

Name: LOKESH

Date:13.03.2023

Task: 3

1.commands execution vulnerability:

Command execution vulnerabilities can arise through the use of external libraries or third-party code. If the code is not properly vetted or updated, it may contain vulnerabilities that can be exploited by an attacker.

To mitigate command execution vulnerabilities, it is important to properly validate user input and to use up-to-date and secure third-party code. Additionally, it is recommended to use sandboxing or other isolation techniques to prevent an attacker from executing arbitrary commands on the system.

Command injection is an attack in which the goal is execution of arbitrary commands on the host operating system via a vulnerable application. Command injection attacks are possible when an application passes unsafe user supplied data (forms, cookies, HTTP headers etc.) to a system shell.

i. Security level: low

The screenshot shows the DVWA Command Execution page. The title bar says "DVWA". The left sidebar has a menu with "Command Execution" highlighted in green. The main content area has a heading "Vulnerability: Command Execution" and a form titled "Ping for FREE" with a text input field and a "submit" button. Below the form, there is some red text: "help", "index.php", and "source". At the bottom of the page, there is a "More info" section with two links: "http://www.scribd.com/doc/25364768/PHP-Endangers-Remote-Code-Execution" and "http://www.safecodebase.com/".

ii. Security level: medium

DVWA

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

submit

help
index.php
source

More info

<http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
<http://www.ss64.com/bash/>
<http://www.ss64.com/nt/>

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

Security level: high

DVWA

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

submit

help
index.php
source

More info

<http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
<http://www.ss64.com/bash/>
<http://www.ss64.com/nt/>

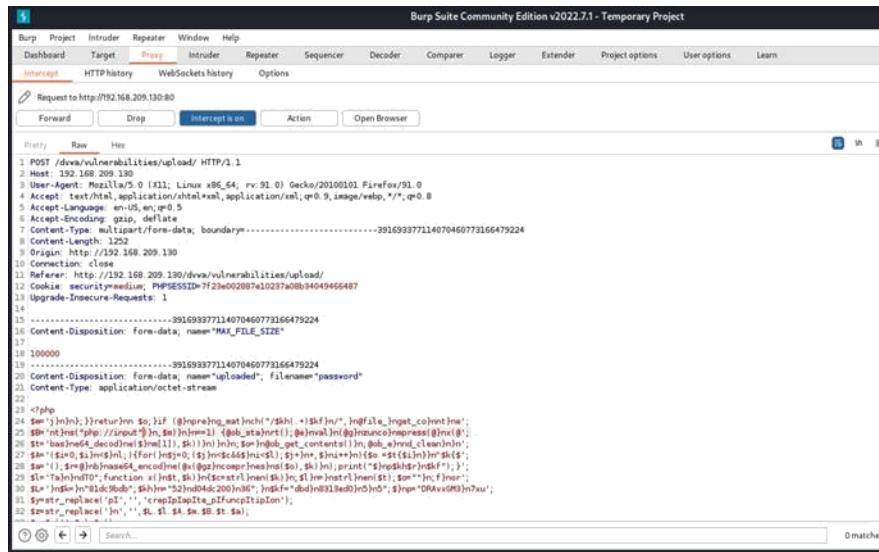
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout

2.File Upload Vulnerability:

File upload vulnerability is a security flaw that allows an attacker to upload malicious files to a web server, which can then be executed on the server or downloaded by other users. This vulnerability can be exploited by attackers to compromise the confidentiality, integrity, and availability of the web application and its data.

There are several ways in which file upload vulnerabilities can arise. One common way is through inadequate file type validation. If a web application allows users to upload files without properly verifying the file type and content, an attacker may be able to upload a malicious file disguised as a harmless file type, such as an image or a PDF.

