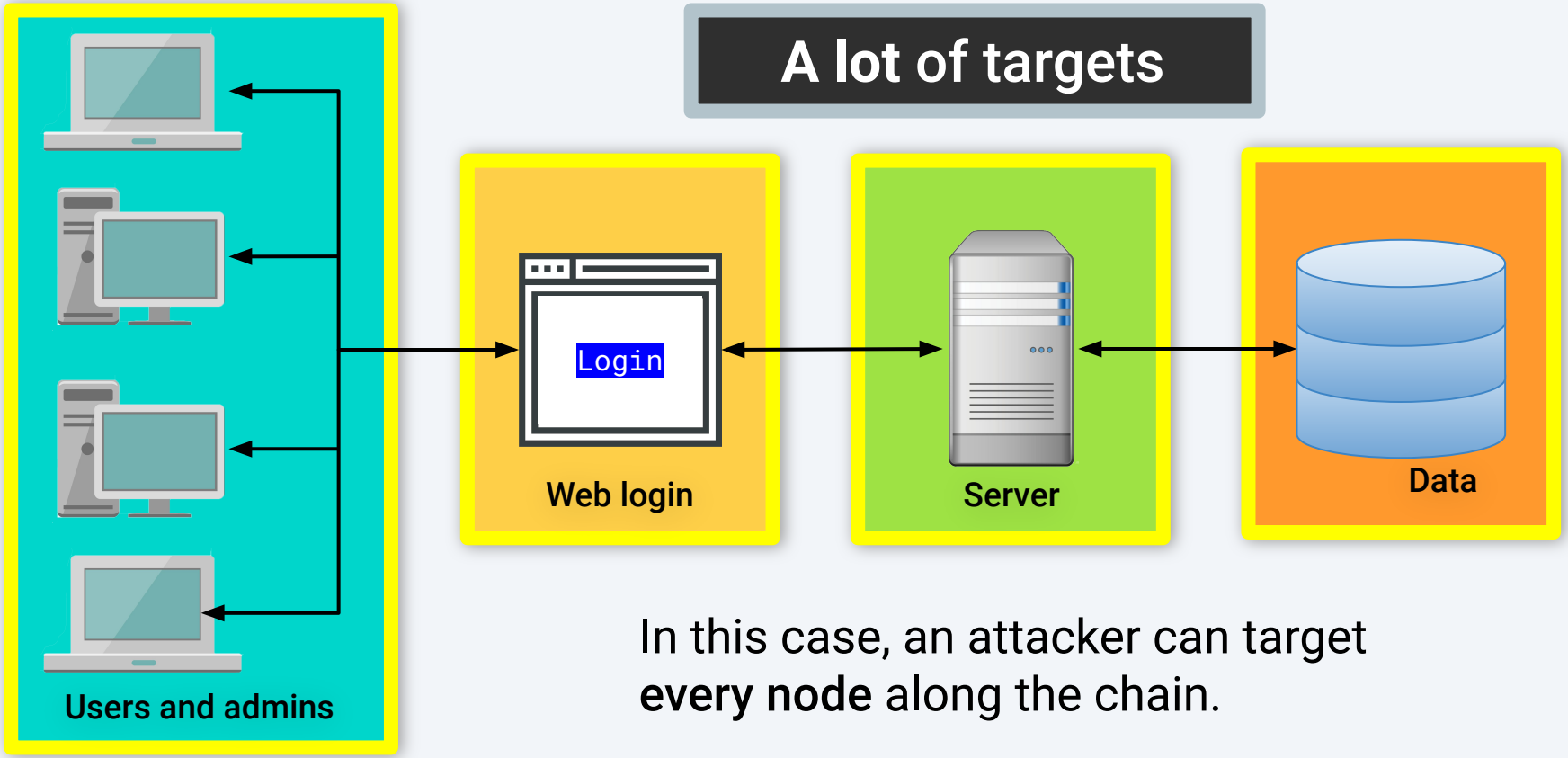


Time’s Up! Let’s Review.

Questions?

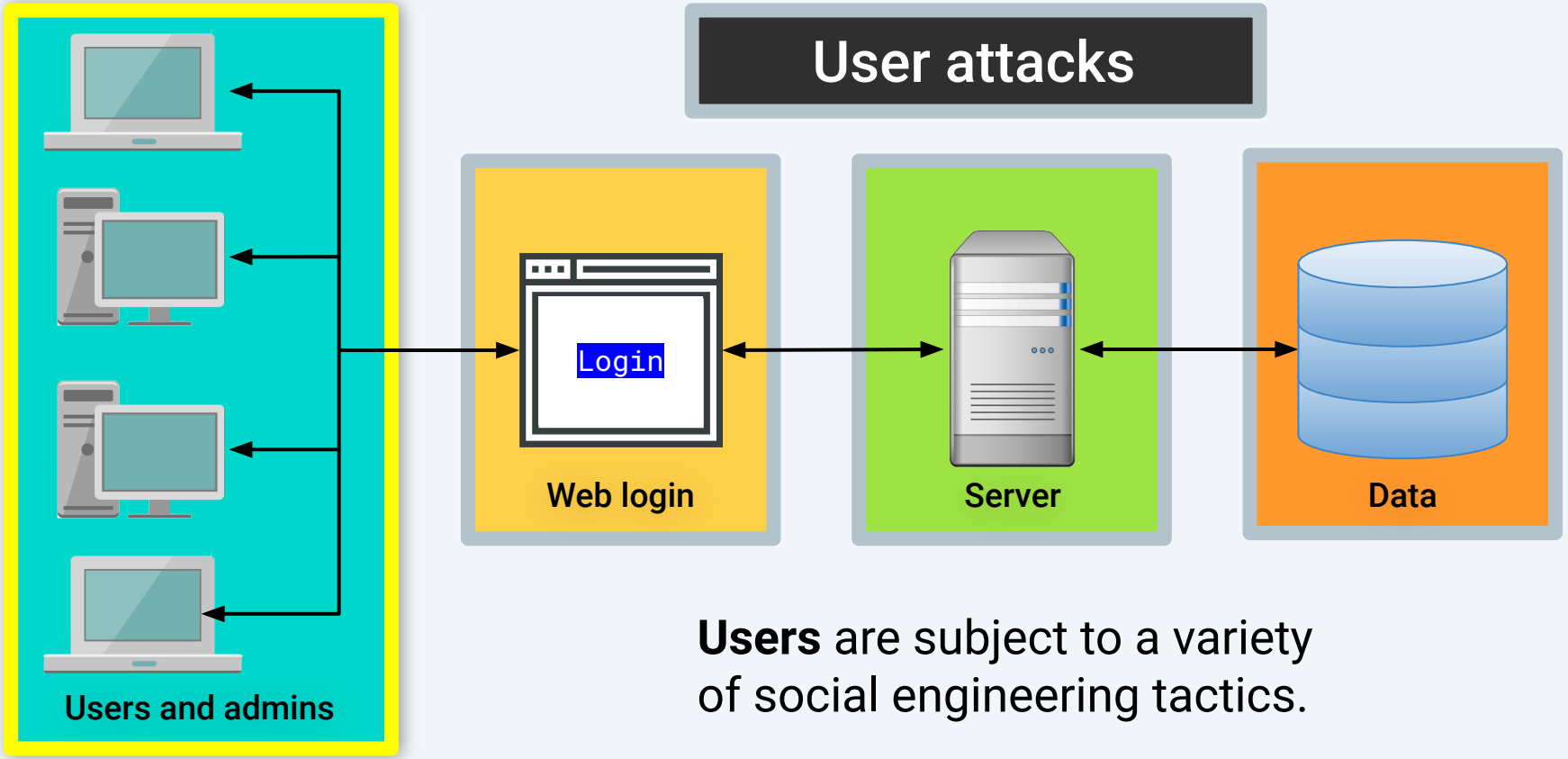
Step #1:
Assess the Target

Activity: Security Task #1: Attacking the Wall



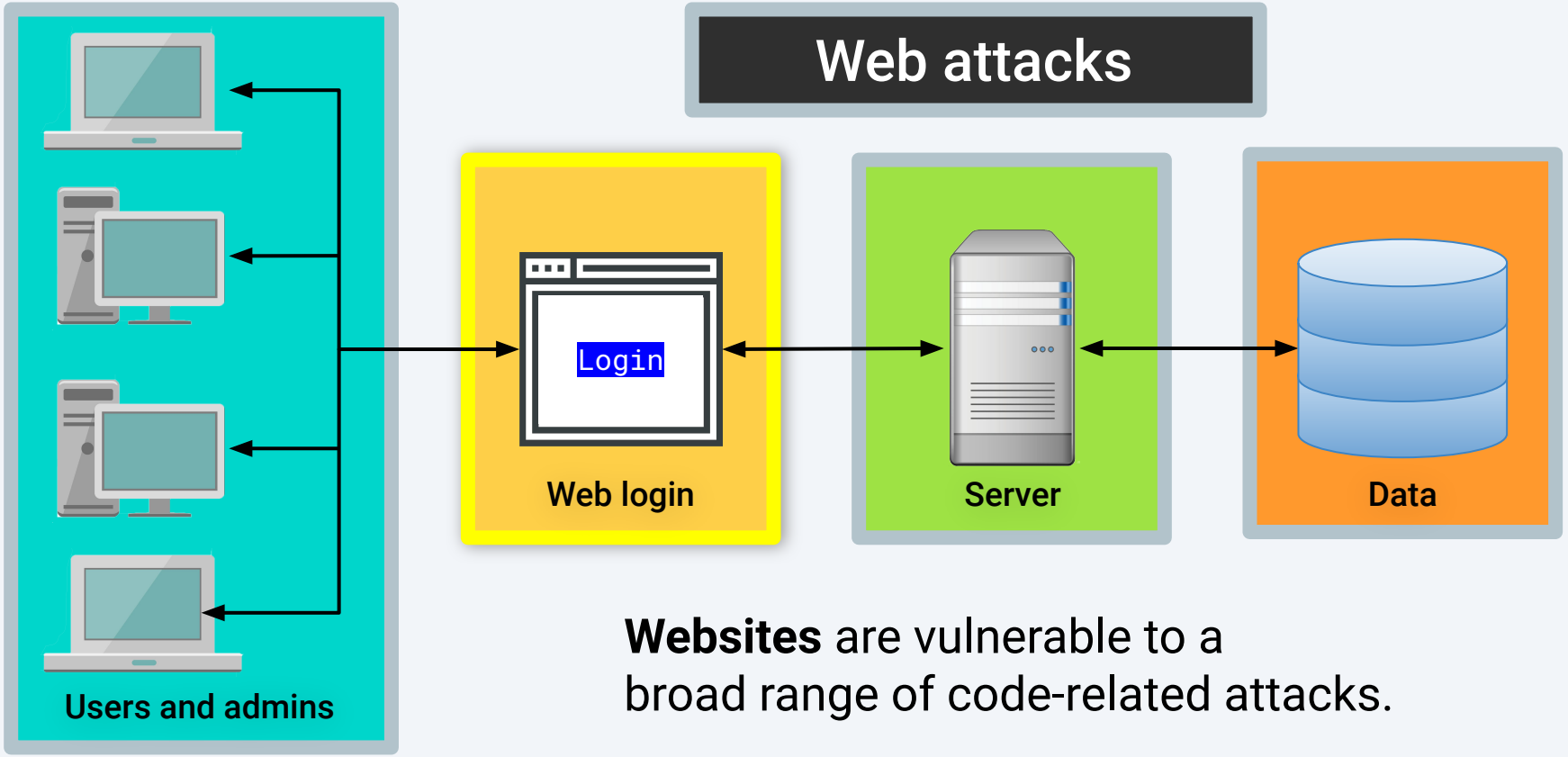
In this case, an attacker can target **every node** along the chain.

