Security Scan Report

Executive Summary

This report summarizes a security scan of the GitHub repository: https://github.com/digininja/DVWA.git. We found 74 potential security issues. Breakdown: 6 Critical, 68 High, 0 Medium, 0 Low, 0 Informational issues.

Overall Risk Assessment

Risk Score: 81.6/100 (Critical Risk)

What We Found

Critical Severity Issues (6)

File	Line	Issue	Context
vulnerabilities/view_hel	p 20 np	Use of eval() - Potential code in	jæxtæin/'ñsk.file_get_contents(DVWA_WEB_PAGE_TO_R
vulnerabilities/view_hel	p 2 22hp	Use of eval() - Potential code in	jeextatin/i'n/isk. file_get_contents(DVWA_WEB_PAGE_TO_R
dvwa/js/dvwaPage.js	7	Use of eval() - Potential code in	je/detioanl(l'ijsstage" + id + " = window.open(URL, '" + id
vulnerabilities/javascrip	t/ 8 3urce	/Hilghe_ onnerbál(\$ ca Redejs tial code in	jevaticoryjato = eval("require('crypto')");
vulnerabilities/javascrip	t/ 84 urce	/H ilshe_onnerbál(\$ ca Redejs tial code ir	jexaticBufitskr = eval("require('buffer').Buffer");
vulnerabilities/javascrip	t/ \$ ource	/Hilstejef eval() - Potential code ir	j e⁄atiœ⊫{lsto mCharCode','toStri ng','replace','BeJ','

High Severity Issues (68)

File	Line	Issue	Context
login.php	39	Potential hardcoded credential	\$query = "SELECT * FROM `users` WHERE user='\$user
dvwa/includes/dvwaPa	g d.018 c.pl	ոխnfiltered POST parameter - In	je i:/(ærrarj<u>sk</u>key_exists ("Login", \$_POST) && \$_POST['

security.php	13	Unfiltered POST parameter - In	jeit(tii sset ¢k
			<pre>\$_POST['seclev_submit'])) {</pre>
external/recaptcha/reca	ap 4 dhalib	. pP optential hardcoded credential	<pre> class='g-recaptcha' data-theme='dark'</pre>
vulnerabilities/brute/sou	ır 3 e/imp	o ssnibiltepச் p POST parameter - In	jeft(tiissert&\$_POST['Login']) && isset (\$_POST['u
vulnerabilities/brute/sou	ır ¢ ᢓ/low	p P ptential hardcoded credential	\$query = "SELECT * FROM `users` WHERE user = '\$us
vulnerabilities/brute/sou	ır ¢⊖ /higł	n. Phop ential hardcoded credential	\$query = "SELECT * FROM `users` WHERE user = '\$us
vulnerabilities/sqli/test.	oh4p	Potential hardcoded credential	\$password = "password";
vulnerabilities/brute/sou	ır ¢é /med	di Protent ial hardcoded credential	\$query = "SELECT * FROM `users` WHERE user = '\$us
vulnerabilities/cryptogra	ap h y/sou	r ee/tecriti<u>a</u>athackioorop ed credential	\$key = "ik ben een aardbei";
vulnerabilities/cryptogra	ap40y/sou	r@e/tekeal_hibrdoodenpossibletjah	p \$token = "userid:2";
vulnerabilities/cryptogra	ap118y/sou	ır €e/tæn<u>ti</u>athehanyd.co oped credential	\$key = "wachtwoord";
vulnerabilities/cryptogra	ap2n7y/sou	r@e/tekeal_hiardoo_deighcpeopential	\$token = "userid:2";
vulnerabilities/cryptogra	ap f6 y/sou	rec/tentiahpardcoded credential	\$key = "wachtwoord";
vulnerabilities/cryptogra	ap219y/sou	ır icle ∕i fötve,nekdp POST parameter - In	jethtingssinge = \$_POST['message'];
vulnerabilities/cryptogra	ap3k0y/sou	Ir lde/fölte:neldp POST parameter - In	jeb (anraj<u>sk</u>key_exists ('direction', \$_POST) && \$_PO
vulnerabilities/cryptogra	ap389y/sou	ır ı≿⊮/fötvenelo pPOST parameter - In	jespiasswiskrd = \$_POST['password'];
vulnerabilities/open_red	dißect/so	u lde/inherdidr©.គ្ lhparameter - Inje	ctfo(arrissk_key_exists ("redirect", \$_GET) && \$_GET[
vulnerabilities/open_red	di # ect/so	u tde/interdidr6.គ្ កែparameter - Inje	c ti⊲pres<u>k</u> match ("/http:\V https:\V/i", \$_GET['
vulnerabilities/open_red	di rle lct/so	u tde/interdidr6.Fin parameter - Inje	ctieadæk("location: " . \$_GET['redirect']);
vulnerabilities/open_red	dißect/so	u lde/iltiglediֆ ET parameter - Inje	ctfoerrisk_key_exists ("redirect", \$_GET) && \$_GET[
vulnerabilities/open_red	di r ect/so	u tde/litighephြ ET parameter - Inje	c tio etrixts(\$_GET['redirect'], "info.php") !== fals
vulnerabilities/open_red	di c ect/so	u tde/ittighephֆ ET parameter - Inje	ctieadisk("location: " . \$_GET['redirect']);