i.Security level: low



The screenshot shows the Burp Suite interface with the 'Replay' tab selected. A POST request is displayed in the message list. The raw request content is as follows:

```
1 POST /dvs/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.209.130
3 User-Agent: Mozilla/5.0 (Linux; x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----39169337711407046073166479224
8 Content-Length: 1252
9 Origin: http://192.168.209.130
10 Connection: close
11 Referer: http://192.168.209.130/dvs/vulnerabilities/upload/
12 Cookie: security=medium; PHPSESSID=D7F2B602087E10237a0B94049466487
13 Upgrade-Insecure-Requests: 1
14
15 -----39169337711407046073166479224
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 1000000
19 -----
20 Content-Disposition: form-data; name="uploaded"; filename="password"
21 Content-Type: application/octet-stream
22
23 <php>
24 $c=$n;});return $c;if ($n[privkey_md5($k)]) {if ($k[0]==$f[0]) {
25 $p='t1est';$p1='';$p2='';$p3='';$p4='';$p5='';$p6='';$p7='';$p8='';$p9='';$p10='';$p11='';$p12='';$p13='';$p14='';$p15='';$p16='';$p17='';$p18='';$p19='';$p20='';$p21='';$p22='';$p23='';$p24='';$p25='';$p26='';$p27='';$p28='';$p29='';$p30='';$p31='';$p32='';$p33='';$p34='';$p35='';$p36='';$p37='';$p38='';$p39='';$p40='';$p41='';$p42='';$p43='';$p44='';$p45='';$p46='';$p47='';$p48='';$p49='';$p50='';$p51='';$p52='';$p53='';$p54='';$p55='';$p56='';$p57='';$p58='';$p59='';$p60='';$p61='';$p62='';$p63='';$p64='';$p65='';$p66='';$p67='';$p68='';$p69='';$p70='';$p71='';$p72='';$p73='';$p74='';$p75='';$p76='';$p77='';$p78='';$p79='';$p80='';$p81='';$p82='';$p83='';$p84='';$p85='';$p86='';$p87='';$p88='';$p89='';$p90='';$p91='';$p92='';$p93='';$p94='';$p95='';$p96='';$p97='';$p98='';$p99='';$p100='';$p101='';$p102='';$p103='';$p104='';$p105='';$p106='';$p107='';$p108='';$p109='';$p110='';$p111='';$p112='';$p113='';$p114='';$p115='';$p116='';$p117='';$p118='';$p119='';$p120='';$p121='';$p122='';$p123='';$p124='';$p125='';$p126='';$p127='';$p128='';$p129='';$p130='';$p131='';$p132='';$p133='';$p134='';$p135='';$p136='';$p137='';$p138='';$p139='';$p140='';$p141='';$p142='';$p143='';$p144='';$p145='';$p146='';$p147='';$p148='';$p149='';$p150='';$p151='';$p152='';$p153='';$p154='';$p155='';$p156='';$p157='';$p158='';$p159='';$p160='';$p161='';$p162='';$p163='';$p164='';$p165='';$p166='';$p167='';$p168='';$p169='';$p170='';$p171='';$p172='';$p173='';$p174='';$p175='';$p176='';$p177='';$p178='';$p179='';$p180='';$p181='';$p182='';$p183='';$p184='';$p185='';$p186='';$p187='';$p188='';$p189='';$p190='';$p191='';$p192='';$p193='';$p194='';$p195='';$p196='';$p197='';$p198='';$p199='';$p200='';$p201='';$p202='';$p203='';$p204='';$p205='';$p206='';$p207='';$p208='';$p209='';$p210='';$p211='';$p212='';$p213='';$p214='';$p215='';$p216='';$p217='';$p218='';$p219='';$p220='';$p221='';$p222='';$p223='';$p224='';$p225='';$p226='';$p227='';$p228='';$p229='';$p230='';$p231='';$p232='';$p233='';$p234='';$p235='';$p236='';$p237='';$p238='';$p239='';$p240='';$p241='';$p242='';$p243='';$p244='';$p245='';$p246='';$p247='';$p248='';$p249='';$p250='';$p251='';$p252='';$p253='';$p254='';$p255='';$p256='';$p257='';$p258='';$p259='';$p260='';$p261='';$p262='';$p263='';$p264='';$p265='';$p266='';$p267='';$p268='';$p269='';$p270='';$p271='';$p272='';$p273='';$p274='';$p275='';$p276='';$p277='';$p278='';$p279='';$p280='';$p281='';$p282='';$p283='';$p284='';$p285='';$p286='';$p287='';$p288='';$p289='';$p290='';$p291='';$p292='';$p293='';$p294='';$p295='';$p296='';$p297='';$p298='';$p299='';$p300='';$p301='';$p302='';$p303='';$p304='';$p305='';$p306='';$p307='';$p308='';$p309='';$p310='';$p311='';$p312='';$p313='';$p314='';$p315='';$p316='';$p317='';$p318='';$p319='';$p320='';$p321='';$p322='';$p323='';$p324='';$p325='';$p326='';$p327='';$p328='';$p329='';$p330='';$p331='';$p332='';$p333='';$p334='';$p335='';$p336='';$p337='';$p338='';$p339='';$p340='';$p341='';$p342='';$p343='';$p344='';$p345='';$p346='';$p347='';$p348='';$p349='';$p350='';$p351='';$p352='';$p353='';$p354='';$p355='';$p356='';$p357='';$p358='';$p359='';$p360='';$p361='';$p362='';$p363='';$p364='';$p365='';$p366='';$p367='';$p368='';$p369='';$p370='';$p371='';$p372='';$p373='';$p374='';$p375='';$p376='';$p377='';$p378='';$p379='';$p380='';$p381='';$p382='';$p383='';$p384='';$p385='';$p386='';$p387='';$p388='';$p389='';$p390='';$p391='';$p392='';$p393='';$p394='';$p395='';$p396='';$p397='';$p398='';$p399='';$p400='';$p401='';$p402='';$p403='';$p404='';$p405='';$p406='';$p407='';$p408='';$p409='';$p410='';$p411='';$p412='';$p413='';$p414='';$p415='';$p416='';$p417='';$p418='';$p419='';$p420='';$p421='';$p422='';$p423='';$p424='';$p425='';$p426='';$p427='';$p428='';$p429='';$p430='';$p431='';$p432='';$p433='';$p434='';$p435='';$p436='';$p437='';$p438='';$p439='';$p440='';$p441='';$p442='';$p443='';$p444='';$p445='';$p446='';$p447='';$p448='';$p449='';$p450='';$p451='';$p452='';$p453='';$p454='';$p455='';$p456='';$p457='';$p458='';$p459='';$p460='';$p461='';$p462='';$p463='';$p464='';$p465='';$p466='';$p467='';$p468='';$p469='';$p470='';$p471='';$p472='';$p473='';$p474='';$p475='';$p476='';$p477='';$p478='';$p479='';$p480='';$p481='';$p482='';$p483='';$p484='';$p485='';$p486='';$p487='';$p488='';$p489='';$p490='';$p491='';$p492='';$p493='';$p494='';$p495='';$p496='';$p497='';$p498='';$p499='';$p500='';$p501='';$p502='';$p503='';$p504='';$p505='';$p506='';$p507='';$p508='';$p509='';$p510='';$p511='';$p512='';$p513='';$p514='';$p515='';$p516='';$p517='';$p518='';$p519='';$p520='';$p521='';$p522='';$p523='';$p524='';$p525='';$p526='';$p527='';$p528='';$p529='';$p530='';$p531='';$p532='';$p533='';$p534='';$p535='';$p536='';$p537='';$p538='';$p539='';$p540='';$p541='';$p542='';$p543='';$p544='';$p545='';$p546='';$p547='';$p548='';$p549='';$p550='';$p551='';$p552='';$p553='';$p554='';$p555='';$p556='';$p557='';$p558='';$p559='';$p560='';$p561='';$p562='';$p563='';$p564='';$p565='';$p566='';$p567='';$p568='';$p569='';$p570='';$p571='';$p572='';$p573='';$p574='';$p575='';$p576='';$p577='';$p578='';$p579='';$p580='';$p581='';$p582='';$p583='';$p584='';$p585='';$p586='';$p587='';$p588='';$p589='';$p590='';$p591='';$p592='';$p593='';$p594='';$p595='';$p596='';$p597='';$p598='';$p599='';$p600='';$p601='';$p602='';$p603='';$p604='';$p605='';$p606='';$p607='';$p608='';$p609='';$p610='';$p611='';$p612='';$p613='';$p614='';$p615='';$p616='';$p617='';$p618='';$p619='';$p620='';$p621='';$p622='';$p623='';$p624='';$p625='';$p626='';$p627='';$p628='';$p629='';$p630='';$p631='';$p632='';$p633='';$p634='';$p635='';$p636='';$p637='';$p638='';$p639='';$p640='';$p641='';$p642='';$p643='';$p644='';$p645='';$p646='';$p647='';$p648='';$p649='';$p650='';$p651='';$p652='';$p653='';$p654='';$p655='';$p656='';$p657='';$p658='';$p659='';$p660='';$p661='';$p662='';$p663='';$p664='';$p665='';$p666='';$p667='';$p668='';$p669='';$p670='';$p671='';$p672='';$p673='';$p674='';$p675='';$p676='';$p677='';$p678='';$p679='';$p680='';$p681='';$p682='';$p683='';$p684='';$p685='';$p686='';$p687='';$p688='';$p689='';$p690='';$p691='';$p692='';$p693='';$p694='';$p695