Activity: Security Task #1: Attacking the Wall



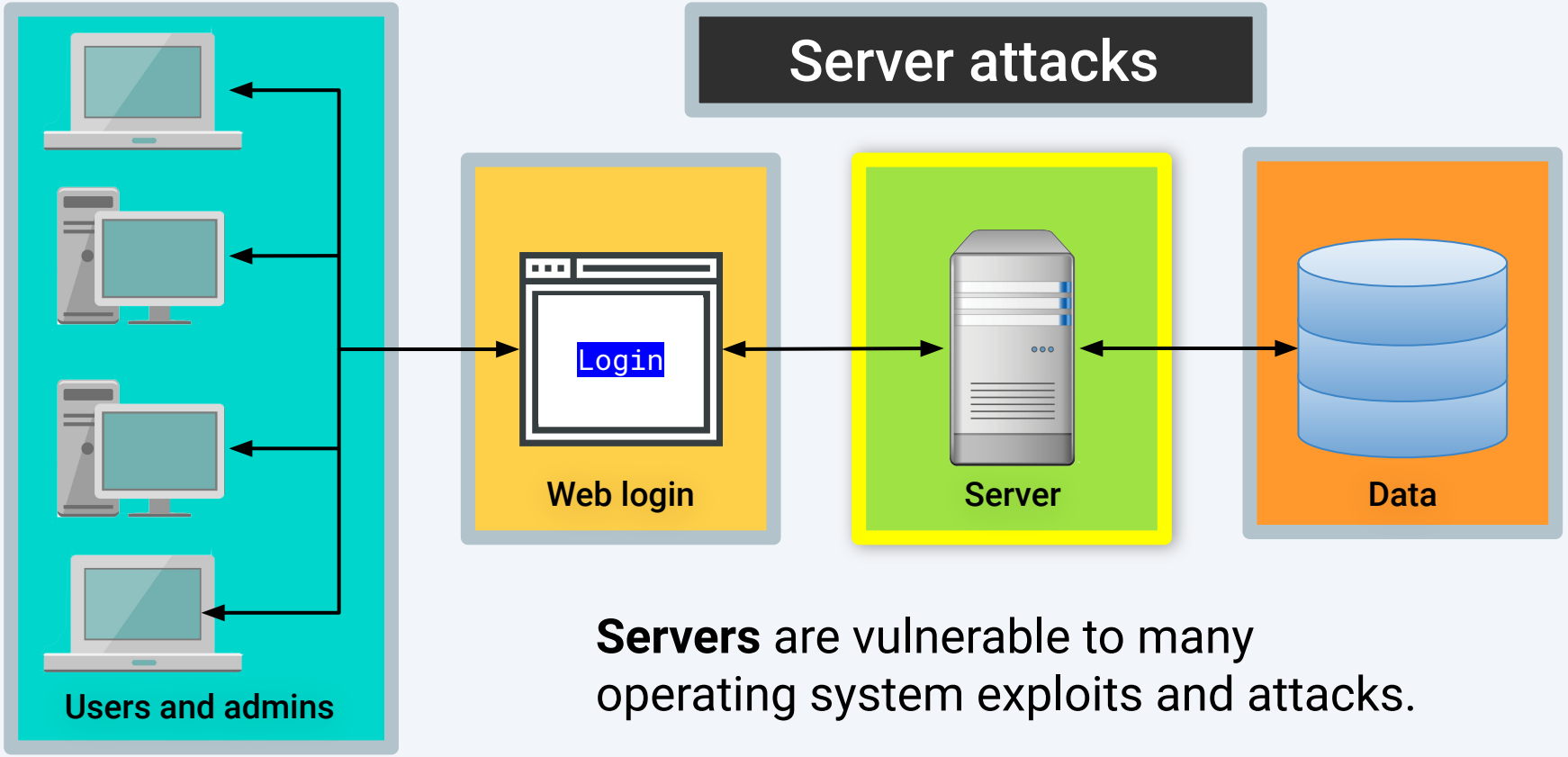
Users are subject to a variety of social engineering tactics.

Activity: Security Task #1: Attacking the Wall



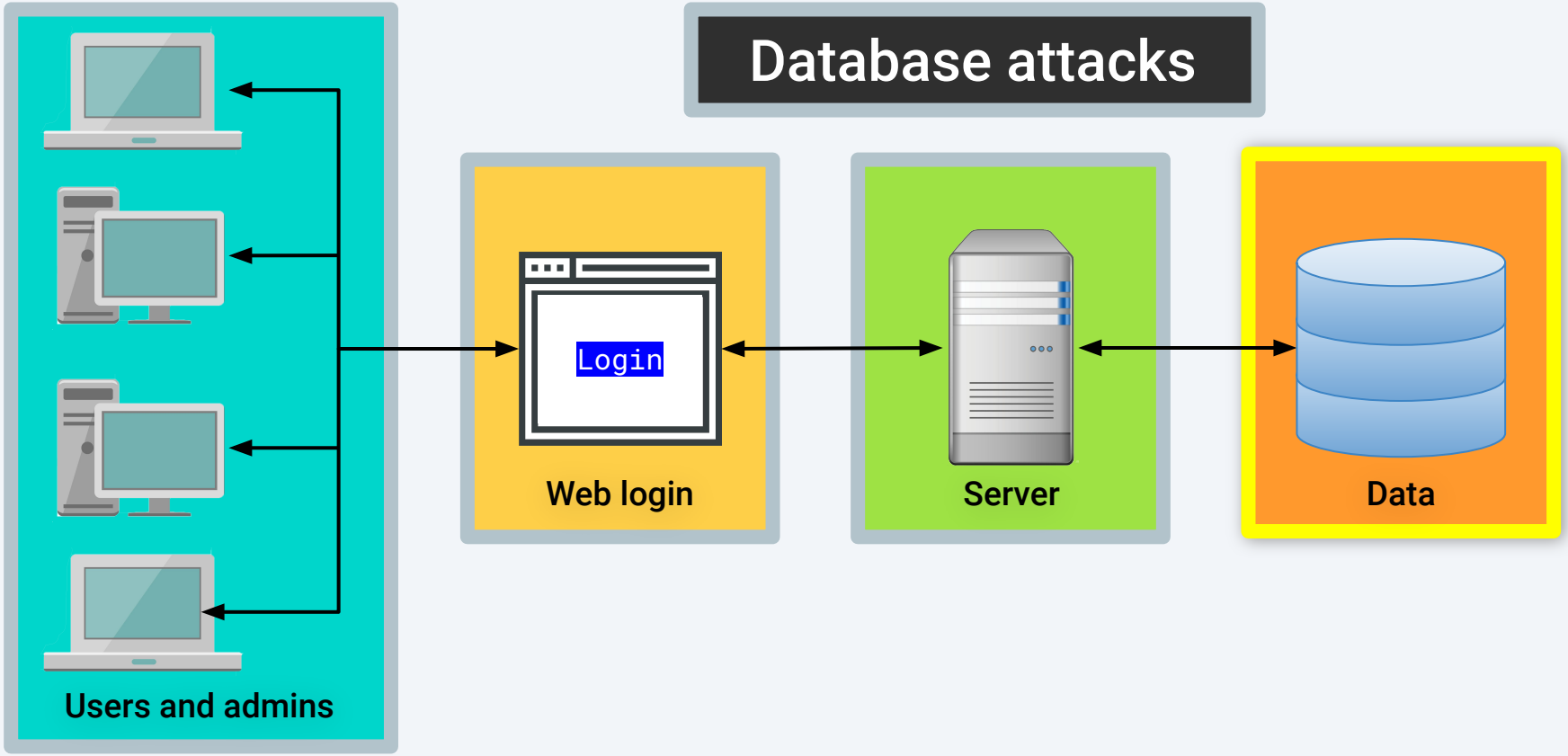
Websites are vulnerable to a broad range of code-related attacks.

Activity: Security Task #1: Attacking the Wall



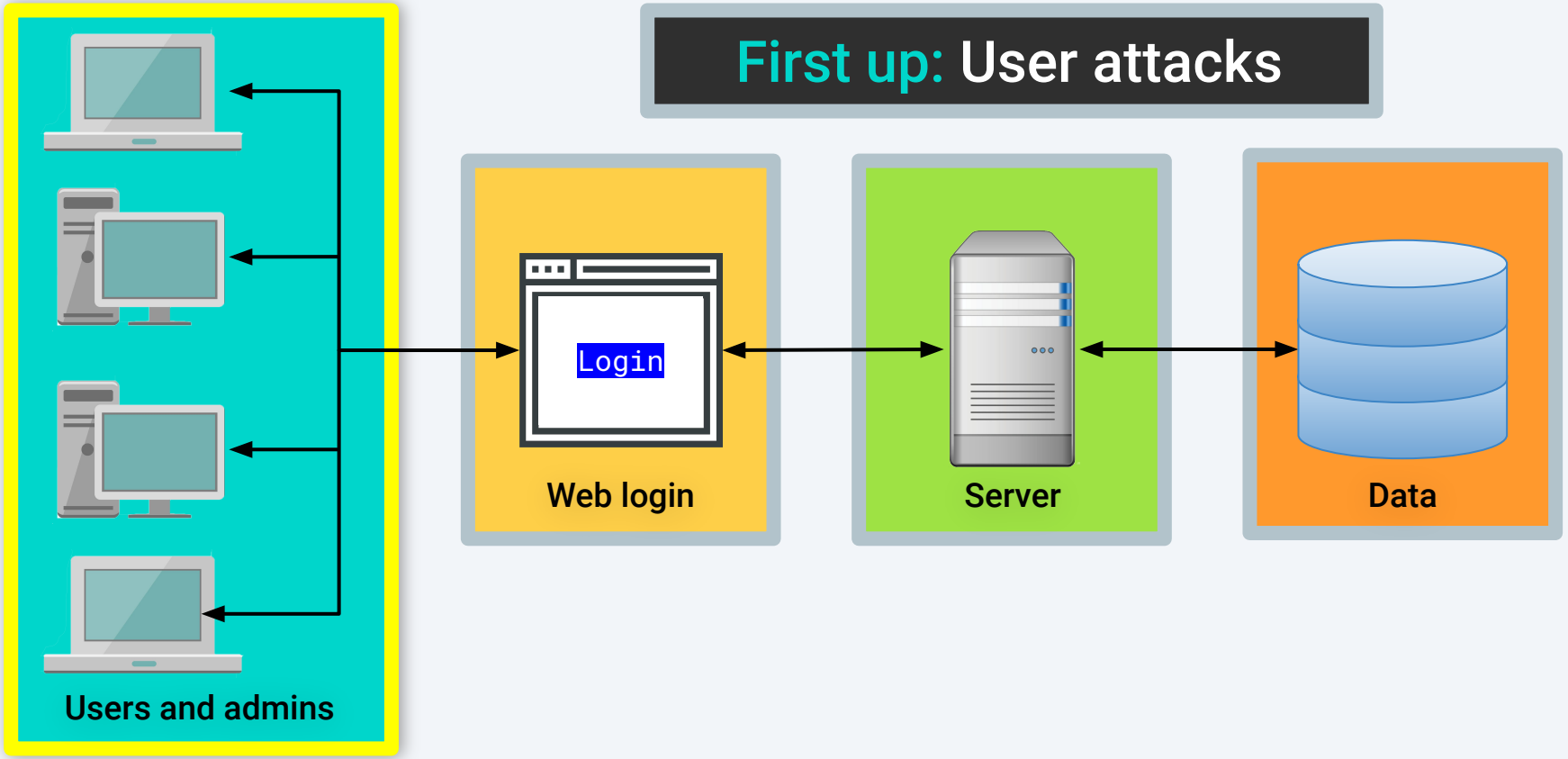
Servers are vulnerable to many operating system exploits and attacks.

