
OWASP 2024
AppSec DAYS
SINGAPORE
TM

OWASP 2024
AppSec DAYS
SINGAPORE

OCT 1-2 2024
OCT 1 TRAINING
OCT 2 CONFERENCE

Unlocking the Gates: Understanding Authentication Bypass Vulnerabilities

Vikas Khanna

Agenda

- **Phase 1 - Introduction**
- **Phase 2 - Techniques & Vulnerabilities to bypass Authentication**
- **Phase 3 - Demo: Authentication Bypass in Apple**

Who Am I?

Vikas Khanna

- **Pentester| Bug Bounty Hunter | Speaker**
- **Focus on Web and API Security**
- **Acknowledged by Google, Apple, Microsoft, Verizon, Sony, and etc.**
- **Contributor in OWASP | WSTG and WWW-Community**
- **CVEs in IBM, Oracle Enterprise Products**

What is Authentication?

- **Authentication is the process of verifying the identity of a user or system to determine if they are who they claim to be.**
- **It is a fundamental security mechanism used to control access to resources and ensure that only authorised individuals or entities are granted access to sensitive information, systems, or services.**
- **Authentication typically involves the use of credentials such as usernames, passwords, biometrics, or digital certificates to validate the identity of the user or system before granting access.**

Functionality Abuse Vectors

- Login (Username:Password)
- Forgot Password
- Sign Up
- Security Question Page
- Login with OTP (Email/Phone Number)
- Update Email/Phone Number/Profile etc.
- Change Password
- Admin Page

Real Abuse Vectors

Authentication Bypass: Session Puzzling

- The user access the forgot password page and enters his/her email/username.
- If the email/username matches with already registered user, the app will redirect to security question page.
- For example, the security question URL is
  - https://www.0xnoob.com/forgotpassword/securityquestion
  - The application will create session for the user mapped with the email/username entered to access the security question page.
  - If the application is misconfigured, it will use same session variable for security question page and the post login modules.
  - As a result, we can access the post login modules of the vulnerable application by just navigate to those modules such as https://www.0xnoob.com/dashboard

Authentication Bypass: Session Fixation

- **Happens when attacker can fix the session and trick victim to use the fixed session.**
- **Pre login and Post login token is same.**
- **Can be performed locally by accessing the victim's device/shared device and remotely if the session token is in URL.**

Authentication Bypass: Access Control Checks

- IDORs (Insecure direct object references)
  - ?userid=1, ?userid=2 etc
- Privilege Escalation
  - Navigate to the modules developed for admin users from non-admin user's account
- Forced Browsing
  - Modules/Pages which are not linked to application but are still present on the server. May disclose sensitive information.

Authentication Bypass: JS Files Analysis

- The JS files may contain the sensitive information such as
  - API Keys
  - Credentials
  - Sensitive API links/ URLs/ Calls

Authentication Bypass: Password Reset Link Hijacking via Host Header Injection

- When application is creating reset token links on the basic of Host header's value
- When the response is 3XX Redirect

Normal Request
Host:0xnoob.com → **Response (3XX)**
Location:0xnoob.com (same value as host header)

After Manipulation of Host Header

Malicious Request
Host:attacker.com → **Response (3XX)**
Location:attacker.com (same value as host header)

Authentication Bypass: Password Reset Link Hijacking via Host Header Injection

- **Navigate to "Password Reset" page and change the Host Header (Use client-side proxy tool such as Burp Suite)**

Password Reset Request
Host:attacker.com → **Response (3XX)**
Location:attacker.com (same value as host header)

Application will send the reset link to Victim's email address with the token by creating link based on "Host" header's value.
such as attacker.com/reset- password/random-token-which-is-unique

- **Once victim will click the link, they will be redirected to attacker's website (token is unused)**
- **Attacker can access the token from the server logs and can reset the password by navigate to https://www.0xnoob.com/random-token-which-is-unique**

Authentication Bypass: Breaking 2FAs/OTPs

- 2FA/OTP Code Leakage in Response
- Missing 2FA/OTP Code Integrity Validation
- 2FA Code/OTP Reusability
- Master OTPs: Try to login with master OTPs such as
  - 111111
  - 000000
  - 123456
  - 999999
  - etc