='';$p696='';$p697='';$p698='';$p699='';$p700='';$p701='';$p702='';$p703='';$p704='';$p705='';$p706='';$p707='';$p708='';$p709='';$p710='';$p711='';$p712='';$p713='';$p714='';$p715='';$p716='';$p717='';$p718='';$p719='';$p720='';$p721='';$p722='';$p723='';$p724='';$p725='';$p726='';$p727='';$p728='';$p729='';$p730='';$p731='';$p732='';$p733='';$p734='';$p735='';$p736='';$p737='';$p738='';$p739='';$p740='';$p741='';$p742='';$p743='';$p744='';$p745='';$p746='';$p747='';$p748='';$p749='';$p750='';$p751='';$p752='';$p753='';$p754='';$p755='';$p756='';$p757='';$p758='';$p759='';$p760='';$p761='';$p762='';$p763='';$p764='';$p765='';$p766='';$p767='';$p768='';$p769='';$p770='';$p771='';$p772='';$p773='';$p774='';$p775='';$p776='';$p777='';$p778='';$p779='';$p780='';$p781='';$p782='';$p783='';$p784='';$p785='';$p786='';$p787='';$p788='';$p789='';$p790='';$p791='';$p792='';$p793='';$p794='';$p795='';$p796='';$p797='';$p798='';$p799='';$p800='';$p801='';$p802='';$p803='';$p804='';$p805='';$p806='';$p807='';$p808='';$p809='';$p810='';$p811='';$p812='';$p813='';$p814='';$p815='';$p816='';$p817='';$p818='';$p819='';$p820='';$p821='';$p822='';$p823='';$p824='';$p825='';$p826='';$p827='';$p828='';$p829='';$p830='';$p831='';$p832='';$p833='';$p834='';$p835='';$p836='';$p837='';$p838='';$p839='';$p840='';$p841='';$p842='';$p843='';$p844='';$p845='';$p846='';$p847='';$p848='';$p849='';$p850='';$p851='';$p852='';$p853='';$p854='';$p855='';$p856='';$p857='';$p858='';$p859='';$p860='';$p861='';$p862='';$p863='';$p864='';$p865='';$p866='';$p867='';$p868='';$p869='';$p870='';$p871='';$p872='';$p873='';$p874='';$p875='';$p876='';$p877='';$p878='';$p879='';$p880='';$p881='';$p882='';$p883='';$p884='';$p885='';$p886='';$p887='';$p888='';$p889='';$p890='';$p891='';$p892='';$p893='';$p894='';$p895='';$p896='';$p897='';$p898='';$p899='';$p900='';$p901='';$p902='';$p903='';$p904='';$p905='';$p906='';$p907='';$p908='';$p909='';$p910='';$p911='';$p912='';$p913='';$p914='';$p915='';$p916='';$p917='';$p918='';$p919='';$p920='';$p921='';$p922='';$p923='';$p924='';$p925='';$p926='';$p927='';$p928='';$p929='';$p930='';$p931='';$p932='';$p933='';$p934='';$p935='';$p936='';$p937='';$p938='';$p939='';$p940='';$p941='';$p942='';$p943='';$p944='';$p945='';$p946='';$p947='';$p948='';$p949='';$p950='';$p951='';$p952='';$p953='';$p954='';$p955='';$p956='';$p957='';$p958='';$p959='';$p960='';$p961='';$p962='';$p963='';$p964='';$p965='';$p966='';$p967='';$p968='';$p969='';$p970='';$p971='';$p972='';$p973='';$p974='';$p975='';$p976='';$p977='';$p978='';$p979='';$p980='';$p981='';$p982='';$p983='';$p984='';$p985='';$p986='';$p987='';$p988='';$p989='';$p990='';$p991='';$p992='';$p993='';$p994='';$p995='';$p996='';$p997='';$p998='';$p999='';$p1000='';$p1001='';$p1002='';$p1003='';$p1004='';$p1005='';$p1006='';$p1007='';$p1008='';$p1009='';$p1010='';$p1011='';$p1012='';$p1013='';$p1014='';$p1015='';$p1016='';$p1017='';$p1018='';$p1019='';$p1020='';$p1021='';$p1022='';$p1023='';$p1024='';$p1025='';$p1026='';$p1027='';$p1028='';$p1029='';$p1030='';$p1031='';$p1032='';$p1033='';$p1034='';$p1035='';$p1036='';$p1037='';$p1038='';$p1039='';$p1040='';$p1041='';$p1042='';$p1043='';$p1044='';$p1045='';$p1046='';$p1047='';$p1048='';$p1049='';$p1050='';$p1051='';$p1052='';$p1053='';$p1054='';$p1055='';$p1056='';$p1057='';$p1058='';$p1059='';$p1060='';$p1061='';$p1062='';$p1063='';$p1064='';$p1065='';$p1066='';$p1067='';$p1068='';$p1069='';$p1070='';$p1071='';$p1072='';$p1073='';$p1074='';$p1075='';$p1076='';$p1077='';$p1078='';$p1079='';$p1080='';$p1081='';$p1082='';$p1083='';$p1084='';$p1085='';$p1086='';$p1087='';$p1088='';$p1089='';$p1090='';$p1091='';$p1092='';$p1093='';$p1094='';$p1095='';$p1096='';$p1097='';$p1098='';$p1099='';$p1100='';$p1101='';$p1102='';$p1103='';$p1104='';$p1105='';$p1106='';$p1107='';$p1108='';$p1109='';$p1110='';$p1111='';$p1112='';$p1113='';$p1114='';$p1115='';$p1116='';$p1117='';$p1118='';$p1119='';$p1120='';$p1121='';$p1122='';$p1123='';$p1124='';$p1125='';$p1126='';$p1127='';$p1128='';$p1129='';$p1130='';$p1131='';$p1132='';$p1133='';$p1134='';$p1135='';$p1136='';$p1137='';$p1138='';$p1139='';$p1140='';$p1141='';$p1142='';$p1143='';$p1144='';$p1145='';$p1146='';$p1147='';$p1148='';$p1149='';$p1150='';$p1151='';$p1152='';$p1153='';$p1154='';$p1155='';$p1156='';$p1157='';$p1158='';$p1159='';$p1160='';$p1161='';$p1162='';$p1163='';$p1164='';$p1165='';$p1166='';$p1167='';$p1168='';$p1169='';$p1170='';$p1171='';$p1172='';$p1173='';$p1174='';$p1175='';$p1176='';$p1177='';$p1178='';$p1179='';$p1180='';$p1181='';$p1182='';$p1183='';$p1184='';$p1185='';$p1186='';$p1187='';$p1188='';$p1189='';$p1190='';$p1191='';$p1192='';$p1193='';$p1194='';$p1195='';$p1196='';$p1197='';$p1198='';$p1199='';$p1200='';$p1201='';$p1202='';$p1203='';$p1204='';$p1205='';$p1206='';$p1207='';$p1208='';$p1209='';$p1210='';$p1211='';$p1212='';$p1213='';$p1214='';$p1215='';$p1216='';$p1217='';$p1218='';$p1219='';$p1220='';$p1221='';$p1222='';$p1223='';$p1224='';$p1225='';$p1226='';$p1227='';$p1228='';$p1229='';$p1230='';$p1231='';$p1232='';$p1233='';$p1234='';$p1235='';$p1236='';$p1237='';$p1238='';$p1239='';$p1240='';$p1241='';$p1242='';$p1243='';$p1244='';$p1245='';$p1246='';$p1247='';$p1248='';$p1249='';$p1250='';$p1251='';$p1252='';$p1253='';$p1254='';$p1255='';$p1256='';$p1257='';$p1258='';$p1259='';$p1260='';$p1261='';$p1262='';$p1263='';$p1264='';$p1265='';$p1266='';$p1267='';$p1268='';$p1269='';$p1270='';$p1271='';$p1272='';$p1273='';$p1274='';$p1275='';$p1276='';$p1277='';$p1278='';$p1279='';$p1280='';$p1281='';$p1282='';$p1283='';$p1284='';$p1285='';$p1286='';$p1287='';$p1288='';$p1289='';$p1290='';$p1291='';$p1292='';$p1293='';$p1294='';$p1295='';$p1296='';$p1297='';$p1298='';$p1299='';$p1300='';$p1301='';$p1302='';$p1303='';$p1304='';$p1305='';$p1306='';$p1307='';$p1308='';$p1309='';$p1310='';$p1311='';$p1312='';$p1313='';$p1314='';$p1315='';$p1316='';$p1317='';$p1318='';$p1319='';$p1320='';$p1321='';$p1322='';$p1323='';$p1324='';$p1325='';$p1326='';$p1327='';$p1328='';$p1329='';$p1330='';$p1331='';$p1332='';$p1333='';$p1334='';$p1335='';$p1336='';$p1337='';$p1338='';$p1339='';$p1340='';$p1341='';$p1342='';$p1343='';$p1344='';$p1345='';$p1346='';$p1347='';$p1348='';$p1349='';$p1350='';$p1351='';$p1352='';$p1353='';$p1354='';$p1355='';$p1356='';$p1357='';$p1358='';$p1359='';$p1360='';$p1361='';$p1362='';$p1363='';$p1364='';$p1365='';$p1366='';$p1367='';$p1368='';$p1369='';$p1370='';$p1371='';$p1372='';$p1373='';$p1374='';$p1375='';$p1376='';$p1377='';$p1378='';$p1379='';$p1380='';$p1381='';$p1382='';$p1383='';$p1384='';$p1385='';$p1386='';$p1387='';$p1388='';$p1389='';$p1390='';$p1391='';$p1392='';$p1393='';$p1394='';$p1395='';$p1396='';$p1397='';$p1398='';$p1399='';$p1400='';$p1401='';$p1402='';$p1403='';$p1404='';$p1405='';$p1406='';$p1407='';$p1408='';$p1409='';$p1410='';$p1411='';$p1412='';$p1413='';$p1414='';$p1415='';$p1416='';$p1417='';$p1418='';$p1419='';$p1420='';$p1421='';$p1422='';$p1423='';$p1424='';$p1425='';$p1426='';$p1427='';$p1428='';$p1429='';$p1430='';$p1431='';$p1432='';$p1433='';$p1434='';
```