Activity: Security Task #1: Attacking the Wall



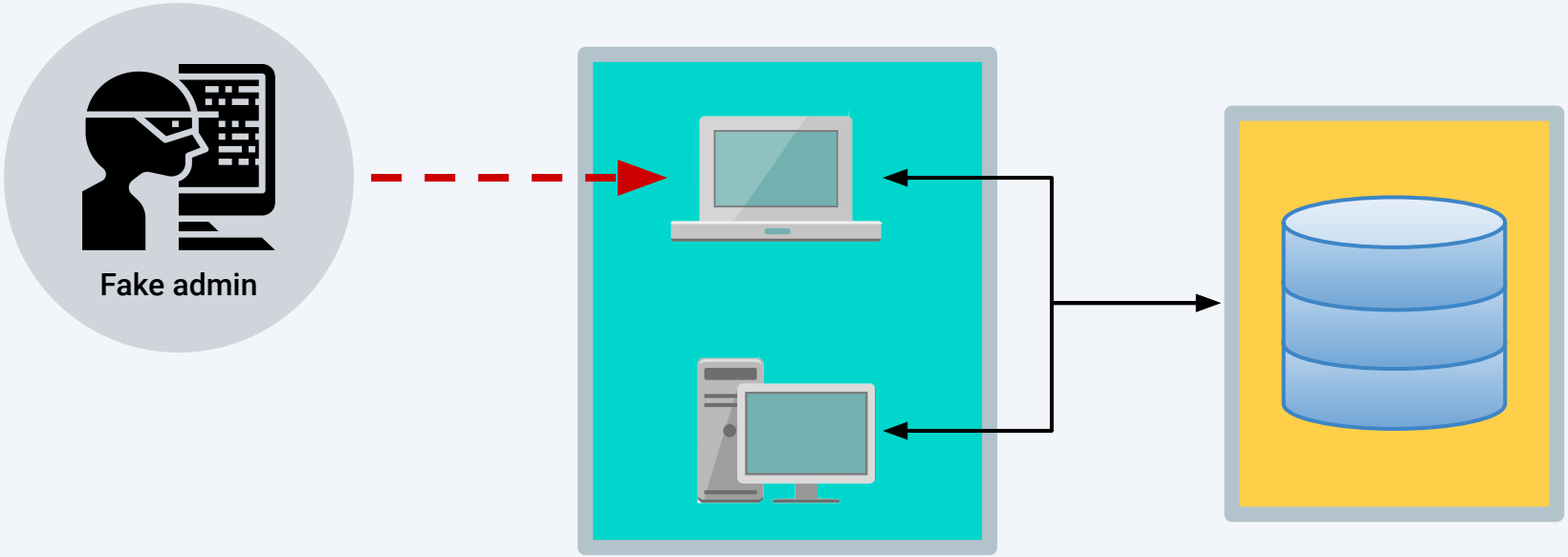
Step #2:
Define Attack Strategy

Activity: Security Task #1: Attacking the Wall



Step 2: Defining Attack Strategies

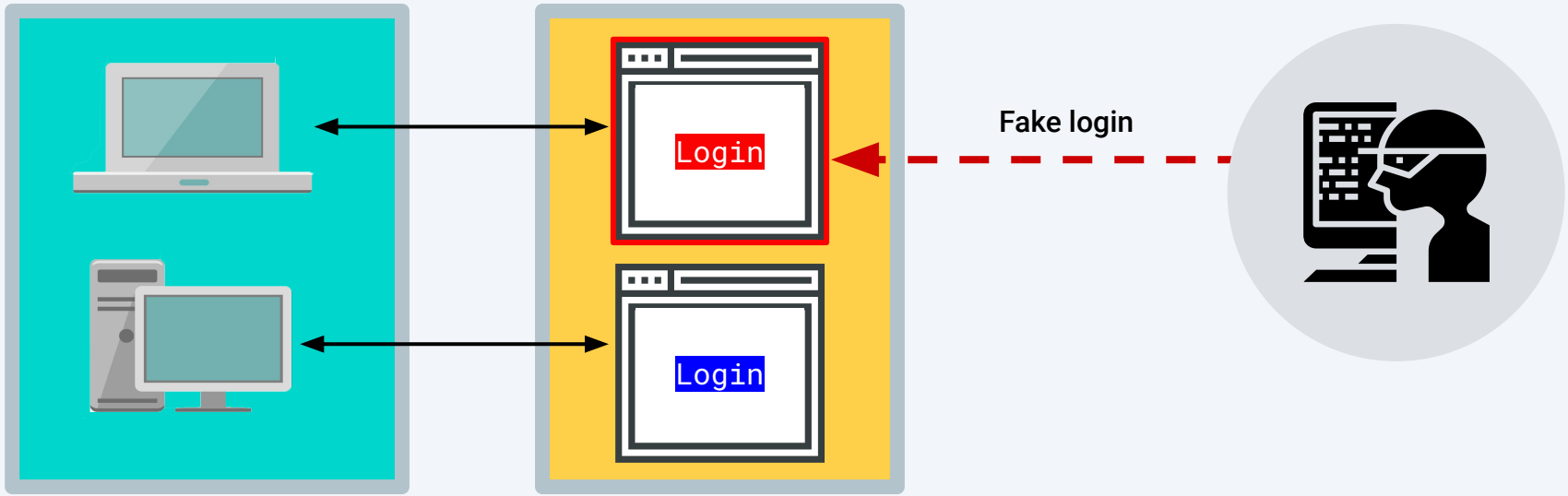
Attack option #1: Social engineering
An attacker can ask users for their credentials by pretending to be an administrator.



Step 2: Defining Attack Strategies

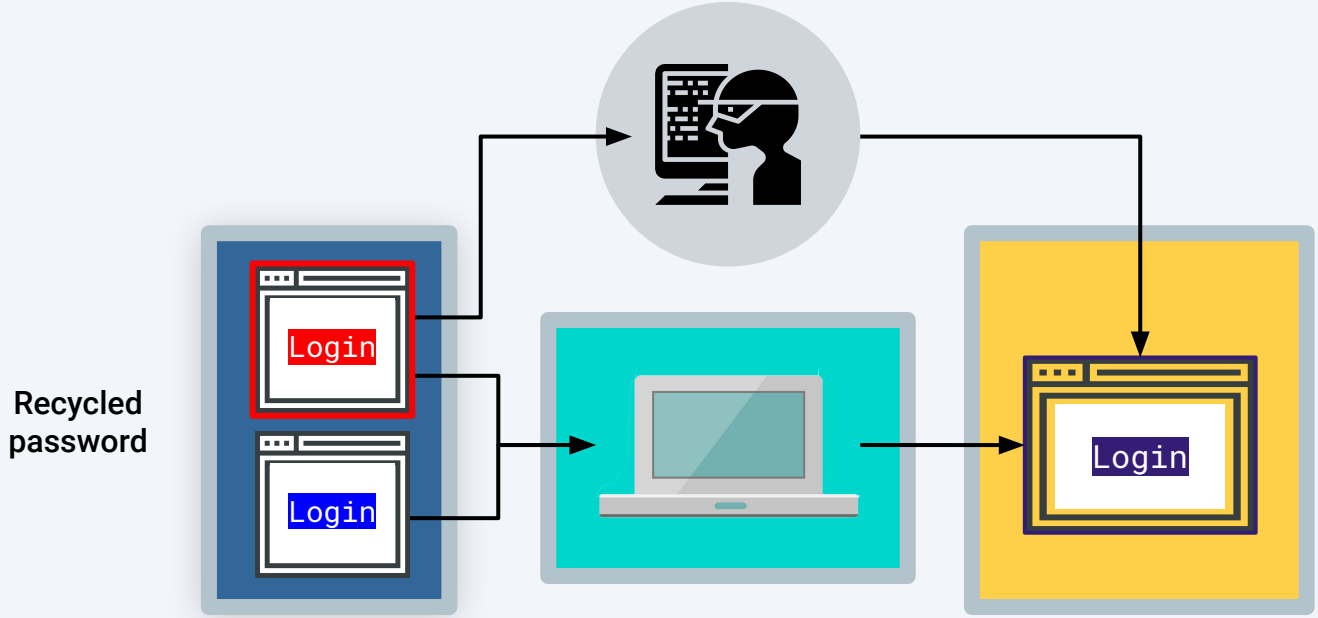
Attack option #2: Phishing

An attacker can attempt a phishing attack, where users are redirected to fake login pages that capture user credentials.



Step 2: Defining Attack Strategies

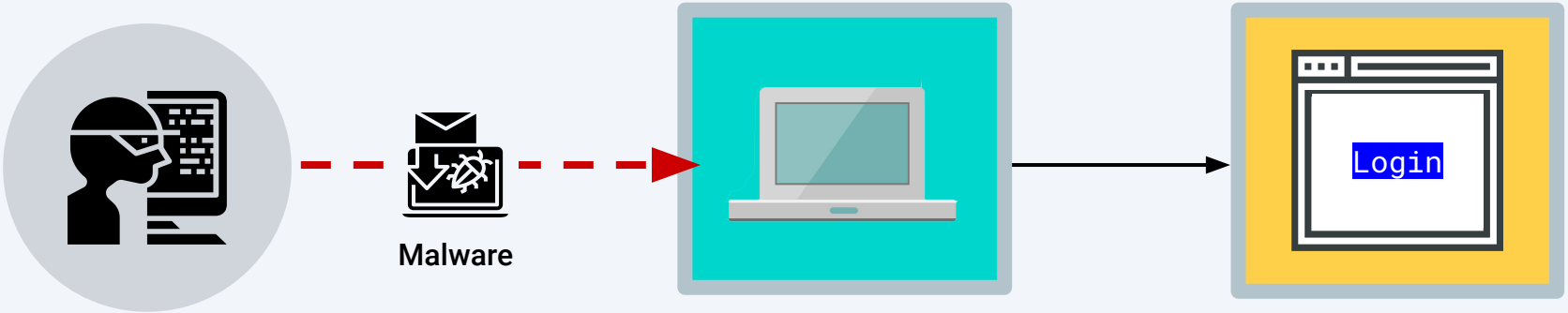
Attack option #3: Credential reuse
An attacker can find users' login and password information on other websites.