Authentication Bypass: PHP Type Juggling

- **It is a feature of PHP that allows automatic conversion of data types when performing operations or comparisons.**
- **PHP is a loosely typed language, meaning that variables do not have explicit data types declared, and PHP tries to interpret the data type based on its context.**

- **Implicit Type Conversion: (Integer)**
  - **$num1 = "10"; // String**
  - **$num2 = 5; // Integer**
  - **result = $num1 + $num2; // $num1 is converted to an integer (10) for the addition**
  - **echo $result; // Output: 15**

Authentication Bypass: PHP Type Juggling

- **PHP will automatically convert the data types of variables to make them compatible for comparison.**
- **Loose Comparisons (‘==‘):attempt to convert the operands to a common data type for comparison.**

```
$num3 = 10; // Integer
$num4 = “10abc”;      // String

if ($num3 == $num4) {
  echo "Equal";
} else {
  echo "Not Equal";
}
// Output: Equal, as $num3 is converted to an integer (10) for the comparison
```

Authentication Bypass: PHP Type Juggling

- **However, if the string being compared does not contain an integer, it will be converted to a "0". As a result, the following comparison will also evaluate to True:**

  (0 == “test”) – True

Authentication Bypass: PHP Type Juggling

How it is a vulnerability? How can we bypass Authentication?

if ($_POST["password"] == "Admin_Password") {login as admin();}

0==“Admin Password”

0==0 – True

Please note:- This vulnerability is not exploitable directly, means user should be able to control input data type:-

For example, if the application accepts the user inputs as JSON, we can convert login request to JSON (a user controllable input data type) and it will compare user’s provided Integer input (0) with a String (Actual password) and this will become a true statement.

Authentication Bypass: CSRF (Cross-Site Request Forgery)

- **Update Mobile Number/Email address**
- **Password change request where old password is not required**

Authentication Bypass: HTTP-Parameter-Pollution

The basic idea behind HTTP Parameter Pollution is that an attacker manipulates the request by sending multiple instances of the same parameter with different values, causing confusion for the application on how to handle and interpret the data. Can try this on sensitive modules such as

- **Reset Password**
- **Update Password**
- **Request OTP**

(https://www.0xnoob.com/sensitive_modules/?user_id=1&user_id=2)

Authentication Bypass:

Authentication Bypass through Response Manipulation

- **In Response if “Success”:false then change to “Success”:true**

Authentication Bypass through Status Code Manipulation

- **Status code is 4XX, change it to 200 OK and see if it bypass the restrictions**

Authentication Bypass: Brute Force

- Password Spraying (Brute Force the usernames/email addresses)
- Check if the rate limit control is implemented
- Broken brute-force protection, IP block
- Broken brute-force protection, multiple credentials per request (Check Portswigger's Web Security Academy)

Authentication Bypass: Default Credentials

- Check for the default credentials
- If WordPress is used, then navigate to /?author=1, /?author=2 as it will disclose the usernames
- Always run wp-scan with the API key, as it will show different results