Vulnerability: File Upload

Choose an image to upload:

No file selected.

.../.../hackable/uploads/pass successfully uploaded!

More Information

- https://www.owasp.org/index.php/Unrestricted_File_Upload
- <https://www.acunetix.com/websitetesting/upload-forms-threat/>

ii. Security level: medium



Vulnerability: File Upload

Choose an image to upload:

No file selected.

.../.../hackable/uploads/pass succesfully uploaded!

iii. Security level: high



Vulnerability: File Upload

Choose an image to upload:

No file selected.

.../.../hackable/uploads/pass successfully uploaded!

More Information

- https://www.owasp.org/index.php/Unrestricted_File_Upload
- <https://www.acunetix.com/websitedevelopment/upload-forms-threat/>

Home
Instructions
Setup / Reset DB

Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript

3.Sql Injection Vulnerability:

SQL injection vulnerability is a type of security flaw that allows an attacker to inject malicious SQL code into an application's database. This vulnerability can be exploited by attackers to compromise the confidentiality, integrity, and availability of the database and its data. Once an attacker has successfully injected malicious SQL code, they can perform various malicious actions such as accessing sensitive data, modifying or deleting existing data, or even taking complete control of the database server.

i.Security level: low

DVWA

Vulnerability: SQL Injection

User ID:

Submit

ID: %' or '0' = '0
First name: admin
Surname: admin

ID: %' or '0' = '0
First name: Gordon
Surname: Brown

ID: %' or '0' = '0
First name: Hack
Surname: Me

ID: %' or '0' = '0
First name: Pablo
Surname: Picasso

ID: %' or '0' = '0
First name: Bob
Surname: Smith

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

ii.Security level: medium

DVWA

Vulnerability: SQL Injection

User ID:

Submit

ID: %' or '0' = '0
First name: admin
Surname: admin

ID: %' or '0' = '0
First name: Gordon
Surname: Brown

ID: %' or '0' = '0
First name: Hack
Surname: Me

ID: %' or '0' = '0
First name: Pablo
Surname: Picasso

ID: %' or '0' = '0
First name: Bob
Surname: Smith

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

iii.Security level: high

DVWA

Vulnerability: SQL Injection

User ID:

Submit

ID: %' or '0' = '0
First name: admin
Surname: admin

ID: %' or '0' = '0
First name: Gordon
Surname: Brown

ID: %' or '0' = '0
First name: Hack
Surname: Me

ID: %' or '0' = '0
First name: Pablo
Surname: Picasso

ID: %' or '0' = '0
First name: Bob
Surname: Smith

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

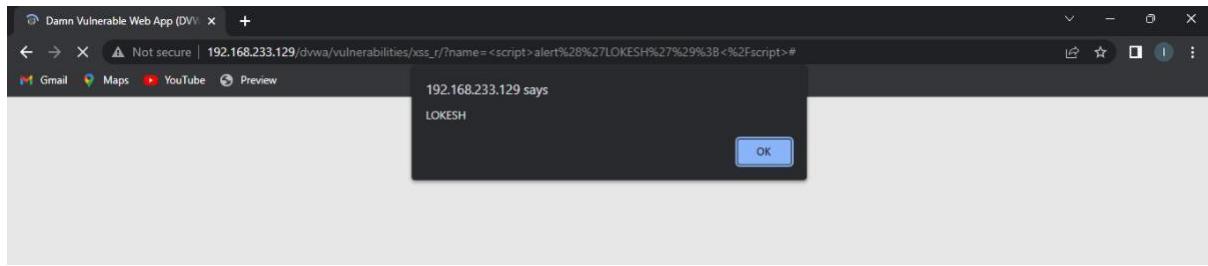
4.Cross-Site Scripting:

Cross-site scripting (XSS) is a type of web security vulnerability where an attacker is able to inject malicious code, usually in the form of scripts, into a web page viewed by other users. This can allow the attacker to steal sensitive information, such as login credentials or personal data, or to modify the content of the page in a way that can harm users.