Step 2: Defining Attack Strategies

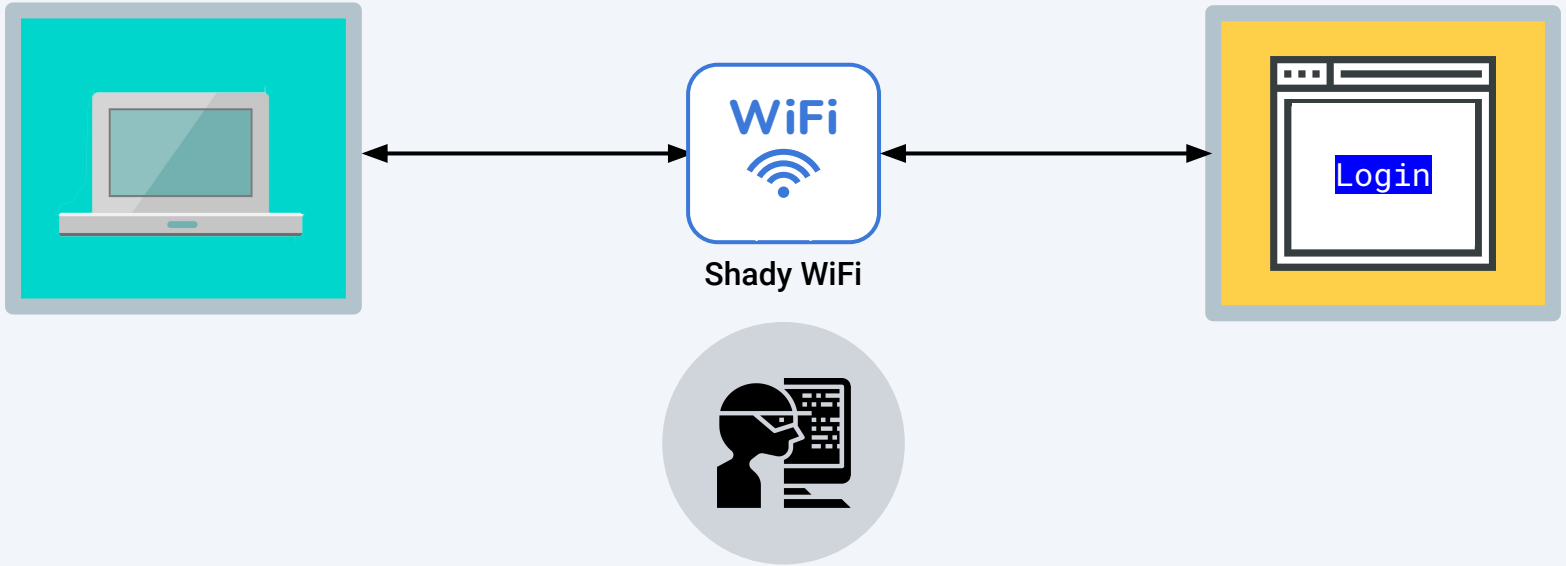
Attack option #4: Malware

An attacker can deploy malware such as spyware or keyloggers to capture daily user activity.



Step 2: Defining Attack Strategies

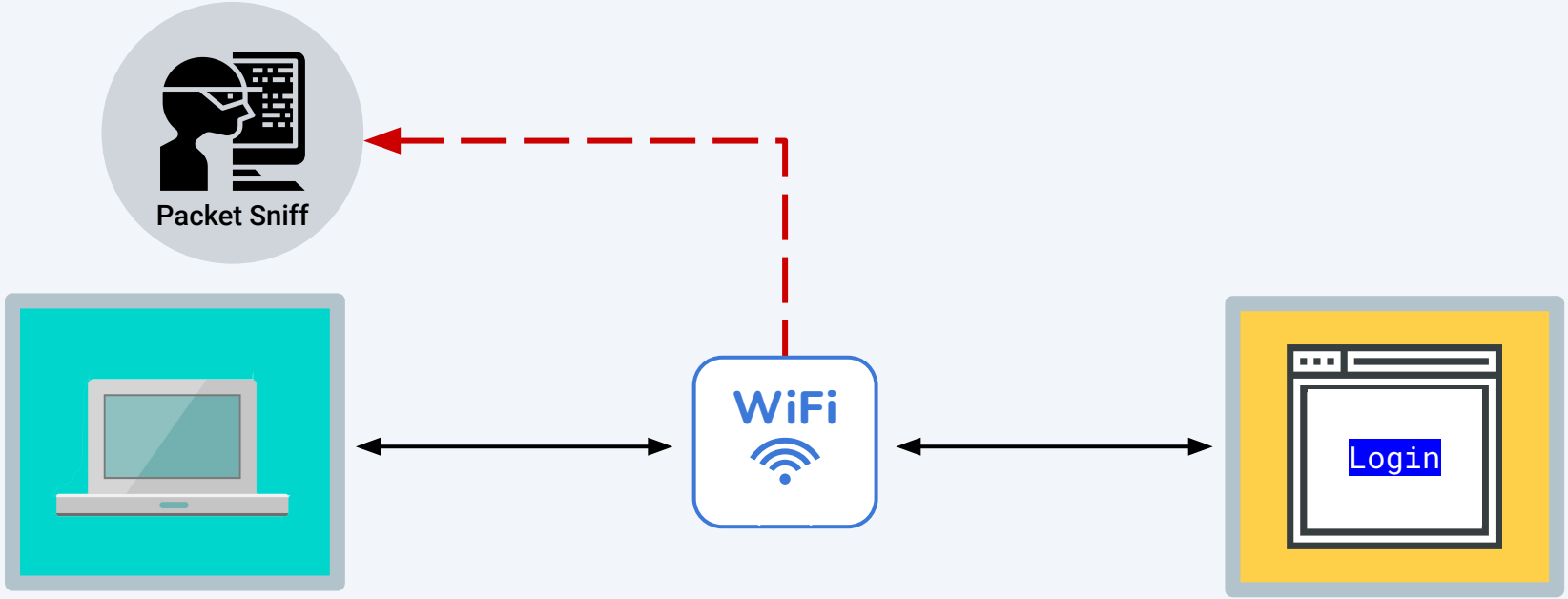
Attack option #5: Man in the middle attack
An attacker can create a man in the middle attack by providing a free WiFi hotspot to capture user credentials.



Step 2: Defining Attack Strategies

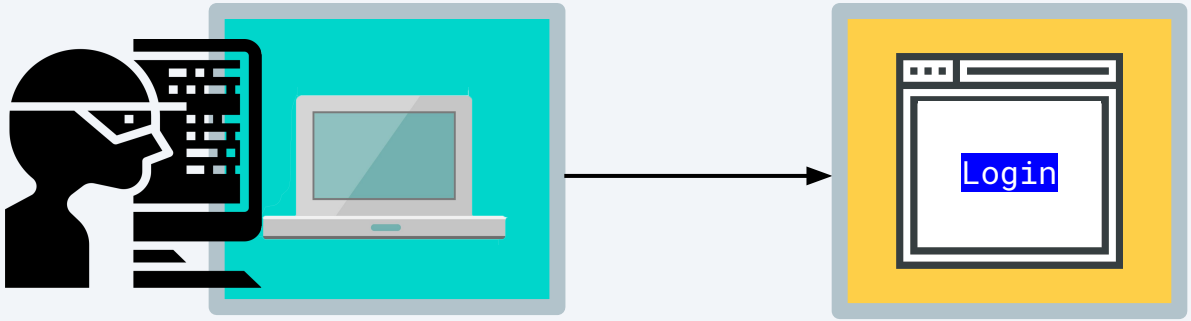
Attack option #6: Sniff packet

An attacker can sniff packet traffic across insecure wireless networks such as a cafe or restaurant.



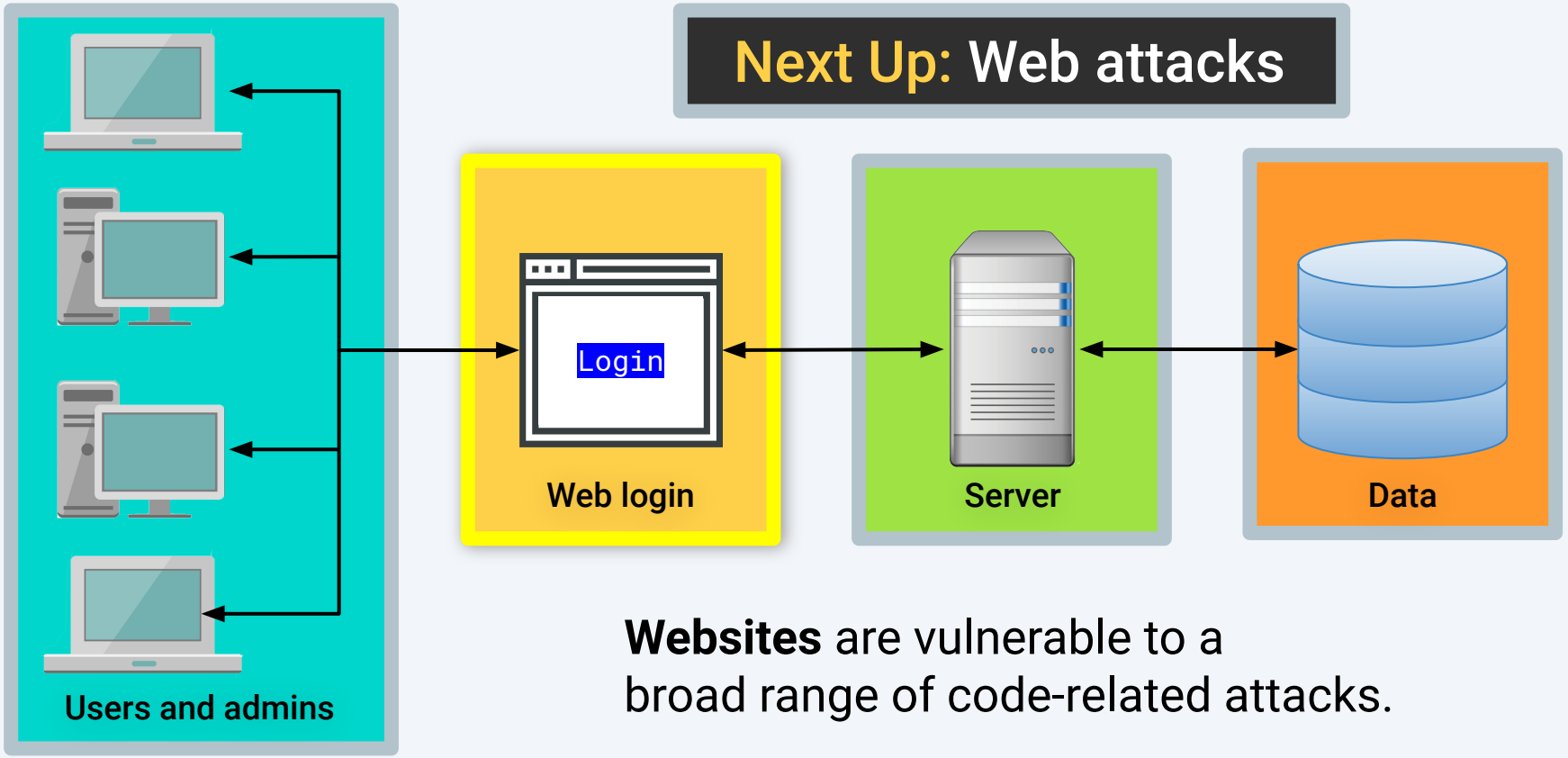
Step 2: Defining Attack Strategies

Attack option #7: Stolen hardware
An attacker can simply steal a computer and use the saved credentials to login.



Stolen laptop

Activity: Security Task #1: Attacking the Wall



Websites are vulnerable to a broad range of code-related attacks.

Step 2: Defining Attack Strategies

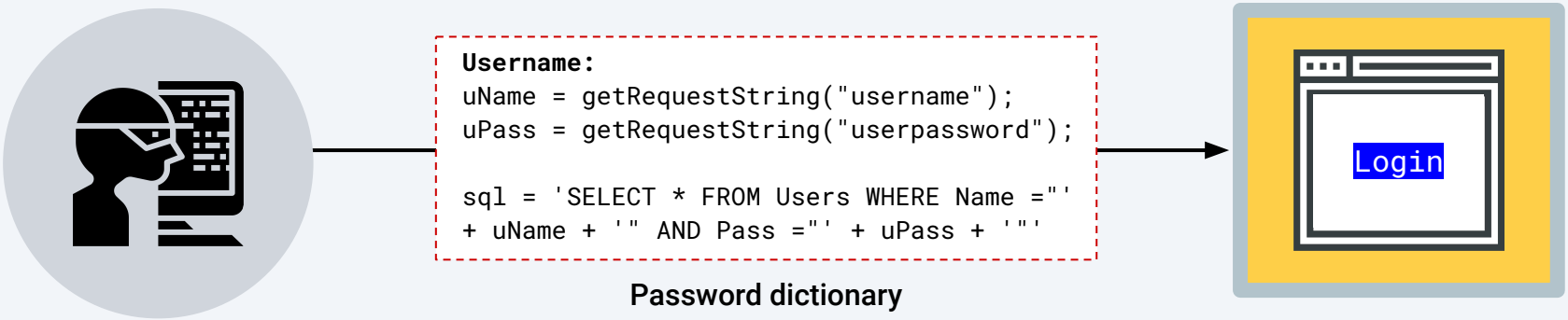
Attack option #8: Brute force attack

An attacker can use a brute force attack to continuously attempt username and password combinations.

