1. **Overall Vulnerability/Risk analysis 0.5p**

   Business risk (Executive summary)

   Data breaches can lead to violations of data protection regulations. This can result in large fines to the company. Other monetary losses can occur if the attacker gains access to payment information. System compromise is also a possibilty. All of these can lead to reputational damage.

   Technical risk (Technical summary)

   File Inclusion (LFI, RFI)

   The vulnerability allows the attacker to read data from the server by utilising file inclusion.

   Command Injection (OS Command Injection, CWE-78)

   The vulnerability allows the attacker to remotely execute code on the system.

   Burp Suite: Intruder (Broken Object Level Authorization (BOLA), CWE-639)

   The vulnerability allows the attacker to access all of the ticket systems tickets with a single account.

2. **Vulnerability impact review 0.5p**

   Risk likelihood

   File Inclusion

   The vulnerability requires little technical know-how to take advantage of.

Command Injection

The vulnerability is extremely easy to take advantage of, even by accident.

Burp Suite: Intruder

The vulnerability requires some technical know-how to take advantage of. No one will do it by accident.

Potential impact

The potential impact of the vulnerabilities are all very critical.

3. Remediation options 0.5p

File Inclusion

Filtering the users input to only allow the user to access the desired files. RCE can be prevented by not allowing urls to be inputted. Requiring proper credentials for admin access and not relying on a plain text value of the cookies.

Command Injection

Reducing the use of functions that interact with the system and filtering the users input to only allow the desired commands to be executed.

Burp Suite: Intruder

Changing all the affected users' passwords is the first step that needs to be taken. Changing the ticket system so that only the tickets associated with the logged in user are shown is also a good idea. In the case of an attack this will reduce the amount of leaked information. To prevent any future attacks storing credentials in an encrypted format is advised.

4. **Technical findings 2.5p**

File Inclusion

Data can be read from the challenge-1 server by making a POST request instead of a GET request.

In challenge-2 modifying the cookies Value parameter from “guest” to “Admin” gives admin access to the site. When admin access is achieved data can be read from the server by replacing the cookies Value paremeter with “../../../../../etc/flag2%00”.

In challenge-3 capturing the request with Burp and changing the request to POST and setting the “file” parameter to “../../../../etc/flag3%00” gives access to the desired file.

RCE can be achieved on the server by inputting an URL in the sites form that points to a file containing PHP code.

For example <?php echo exec(“hostname”);?> sent to the server will run the hostname command on the remote server showing its output on the page.

Command Injection

Commands can be run on the remote server by instead of inputting an IP to the application, giving it the desired linux command in the following format “|(command)”.

Burp Suite: Intruder

Burp suite can be used to detect all the tickets in the system by launching a sniper attack on the site “**GET /support/ticket/§number§ HTTP/1.”** where “number” is the ticket number in the system. Number is incremented from 1 to 100. Five tickets are found in the system. Tickets not shown on the users dashboard are also found with the same credentials.

5. **Original cause 0.5p**
   File Inclusion
   The root cause of the vulnerability is the lack of sanitisation of the users input.

   Command Injection
   The root cause of the vulnerability is the use of functions that interact with the system and the lack of sanitisation of the users input.

   Burp Suite: Intruder
   The root cause of the vulnerability is leaked credentials which give the attacker access to the application. The attackers ability to see all of the tickets in the system after gaining access is caused by a **Insecure Direct Object References** (IDOR) vulnerability.

6. **Conclusions 0.5p**

   Most likely compromise scenarios

   File Inclusion

   The vulnerability would most likely be exploited by an actor wanting to access sensitive information on the server. For example account information, packages installed etc.

   Command Injection

   Would most likely be exploited an actor wanting to export sensitive information or install malicious packages to the server.

   Burp Suite: Intruder

   Would most likely be exploited by an actor wanting access the ticket system in the hope of finding sensitive information in some of the tickets.

Implications

File Inclusion

Unauthorized exportation of information caused by unsanitized user input.

Command Injection

Remote Code Execution caused by use of functions that interact with the OS and lack of filtering for the users inputs.

Burp Suite: Intruder

Tickets shown to users that they are not associated with caused by a IDOR vulnerability.