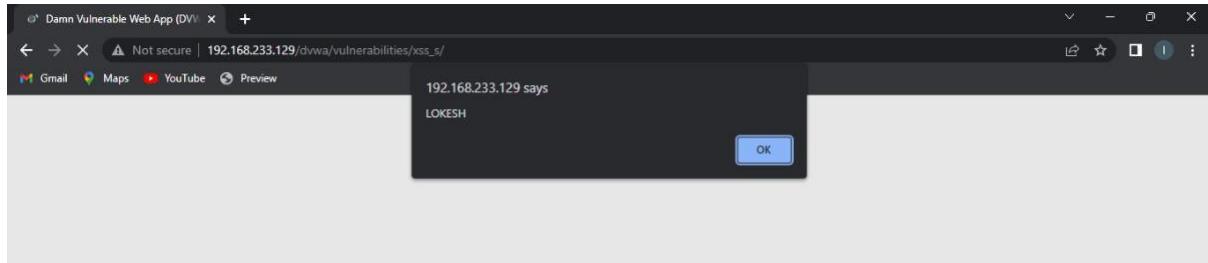
XSS attacks can occur when a web application does not properly validate or sanitize user input, such as in the case of comment boxes or search bars. An attacker can inject malicious code, such as a script that steals cookies or submits a form, into the web page by entering it as user input. This code is then executed by the browser of other users viewing the web page

i. Security level: low

XSS-reflected:

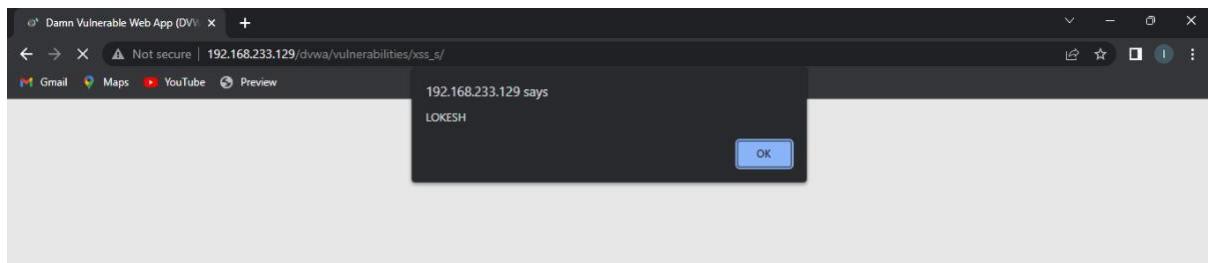


XSS-stored:

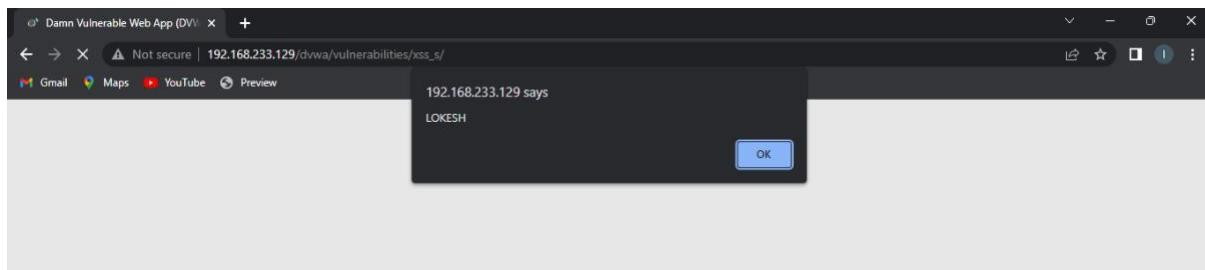


ii. Security level: medium

XSS-reflected:

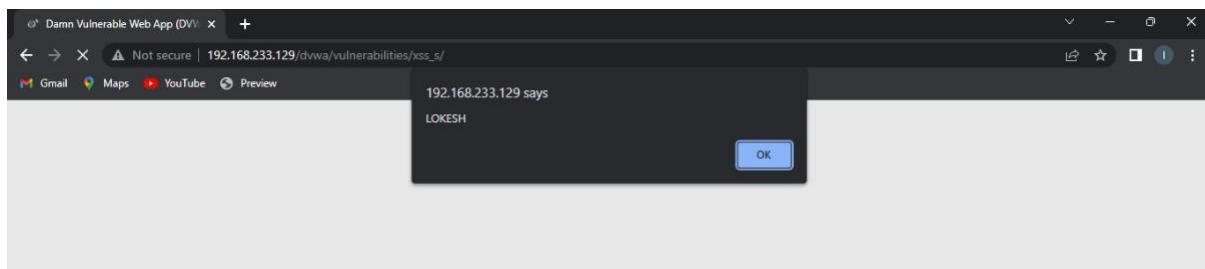


XSS-stored:

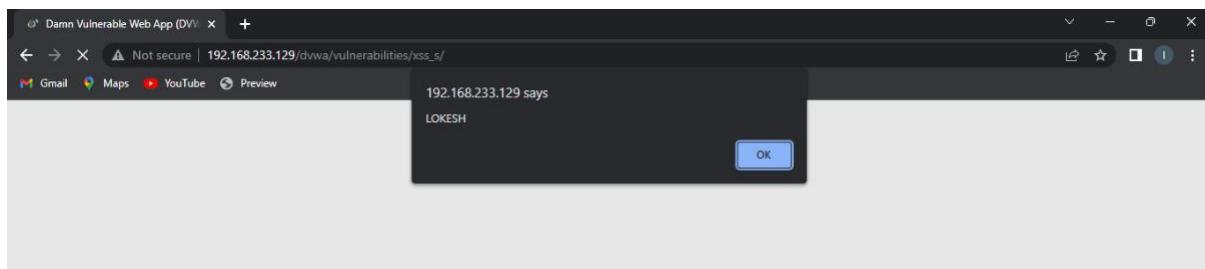


iii.Security level: high

XSS-reflected:



XSS-stored:



5.Sensitive Information Disclosure:

Information disclosure, also known as information leakage, is when a website unintentionally reveals sensitive information to its users. Depending on the context, websites may leak all kinds of information to a potential attacker.

Sensitive information disclosure is a type of security flaw that occurs when sensitive data, such as personal information or confidential business data, is exposed to unauthorized users or parties. This vulnerability can be exploited by attackers to compromise the confidentiality, integrity, and availability of the sensitive data.

i.Security level: low

File Edit View Analyse Report Tools Import Export Online Help

Standard Mode < Quick Start Request Response Requester +

Sites + Header: Text Body: Text

Contexts Default Context

HTTP/1.1 200 OK
Date: Wed, 08 Mar 2023 14:05:11 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Set-Cookie: PHPSESSID=d79fb6589e9099e0057d3d09c1784ce5; path=/
Set-Cookie: security=high
Content-Type: text/html;charset=utf-8
Content-Length: 1289

DVWA

DVWA Security

Script Security
Security Level is currently **low**.
You can set the security level to low, medium or high.
The security level changes the vulnerability level of DVWA.