Step 2: Defining Attack Strategies

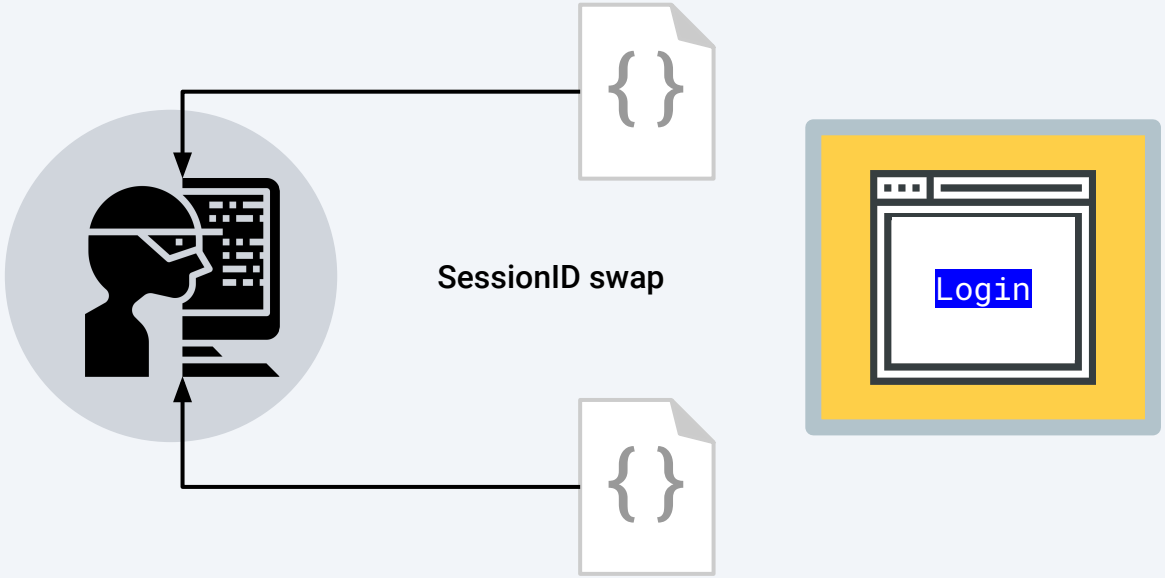
Attack option #9: Code injection

An attacker can use a code-injection attack to inject malicious code directly into username or password fields.

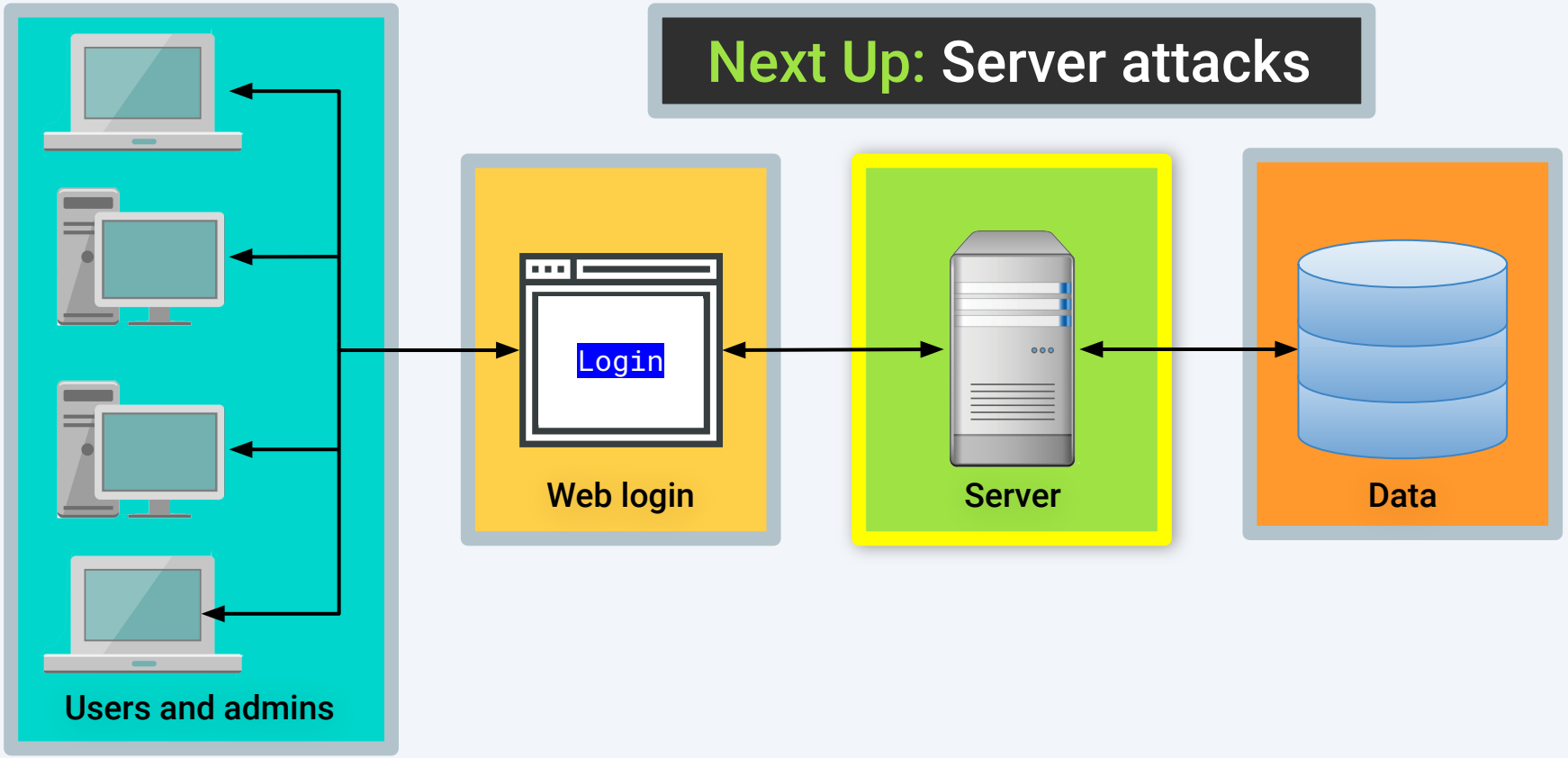


Step 2: Defining Attack Strategies

Attack option #10: Faulty session management
An attacker can exploit faulty session management, when developers incorrectly implement code used to maintain login and logouts.

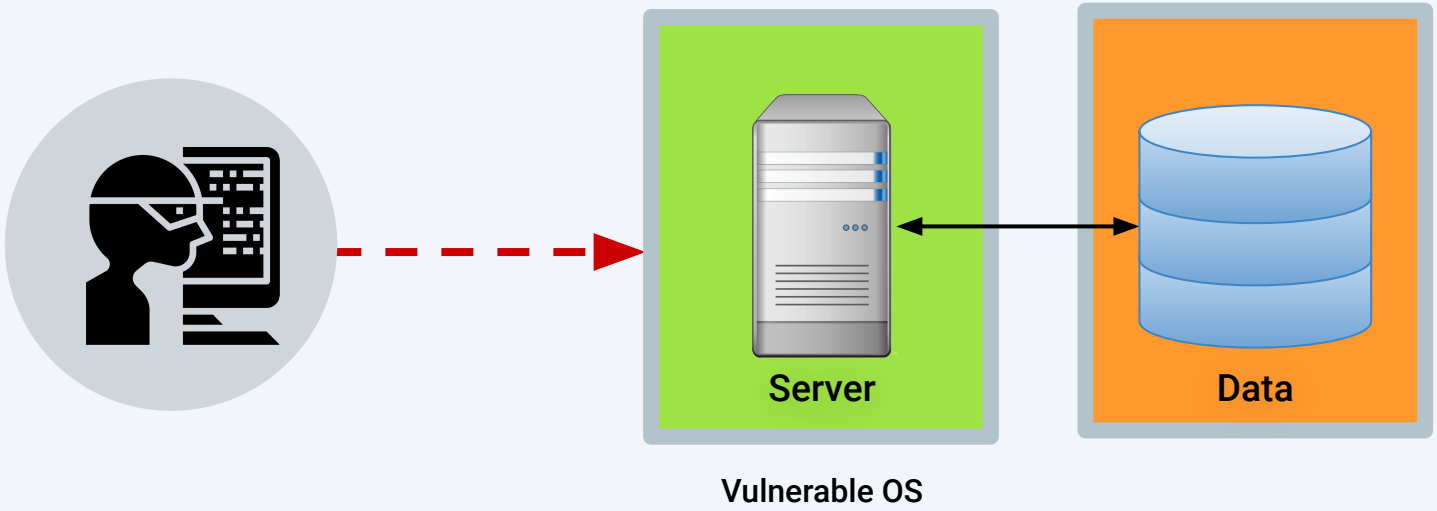


Activity: Security Task #1: Attacking the Wall



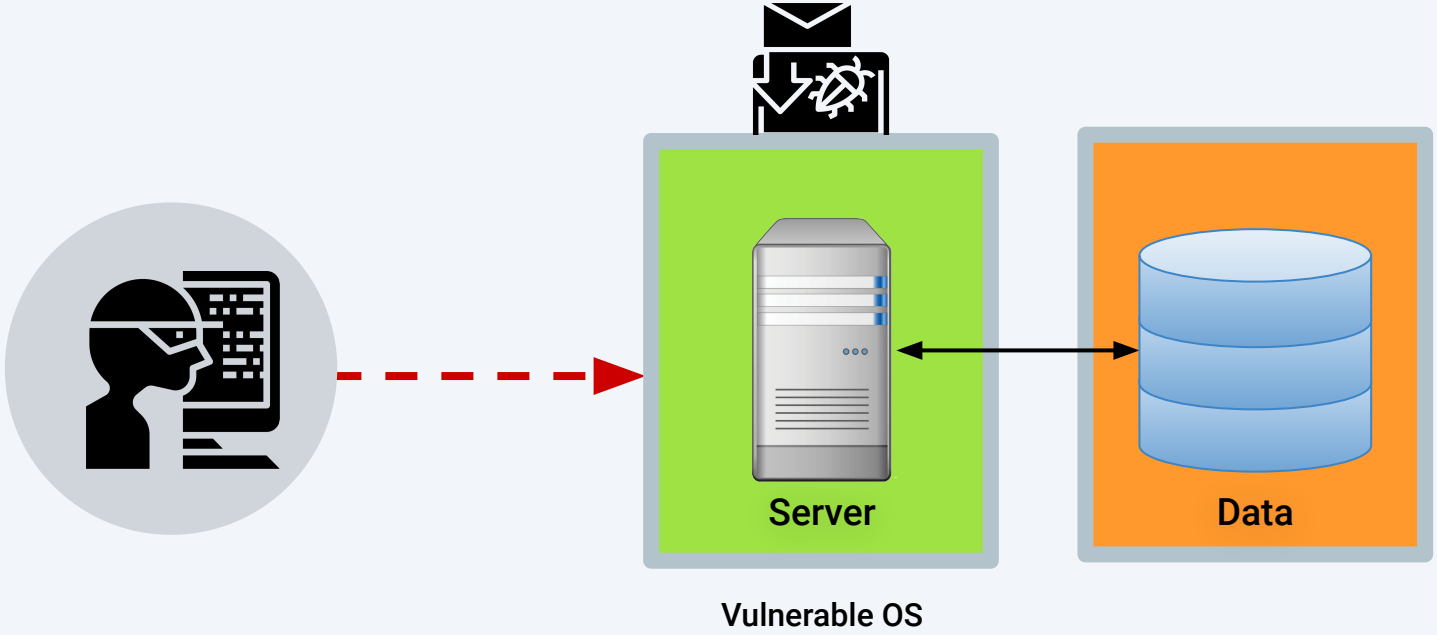
Step 2: Defining Attack Strategies

Attack option #11: OS exploits
Servers, which run on operating systems (OSs) like Windows and Linux, are subject to OS exploits when incorrectly patched.

