Relatório Técnico – Ataque de SQL Injection

Introdução

Este relatório apresenta a análise do ataque de SQL Injection realizado em ambiente controlado, conforme tutorial de Segurança Cibernética. O objetivo foi demonstrar como uma falha no código de autenticação pode permitir que usuários mal-intencionados obtenham acesso indevido ao sistema sem possuir credenciais válidas.

Contexto da Vulnerabilidade

Durante o teste, foi disponibilizado um sistema web simples, rodando em XAMPP com banco de dados MySQL. O formulário de login possuía a seguinte consulta SQL em PHP:

```
SELECT * FROM users WHERE username='$username' AND password='md5($pass)';
```

Problema Identificado

- O código concatena diretamente as entradas do usuário na query SQL.
- Não há validação ou sanitização dos dados inseridos.
- Isso permite a inserção de comandos maliciosos diretamente na consulta ao banco de dados.

Execução do Ataque

O invasor inseriu, no campo de usuário, o seguinte payload:

```
' OR 1=1 -- //
```

Este comando torna a condição sempre verdadeira e ignora a validação de senha, permitindo login sem credenciais válidas.

Problema Identificado

Impacto	Descrição
Acesso não autorizado	Login no sistema sem credenciais válidas.
Exfiltração de dados	Possibilidade de obter informações sensíveis.
Modificação de registros	Alterar ou excluir dados armazenados.
Escalada de privilégios	Acesso a funções administrativas sem permissão.

Mitigação Aplicada

O tutorial demonstrou uma correção inicial utilizando a função addslashes nos campos de entrada. Isso impede que caracteres especiais sejam interpretados como parte da query.

Melhorias Recomendadas:

1 - Uso de consultas preparadas (Prepared Statements) com PDO ou MySQLi.
2 - Validação e sanitização de entrada (whitelists, regex, escape de caracteres).
3 - Política de privilégios mínimos no banco de dados.
4 - Monitoramento de logs para detecção de tentativas suspeitas.

Conclusão

O experimento demonstrou como falhas simples podem abrir brechas graves de segurança. O SQL Injection ainda é uma das vulnerabilidades mais exploradas, mas pode ser prevenido com boas práticas de desenvolvimento seguro.