PHPIDS
PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.
You can enable PHPIDS across this site for the duration of your session.
PHPIDS is currently **disabled**. [[enable PHPIDS](#)] [[Simulate attack](#)] - [[View IDS log](#)]

ii.Security level: medium

File Edit View Analyse Report Tools Import Export Online Help

Standard Mode < Quick Start Request Response Requester +

Sites + Header: Text Body: Text

Contexts Default Context

HTTP/1.1 200 OK
Date: Wed, 08 Mar 2023 14:05:11 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Set-Cookie: PHPSESSID=d79fb6589e9099e0057d3d09c1784ce5; path=/
Set-Cookie: security=high
Content-Type: text/html;charset=utf-8
Content-Length: 1289

The DVWA Security page displays a sidebar menu on the left with various attack options. The 'Script Security' section shows the security level is currently set to 'low'. A dropdown menu allows changing the security level to 'medium' or 'high'. The 'PHPIDS' section indicates PHPIDS v.0.6 is disabled. Buttons for 'Simulate attack' and 'View IDS log' are present.

iii.Security level: high

A screenshot of the Fiddler web debugger. It shows a captured request from the DVWA application. The response header includes the following information:

```
HTTP/1.1 200 OK
Date: Wed, 08 Mar 2023 14:05:11 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Set-Cookie: PHPSESSID=d79fb6589e9099e0057d3d09c1784ce5; path=/
Set-Cookie: security=high
Content-Type: text/html; charset=utf-8
Content-Length: 1289
```

The DVWA Security page displays a sidebar menu on the left with various attack options. The 'Script Security' section shows the security level is currently set to 'medium'. A dropdown menu allows changing the security level to 'low' or 'high'. The 'PHPIDS' section indicates PHPIDS v.0.6 is disabled. Buttons for 'Simulate attack' and 'View IDS log' are present.

6.Local File Inclusion: A File Inclusion Vulnerability is a type of Vulnerability commonly found in PHP based websites and it is used to affect the web applications. This issue generally occurs when an application is trying to get some information from a particular server where the inputs for getting a particular file location are not treated as a trusted source.

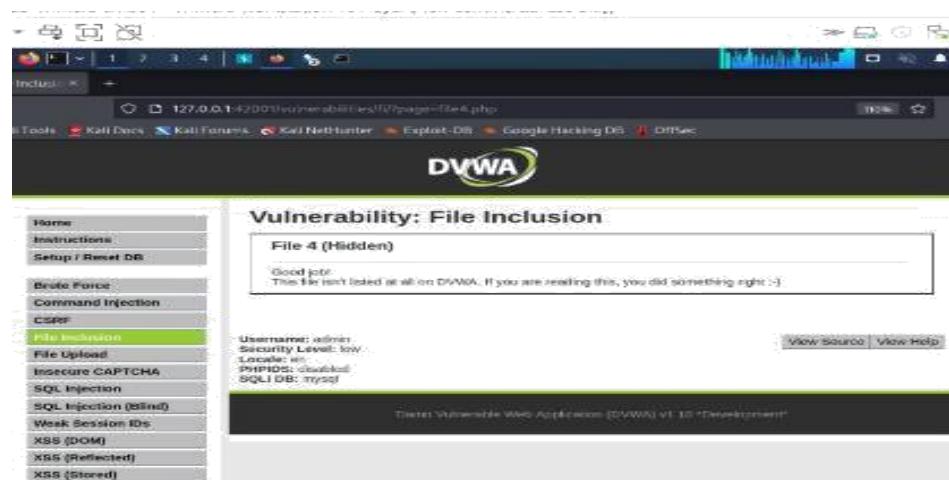
It generally refers to an inclusion attack where an attacker can supply a valid input to get a response from a web server. In response, an attacker will be able to judge whether the input which he supplied is valid or not. If it is valid, then whatever/whichever file an attacker wants to see they can easily access it.

i.Security level: low



A screenshot of the DVWA (Damn Vulnerable Web Application) interface. The URL in the browser is 127.0.0.1:8001/vulnerabilities/10/?page=file4.php. The main content area displays the message "File 4 (Hidden)" and "Good job! This file isn't listed at all on DVWA. If you are reading this, you did something right :-)." Below this, there is a table showing session details: Username: admin, Security Level: low, Locale: en, PHPIDS: disabled, SQLI DB: mysql. At the bottom, it says "Powered by Damn Vulnerable Web Application (DVWA) v1.10 'Development'". On the left sidebar, the "File Inclusion" option is highlighted in green, indicating it's the current exploit being demonstrated.

ii.Security level: medium



A screenshot of the DVWA File Inclusion (Medium) page. The URL is identical to the low-level version: 127.0.0.1:8001/vulnerabilities/10/?page=file4.php. The content and session details are the same as the low-level version. The left sidebar shows the "File Inclusion" option is also highlighted in green.

iii.Security level: high

A screenshot of a web browser showing the DVWA (Damn Vulnerable Web Application) interface. The URL is 127.0.0.1:8001/vulnerabilities/10/page=File6.php. The page title is "Vulnerability: File Inclusion". A message box says "Good job! This file isn't listed at all on DVWA. If you are reading this, you did something right :-)." Below it, a table shows session information: Username: admin, Security Level: low, Locale: en, PHPIDS: disabled, SQL DB: mysql. At the bottom, a footer bar reads "Damn Vulnerable Web Application (DVWA) v1.10 'Development'".

7.Remote File Inclusion:

Remote File Inclusion (RFI) is a type of security vulnerability that occurs when an application allows an attacker to include and execute remote files on the server. The vulnerability typically arises from insufficient input validation or poor coding practices in the application.

RFI typically occurs when a web application allows user input to control the path or URL of a file that is included or executed on the server. An attacker can exploit this vulnerability by manipulating the input to point to a remote file that they control, which can be used to inject malicious code onto the server.

i.Security level: low

A screenshot of a web browser showing the DVWA (Damn Vulnerable Web Application) interface. The URL is 127.0.0.1:8001/vulnerabilities/10/page=File6.php. The page title is "Vulnerability: File Inclusion". A message box says "Good job! This file isn't listed at all on DVWA. If you are reading this, you did something right :-)." Below it, a table shows session information: Username: admin, Security Level: low, Locale: en, PHPIDS: disabled, SQL DB: mysql. At the bottom, a footer bar reads "Damn Vulnerable Web Application (DVWA) v1.10 'Development'".

ii.Security level: medium

The screenshot shows a browser window with the URL `127.0.0.1:4200/vulnerabilities/file-inclusion/page-file-4.php`. The DVWA logo is at the top. The main content area is titled "Vulnerability: File Inclusion" and displays "File 4 (Hidden)". A message box says "Good job! This file isn't listed at all on DVWA. If you are reading this, you did something right :-)".