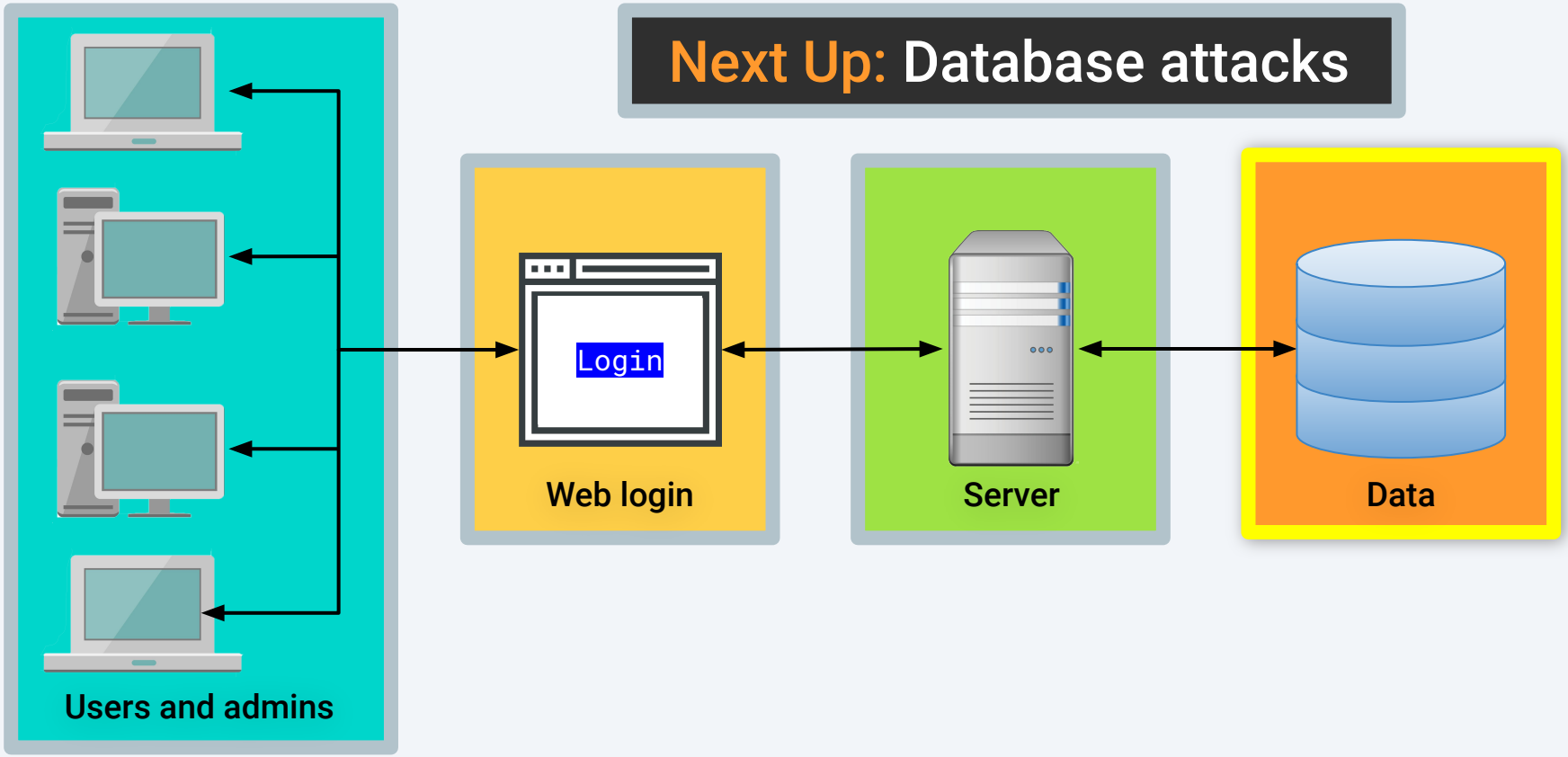


Step 2: Defining Attack Strategies

Attack option #12: Malicious software
Malicious software can be directly loaded onto the server by USB or other means.



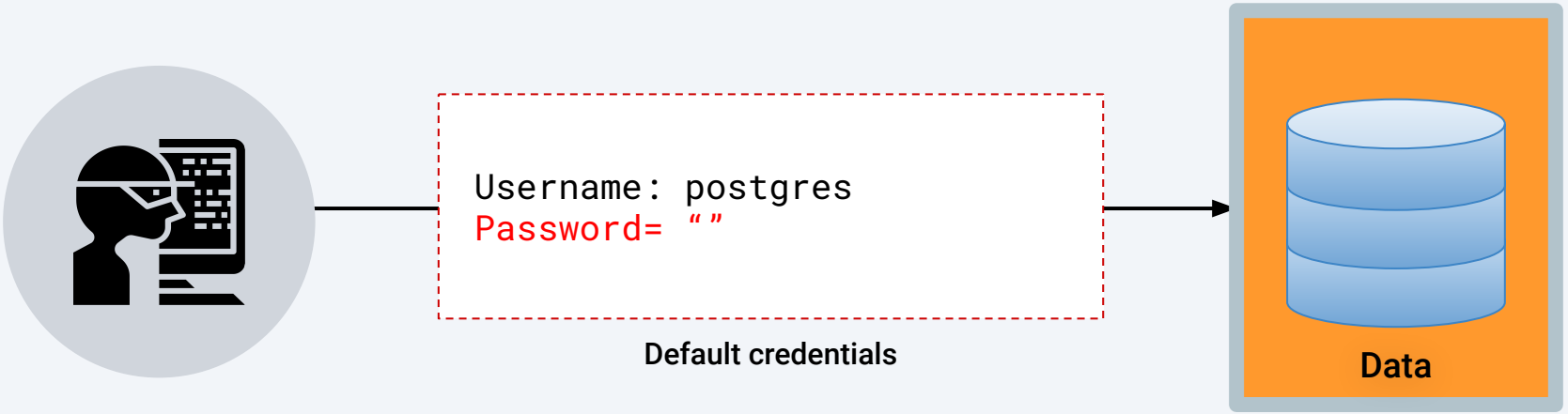
Activity: Security Task #1: Attacking the Wall



Step 2: Defining Attack Strategies

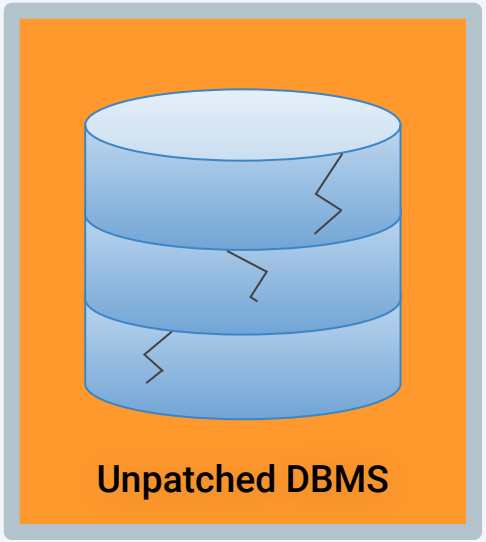
Attack option #13: Default credentials

Database management systems often come with default credentials, which might be left unchanged.



Step 2: Defining Attack Strategies

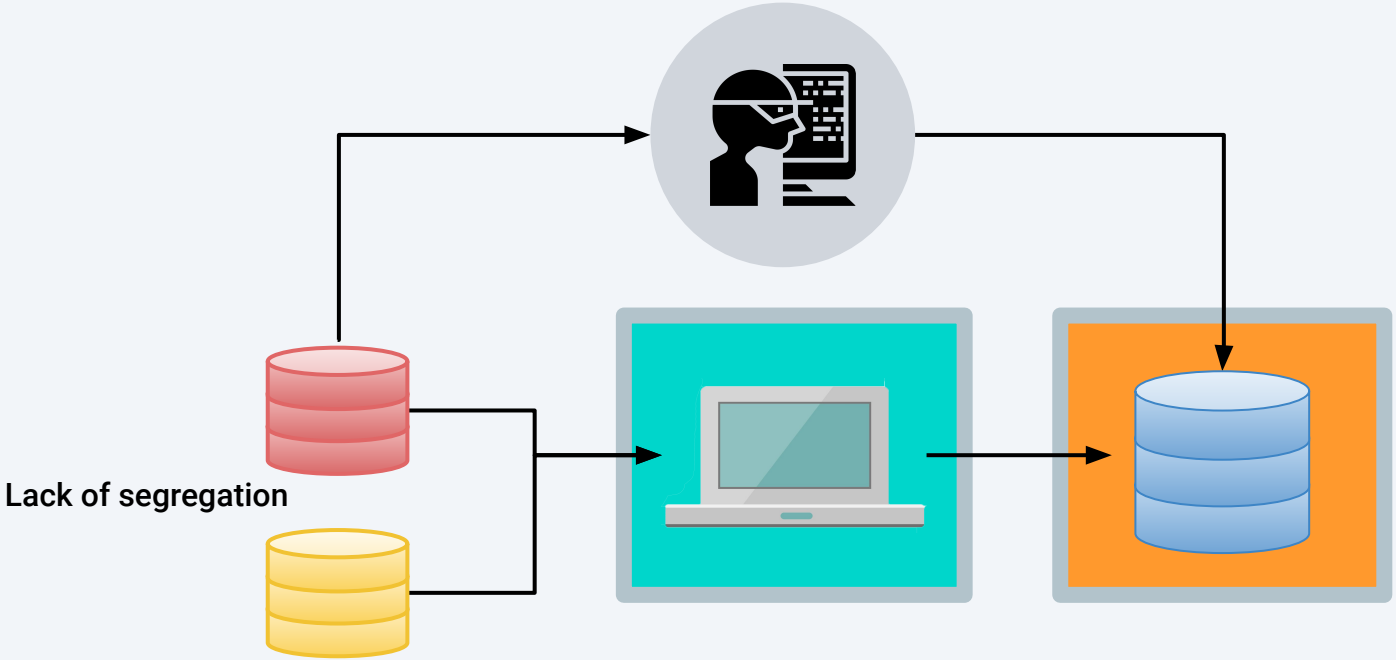
Attack option #14: Unpatched database
Database management systems might be unpatched against publicly known vulnerabilities.



Step 2: Defining Attack Strategies

Attack option #15: Lack of segregation

The database might allow a client to look at another client's data.



