# Natas

## 11->12

level exige que se faça upload de um jpeg, mas não checa se é mesmo um jpeg; dá pra fazer upload de um script php

- O problema é, esse script é sempre renomeado para um string com .jpg no fim; na hora de clicar no link o browser não executa o script
- MAS, a string com o nome final do arquivo está na verdade em um <input> com atributo hidden, dá pra editar e colocar extensão .php
- Com isso dá pra printar a senha em /etc/natas_webpass/natas13

## 12->13

- Parecido com o anterior, mas ele realmente checa se a imagem é um jpeg
- Ainda dá pra upar um script editando os primeiros bytes do script pra ser igual ao magic number do jpeg
- Pra adicionar bytes num arquivo, o melhor jeito é usar o bvi (binary vi)

## 13->14

envolve SQL injection; é preciso injetar código SQL (parecido com XSS) no campo username para garantir que a query vai retornar pelo menos uma linha (mysql_num_rows() > 0). O melhor jeito de fazer isso é garantir que a condição da query é sempre true. Para isso, podemos fechar as aspas, colocar OR true e # (comentário).

## 14->15

SQL injection, LIKE operator, LIKE BINARY (case-sensitive); vide script15

## 15->16

brutar cada caractere da senha, com grep e regex; vide script16

## 16->17

parecido com o 16, mas com sql (RLIKE BINARY). Usamos sleep() para saber se o caractere da senha está certo. Vide script17.py

## 17->18

como o ID máximo de usuário é 640, fizemos um javascript para atualizar o valor do cookie

PHPSESSID, para tentar todos os IDs possíveis até chegar no ID do admin. (138?)

---

Revision #6

Created Fri, Oct 26, 2018 7:38 PM by Luana

Updated Wed, Dec 12, 2018 7:54 PM by Luana