The left sidebar menu is visible, with "File Inclusion" highlighted. Below the message box, there is a "View Source" and "View Help" link. At the bottom, it says "Dvwa Vulnerable Web Application (DVWA) v1.10 'Development'".

iii.Security level: high

This screenshot is identical to the one above, showing the DVWA File Inclusion page with a successful exploit message. The URL is `127.0.0.1:4200/vulnerabilities/file-inclusion/page-file-4.php`.

8.Bruteforce Attack:

A brute force attack is a type of cyber attack in which an attacker tries to guess a password or encryption key by systematically trying every possible combination until the correct one is found. This type of attack can be used to gain unauthorized access to a system, steal sensitive data, or launch further attacks.

Brute force attacks are typically automated and use specialized software or scripts to generate and try large numbers of password or key combinations. The attack can take a long time, depending on the length and complexity of the password or key being targeted, and the computing power available .

i.Security level: low

```

1 GET /dvs/vulnerabilities/brute/?username=lokesh&password=password&Login=Login HTTP/1.1
2 Host: 192.168.239.129
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Pragma: no-cache
11 Cache-Control: no-cache
12
13

```

Vulnerability: Brute Force

Login

Username: lokesh
Password: *****
Login

Welcome to the password protected area admin

ii. Security level: medium

```

1 GET /dvs/vulnerabilities/brute/?username=lokesh&password=password&Login=Login HTTP/1.1
2 Host: 192.168.239.129
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Pragma: no-cache
11 Cache-Control: no-cache
12
13

```

iii. Security level: high

The Burp Suite interface shows a captured request:

```

1 GET /dwa/vulnerabilities/brute/?username=lokesh&password=password&Login=Login HTTP/1.1
2 Host: 192.168.239.129
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Upgrade-Insecure-Requests: 1
L0 Pragma: no-cache
L1 Cache-Control: no-cache
L2
L3

```

9. forced browsing vulnerability:

Forced browsing, also known as directory traversal, is a type of security vulnerability that occurs when an attacker is able to access files and directories on a web server that are not intended to be publicly accessible. This vulnerability can be exploited by attackers to gain unauthorized access to sensitive information or to launch further attacks.

To mitigate forced browsing vulnerabilities, it is important to implement proper access controls and to validate user input. Web servers should be configured to restrict

access to sensitive files and directories and to use secure file permissions. Input validation and sanitization can help prevent attacks that attempt to modify URLs or inject malicious code. Web application firewalls can also be used to detect and block forced browsing attacks.

10.components with known vulnerability:

Using Components with Known Vulnerabilities According to OWASP: Using Components with Known Vulnerabilities Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate severe data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine the app.

These attacks have become commonplace because it is far easier for an attacker to use a known weakness than create a specific program or attack methodology to search out vulnerabilities themselves. This fact should put known component vulnerabilities high on your security priority list to mitigate.

i.Security level: Low:

The screenshot shows the CVE Details website interface. At the top, there's a navigation bar with links for Log In, Register, and a third-party risk management course. A search bar at the top right contains placeholder text '(e.g.: CVE-2009-1234 or 2010-1234 or 20101234)'. Below the search bar are buttons for 'Search' and 'View CVE'. The main content area displays a vulnerability detail page for CVE-2016-4975. The title 'Vulnerability Details : CVE-2016-4975' is followed by a brief description: 'Possible CRLF injection allowing HTTP response splitting attacks for sites which use mod_userdir. This issue was mitigated by changes made in 2.4.25 and 2.2.32 which prohibit CR or LF injection into the "Location" or other outbound header key or value. Fixed in Apache HTTP Server 2.4.25 (Affected 2.4.1-2.4.23). Fixed in Apache HTTP Server 2.2.32 (Affected 2.2.0-2.2.31).' Below this is a note about publication and update dates. The page includes sections for CVSS Scores & Vulnerability Types, showing a score of 4.3, and Related OVAL Definitions. On the left sidebar, there are links for various search and browse functions like Home, Vendors, Products, and CVSS Score Distribution.

ii.Security level: Medium:

CVE Details
The ultimate security vulnerability datasource

Log In Register Take a third party risk management course for FREE

Switch to https:// Home

Browse :
[Vendors](#)
[Products](#)
[Vulnerabilities By Date](#)
[Vulnerabilities By Type](#)

Reports :
[CVSS Score Report](#)
[CVSS Score Distribution](#)

Search :
[Vendor Search](#)
[Product Search](#)
[Version Search](#)
[Vulnerability Search](#)
[By Microsoft References](#)

Top 50 :
[Vendors](#)
[Vendor Cvss Scores](#)
[Products](#)
[Product Cvss Scores](#)
[Versions](#)

Other :
[Microsoft Bulletins](#)
[Bugtraq Entries](#)
[CWE Definitions](#)
[About & Contact](#)

Vulnerability Details : [CVE-2016-4975](#)

Possible CRLF injection allowing HTTP response splitting attacks for sites which use mod_userdir. This issue was mitigated by changes made in 2.4.25 and 2.2.32 which prohibit CR or LF injection into the "Location" or other outbound header key or value. Fixed in Apache HTTP Server 2.4.25 (Affected 2.4.1-2.4.23). Fixed in Apache HTTP Server 2.2.32 (Affected 2.2.0-2.2.31).

Published Date : 2018-08-14 Last Update Date : 2021-06-06

Collapse All Expand All Select Select&Copy Scroll To Comments External Links

Search Twitter Search YouTube Search Google

- CVSS Scores & Vulnerability Types

CVSS Score	4.3
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Http response splitting
CWE ID	93

- Related OVAL Definitions

iii.Security level: High

CVE Details
The ultimate security vulnerability datasource

Log In Register Take a third party risk management course for FREE

Switch to https:// Home

Browse :
[Vendors](#)
[Products](#)
[Vulnerabilities By Date](#)
[Vulnerabilities By Type](#)

Reports :
[CVSS Score Report](#)
[CVSS Score Distribution](#)

Search :
[Vendor Search](#)
[Product Search](#)
[Version Search](#)
[Vulnerability Search](#)
[By Microsoft References](#)

Top 50 :
[Vendors](#)
[Vendor Cvss Scores](#)
[Products](#)
[Product Cvss Scores](#)
[Versions](#)

Other :
[Microsoft Bulletins](#)
[Bugtraq Entries](#)
[CWE Definitions](#)
[About & Contact](#)

Vulnerability Details : [CVE-2016-4975](#)

Possible CRLF injection allowing HTTP response splitting attacks for sites which use mod_userdir. This issue was mitigated by changes made in 2.4.25 and 2.2.32 which prohibit CR or LF injection into the "Location" or other outbound header key or value. Fixed in Apache HTTP Server 2.4.25 (Affected 2.4.1-2.4.23). Fixed in Apache HTTP Server 2.2.32 (Affected 2.2.0-2.2.31).