Security Task #2
Defending the Wall

SECURITY TASK #2

Defending the Wall

In the same groups, you will now use your list of potential attacks from the previous activity to develop a list of at least 10 strategies to mitigate the website's risk.

As you work through this scenario, consider the following:

Can any mitigation strategies be used to handle multiple threats at once?

Activity: Security Task #2: Defending the Wall

Now that we've assembled a list of potential attacks, your next task is to develop a list of at least 10 strategies to mitigate the website's risk of unauthorized access. **Be prepared to share!**

Suggested Time:

20 Minutes

Time's Up! Let's Review.

Questions?

Activity: Security Task #2: Defending the Wall

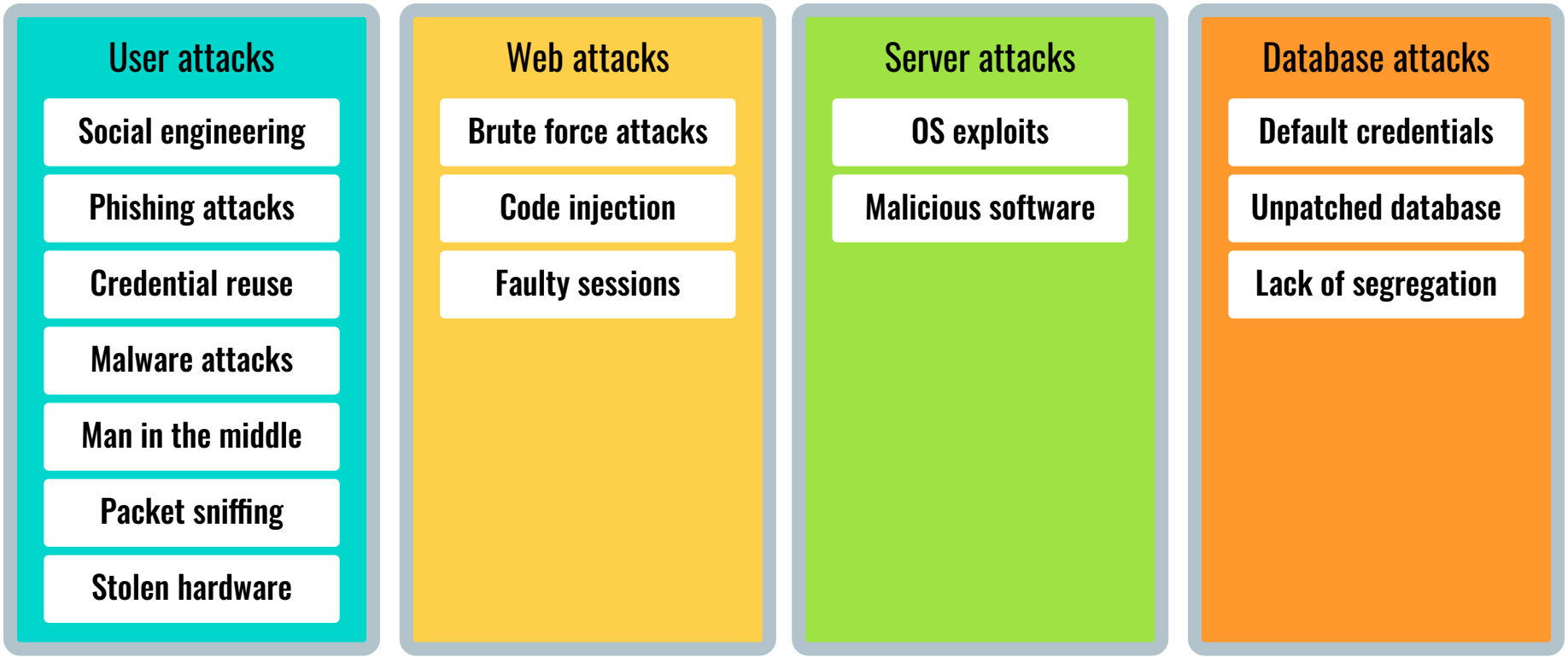
To get started, review this list of identified attack types.

User attacks	Web attacks	Server attacks	Database attacks
Social engineering	Brute force attacks	OS exploits	Default credentials
Phishing attacks	Code injection	Malicious software	Unpatched database
Credential reuse	Faulty sessions		Lack of segregation
Malware attacks			
Man in the middle			
Packet sniffing			
Stolen hardware			

Step #3:
Risk Mitigation Plan

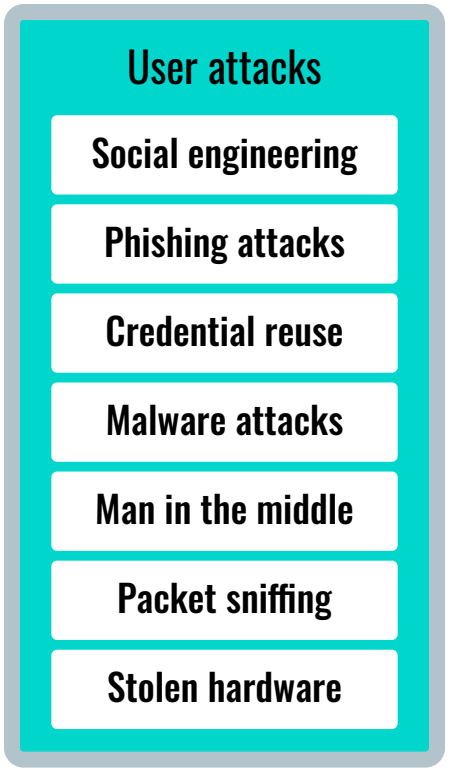
Step 3: Risk Mitigation Plan

Risk mitigation begins by assessing all risks and looking for parallels.



Step 3: Risk Mitigation Plan

Risk mitigation begins by assessing all risks and looking for parallels.



01 Educate all users on the danger of phishing and social engineering.

02 Use randomly generated passwords.

03 Ensure users are using multi-factor authentication (password + phone confirmation).

04 Use HTTPS and only access sensitive content over secure channels.

Step 3: Risk Mitigation Plan

Web and Server Risk Mitigation

Web attacks

- Brute force attacks
- Code injection
- Faulty sessions

Server attacks

- OS exploits
- Malicious software

01 Ensure strong passwords are used (alphanumeric + symbol + special characters).

02 Sanitize any input in the web application form fields and filter the output.

03 Ensure users are immediately logged out when closing a browser. (Logins are erased after 30 seconds of inactivity.)

04 Ensure all servers are routinely patched against latest known vulnerabilities.

05 Incorporate antivirus and user education.

Step 3: Risk Mitigation Plan

Suggested Plan

01 Educate all users on the dangers of phishing and social engineering.

02 Require randomly generated passwords.

03 Ensure users have multi-factor authentication (password + phone confirmation).

04 Use HTTPS and only access sensitive content over secure channels.

05 Ensure strong passwords are used (alphanumeric + symbols).

06 Sanitize any input in the web application form fields and filter the output.

07 Ensure users are immediately logged off when closing a browser.

08 Ensure all servers are routinely patched against latest known vulnerabilities.

09 Ensure physical access to servers is protected by multiple forms of authentication (login + biometric).

10 Ensure all data stored in the database is encrypted and cannot be read without additional login information.

11 Provide database access on need-to-know basis.

12 Log and monitor all database access.

13 Ensure all cloud security platforms follow best practices for security implementation.


15:00
Break

For the rest of today's class, we will set up the virtual environments that we'll use for the majority of technical activities in the course.

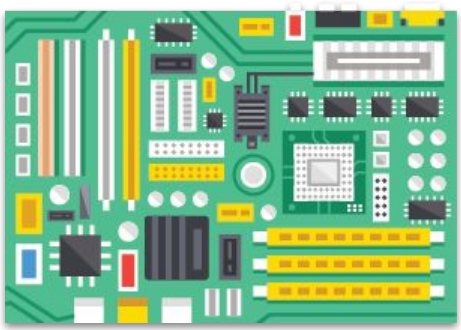
Introduction to Virtual Machines

Physical Machines

To most people, a computer means a physical laptop or desktop computer made of hardware, such as:

Monitors

Graphic cards

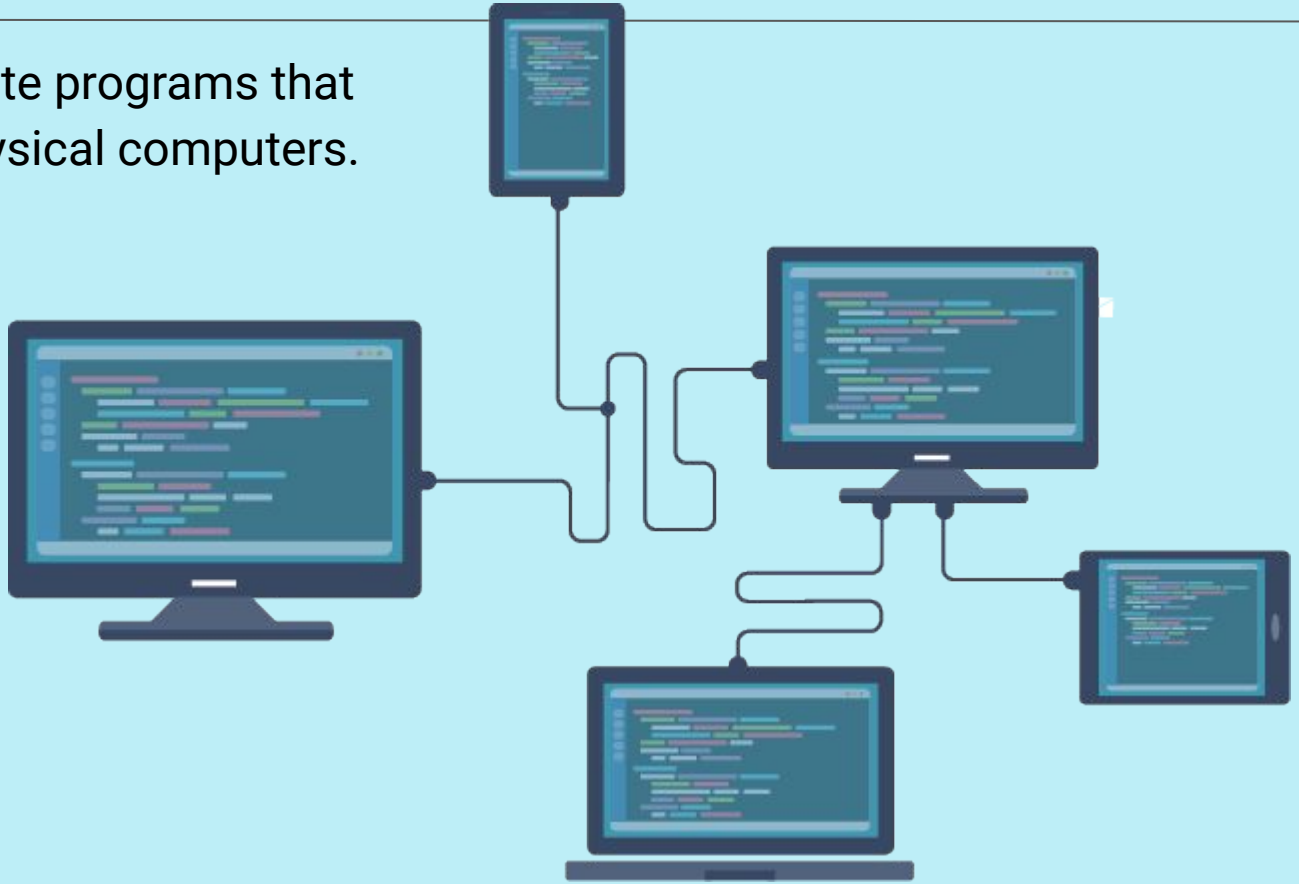


Hard drives

etc.

