Phising Report of https://demo.testfire.net/

Vulnerability Name: Authentication Bypass

Description

Authentication bypass occurs when an attacker gains unauthorized access to a system without completing the intended authentication process. This typically exploits flaws in input validation, session management, or logic errors in authentication workflows.

A common example is using SQL injection to manipulate login queries. If multi-factor authentication (MFA) is not enforced, attackers can gain access using only compromised credentials.

Attack Steps

Step 1:
Searched for vulnerable login pages using Google dork:
inurl:admin login php site:demo.testfire.net

Step 2:
Accessed the admin login page of the target website.

Step 3:
Analyzed login logic:

User Id	Password	Result
True	True	True
True	False	False
False	True	False
False	False	False

Step 4:
Injected SQL payload in the login field:
`1' OR '1'='1`
This bypassed authentication and granted admin access.

System State

Before Exploit:

- Login page accessible without input validation
- No multi-factor authentication
- Database vulnerable to SQL injection

After Exploit:

- Admin access gained without valid credentials
- Sensitive data exposed
- System integrity compromised

⚠ Mistakes by Roles & Suggested Fixes

🧑‍💻 Web Developer Mistake

Issue: Input fields allow special characters, enabling SQL injection.
Suggestions:

- ✓ Use input validation (e.g., regex for alphanumeric usernames)
- ✓ Implement server-side validation
- ✓ Use prepared statements or ORM to prevent SQL injection
- ✓ Integrate security libraries (e.g., OWASP ESAPI)

🗄 Database Administrator Mistake

Issue: Database lacks encryption, exposing sensitive data.
Suggestions:

- ✓ Encrypt data at rest using AES-256
- ✓ Hash passwords with bcrypt or Argon2
- ✓ Enable Transparent Data Encryption (TDE)
- ✓ Rotate encryption keys regularly

🌐 Network Administrator Mistake

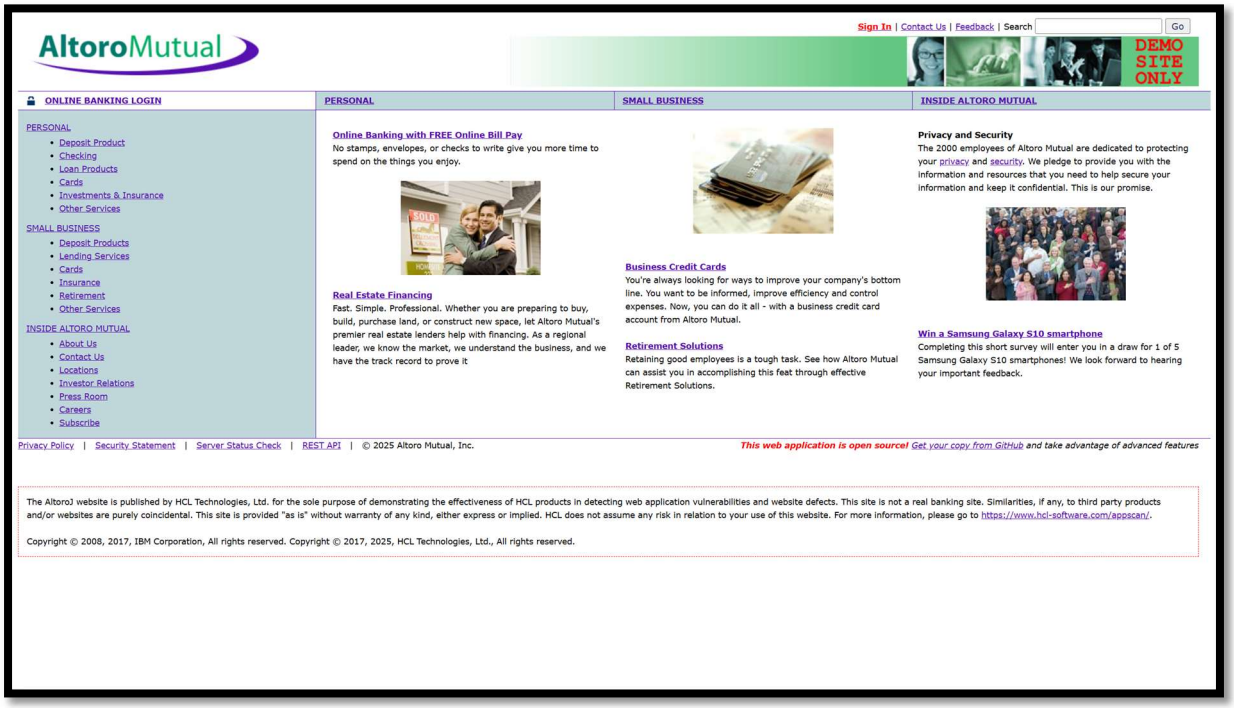
Issue: Firewall misconfiguration allows unauthorized access.
Suggestions:

- ✓ Apply least privilege rules for inbound/outbound traffic
- ✓ Block unused ports and monitor logs
- ✓ Deploy IDS/IPS systems
- ✓ Conduct regular firewall audits and penetration tests

✓ Final Recommendations

- Enforce **multi-factor authentication (MFA)**
- Conduct **regular vulnerability assessments**
- Train developers on **secure coding practices**
- Maintain **incident response plans** for breach scenarios

Before Attack:



After Attack: