Effectiveness of Automated Application Penetration Testing Tools

ALEXANDRE FERREIRA

HARALD KLEPPE
Overview

- Introduction
- Background
- Target Application
- Vulnerability Scanners
- Test Results
- Conclusion
- Questions
Introduction

• Are automated penetration testing tools effective?
  – What and how is automated with these tools?
  – How much manual intervention is required from the results? (false positives / negatives)
  – What are the most effective tools?
  – What level of effectiveness is acceptable / necessary to properly support pentesters?
Background

- OWASP Top 10 Project
- What is a Penetration Test?
- What is a Penetration Testing Tool?
Target Application

• Why a new application?
  – Other tools (HacmeBank, WebGoat, ...)
  – Known implementations

• How and which vulnerabilities are implemented?
  – Let's have a look!
Target Application (2)

- SQL Injection
  - In URL and in HTML form

- Cross Site Scripting (XSS)
  - Stored and reflected

- Cross Site Request Forgery (CSRF)

- Path traversal

- Failure to restrict URL access

- Printed error
Vulnerability Scanners

- Tool selection
  - Both open source and commercial tools
  - Established tools
  - New players
  - Some tools: €10 000 per year
## Vulnerability Scanners (2)

<table>
<thead>
<tr>
<th>Commercial</th>
<th>Open Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acunetix</td>
<td>Paros</td>
</tr>
<tr>
<td>BurpSuite Pro</td>
<td>Skipfish</td>
</tr>
<tr>
<td>Core Impact</td>
<td>w3af</td>
</tr>
<tr>
<td>IBM AppScan</td>
<td>ZAProxy</td>
</tr>
<tr>
<td>NTOSpider</td>
<td></td>
</tr>
<tr>
<td>ParosPro</td>
<td></td>
</tr>
<tr>
<td>Qualys</td>
<td></td>
</tr>
</tbody>
</table>
Vulnerability Scanners (3)
Vulnerability Scanners (4)

Scan statistics:
- Scan time: 0:03:45.0606
- HTTP requests: 58228 (258.1/s), 44554 kB in, 13774 kB out (258.5 kB/s)
- Compression: 11179 kB in, 34554 kB out (51.1% gain)
- HTTP faults: 4 net errors, 0 proto errors, 0 retried, 0 drops
- TCP handshakes: 590 total (98.7 req/conn)
- TCP faults: 0 failures, 4 timeouts, 12 purged
- External links: 238 skipped
- Reqs pending: 0

Database statistics:
- Pivots: 125 total, 124 done (99.20%)
- In progress: 0 pending, 0 init, 0 attacks, 1 dict
- Missing nodes: 24 spotted
- Node types: 1 serv, 48 dir, 37 file, 0 pinfo, 2 unkn, 31 par, 6 val
- Issues found: 198 info, 1 warn, 7 low, 3 medium, 1 high impact
- Dict size: 58 words (58 new), 6 extensions, 256 candidates

[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 125
[+] Looking for duplicate entries: 125
[+] Counting unique nodes: 97
[+] Writing scan description...
[+] Writing crawl tree: 125
[+] Generating summary views...
[+] Report saved to 'test/index.html' [0x1555b1e6].
[+] This was a great day for science!

root@mx:~/skipfish-1.84b#
Test Results

- Low hitrate, differ from other research
- None of the tools “passed” this test
## Test Results (2)

<table>
<thead>
<tr>
<th>Vulnerability Type</th>
<th>Tools</th>
<th>Path traversal</th>
<th>CSRF</th>
<th>Reflected XSS</th>
<th>Stored XSS</th>
<th>Failure to restrict URL access</th>
<th>SQL Injection (in URL)</th>
<th>SQL Injection (in HTML form)</th>
<th>Printed error message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercial</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercial</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercial</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Source</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercial</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercial</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Source</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Source</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Source</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Source</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Source</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Source</td>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Test Results (3)

- Insufficient dataset to compare the tools generally
- Relying on crawling engines proves to be dangerous
Conclusion

- Scanners are conditionally effective
  - Nearly the entire scan can be automated
  - Quite some intervention is required
  - For our application: Skipfish + BurpSuite
  - Necessary effectiveness
Conclusion (2)

• Further research
  - Crawling abilities of different scanners
  - Selective scanning
Questions