Virtual Machines

It is possible to write programs that simulate entire physical computers.



Virtual Machines

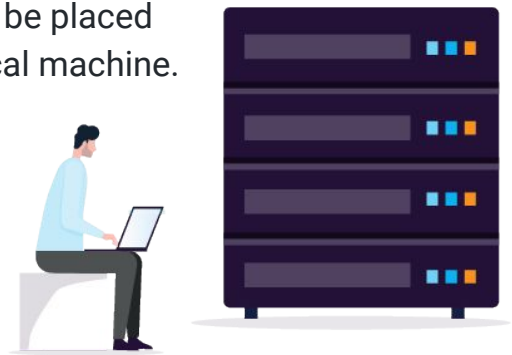
VMs are programs that simulate entire computers.

- VMs can operate as entirely different computers than the one they're running on.
- We can run many different VMs on a single physical computer.

Physical vs. Virtual Machines

Virtual

- Easy and inexpensive (or free) to set up.
- Easily distributed. In this class, we'll distribute VMs so that each student runs the exact same setup.
- Multiple VMs can be placed on a single physical machine.



Physical

Typically more efficient because they have direct access to hardware components.

Let’s get started setting up our virtual environments.

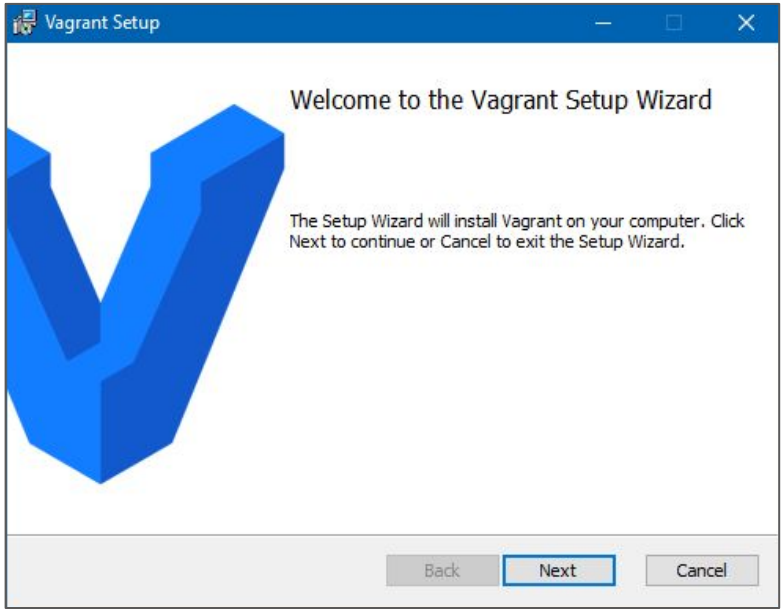
Setting Up Our Virtual Machines

For the rest of the class, we'll focus on the three-step installation process outlined in the **Using Vagrant** document:

01 Accessing the command line and downloading VirtualBox and Vagrant.

02 Downloading the VM using Vagrant files and scripts.

03 Accessing the VM.



Everyone Do: Local Machine Setup

As a class, we will set up our VMs.

Make sure you have access to the **Using Vagrant** documentation.

Suggested Time:

Step 1

Accessing the command line and downloading VirtualBox and Vagrant.

To run VMs, we will first need to make sure that we have the following tools installed:

Git Bash
for Windows users

Terminal
for Mac users

VirtualBox
A virtualization tool that we'll use to run various lab activities. VirtualBox allows us to run different operating systems on our local machines.

Vagrant
A tool we'll use to build and set up these virtual environments. Vagrant will allow us to run scripts that install these VMs, which will then be run using VirtualBox. We will run these scripts using Git Bash or Terminal.

Does everyone have Git Bash or Terminal, VirtualBox, and Vagrant installed?

If so, let's move on.

Step 2

Downloading the VM using Vagrant files and scripts.

Now that we have our tools installed, we need to download the following files:

`vagrant-linux.sh`

A script file that ensures your VM is installed properly on your computer.

`Vagrantfile`

A file that configures and defines your virtual machine setup.

In our case, this Vagrantfile, when executed via the `vagrant-linux.sh` script, configures the custom Linux Ubuntu VM you are using.

Has everyone set up their VM using `vagrant.linux.sh` and `Vagrantfile`?

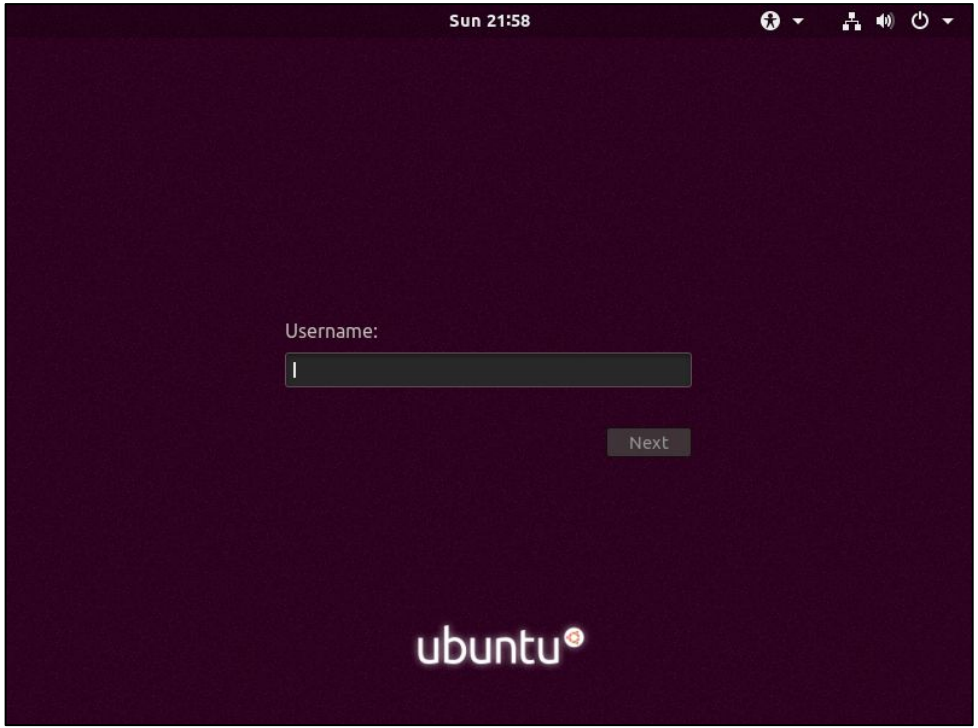
If so, let's move on.

Step 3

Accessing the VM.

Now we will access the **graphical user interface (GUI)** of our VM.

Use **Part 3** of the Using Vagrant documentation to access your VM.



VM Setup Confirmation

At this point, everyone should have:

01

Accessed the command line and downloaded VirtualBox and Vagrant.

02

Downloaded the VM using Vagrant files and scripts.

03

Accessed the VM.

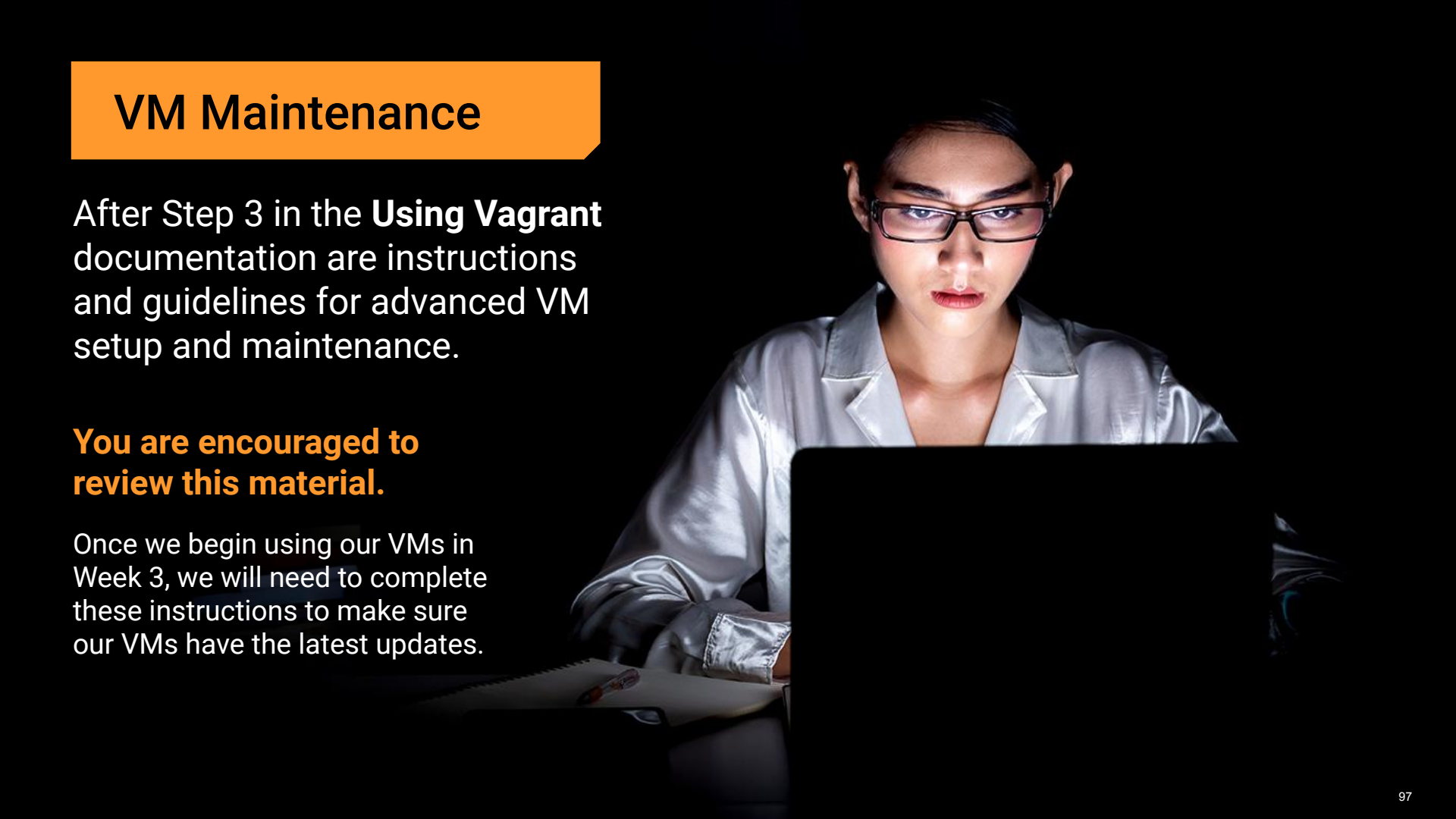
VM Maintenance
(if time allows)

VM Maintenance

After Step 3 in the **Using Vagrant** documentation are instructions and guidelines for advanced VM setup and maintenance.

You are encouraged to review this material.

Once we begin using our VMs in Week 3, we will need to complete these instructions to make sure our VMs have the latest updates.



Important:

Do not run apt-update,
or update guest additions.

VM Maintenance

Specifically, we will run the following commands to get our VMs up to date:

`vagrant box update` to get the most recently updated VM.
This might take several minutes or longer, depending on your internet connection.

`vagrant destroy` within the directories where your Vagrantfiles are installed, to ensure that the VMs are stopped and all associated files are removed.

`vagrant up`, to launch the newer version.

`vagrant box prune` (optional) afterwards, to delete all old, unused versions of the VM.

Questions?


The End