Authentication Bypass: Insecure Logout Management

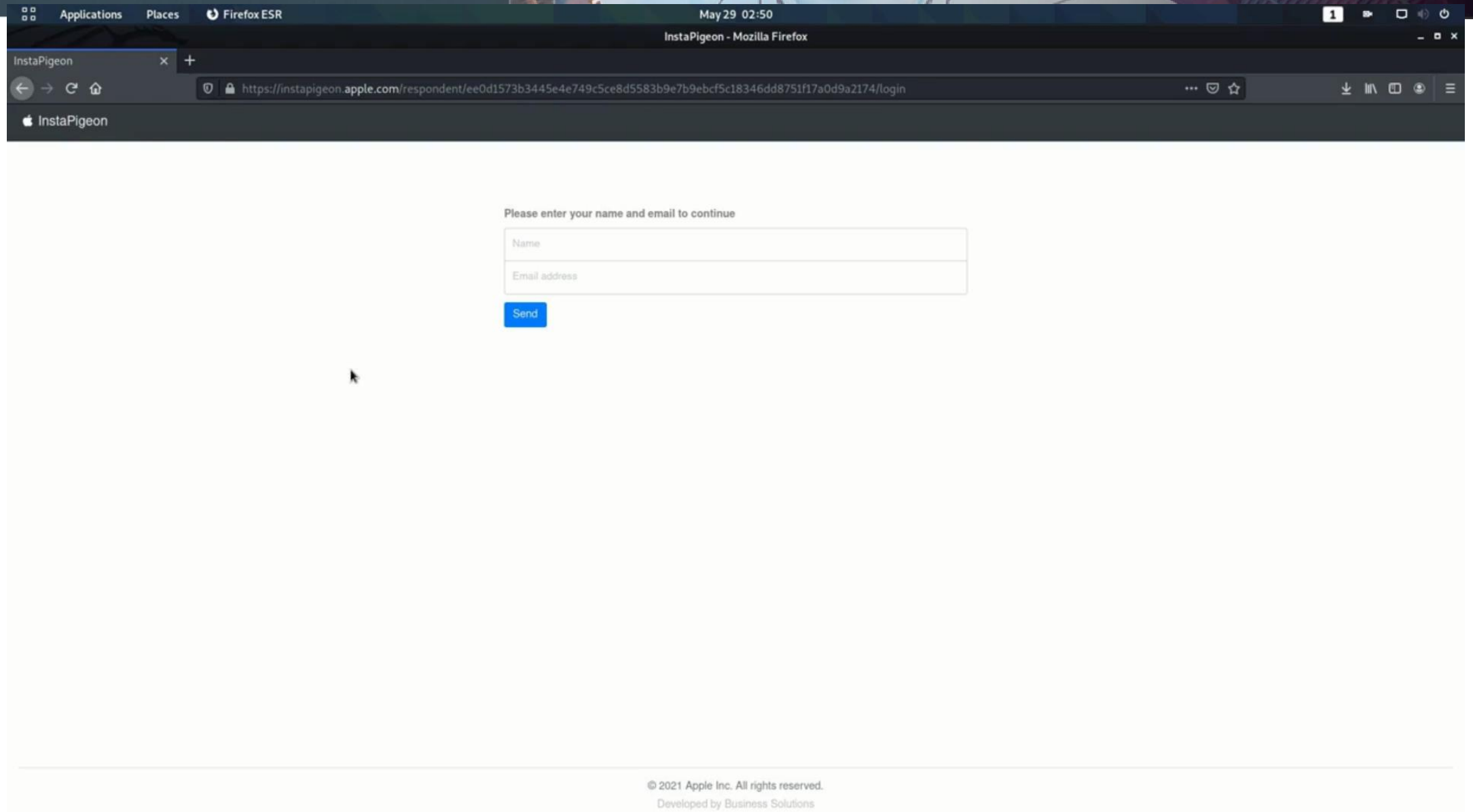
- **Specially for banking applications and other sensitive applications check below mentioned test cases**
  - **Reuse the session – Logout from the application and replay the post login requests with old session value.**
  - **Close the browser and check if you are still logged in**
  - **Check if the session is bound to IP address**

Authentication Bypass: Tips

- **Go out of the box (Red Teamers, Penetration Testers)**
  - **Pastebin (http://pastebin.com)**
  - **Dehashed (http://dehashed.com)**
  - **Open Bug Bounty**

(Free Bug Bounty Program and Coordinated Vulnerability Disclosure | Open Bug Bounty)

My First Apple Bug :)


Applications
Places
Firefox ESR
May 29 02:50
InstaPigeon - Mozilla Firefox
InstaPigeon
https://instapigeon.apple.com/respondent/ee0d1573b3445e4e749c5ce8d5583b9e7b9ebcf5c18346dd8751f17a0d9a2174/login
InstaPigeon
Please enter your name and email to continue
Name
Email address
Send
© 2021 Apple Inc. All rights reserved.
Developed by Business Solutions

Let’s Discuss!

Thank you for your time!

I hope we learned something new today


OWASP 2024
AppSec DAYS
SINGAPORE
THANK
YOU!