vulnerabilities/open_red	li <i>fle</i> ct/sou tde/ih é	orephடுET parameter - Inje	c tfo(arrissk_ key_exists ("id", \$_GET) && is_numeric(\$
vulnerabilities/open_red	linte&ct/soutden/inte	ரஷ்ந் ET parameter - Inje	csiowitorhsikintval (\$_GET['id'])) {
vulnerabilities/open_red	lißect/sou tdefilbe	vrepth டு ET parameter - Inje	c tio (ar ris) k_key_exists ("redirect", \$_GET) && \$_GET[
vulnerabilities/open_red	li r ect/sou tde/ilbe	wr epth⊚ ET parameter - Inje	ctieadໝk("location: " . \$_GET['redirect']);
vulnerabilities/open_red	litect/soutdefilte	p ed s௸ ௱ .p pa rameter - Inje	c tioerrisk _key_exists ("redirect", \$_GET) && is_num
vulnerabilities/open_red	li <i>t</i> ect/sou tde/ilte	р еds&⊞.риър rameter - Inje	c siwitofs kintval (\$_GET['redirect'])) {
vulnerabilities/authbypa	s 43 auth by Diasst	j s nnerHTML manipulation	-ckssirislerHTML = user['user_id'] + ' <input aardbei";<="" ben="" button"="" een="" id="surname</td></tr><tr><td>vulnerabilities/authbypa</td><td>s&Pauthby Ppiasst</td><td>jsnnerHTML manipulation</td><td>-c%\$\$\$irislerHTML = '<input
type=" ik="" td="" text"="" type="</td></tr><tr><td>vulnerabilities/authbypa</td><td>s45auth b/Длаsst</td><td>jsnnerHTML manipulation</td><td>-cessirialerHTML = '<input
type=" value="Upd</td></tr><tr><td>vulnerabilities/cryptogra</td><td>ph0y/sour@e/tee</td><td>tilialırlmapıld poded credential</td><td>\$key = "/>
vulnerabilities/cryptogra	p2nly/sourtcen/filte	de.dr Pውፍ T parameter - In	je\$ttiokærnisk\$_POST['token'];
vulnerabilities/captcha/	sduirce/medhititte	р еф POST parameter - In	je <u>tti</u> B ର83 Қ'g-recaptcha-respon se']
vulnerabilities/captcha/	s@&rce/rn&rditæn:	tipal φ ardcoded credential	\$insert = "UPDATE `users` SET password = '\$pass_ne
vulnerabilities/captcha/	solutrce/higbhapittp	red POST parameter - In	je <u>tti</u> B ର83 Ķ'g-recaptcha-respon se']
vulnerabilities/captcha/	o o o o o o o o o o o o o o o o o o o	tial hardcoded credential	\$insert = "UPDATE `users` SET password = '\$pass_ne
vulnerabilities/captcha/	sduirce/low/lpfilipe	red POST parameter - In	j e<u>ti</u>lBΩ83 l≰'g-recaptcha-respon se']
vulnerabilities/captcha/	son ce/low?.ptepo	tial hardcoded credential	\$insert = "UPDATE `users` SET password = '\$pass_ne
vulnerabilities/captcha/	s 20.97 ce/imploosistile	νίεφΡ ΩST parameter - In	j લ⊵<u>ti</u>lθΩ83l kˈg-recaptcha-respon se']
vulnerabilities/csrf/sour	cel/mediumPoptep	tial hardcoded credential	\$insert = "UPDATE `users` SET password = '\$pass_ne
			o⊏ i password = *pass_ne

vulnerabilities/csrf/test_	_c2edent	a Ropen tial hardcoded credential	\$query = "SELECT * FROM `users` WHERE user='\$user
vulnerabilities/csrf/sour	c el/3 nigh.∣	ol∰otential hardcoded credential	\$insert = "UPDATE `users` SET password = '" . \$pas
vulnerabilities/xss_d/so	u 5 ce/me	d lumfil teme d GET parameter - Inje	c \$defask = \$_GET['default'];
vulnerabilities/xss_d/so	u r ce/hig	հ ւխիւ iitered GET parameter - Inje	ctsiowitofis (<\$_GET['default']) {
vulnerabilities/csp/sour	c e ∕high.j	sDirect innerHTML manipulation	-d 669misk t.getElementById("a nswer").innerHTML = obj[
vulnerabilities/csp/sour	c e ⁄jsonp	.խՍարfiltered GET parameter - Inje	c \$oalltisk ck = \$_GET['callback'];
vulnerabilities/csrf/sour	cel/6ow.p	h p otential hardcoded credential	\$insert = "UPDATE `users` SET password = '\$pass_ne
vulnerabilities/csp/sour	cel/#nedio	u rblipfiriþ ered POST parameter - In	jeic(iissets(\$_POST['include'])) {
vulnerabilities/csp/sour	cel/6nedi	urbl ıpfiriț ered POST parameter - In	jeˈcti� <u>n</u> P@6T[ˈincludeˈ] . "
vulnerabilities/csp/sour	cel/Bow.p	hխnfiltered POST parameter - In	jeic(iissets(\$_POST['include'])) {
vulnerabilities/csp/sour	cel/5ow.p	h p Infiltered POST parameter - In	je csicniptisk c='" . \$_POST['include'] . "'>
vulnerabilities/csp/sour	c e /impos	sti lbiléiteire d POST parameter - In	jei ct(iisset s(\$_POST['include'])) {
vulnerabilities/csp/sour	cel/ilmpos	sti lbiléiteire d POST parameter - In	jeˈcti\$ <u>n</u> Pi9IST['include'] . "
vulnerabilities/csp/sour	c e ⁄high. _l	րկփոfiltered POST parameter - In	jeict(iissets(\$_POST['include'])) {
vulnerabilities/csp/sour	cel/bigh. _l	րկարfiltered POST parameter - In	je'cti\$ <u>n</u> R'9 ST['include'] . "
vulnerabilities/csp/sour	c e /impos	s யிர்e ் s innerHTML manipulation	-d ‰smisk t.getElementById("a nswer").innerHTML = obj[
vulnerabilities/api/src/L	o@i®iCon	tr blit∉iltehe d POST parameter - In	jeso liitio hrisk (\$_POST['grant_type']) {
vulnerabilities/api/src/L	o@i®iCon	tr blit∉iltehe d POST parameter - In	j etuiserniak ne = \$_POST['username'];
vulnerabilities/api/src/L	o ⊗ iAiCon	tr blit∉iltehe d POST parameter - In	j ூற்கைங்கா d = \$_POST[ˈpasswordˈ];
vulnerabilities/api/src/L	o gifh© on	tr blit∉ilteh ed POST parameter - In	j ¢ឯដែរ resisk token = \$_POST['refresh_token'];
vulnerabilities/api/src/L	ogiØ.php	Potential hardcoded credential	private const ACCESS_TOKEN_SECRET = "12345";
vulnerabilities/api/src/L	ogia.php	Potential hardcoded credential	private const REFRESH_TO KEN_SECRET = "98765";
vulnerabilities/api/src/T	okle⊕n.ph	p Potential hardcoded credential	private const ENCRYPTION_KEY = "Paintbrush";

vulne	rabilities/javascrip	t/ ß77 dex.p	h ø nfiltered POST parameter - In	je\$pilonasisk⊨ \$_POST['phrase'];
vulne	rabilities/javascrip	t/ 363 dex.p	hønfiltered POST parameter - In	je\$ttioknermisk\$_POST['token'];

What You Can Do

- Avoid using risky functions like 'eval()' that can run harmful code.
- Keep passwords and secrets out of your code—use secure storage instead.
- Run this scan regularly to catch issues early.
- Ask your team to follow safe coding habits.

Note: This tool is very thorough, but it's always a good idea to have an expert double-check important systems.