Published Date : 2018-08-14 Last Update Date : 2021-06-06

Collapse All Expand All Select Select&Copy Scroll To Comments External Links

Search Twitter Search YouTube Search Google

- CVSS Scores & Vulnerability Types

CVSS Score	4.3
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Http response splitting
CWE ID	93

- Related OVAL Definitions

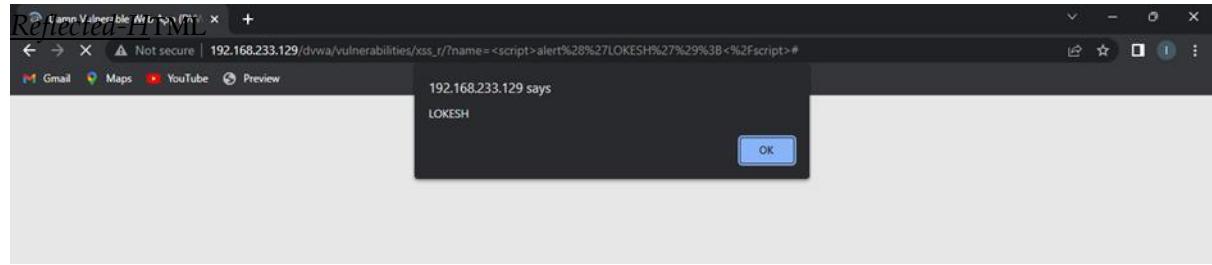
Security level: high

11.HTML injection:

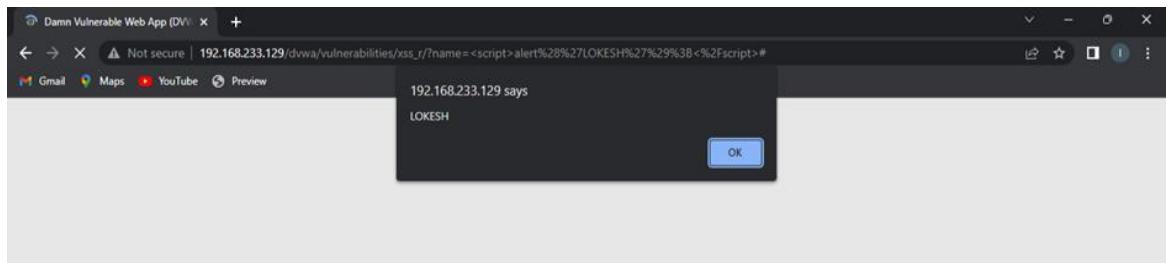
HTML injection vulnerabilities typically arise from insufficient input validation or sanitization in a web application. Attackers can inject malicious code into input fields, such as login forms, comment sections, or search fields, and the code is then displayed to other users who visit the page. When the code is executed in the browser of the victim, it can steal cookies, session tokens, or other sensitive information, or redirect the victim to a malicious website.

Developers should also avoid using unsanitized user input in HTML output, and use frameworks that provide built-in security features, such as template engines that automatically sanitize user input. It is also important to provide security awareness training to users to help them identify and avoid phishing attacks and other forms of social engineering.

i.Security level: Low:

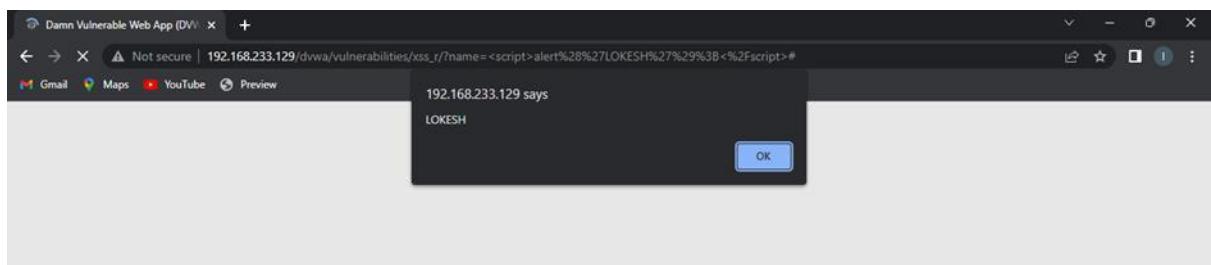


Stored-HTML:

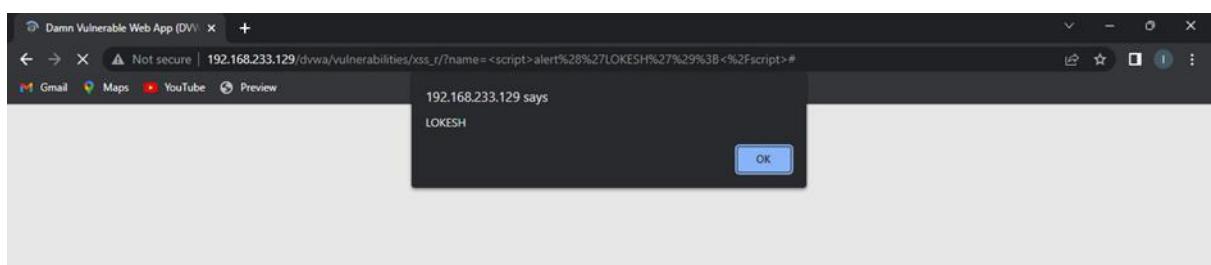


ii.Security level: Medium:

Reflected-HTML

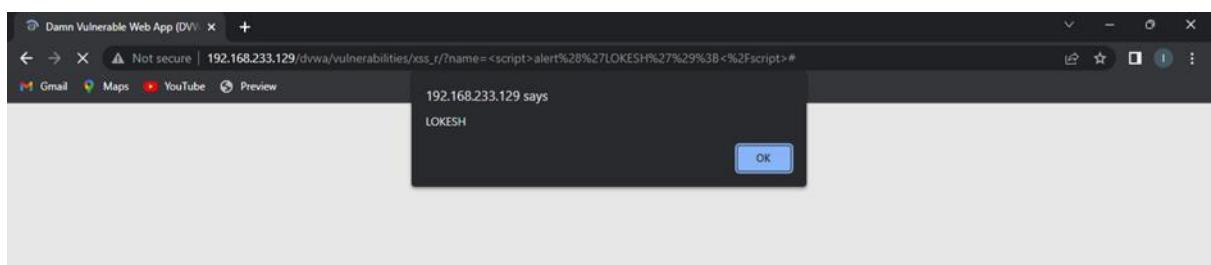


Stored-HTML:



iii.Security level: High

Reflected-HTML



Stored-HTML:

