W6
Security Testing
Wednesday, May 2nd, 2018
11:30 AM

Automated Security Scanning for Your Delivery Pipeline

Presented by:

Matthew Grasberger

Coveros, Inc.

Brought to you by:

350 Corporate Way, Suite 400, Orange Park, FL 32073
888-268-8770 ·· 904-278-0524 - info@techwell.com - http://www.stareast.techwell.com/

Matthew Grasberger

Coveros, Inc.

Matthew Grasberger is an associate consultant at Coveros with a specialty in DevOps engineering, test automation, and security automation. He works with clients to build and develop robust test automation suites that are integrated into DevOps pipelines based on industry leading practices. In addition, Matt has leveraged open source security tools like OWASP ZAP to automate routine security scans, adding value in the form of software quality and application security.

Agility. Security. Delivered.

Automated Security Scanning for Your Delivery Pipeline

Matthew Grasberger

About Coveros

- Coveros builds security-critical applications using agile methods.
- Coveros Services
  - Agile transformations
  - Agile development and testing
  - DevOps and continuous integration
  - Application security analysis
- Agile & Security training
- Government qualifications
  - DCAA approved rates and accounting
  - TS facility clearance

Areas of Expertise

coveros

Select Clients

RSA
The Security Division of EMC

BNP PARIBAS

UNOS DONATE LIFE

IFC International Finance Corporation
World Bank Group

National Cancer Institute

Department of the Air Force · United States of America

Liberty Mutual. INSURANCE

Symantec.

ADVENT®

DELTA DENTAL®

UnitedHealth Group®

EMPLOYERS®

United States Army

Aol.

EMC²

command information

Forge.mil

U.S. Department of Homeland Security

Department of Health & Human Services · USA

About Me

- Matthew Grasberger is a Software Consultant at Coveros with a specialty in DevOps Engineering, Test Automation, and Security Automation. Matt has worked with clients to build and develop robust test automation suites that are integrated into DevOps pipelines based on industry leading practices. In addition, Matt has leveraged open source security tools like OWASP ZAP to automate routine security scans, adding value in the form of software quality and application security.

Security Testing

- Often approached late in the dev cycle
- Issues are not addressed
- Needs ROI: lower cost, better results, or both

Security Testing

- Explore open-source and free tools
- Opportunities in Continuous Delivery and Cloud

Identify low effort opportunities using free and open-source tools

Sqlmap

- Open source penetration tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers
- Python
- Command Line tool

How to Use Sqlmap

- Many command line options →
- **Most notable options**
  - -u URL, --url=URL
    Target URL
    (e.g."http://www.site.com/vuln.php?id=1")
  - --batch Never ask for user input, use the default behaviour
  - -r REQUESTFILE Load HTTP request from a file

```
Usage: python sqlmap.py [options]

Options:
  -h, --help            Show basic help message and exit
  -hh                   Show advanced help message and exit
  --version             Show program's version number and exit
  -v VERBOSE            Verbosity level: 0-6 (default 1)

  Target:
    At least one of these options has to be provided to define the
    target(s)

    -d DIRECT           Connection string for direct database connection
    -u URL, --url=URL   Target URL (e.g. "http://www.site.com/vuln.php?id=1")
    -l LOGFILE          Parse target(s) from Burp or WebScarab proxy log file
    -x SITEMAPURL       Parse target(s) from remote sitemap(.xml) file
    -m BULKFILE         Scan multiple targets given in a textual file
    -r REQUESTFILE      Load HTTP request from a file
    -g GOOGLEDORK       Process Google dork results as target URLs
    -c CONFIGFILE       Load options from a configuration INI file

  Request:
    These options can be used to specify how to connect to the target URL

    --method=METHOD     Force usage of given HTTP method (e.g. PUT)
    --data=DATA         Data string to be sent through POST
    --param-del=PARA..  Character used for splitting parameter values
    --cookie=COOKIE     HTTP Cookie header value
    --cookie-del=COO..  Character used for splitting cookie values
    --load-cookies=L..  File containing cookies in Netscape/wget format
    --drop-set-cookie   Ignore Set-Cookie header from response
    --user-agent=AGENT  HTTP User-Agent header value
    --random-agent      Use randomly selected HTTP User-Agent header value
    --host=HOST         HTTP Host header value
    --referer=REFERER   HTTP Referer header value
    -H HEADER, --hea..  Extra header (e.g. "X-Forwarded-For: 127.0.0.1")
    --headers=HEADERS   Extra headers (e.g. "Accept-Language: fr\nETag: 123")
    --auth-type=AUTH..  HTTP authentication type (Basic, Digest, NTLM or PKI)
    --auth-cred=AUTH..  HTTP authentication credentials (name:password)
    --auth-file=AUTH..  HTTP authentication PEM cert/private key file
    --ignore-code=IG..  Ignore HTTP error code (e.g. 401)
    --ignore-proxy      Ignore system default proxy settings
    --ignore-redirects  Ignore redirection attempts
    --ignore-timeouts   Ignore connection timeouts
    --proxy=PROXY       Use a proxy to connect to the target URL
    --proxy-cred=PRO..  Proxy authentication credentials (name:password)
    --proxy-file=PRO..  Load proxy list from a file
    --tor               Use Tor anonymity network
    --tor-port=TORPORT  Set Tor proxy port other than default
    --tor-type=TORTYPE  Set Tor proxy type (HTTP, SOCKS4 or SOCKS5 (default))
    --check-tor         Check to see if Tor is used properly
    --delay=DELAY       Delay in seconds between each HTTP request
    --timeout=TIMEOUT   Seconds to wait before timeout connection (default 30)
    --retries=RETRIES   Retries when the connection timeouts (default 3)
    --randomize=RPARAM  Randomly change value for given parameter(s)
    --safe-url=SAFEURL  URL address to visit frequently during testing
    --safe-post=SAFE..  POST data to send to a safe URL
    --safe-req=SAFER..  Load safe HTTP request from a file
    --safe-freq=SAFE..  Test requests between two visits to a given safe URL
    --skip-urlencode    Skip URL encoding of payload data
    --csrf-token=CSR..  Parameter used to hold anti-CSRF token
    --csrf-url=CSRFURL  URL address to visit to extract anti-CSRF token
    --force-ssl         Force usage of SSL/HTTPS
    --hpp               Use HTTP parameter pollution method
    --eval=EVALCODE     Evaluate provided Python code before the request (e.g.
                        "import hashlib;id2=hashlib.md5(id).hexdigest()")

  Optimization:
    These options can be used to optimize the performance of sqlmap

    -o                  Turn on all optimization switches
```

Where to Start?

- Identify injection points
  - Login
  - Forms
  - API Endpoints
- Build command to test injection points

```
./sqlmap.py --headers="User-Agent: Mozilla/5.0 (X11; Ubuntu;
Linux i686; rv:25.0) Gecko/20100101 Firefox/25.0"
--cookie="security=low; PHPSESSID=oikbs8qcic2omf5gnd09kihsm7"-u
'http://localhost/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=S
ubmit#' --level=5 --risk=3 -p id
```

Putting it all in Jenkins

Once you’ve come up with a command that is able to test your possible injection sites, the next thing to do is automate it.

- Any CI system with shell access can be used
- Decide how security critical your application is to determine the frequency of your scans

Let's see it in action

OWASP ZAP

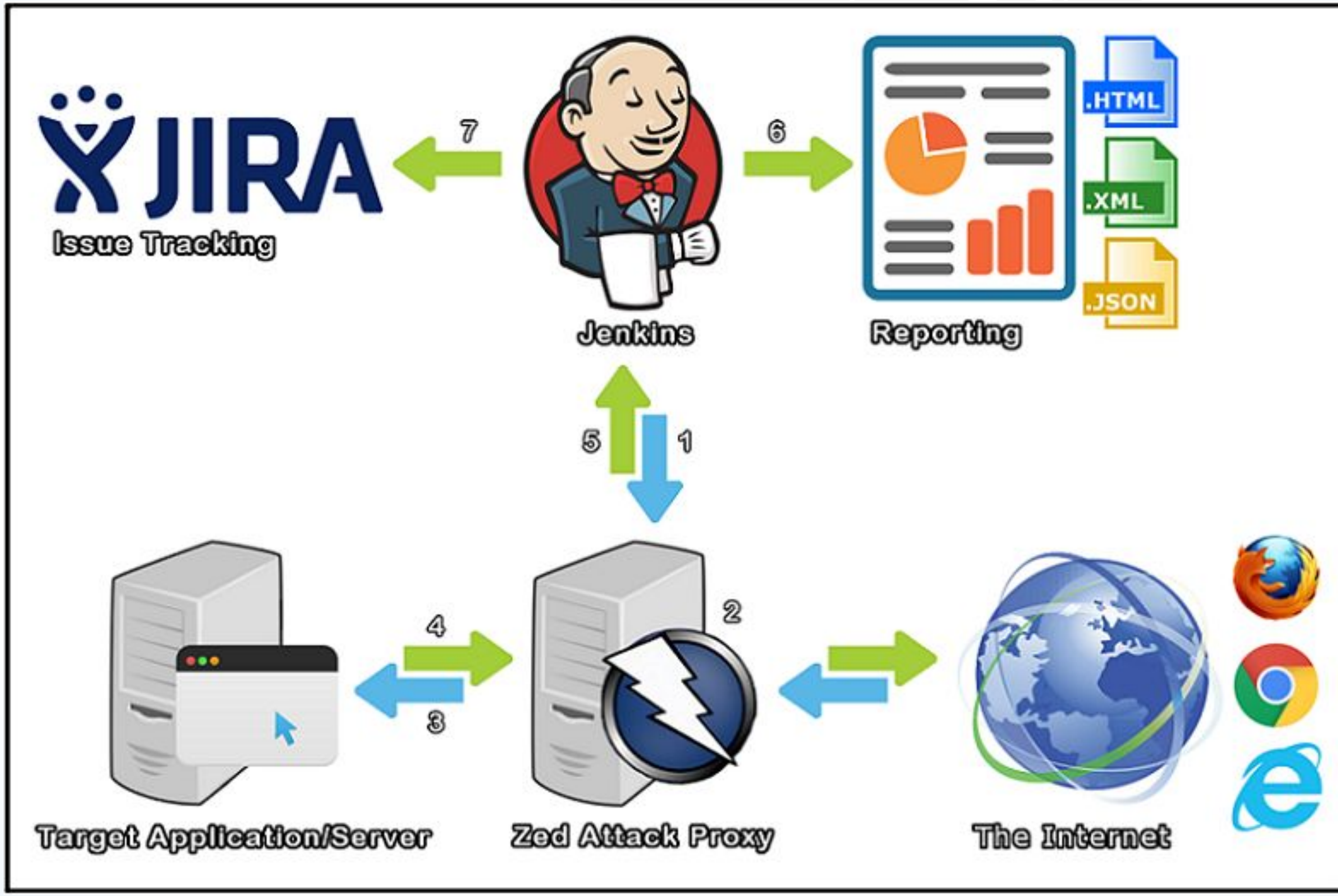
OWASP Zed Attack Proxy is a popular open source focused on finding security vulnerabilities in web applications.

- Can be used as a proxy server
- Automated scanning
- Fuzzer
- Supports scripting

How to Use OWASP ZAP

- GUI
  - Provides access to Active Scanning, Spidering, Fuzzing
- ZAP Daemon
  - Can be used in Jenkins to perform different scans or launch ZAP in proxy mode
- Official OWASP Zed Attack Proxy Jenkins Plugin

Putting it all in Jenkins



Let’s see it in action

Questions?